A 3D 6-subiteration curve thinning algorithm based on \( P \)-simple points

Christophe Lohou\(^a\),*, Gilles Bertrand\(^b\)

\(^a\)Laboratoire de Logique, Algorithmique, Informatique de Clermont (L3AIC), Département d’Informatique, Institut Universitaire de Technologie, Ensemble Universitaire des Cézeaux, B.P. 86, F-63172 Aubière Cedex, France

\(^b\)Laboratoire d’Algorithmique et Architecture des Systèmes Informatiques (L2A), École Supérieure d’Ingénieurs en Électrotechnique et Électronique (ESIEE), 2, bd Blaise Pascal, Cité Descartes, B.P. 99, F-93162 Noisy-le-Grand Cedex, France

Received 26 September 2003; received in revised form 29 April 2004; accepted 10 February 2005

Available online 25 July 2005

Abstract


In this paper, by applying this methodology, we propose a 6-subiteration curve thinning algorithm which deletes at least all the points removed by two 6-subiteration curve thinning algorithms: either the one proposed by Palágyi and Kuba [A 3D 6-subiteration thinning algorithm for extracting medial lines, Pattern Recogn. Lett. 19(7) (1998) 613–627.], or the one proposed by Gong and Bertrand

* Corresponding author: Département d’Informatique, Institut Universitaire de Technologie, 8, rue Jean-Baptiste Fabre, B.P. 219, F-43006 Le Puy-en-Velay Cedex, France.

E-mail addresses: christophe.lohou@iut.u-clermont1.fr (C. Lohou), g.bertrand@esiee.fr (G. Bertrand).
1. Introduction

1.1. Simple points and thinning algorithms

Some graphical applications require to transform objects while preserving their topology [27, 33]. This leads to the well-known notion of simple point: a point in a binary image is said to be simple if its deletion from the image “preserves the topology” [1, 13, 15–20, 25, 32, 38, 39, 42–44]. A simple point may be locally characterized (i.e. the examination of the only $3 \times 3$ neighborhood centered around a point is enough to decide whether this point is simple or not).

Let us consider Fig. 1 which depicts a 2D object in a square grid: the point $a$ is not simple since its removal leads to disconnect the object; the point $b$ is also not simple since its removal leads to merge two connected components of the complementary of the object; the encircled points are simple (when the so-called 4-adjacency and 8-adjacency are, respectively, used for the object and its complementary, see [20], we also suppose that points outside this figure belong to the complementary of the object).

The notion of simple point is fundamental for all transformations where some topological features are to be preserved. Thinning algorithms are usually designed as processes which remove simple points and obey several other criteria. In fact, during the thinning process, certain simple points are kept in order to preserve some geometrical properties of the object. Such points are called end points. For the 3D case, we can define two different kinds of
end points: curve-end points \([5, 6, 9, 22–24, 27, 28, 33–35, 37, 45, 47]\) and surface-end points \([2, 5, 6, 9, 12, 14, 23, 24, 26, 28, 30, 35, 45]\). A thinning process which preserves curve-end points (resp. surface-end points) is called a curve thinning algorithm (resp. a surface thinning algorithm). The result obtained by a curve thinning algorithm (resp. a surface thinning algorithm) is called a curve skeleton (resp. a surface skeleton).

1.2. Parallel thinning algorithms

A major problem which arises when designing thinning algorithms is that the simultaneous removal of simple points may change the topology of an object: e.g. we see that, if we delete in parallel all simple points of the object depicted in Fig. 1, it will be disconnected.

Therefore, a parallel thinning algorithm must use a “certain deletion strategy” in order to preserve the topology. A popular way for overcoming this problem in 2D is to consider a directional strategy for removing points in parallel \([40, 41]\): 2D points are classified into four types corresponding to the four directions \(\text{North, South, East, West}\). A point of type \(\text{North}\) is a point of the object which has its immediate neighbor in the \(\text{North}\) direction which belongs to the complementary of the object. At each iteration, only simple points of a given type are considered for deletion. The four directions are alternatively used so that the thinning process is as symmetrical as possible. This directional strategy has, in 2D, good topological properties: the topology of the object is preserved except that connected components of two points may be erased. When designing a 2D thinning algorithm, it is therefore sufficient to check that these particular patterns are not deleted to have a sound algorithm.

Let us consider now the 3D case, e.g. let us consider an object in a cubic grid like the one depicted in Fig. 2. For implementing a directional strategy, six directions \(\text{North, South, East, West}\) are now to be used. The encircled points are points of type Up which are simple (when the so-called 26-adjacency and 6-adjacency are, respectively, used for the object and its complementary, see \([20]\), we also suppose that points outside this figure belong to the complementary of the object); note also that for this considered object, any point of type Up is a simple point. We see that, if all simple points of type Up are removed, the object \(X\) will be disconnected. Thus, the classical directional strategy does not work in 3D.
To solve this problem, two different solutions may be considered:

- Either we consider a deletion strategy based on subiterations, which consists in dividing a deletion iteration into several subiterations. These subiterations may be based on directions \([14,33–35,37,47]\) (as we have just seen before) or on subgrids \([5,28,34,45]\). Usually, points which may be deleted by such an algorithm must match at least one amongst several given masks or templates. These templates are proposed in such a way that the algorithm based on these templates is ensured to preserve the topology; i.e. the templates must be chosen in order to “detect” the configuration like the one in Fig. 2. Another example of deletion strategy consists in using an extended neighborhood (i.e. a neighborhood which strictly includes the \(3 \times 3 \times 3\) neighborhood centered around a considered point); such a strategy may lead to fully parallel thinning algorithms \([26,27,31]\).

- Or another class of simple point must be found in such a way that if we delete in parallel such points, then the topology is ensured to be preserved. This is what has been realized by the introduction of \(P\)-simple points \([2]\). In fact, this notion is very general and leads to different thinning schemes according to a certain strategy \([3]\) (directional \([2,6,12,21,22,24,30]\), symmetrical \([6,23]\), …).

### 1.3. \(P\)-simple points

One of the authors has proposed the notion of \(P\)-simple point \([2]\). Let us consider a subset \(X\) of \(\mathbb{Z}^3\), a subset \(P\) of \(X\), and a point \(x\) of \(P\). The point \(x\) is \(P\)-simple for \(X\) if for each subset \(S\) of \(P\) \(\{x\}\), \(x\) is simple for \(X \setminus S\).

We have the property that any algorithm removing only \(P\)-simple subsets (i.e. subsets composed solely of \(P\)-simple points) is guaranteed to keep the topology unchanged \([2]\). Thus, for a given \(P\), a thinning algorithm deleting \(P\)-simple points is guaranteed to preserve the topology; no proof is required in contrast to most of the already proposed thinning algorithms which do not use \(P\)-simple points (e.g. see \([26,27,31,33,35]\)).

Furthermore, a \(P\)-simple point may be locally characterized (i.e. the examination of the only \(3 \times 3 \times 3\) neighborhood centered around a point is enough to decide whether this point is \(P\)-simple or not). A thinning scheme, based on \(P\)-simple points, may be described by a two-step procedure: in the first step, points which belong to \(P\) are labelled and in the second step, points of \(P\) which are \(P\)-simple (and not end) are deleted (each of these two steps may be carried out in parallel). This deletion is made according to a certain strategy (directional, symmetrical, …), see \([3]\). Note that to check whether a point \(x\) is \(P\)-simple or not, we must know which points belong to \(P\) in the local neighborhood of \(x\). This is the reason why this scheme needs either a preliminary step of labelling (at each subiteration \([2]\), see also \([12,30]\)) or the examination of an extended neighborhood (to avoid the labelling).

We have seen that the directional strategy does not work in 3D. In fact, with the use of \(P\)-simple points, it is possible to derive a sound directional strategy, i.e. a strategy based on directions and which preserves the topology of the object. For example, let us consider again the object depicted in Fig. 2. Let \(X\) be this object. Let us precisely consider the set \(P\) given by \(P = \{x \in X; Up(x) \in \tilde{X}\}\). In Fig. 3(a), points of the initial object \(X\) which belong to \(P\) are depicted by black stars (we suppose that points outside this figure belong to \(\tilde{X}\)). If we delete \(y\), the point \(x\) is no longer simple, thus \(x\) is not \(P\)-simple (by taking \(S = \{y\}\) in the
Fig. 3. The set $P$ is given by $P = \{ x \in X; U_p(x) \in X \}$. (a) Points of the object $X$ of Fig. 2 which belong to $P$ are depicted by black stars, (b) the remaining object after the first parallel deletion of $P$-simple points.

definition of a $P$-simple point, given at the beginning of this section); the same reasoning holds for $y$. We may verify that all points of $P$ are $P$-simple, except $x$ and $y$. Thus, after the first parallel deletion of $P$-simple points, the remaining object is the one depicted in Fig. 3(b).

1.4. General notions about $P^X$-simple points and our methodology to conceive thinning algorithms

In [21,24], we have introduced a set $P^X$ derived from a given set $P$, which permits us to propose a new thinning scheme, based on the parallel deletion of $P^X$-simple points, and such that this scheme needs neither a preliminary step of labelling nor the examination of an extended neighborhood, in contrast to the already proposed thinning algorithms based on $P$-simple points.

In addition, we have proposed a general methodology to build a thinning algorithm $A'$ deleting $P^X$-simple points, from an existent thinning algorithm $A$, while preserving the same end points. This methodology consists in proposing successive “refinements” of $P$, until a certain $P$ is obtained such that at least all points deleted by $A$ are $P^X$-simple. This also implies that $A$ preserves the topology.

We have already proposed a 12-subiteration thinning algorithm [21,24], by the use of this methodology. It was natural to check our methodology on another thinning algorithm. Due to its simplicity to be encoded (mainly due to the fact that it is described by a few deleting templates), one of the most famous thinning algorithm is the 6-subiteration curve thinning algorithm proposed by Palágyi and Kuba [33]. Thus, in this paper, our purpose is to design a new 3D 6-subiteration curve thinning algorithm based on the parallel deletion of $P^X$-simple points, by applying our methodology. From the 6-subiteration curve thinning algorithm proposed by Palágyi and Kuba, we conceive a first thinning algorithm deleting $P^X$-simple points. Then, we “fit” it in such a way that it can delete at least all the points removed by Palágyi and Kuba’s 6-subiteration thinning algorithm, while preserving the same end points.
As a result, Palágyi and Kuba’s curve thinning algorithm deletes 9,916,926 points, our proposed algorithm deletes 23,721,982 points. These results have to be compared with this number: 25,985,118 simple and noncurve-end points amongst the 67,108,864 \((=2^{26})\) possible \(3 \times 3 \times 3\) configurations. We recall that this number is not reachable by any parallel thinning algorithm preserving topology, as we have seen before (in Section 1.2) during the discussion about the parallel deletion of simple points. Furthermore, we have observed that our algorithm also deletes at least all the points removed by Gong and Bertrand’s 6-subiteration thinning algorithm [14] in its curve variant proposed by Rolland, Chassery and Montanvert [37].

1.5. Contents

This paper is organized as follows. In Section 2, several notions of digital topology are recalled (more particularly, the notion of simple point and the topological numbers). The notion of \(P\)-simple point is recalled in Section 3. In Section 4, the notion of \(P^x\)-simple point is introduced, this notion permits to conceive thinning algorithms described by a one-step procedure. In Section 5, we give the general scheme used by the various 6-subiteration thinning algorithms and describe Palágyi and Kuba’s and Gong and Bertrand’s thinning algorithms. In Section 6, we introduce our recently proposed general methodology which permits to derive thinning algorithms based on \(P^x\)-simple points. The achieving of a thinning algorithm built with this methodology from Palágyi and Kuba’s 6-subiteration thinning algorithm is detailed in Section 7. In Section 8, we compare our algorithm with Gong and Bertrand’s thinning algorithm too, and give results on several images obtained by these three algorithms. We finally conclude and present some perspectives for future research in Section 9.

2. Basic notions

A point \(x \in \mathbb{Z}^3\) is defined by \((x_1, x_2, x_3)\) with \(x_i \in \mathbb{Z}\). We consider the three neighborhoods: \(N_{26}(x) = \{x' \in \mathbb{Z}^3 : \max(|x_1 - x'_1|, |x_2 - x'_2|, |x_3 - x'_3|) \leq 1\}\), \(N_6(x) = \{x' \in \mathbb{Z}^3 : |x_1 - x'_1| + |x_2 - x'_2| + |x_3 - x'_3| \leq 1\}\), and \(N_{18}(x) = \{x' \in \mathbb{Z}^3 : |x_1 - x'_1| + |x_2 - x'_2| + |x_3 - x'_3| \leq 2\} \cap N_{26}(x)\). We define \(N^*_n(x) = N_n(x) \setminus \{x\}\). We call, respectively, 6-, 18-, 26-neighbors of \(x\) the points of \(N^*_6(x), N^*_18(x) \setminus N^*_6(x), N^*_26(x) \setminus N^*_18(x)\); these points are, respectively, represented in Fig. 4(a) by black triangles, black squares, and black circles. The 6-neighbors of \(x\) determine six major directions (Fig. 4(b)): Up, Down, North, South, West, East, respectively, denoted by \(U, D, N, S, W, E\). Each point of \(N^*_26(x)\) may characterize one direction amongst the 26 that we can obtain from the 6 major ones, e.g. \(SW, USW, \ldots\). Let \(Dir\) denote one of these 26 directions. The point in \(N^*_26(x)\) along the direction \(Dir\) is called the Dir-neighbor of \(x\) and is denoted by \(Dir(x)\). In the following, points in \(N^*_26(x)\) are often denoted by \(p_i\) with \(0 \leq i \leq 26\), see Fig. 4(c); e.g. \(p_0\) is the USW-neighbor of \(p_{13}\), i.e. \(p_0 = USW(p_{13})\). Let \(X \subseteq \mathbb{Z}^3\). The points belonging to \(X\) (resp. \(\overline{X}\), the complement of \(X\) in \(\mathbb{Z}^3\)) are called black points (resp. white points).
Two points \( x \) and \( y \) are said to be \( n \)-adjacent if \( y \in N^*_n(x) \) \( (n = 6, 18, 26) \). An \( n \)-path is a sequence of points \( x_0, \ldots, x_k \), with \( x_i \) \( n \)-adjacent to \( x_{i-1} \) and \( 1 \leq i \leq k \). If \( x_0 = x_k \), the path is closed. Let \( X \subseteq \mathbb{Z}^3 \). Two points \( x \in X \) and \( y \in X \) are \( n \)-connected if they can be linked by an \( n \)-path included in \( X \). The equivalence classes relative to this relation are the \( n \)-connected components of \( X \). If \( X \) is finite, the infinite connected component of \( \overline{X} \) is the background, the other connected components of \( \overline{X} \) are the cavities. In order to have a correspondence between the topology of \( X \) and the one of \( \overline{X} \), we have to consider two different kinds of adjacency for \( X \) and \( \overline{X} \) [20]: if we use an \( n \)-adjacency for \( X \), we have to use another \( \overline{n} \)-adjacency for \( \overline{X} \). In this paper, we only consider \((n, \overline{n}) = (26, 6)\). The presence of an \( n \)-hole in \( X \) is detected whenever there is a closed \( n \)-path in \( X \) that cannot be deformed, in \( X \), into a single point (see [16], for further details). For example, a hollow ball has one cavity and no hole, a solid torus has one hole and no cavity, and a hollow torus has one cavity and two holes.

Let \( X \subseteq \mathbb{Z}^3 \). A point \( x \in X \) is said to be \( n \)-simple if its removal does not “change the topology” of the image, in the sense that there is a one-to-one correspondence between the components, the holes of \( X \) and \( \overline{X} \) and the components, the holes of \( X \setminus \{x\} \) and \( \overline{X} \cup \{x\} \) (see [16], for a precise definition). The set composed of all \( n \)-connected components of \( X \) is denoted by \( \mathcal{C}_n(X) \). The set of all \( n \)-connected components of \( X \) and \( n \)-adjacent to a point \( x \) is denoted by \( \mathcal{C}_n^+(X) \). Let \( \#X \) denote the number of elements which belong to \( X \). The topological numbers relative to \( X \) and \( x \) are the two numbers [1]: \( T_6(x, X) = \#\mathcal{C}_6^+[N^*_6(x) \cap X] \) and \( T_{26}(x, X) = \#\mathcal{C}_{26}^+[N^*_{26}(x) \cap X] \) (in fact, \( T_{26}(x, X) = \#\mathcal{C}_{26}[N^*_{26}(x) \cap X] \)), since any 26-connected component of black points in \( N^*_{26}(x) \) is inevitably 26-adjacent to \( x \). These numbers lead to a very concise characterization of 3D simple points [7,29]: \( x \in X \) is \( 26 \)-simple for \( X \) if and only if \( T_{26}(x, X) = 1 \) and \( T_6(x, \overline{X}) = 1 \). Note that this characterization is equivalent to the one proposed by Saha et al. [42–44].

Some examples are given in Fig. 5. The topological numbers relative to \( x \) and \( X \) or \( \overline{X} \) are \((T_{26}(x, X)), T_6(x, \overline{X}) = (1, 2), (2, 1), (1, 2), (1, 1)\) for the configurations (a), (b), (c), and (d), respectively. Only the configuration in Fig. 5(d) corresponds to a 26-simple point.
3. P-simple points

Let us introduce the notions of P-simple point and P-simple set [2]. In the following, we consider a subset $X$ of $\mathbb{Z}^3$, a subset $P$ of $X$, and a point $x$ of $P$.

**Definition 1.** The point $x$ is P-simple (for $X$) if for each subset $S$ of $P \setminus \{x\}$, $x$, is 26-simple for $X \setminus S$. Let $PSP(P)$ denote the set of all $P$-simple points. A subset $D$ of $X$ is P-simple if $D \subseteq PSP(P)$.

We have the property that any algorithm removing only P-simple subsets (i.e. subsets composed solely of P-simple points) is guaranteed to keep the topology unchanged [2].

We give a local characterization of a P-simple point [4] (see also [3,8]):

**Proposition 2.** Let $R$ denote the set $X \setminus P$. The point $x$ is P-simple iff:

$$
\begin{align*}
T_{26}(x, R) &= 1, \\
T_6(x, \overline{X}) &= 1, \\
\forall y \in N_{26}^+(x) \cap P, \exists z \in R & \text{ such that } z \text{ is } 26\text{-adjacent to } x \text{ and to } y, \\
\forall y \in N_6^+(x) \cap P, \exists z \in \overline{X} & \text{ and } \exists t \in \overline{X} \text{ such that } \{x, y, z, t\} \text{ is a unit square.}
\end{align*}
$$

Some examples are given in Fig. 6: only the points $x$ in (a) and (b) are P-simple. Let us consider the subset $X$ depicted in Fig. 6(c). The subset $S = \{p, q, r\}$ is a subset of $P \setminus \{x\}$, and $x$ is not simple for $X \setminus S$. Therefore by Definition 1, the point $x$ cannot be a P-simple point; or directly with Proposition 2, the first P-simplicity condition is not verified because $T_{26}(x, R) = 2$.

Let us consider the subset $X$ depicted in Fig. 3(a) again. We may verify that the two points $x$ and $y$ are not P-simple because the first P-simplicity condition of Proposition 2 is not verified ($T_{26}(x, R) = T_{26}(y, R) = 2$).

![Fig. 5. Points belonging to $X$ and $\overline{X}$ are, respectively, represented by black disks and white circles. Only the point $x$ in (d) is 26-simple.](image-url)
In the rest of this paper, we will propose a thinning algorithm based on the parallel deletion of $P$-simple points. With regard to the previous definition, $P$ will be the set of points which are candidates to be deleted; $P$ being defined according to a certain strategy of deletion [3]: directional [2,6,12,21,22,24,30] (this is the case for the algorithm proposed in this paper), symmetrical [6,23], . . . ; and $R$ will be the set of points which are not candidates for the deletion.

4. Strategies to detect $P$-simple points

For each $x$ of $\mathbb{Z}^3$, we consider a finite family $\mathcal{T}$ of pairs of subsets of $\mathbb{Z}^3$ $(B^k(x), W^k(x))$ with $1 \leq k \leq l$, such that $B^k(x) \cap W^k(x) = \emptyset$ and $x$ belongs to $B^k(x)$; $\mathcal{T}$ is said to be a family of templates.

In the following, we consider a subset $X$ of $\mathbb{Z}^3$. Let $P(\mathcal{T}, X) = \{x \in \mathbb{Z}^3 : \exists k \text{ with } 1 \leq k \leq l \text{ such that } B^k(x) \subseteq X \text{ and } W^k(x) \subseteq \overline{X} \}$. In fact, $P(\mathcal{T}, X)$ corresponds to a hit or miss transform of $X$ by $\mathcal{T}$ [46].

A thinning algorithm, based on the deletion of $P$-simple points, could consider subsets $P$ which would be characterized by a certain family $\mathcal{T}$ of templates. Such an algorithm must decide whether a point $x$ is $P(\mathcal{T}, X)$-simple or not: it must check if the point $x$ belongs to $P(\mathcal{T}, X)$, and in order to verify the four conditions of Proposition 2, it must check if the points $y$ of $N^+_2(x)$ belong to $P(\mathcal{T}, X)$. Such an algorithm may operate according to different ways to detect the points belonging to $P(\mathcal{T}, X)$ and the points being $P(\mathcal{T}, X)$-simple:

- The first strategy consists of the repetition of two steps [2]. During the first step, the points belonging to $P(\mathcal{T}, X)$ are labelled, through the access of $B^k(x)$, and of $W^k(x)$, for all points $x$ of $\mathbb{Z}^3$; at most $l$ pairs $(B^k(x), W^k(x))$ have to be checked. During the second step, the four conditions of $P$-simplicity of Proposition 2 are checked for all points of
4.1. Notion of $P^x$-simple points and detection of such points

Let $P$ be a subset of $X \subseteq \mathbb{Z}^3$ and let $x$ be a point of $\mathbb{Z}^3$. In this section, we introduce the subset $P^x$, locally defined for each point $x$ of $\mathbb{Z}^3$ and from $P$. We will consider as before that the set $P$ is described by a family $\mathcal{T}$ of templates. From this subset $P^x$, we will derive the notion of a $P^x$-simple point.

For each point $x$ of $\mathbb{Z}^3$, we define the subset $P^x(\mathcal{T}, X)$ of $\mathbb{Z}^3$, determined by $P^x(\mathcal{T}, X) = \{y \in N_{26}(x) : \exists k \text{ with } 1 \leq k \leq l \text{ such that } [B^k(y) \cap N_{26}(x)] \subseteq X \text{ and } [W^k(y) \cap N_{26}(x)] \subseteq X \}$. In fact, $P^x(\mathcal{T}, X)$ is constituted by the points $y$ of $N_{26}(x) \cap X$ which “may belong” to $P(\mathcal{T}, X)$, by the only inspection of membership to $X$ or to $\overline{X}$ of points belonging to $[B^k(y) \cup W^k(y)] \cap N_{26}(x)$. We have $P^x(\mathcal{T}, X) \supseteq \{P(\mathcal{T}, X) \cap N_{26}(x)\}$.

In [24], we have proven that a $P^x(\mathcal{T}, X)$-simple point $x$ is $P(\mathcal{T}, X)$-simple. This implies that an algorithm deleting in parallel $P^x(\mathcal{T}, X)$-simple points is guaranteed to preserve the topology, because it deletes $P(\mathcal{T}, X)$-simple subsets. In addition, since $P^x(\mathcal{T}, X)$ is completely known in $N_{26}(x)$ for each point $x$, this permits us to propose a new thinning scheme, based on the parallel deletion of $P^x(\mathcal{T}, X)$-simple points $x$, which needs neither a preliminary step of labelling nor the examination of an extended neighborhood, in contrast to the already proposed thinning algorithms based on $P(\mathcal{T}, X)$-simple points (see the one proposed in [2], for example). The thinning algorithm, which we will propose in Sections 5.4 and 7, deletes $P^x(\mathcal{T}, X)$-simple points.

Notations: In the following, we write $P$ (resp. $P^x$) instead of $P(\mathcal{T}, X)$ (resp. $P^x(\mathcal{T}, X)$) and “$x$ is a $P$-simple (resp. $P^x$-simple) point” means “$x$ is a $P(\mathcal{T}, X)$-simple (resp. $P^x(\mathcal{T}, X)$-simple) point”.

4.2. Example

In this section, we give an example that illustrates there exist points $x$ which are $P$-simple but not $P^x$-simple, for the same family $\mathcal{T}$.

The 6-subiteration thinning algorithm proposed by Palágyi and Kuba deletes certain simple points whose neighbor, according to a considered direction, belongs to $\overline{X}$ (see Section 5.2). So, we propose to consider the subset $P$ such that $P = \{x \in X : \text{the } U\text{-neighbor of } x$
belongs to $\overline{X}$ (see also Section 7.1). It may be described by a family $\mathcal{F}$ constituted by only one template $(B^1(x), W^1(x))$ with $B^1(x) = \{x\}$ and $W^1(x) = \{U(x)\}$.

Let us consider a $3 \times 3 \times 3$ neighborhood of a point $x$. Let $I(x)$ be the set of points in $N_{26}(x) \cap X$ for which the $U$-neighbor belongs to $N_{26}(x)$ and where $x = p_{13}$ (i.e. with notations of Fig. 4(c), $I(x) = \{p_3, \ldots, p_8, p_{12}, \ldots, p_{17}, p_{21}, \ldots, p_{26}\} \cap X$). Let $J(x) = [N_{26}(p_{13}) \cap X] \setminus I(x)$ where $x = p_{13}$ (i.e. $J(x) = \{p_0, \ldots, p_2, p_9, \ldots, p_{11}, p_{18}, \ldots, p_{20}\} \cap X$). We have:

- For $y \in I(x)$, $y \in P^x$ iff $B^1(y) \cap N_{26}(x)(=\{y\})$ is included in $\overline{X}$ (always verified for any $y \in I(x)$) and $W^1(y) \cap N_{26}(x)(=\{U(y)\})$ is included in $\overline{X}$. Therefore for $y \in I(x)$, we have $y \in P^x$ if $U(y)$ belongs to $\overline{X}$.
- For $y \in J(x)$, $y \in P^x$ iff $B^1(y) \cap N_{26}(x)(=\{y\})$ is included in $\overline{X}$ (always verified for any $y \in J(x)$) and $W^1(y) \cap N_{26}(x)(=\emptyset)$ is included in $\overline{X}$ (always verified for any $y$). Therefore $y \in P^x$ for any $y \in J(x)$.

In summary, for each point $x$ of $\mathbb{Z}^3$, $P^x = \{y \in I(x) : U(y) \in \overline{X}\} \cup J(x)$.

Let us consider the configuration depicted in Fig. 7(a) and let us apply the previous remarks. The points of $P$ (resp. $P^x$) are represented by a star in Fig. 7(b) (resp. Fig. 7(c)). In Fig. 7(b), the point $x$ belongs to $P$ since $x$ belongs to $X$ and the $U$-neighbor of $x$ belongs to $\overline{X}$. The point $y$ belongs to $R$, with $R = X \setminus P$, since $z (= U(y))$ belongs to $X$. We highlight that we must examine points outside the local neighborhood of $x$ (or we must use a preliminary labelling—that corresponds to the two first strategies described in the beginning of this section) to check whether $y$ belongs to $P$ or not; and this check is needed to verify the $P$-simplicity of $x$. In this case, $x$ is a $P$-simple point. In Fig. 7(c), the point $x$ belongs to $P^x$ since $x$ belongs to $I(x)$ and the $U$-neighbor of $x$ belongs to $\overline{X}$. The point $y$ belongs to $P^x$ as $y$ belongs to $J(x)$. In this case, $x$ is not a $P^x$-simple point because the first and third $P^x$-simplicity conditions are not verified: with $R^x = X \setminus P^x$, $T_{26}(x, R^x) = 0$ and there is no point of $R^x$ 26-adjacent to $x$ and to $y$. We highlight that the only examination of the local neighborhood is enough to know which points belong to $P^x$ and therefore to check the $P^x$-simplicity of $x$.

![Fig. 7. (a) Initial configuration. (b) The point x is P-simple. (c) is not P-x-simple.](image-url)
5. Description of the used thinning algorithms

In this section, we recall the general scheme for 6-subiteration thinning algorithms and then we specify it more precisely for the algorithm proposed by Palágyi and Kuba [33], for the algorithm proposed by Gong and Bertrand [14,37], and partially for our algorithm deleting \(P^x\)-simple points (in fact, the set \(P\) of our final algorithm will be defined in Section 7).

5.1. General scheme

A thinning scheme consists of the repetition until stability of deletion iterations. In the case of 6-subiteration thinning algorithms, an iteration is divided into 6 subiterations, each of them successively corresponding to 1 of the 6 following directions: Up, Down, North, South, East, and West (see Fig. 4 (b)). Let \(\alpha\) denote such a direction. The stability is obtained when there is no more deletion during 6 successive subiterations. Such a thinning scheme can be described by

\[
X^i = X^{i-1} \setminus \text{DEL}(X^{i-1} \cup \alpha) \quad \text{for the } i\text{th deletion subiteration (}i > 0),
\]

with \(X^0 = X\), and \(\text{DEL}(Y, \alpha)\) being the set of points to be deleted from \(Y\), according to the direction \(\alpha\) corresponding to the \(i\)th subiteration. The stability is obtained when \(X^k = X^{k+6}\).

5.2. Palágyi and Kuba’s thinning algorithm

Palágyi and Kuba have proposed a 6-subiteration curve thinning algorithm [33], denoted by \(\text{PK}\) in the following. A set of \(3 \times 3 \times 3\) matching templates is given for each direction. For a given direction \(\alpha\), a point is deletable by \(\text{PK}\) if at least one template (or theirs rotations around the axis along the direction \(\alpha\)) in the set of templates matches it. The set of templates used by \(\text{PK}\) along the direction \(\alpha\) is denoted by \(T_\alpha\) and is represented in Fig. 9 for the direction \(\alpha = U\); see notations in Fig. 8. The templates for the other directions can be obtained by appropriate rotations and reflections of these templates.

We recall the definition of an end point, used in \(\text{PK}\), which is also adopted in our proposed algorithm. A black point \(x\) is an end point if the set \(N^{\pm}_x(x)\) contains exactly one black point. We may verify that end points are prevented to be deleted by the templates of \(T_\alpha\) (Fig. 9).

According to the previous general thinning scheme of Section 5.1, \(\text{DEL}(Y, \alpha)\) is the set of points of \(Y\) which are matched by at least one template of \(T_\alpha\), for the direction \(\alpha\) corresponding to the deletion subiteration.

5.3. Gong and Bertrand’s thinning algorithm

We first introduce some notations. We recall that \(\alpha\) denotes one of the six deletion directions (see Fig. 4(b)). Let \(\bar{\alpha}\) denote the opposite direction. Let \(N^9_2(x)\) (resp. \(N^{18}_2(x)\))
Fig. 9. The set \( \mathcal{T}_U \) of thinning templates for the direction \( U \), up to rotations around the vertical axis (see notations in Fig. 8).

denote the four 6-neighbors (resp. 18-neighbors) of \( x \) which belong to the \( 3 \times 3 \) window perpendicular to the direction \( z \) and containing \( x \) (in fact, \( N^6_3(x) = N^x_6(x) \backslash \{x(x), \overline{x}(x)\} \)).

Let \( X \subseteq \mathbb{Z}^3 \). A point \( x \in X \) is said to verify the condition \( GB \) iff (see also Fig. 10):

\[
\begin{align*}
GB_1 & : \pi(x) \in \overline{X}, \\
GB_2 & : \overline{x}(x) \in X, \\
GB_3 & : \forall y \in N^6_3(x), \text{ if } y \in \overline{X} \text{ then } \pi(y) \in \overline{X}, \\
GB_4 & : \forall y \in N^6_2(x), \forall z \in N^{18}_2(y) \cap N^6_2(x), \text{ and } t = N^6_2(z) \cap N^6_2(y) \cap N^{18}_2(x), \text{ if } y \in \overline{X} \text{ and } z \in \overline{X} \text{ and } t \in \overline{X} \text{ then } \pi(t) \in \overline{X}.
\end{align*}
\]

Gong and Bertrand have proposed a 6-subiteration surface thinning algorithm [14]. For a given direction \( z \), a point is deletable by this algorithm if it verifies the previous condition \( GB \) and another Boolean condition \( C_1 \) which ensures that this point is not a surface-end point. By replacing the Boolean condition \( C_1 \) that a point must not be a surface-end point by the one \( C_2 \) which avoids this point to be a curve-end point (with the same characterization which has been adopted in Section 5.2), we obtain the 6-subiteration curve thinning algorithm, proposed by Rolland, Chassery and Montanvert [37]. According to the previous general thinning scheme of Section 5.1, for this last algorithm, \( \text{DEL}(Y, z) \) is the set of points of \( Y \) which verify both \( GB \) and \( C_2 \) according to the direction \( z \) corresponding to the deletion subiteration. This algorithm is denoted by \( GB \) in the following.
Fig. 10. “Illustration” of the condition $GB$ which is one of the two conditions that a point must verify to be deleted by Gong and Bertrand’s algorithm. More precisely, (a) (resp. (b), (c), (d)) “depicts” the condition $GB_1$ (resp. $GB_2$, $GB_3$, $GB_4$) of the condition $GB$, for the direction $x = U$.

Because this study only concerns with curve thinning algorithms, from now, we write end points instead of curve-end points.

5.4. Algorithm deleting $P^x$-simple points

A 6-subiteration thinning algorithm removing $P$-simple points has already been proposed [2,22] (see also [12,30]). Now, we give a general scheme for 6-subiteration thinning algorithms deleting $P^x$-simple points. It can be described by the scheme of Section 5.1, with $DEL(Y, x) = PSP(P^x)$; $PSP(P^x)$ being the set of $P^x$-simple points for $Y$ which are not end points and according to the direction $x$ corresponding to the deletion subiteration. From this scheme, we will propose our algorithm by defining an appropriate $P$, in the sense that we investigate $P$ such that our algorithm deletes at least the points removed by $PK$ (the achieving of such a set $P$ is detailed in Section 7). In the following, we write LB to indicate our final algorithm which deletes $P^x$-simple points, while preserving end points.
5.5. Implementation

A preliminary step to the use of PK, GB, or LB on real 3D binary images consists in producing all possible 67 108 864 (\(=2^{26}\)) configurations of the \(3 \times 3 \times 3\) neighborhood of a point \(x\) (i.e. \(N_{26}^x(x)\)) and to retain only:

- either those ones verifying at least one of the thinning templates in the case of PK,
- either those ones verifying both the four conditions of GB and the condition \(C_2\) in the case of GB,
- or those ones which correspond to a \(P^x\)-simple and nonend point in the case of LB (once a satisfying set \(P\) has been found).

This must be done for each deletion direction. Then, we use a binary decision diagram (BDD) [10,11] to encode these deletable configurations. A BDD can be seen as a compressed graph which permits to know here (see also [36]) whether a configuration, only described through the points of \(X\) and \(\overline{X}\) in a \(3 \times 3 \times 3\) neighborhood, is deletable or not; this decision being made by a simple inspection of the neighborhood without any other computation.

In other words, the use of the associated BDD avoids checking:

- the matching of a configuration with the thinning templates, in the case of PK;
- whether a point verifies both the four conditions of GB and the condition \(C_2\), in the case of GB;
- whether the points in \(N_{26}^x(x)\) belong to \(P^x\) or not, to check the four \(P^x\)-simplicity conditions on \(x\) to know whether \(x\) is \(P^x\)-simple or not, and to check whether \(x\) is an end point or not, for a considered configuration whose central point is \(x\), in the case of LB.

In summary, once the BDDs are obtained, the implementation is the same for the algorithms PK, GB, and LB, only the size of “storage” of the called BDDs is different. In other words, PK, GB, and LB have the same computational complexity.

6. Methodology to design 3D thinning algorithms based on \(P^x\)-simple points

We have recently proposed a methodology in order to conceive thinning algorithms, based on \(P^x\)-simple points [21,22,24].

From an existent algorithm \(\mathcal{A}\), given by a set of templates, this methodology consists in proposing successive “refinements” of \(P\), until a set \(P\) is obtained such that at least all points \(x\) deleted by \(\mathcal{A}\) are \(P^x\)-simple. Let \(S_0\) be the set composed of all the local configurations which match at least one of the templates describing \(\mathcal{A}\). More precisely:

- Initially, by the examination of the templates, we extract a simple pattern which occurs in each of the templates. With this pattern, we initialize the first set \(P\).
- We automatically generate all possible configurations in a \(3 \times 3 \times 3\) neighborhood, and we retain only the ones corresponding to \(P^x\)-simple and nonend points \(x\). Let \(S_1\) be the set of these configurations.
• We check whether $S_0$ is contained in $S_1$:
  • If it not the case, then there exists at least one configuration $C$ corresponding to a point $x$ deleted by $\mathcal{A}$ but not $P^x$-simple. We examine the “behavior” of the points in this configuration $C$ (deletable or not by $\mathcal{A}$, belonging to $P^x$ or not). Then we adapt the set $P$ in such a way that this configuration becomes $P^x$-simple (i.e. by adding or removing black or white points in the templates which prevented the central point $x$ of $C$ from being $P^x$-simple; further details are given in Section 7). As before, we generate a new set $S_1$ and we repeat this procedure until $S_0$ is contained in $S_1$.
  • If it is the case, then all configurations deleted by $\mathcal{A}$ are $P^x$-simple; we call $\mathcal{A}'$ the algorithm which deletes $P^x$-simple points.

If the criterion is satisfied ($S_0$ is contained in $S_1$), then we have succeeded in deriving an algorithm $\mathcal{A}'$ (removing $P^x$-simple points) which deletes at least all points removed by $\mathcal{A}$ while preserving the same end points. The properties of such an algorithm are:

• The achieving of a skeleton usually requires less iterations with $\mathcal{A}'$ than with $\mathcal{A}$. It is not always the case; e.g., some configurations may appear after the first iteration of the algorithm $\mathcal{A}$ but they do not after the first iteration of the algorithm $\mathcal{A}'$; thus, we cannot compare the behavior of these algorithms at the second iteration (see also Section 8.2 for the explanations and the results obtained by the three algorithms used in this paper on several images). Moreover, we will show that there exist objects for which our algorithm needs more deletion iterations than the one, from ours derives, requires (see Section 7.2, more precisely the object depicted in Fig. 13(c), and Section 8.1).
  • When $\mathcal{A}'$ is obtained, then both $\mathcal{A}'$ and $\mathcal{A}$ are guaranteed to preserve the topology.

With this methodology, we have proposed a 12-subiteration thinning algorithm [21,24]. In this paper, which is an extension of [22], we propose a 6-subiteration thinning algorithm also designed with this methodology (Section 7).

7. Our thinning algorithm (LB)

In this section, by using the methodology presented above, we give the entire reasoning, which leads us to propose successive conditions of membership to a set $P$, in order that at least all points deleted by $pK$ are $P$-simple. This is achieved with our second proposal $P_2$.

We first deal with the direction $U$ until a general comparison of our results. In the following, when we write “a point belongs to $P^K$” then $x$ is the point $p_{13}$ for the considered configuration (see Fig. 4(c)). We write “a configuration is $P^x$-simple” to mean that the central point $x (= p_{13})$ of this configuration is $P^x$-simple. Let $y$ be a point of a configuration, $y$ belongs to $\{p_0, \ldots, p_{26}\}$, see Fig. 4(c); we write “a point $y$ verifies a template $T$” to mean that the template $T$ matches the configuration whose central point is $y$. 


7.1. First membership condition

We observe that any point of \( X \) deleted by \( \mathcal{T}_U \) is such that its \( U \)-neighbor belongs to \( \overline{X} \) (see templates in Fig. 9). Thus, we propose to consider \( P_1 = \{ x \in X : \text{the } U \text{-neighbor of } x \text{ belongs to } \overline{X} \} \), already studied in Section 4.2. Among all \( 2^{26} \) possible configurations, we obtain 4,423,259 ones corresponding to \( P_1 \)-simple and non-end points, for the direction \( U \).

Let us consider the configuration \( \mathcal{C} \) in Fig. 11 (a). The three points \( p_3, p_7, \) and \( p_{13} \) belong to \( P_1 \) (Fig. 11 (b)) because they belong to \( X \), and the \( U \)-neighbor of these points belongs to \( \overline{X} \). The first and the third \( P_1 \)-simplicity conditions are not verified for the central point \( p_{13} \). Thus, the point \( p_{13} \) is not \( P_1 \)-simple. Nevertheless, it is matched by a rotation around the vertical axis of \( M_5 \) of \( \mathcal{T}_U \). Therefore, it should be deleted by the algorithm we want to obtain.

Let us examine the behavior of the other points of this configuration with the templates \( \mathcal{T}_U \) (see Fig. 11(a)). The point \( p_3 \) may verify a rotation around the vertical axis of \( M_5 \) or \( M_6 \). The point \( p_7 \) cannot be deleted because \( p_6 (=W(p_7)) \) belongs to \( \overline{X} \) and \( p_3 (=U(p_6)) \) belongs to \( X \), and the templates are such that for any point \( x \) deleted by \( \mathcal{T}_U \) and for any \( y \) belonging to \( N^*_6(x) \cap \overline{X} \), \( y \) being neither \( U(x) \) nor \( D(x) \), the point \( U(y) \) must belong to \( \overline{X} \); but \( p_7 \) belongs to \( P_1 \).

In other words, the point \( p_7 \) is candidate for deletion with our present proposal but it cannot be deleted by \( PK \). As it is the only difference concerning the “behavior” of points of \( \mathcal{C} \) for \( PK \) and our proposal, we now propose a second condition of membership to a set \( P \) (i.e. \( P_2 \)) such that \( p_7 \) does not belong to \( P_2 \) in order that \( \mathcal{C} \) becomes \( P_2 \)-simple (if the updated membership of the other points of \( \mathcal{C} \) permits it).

7.2. Second membership condition

We first introduce some notations (already seen in Section 5.3). We recall that \( \alpha \) denotes one of the six deletion directions. Let \( \overline{\alpha} \) denote the opposite direction. Let \( N^*_6(x) \) denote the four 6-neighbors of \( x \) which belong to the \( 3 \times 3 \) window perpendicular to the direction \( \alpha \) and containing \( x \) (in fact, \( N^*_6(x) = N_6^*(x) \setminus \{ \alpha(x), \overline{\alpha}(x) \} \)).
We propose to consider \( P_2 = \{ x \in X : \text{the } \alpha\text{-neighbor of } x \text{ belongs to } \overline{X} \text{ and for any point } y \text{ belonging to } N_x^0 \} \), according to the considered direction \( \alpha \).

With notations used in Section 4, the set \( P_2 \) can be described by the family composed of 16 pairs of subsets of \( \mathbb{Z}^3 \) \( (B^k(x), W^k(x)) \) with \( 1 \leq k \leq 16 \), depicted in Fig. 12 for the direction \( \alpha = U \); in fact, there are six main templates ((a), (b), (d), (f), (h), (p)), up to rotations around the axis \((x, y)\).

Let us consider the non-\( P_{1x} \)-simple configuration in Fig. 11(b) (see notations in Fig. 11(c)). The point \( p_{13} \) belongs to \( P_{x2} \), as it verifies the template in Fig. 12(a). The point \( p_3 \) belongs to \( P_{x2} \), as \( p_3 \) may verify the templates in Fig. 12(a), (c), (e), or (g). The point \( p_7 \) does not belong to \( P_{x2} \) because there exists a point \( y (= p_6 (= W(p_7))) \) in \( N_0^U(p_7) \cap N_26(x) \) which belongs to \( \overline{X} \), and such that \( p_3 (= U(y)) \) belongs to \( X \); or more directly because \( p_7 \) verifies no template in Fig. 12. So, this non-\( P_{1x} \)-simple configuration (Fig. 11(b)) is now \( P_{x2} \)-simple (Fig. 11(c)).

We obtain 6 129 527 configurations corresponding to \( P_{x2} \)-simple and non-end points, for the direction \( U \). The 2 124 283 configurations deleted by \( \mathcal{F}_U \) are also \( P_{x2} \)-simple. The fact
that the configurations deletable by PK are $P^x_2$-simple (for each direction and therefore for the whole algorithm) guarantees that the topology is well preserved by PK (as PK deletes subsets of $P^x_2$-simple points, see Section 3).

**Input:** binary array $(X, \bar{X})$ representing a picture with $(n, \bar{n}) = (26, 6)$.

**Output:** binary array $(X, \bar{X})$ corresponding to the skeleton obtained by LB.

```
{ integer i ← 0; (number of the current deletion subiteration)
  integer k ← 0; (number of successive subiterations with no deletion)
  do
    i ← i + 1;
    direction α ← f(i);
    with: • $f(i)$ returns Up (resp. Down, North, South, East and West)
      when $i - 1 \mod 6$ is equal to 0 (resp. 1, 2, 3, 4, 5).
    subset $Y ← DEL(X, α)$;
    with: • $DEL(X, α)$ being the set of $P^x_2$-simple
       and noncurve-end points for $X$,
       and according to the direction $α$;
     • a black point $x$ is a curve-end point if $|N_{26}^x(x) \cap X| = 1$;
     • $P_2 = \{x ∈ X :$ the $α$-neighbor of $x$ belongs to $\bar{X}$
       and for any point $y$ belonging to $N_2^x(x)$,
       if $y$ belongs to $\bar{X}$ then $α(y)$ must belong to $\bar{X}\}$;
     in fact, a point $x$ belongs to $P_2$ iff it verifies at least
     one of the templates in Fig. 12, according to the direction $α$.
    if ($Y ≠ ∅$) then
      { k ← 0;
        ($X ← X \cup Y;$) in fact, this operation may be automatically
        performed by the following instruction:
        $X ← X \setminus Y$;
      }
    else
      k ← k + 1;
  } while ($k ≠ 6$);
```

**Algorithm 1.** Our 3D 6-subiteration curve thinning algorithm LB based on $P^x_2$-simple points.

In summary, according to the general scheme of 6-subiteration thinning algorithm described in Section 5.1, and the one adopted for such algorithms based on the deletion of
Fig. 13. (a) This configuration cannot be deleted by PK whatever the deletion direction, and (b) is $P_2^x$-simple; in (c) (obtained from (a)) no point is deleted by PK, nevertheless $x$ is deleted by LB.

$P_2^x$-simple points described in Section 5.4, our proposed 3D 6-subiteration curve thinning algorithm LB, consisting in deleting the $P_2^x$-simple and noncurve-end points and according to the considered direction $x$, is described by Algorithm 1. Recall that the use of a BDD leads to a very efficient implementation of this algorithm (see Section 5.5).

For a better comparison between PK and LB, we generate the configurations deleted by these algorithms for each direction: PK deletes 9 916 926 configurations, i.e. there exists at least one deletion direction such that a given configuration among these ones is deleted for this direction by PK; LB deletes 23 721 982 configurations (139.2% “better”). In fact, there are 25 985 118 simple and nonend points amongst the 67 108 864 (= $2^{26}$) possible $3 \times 3 \times 3$ configurations and this number is not reachable by any parallel thinning algorithm which preserves topology (see the discussion about the parallel deletion of simple points in Section 1.2).

The configuration depicted in Fig. 13(a) cannot be deleted by PK, whatever the deletion direction. This configuration is $P_2^x$-simple (Fig. 13(b)), with $x = U$. Indeed, the point $p_2$ belongs to $P_2^x$ as $p_2$ may verify the templates in Fig. 12(a), (b), (c), or (d); $p_3$ belongs to $P_2^x$ as $p_3$ may verify the templates in Fig. 12(a), (c), (e), or (g); $p_{13}$ belongs to $P_2^x$ as it verifies the template in Fig. 12(a); $p_5$ does not belong to $P_2^x$ as $p_2 (= U(p_5))$ belongs to $X$ (or more directly, as $p_5$ verifies no template in Fig. 12); and $p_7$ does not belong to $P_2^x$ as there exists a point $y (= p_6 (= W(p_7)))$ in $N_0^y(p_7) \cap N_{26}(x)$ which belongs to $X$ and such that $U(y) (= p_3)$ belongs to $X$ (or more directly, as $p_7$ verifies no template in Fig. 12).

Fig. 13(c) shows an image built from the configuration in Fig. 13(a) such that each point is either a nonsimple point (except $x$) or an end point, and no point can be deleted by PK, nevertheless the point $x$ is deleted by LB. This example shows that there exist objects for which our algorithm needs more deletion subiterations than PK requires, to produce a skeleton. We said in Section 6 that the achieving of a skeleton usually requires less iterations with LB than with PK. This example shows that it is not always the case.
Fig. 14. Two times by row, an initial object followed by the skeleton obtained by GB, PK, LB (the number of deletion iterations and of deleted points are the same with these three algorithms for these images). Under each figure of skeleton are given the number of the last subiteration of deletion and the number of deleted points (see also Table 1).
8. Other results

8.1. Comparisons between GB and LB

We recall we have obtained 6 129 527 configurations corresponding to \( P_2^2 \)-simple and nonend points, for the direction \( U \). The 4 772 095 configurations deleted by GB, for the direction \( U \), are also \( P_2^3 \)-simple. The fact that the configurations deletable by GB are \( P_2^3 \)-simple (for each direction and therefore for the whole algorithm) guarantees that the topology is preserved by GB (as GB deletes subsets of \( P_2^2 \)-simple points, see Section 3).

For a better comparison between GB and LB, we have also generated the configurations deleted by these algorithms for each direction: GB deletes 21 194 234 configurations, i.e.

---

Fig. 15. By row, respectively, the initial object, the curve skeletons for GB, PK, LB. Under each figure of skeleton are given the number of the last subiteration of deletion and the number of deleted points (see also Table 2).
there exists at least one deletion direction such that a given configuration among these ones is deleted for this direction by GB; LB deletes 23 721 982 configurations (11.9% “better”).
The configuration depicted in Fig. 13(a) cannot be deleted by GB, whatever the deletion direction. Indeed, the point $p_{13}$ is such that $N^*_6(p_{13}) \cap X = \emptyset$, thus the condition $GB_2$ (described in Section 5.3) cannot be verified (in other words, there is no deletion direction $\pi$ such that $\overline{\pi}(x) \in X$).
As GB deletes only simple and nonend points, thus, once again, no point can be deleted by GB in the object depicted in Fig. 13(c).

8.2. Results on several 3D binary images

The skeletons of some 3D binary images, obtained with these three algorithms GB, PK, and LB, are shown in Figs. 14–17. Information about initial objects and the achieving of the skeletons are given in Tables 1–3 (the most useful ones are also taken up in Figs. 14–17). In Tables 1–3, the best performances are written in bold.

We observe that:

- For synthetic objects (Figs. 14 and 15—see resp. Tables 1–2), the geometrical appearance, the number of deletion subiterations, and the number of deleted points are almost the same between the three studied algorithms GB, PK, and LB. In fact, the number of deletion iterations and of deleted points are the same for these three algorithms for the objects depicted in Fig. 14.
- For the real image of vertebra (size: \(256 \times 256 \times 51\) voxels) depicted in Fig. 16, the biggest number of deleted points is obtained by PK. Our algorithm seems to be the “best compromise” among the three studied algorithms. Indeed, the number of deleted points is almost the same as the one obtained by PK, but the number of deletion subiterations of our algorithm is almost the half of the one required by PK.

The initial image of vertebra (object \(O_{9}\), Fig. 16) contains 95 small cavities to be preserved by topology-preserving algorithms. Therefore, each cavity is to be transformed into a closed surface segment. Those bubbles are created by peeling the objects from outside and inside. Due to their deletion rules, the three algorithms proposed in the paper have various “bubble blowing ability” (i.e. enlarging cavities from inside). For example, all thinning templates of PK contain numerous points which belong to the complementary of the object; therefore, small cavities (formed by a few points) cannot be enlarged. This implies that the “front propagation” is not isotropic, thus concurrent fronts do not meet half-way. This may explain the reason why PK deletes more points, while requiring more subiterations, to obtain a skeleton of the vertebra, than ours does. To back up what is previously said, our algorithm LB removes more points for the cavity-filled vertebra object (object \(O_{10}\), Fig. 17) while requiring less subiterations of deletion than by PK (see also Table 3).

In fact, it is difficult to assert that such an algorithm requires less subiterations or deletes more points than another one. The only statement is that for a same image, during the first deletion subiteration, by its very conception, our algorithm is ensured to delete more points than the other two algorithms (see results in the ninth column of Tables 1–3). We may also say that at the end of the first deletion subiteration, some configurations corresponding to simple but not deletable points (e.g. end points) may occur with our algorithm, avoiding that one to progress in this area, and perhaps with another algorithm either such configurations (“blocking” for ours) do not occur (because it deletes less points than ours) or they occur later.

In summary, as we said in Section 6, we cannot predict the behavior of these algorithms from the second deletion subiteration. We recall that there exist objects for which our
Table 1
Results obtained by the three thinning algorithms on several synthetic 3D binary images

<table>
<thead>
<tr>
<th>Object</th>
<th>Fig.</th>
<th>Ref.</th>
<th>Size</th>
<th>n.b.p.</th>
<th>t.a.</th>
<th>n.d.p.</th>
<th>n.l.d.s.</th>
<th>n.d.p.f.s.</th>
<th>r.c. (%)</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>Fig. 14</td>
<td>[33]</td>
<td>$26 \times 26 \times 26$</td>
<td>6144</td>
<td>GB, PK, LB</td>
<td>6055</td>
<td>25</td>
<td>512</td>
<td>98.55</td>
<td>242.2</td>
</tr>
<tr>
<td>$O_2$</td>
<td>Fig. 14</td>
<td>[24,33]</td>
<td>$26 \times 26 \times 26$</td>
<td>8192</td>
<td>GB, PK, LB</td>
<td>8074</td>
<td>25</td>
<td>768</td>
<td>98.56</td>
<td>322.96</td>
</tr>
<tr>
<td>$O_3$</td>
<td>Fig. 14</td>
<td>[33]</td>
<td>$26 \times 26 \times 26$</td>
<td>10240</td>
<td>GB, PK, LB</td>
<td>10065</td>
<td>25</td>
<td>768</td>
<td>98.29</td>
<td>402.6</td>
</tr>
<tr>
<td>$O_4$</td>
<td>Fig. 14</td>
<td>[24,47]</td>
<td>$40 \times 40 \times 29$</td>
<td>2050</td>
<td>GB, PK, LB</td>
<td>1995</td>
<td>28</td>
<td>310</td>
<td>97.32</td>
<td>71.25</td>
</tr>
<tr>
<td>$O_5$</td>
<td>Fig. 14</td>
<td>[24,27]</td>
<td>$23 \times 44 \times 23$</td>
<td>1550</td>
<td>GB, PK, LB</td>
<td>1490</td>
<td>12</td>
<td>145</td>
<td>96.13</td>
<td>124.17</td>
</tr>
<tr>
<td>$O_6$</td>
<td>Fig. 14</td>
<td>[24,26,27]</td>
<td>$24 \times 41 \times 15$</td>
<td>2079</td>
<td>GB, PK, LB</td>
<td>2017</td>
<td>22</td>
<td>270</td>
<td>97.02</td>
<td>91.68</td>
</tr>
</tbody>
</table>

*a*Figure in which the object is rendered.
*b*Papers in which we found such objects.
*c*Number of black points.
*d*Thinning algorithm.
*e*Number of deleted points.
*f*Number of the last deleting subiteration.
*g*Number of deleted points during the first subiteration.
*h*Rate of compression ($=100 \times (n.d.p./n.b.p.)$).
*i*Mean of the number of deleted points by subiteration ($=n.d.p./n.l.d.s.$).
Table 2
Results obtained by the three thinning algorithms on several synthetic 3D binary images

<table>
<thead>
<tr>
<th>Object</th>
<th>Fig.</th>
<th>Ref.</th>
<th>Size</th>
<th>n.b.p.</th>
<th>t.a.</th>
<th>n.d.p.</th>
<th>n.l.d.s.</th>
<th>n.d.p.f.s.</th>
<th>r.c. (%)</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_7$</td>
<td>Fig. 15</td>
<td>[26,27]</td>
<td>$24 \times 38 \times 16$</td>
<td>2560</td>
<td>GB</td>
<td>2503</td>
<td>27</td>
<td>220</td>
<td>97.77</td>
<td>92.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PK</td>
<td>2511</td>
<td>27</td>
<td>220</td>
<td>98.09</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LB</td>
<td>2511</td>
<td>27</td>
<td>220</td>
<td>98.09</td>
<td>93</td>
</tr>
<tr>
<td>$O_8$</td>
<td>Fig. 15</td>
<td>[5]</td>
<td>$27 \times 27 \times 27$</td>
<td>4299</td>
<td>GB</td>
<td>4203</td>
<td>51</td>
<td>373</td>
<td>97.77</td>
<td>82.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PK</td>
<td>4251</td>
<td>54</td>
<td>373</td>
<td>98.88</td>
<td>78.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LB</td>
<td>4248</td>
<td>45</td>
<td>373</td>
<td>98.81</td>
<td>94.40</td>
</tr>
</tbody>
</table>

a Figure in which the object is rendered.
b Papers in which we found such objects.
c Number of black points.
d Thinning algorithm.
e Number of deleted points.
f Number of the last deleting subiteration.
g Number of deleted points during the first subiteration.
h Rate of compression ($=100 \times (n.d.p./n.b.p.)$).
i Mean of the number of deleted points by subiteration ($=n.d.p./n.l.d.s.$).
Table 3
Results obtained by the three thinning algorithms on two 3D binary images of vertebra

<table>
<thead>
<tr>
<th>Object</th>
<th>Fig.⁵</th>
<th>Ref.⁶</th>
<th>Size</th>
<th>n.b.p.⁷</th>
<th>t.a.⁸</th>
<th>n.d.p.⁹</th>
<th>n.l.d.s.¹₀</th>
<th>n.d.p.f.s.¹¹</th>
<th>r.c. (%)¹²</th>
<th>Mean¹³</th>
</tr>
</thead>
<tbody>
<tr>
<td>C⁹</td>
<td>Fig. 16</td>
<td>[22]</td>
<td>256 × 256 × 51</td>
<td>179 546</td>
<td>GB</td>
<td>156 765</td>
<td>78</td>
<td>6151</td>
<td>87.31</td>
<td>2009.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PK</td>
<td>170 001</td>
<td>142</td>
<td>5855</td>
<td>94.68</td>
<td>1197.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LB</td>
<td>163 874</td>
<td>73</td>
<td>6 225</td>
<td>91.27</td>
<td>2244.85</td>
</tr>
<tr>
<td>C¹⁰</td>
<td>Fig. 17</td>
<td>New object</td>
<td>256 × 256 × 51</td>
<td>180 162</td>
<td>GB</td>
<td>167 268</td>
<td>95</td>
<td>5 818</td>
<td>92.84</td>
<td>1760.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PK</td>
<td>179 640</td>
<td>152</td>
<td>5 755</td>
<td>99.71</td>
<td>1181.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LB</td>
<td>179 675</td>
<td>122</td>
<td>5 890</td>
<td>99.73</td>
<td>1472.75</td>
</tr>
</tbody>
</table>

⁵Figure in which the object is rendered.
⁶Papers in which we found such objects.
⁷Number of black points.
⁸Thinning algorithm.
⁹Number of deleted points.
¹₀Number of the last deleting subiteration.
¹¹Number of deleted points during the first subiteration.
¹²Rate of compression (=100 × (n.d.p./n.b.p.)).
¹³Mean of the number of deleted points by subiteration (=n.d.p./n.l.d.s.).
algorithm needs more deletion subiterations than the two others to produce a skeleton (see Fig. 13(c)).

9. Conclusion

We have conceived a new 6-subiteration thinning algorithm, based on the deletion of \( P^x \)-simple points, by applying a recent methodology that we proposed in [21,22,24]. As it deletes solely \( P^x \)-simple points, this algorithm is guaranteed to preserve the topology. Furthermore, we have proposed various sets \( P \) such that our final algorithm deletes at least all the points removed by Palágyi and Kuba’s algorithm (PK), while preserving the same end points. This also implies that PK is guaranteed to preserve the topology. In addition, our final algorithm also deletes points removed by Gong and Bertrand’s algorithm [14], in the variant proposed by Rolland et al. [37], while preserving the same end points.

In another study [21,24], we succeeded in proposing a new 12-subiteration thinning algorithm for 3D binary images, which produces curve or surface skeletons, such that it deletes at least the points removed by one other 12-subiteration thinning algorithm [35]. Several remarks may be made on the study performed in this paper. The study made in this paper is easier than the one in which the used methodology has been introduced [21,24]. Indeed, only 6 templates (up to symmetries) along 6 directions have to be checked in this study during the examination of a configuration which corresponds to a non-\( P^x \)-simple point, in contrary to the study made in our previous works [21,24] for which 14 templates (up to symmetries) along 12 directions have to be checked. In addition, we obtain our final set \( P \) (in fact \( P_2 \), detailed in Section 7.2) “more easily” with two proposals in contrary to the three required in our previous study [21,24]. In this paper, we wholly use the formal description of \( P \) or \( P^x \) by a family of templates (see Section 7.2 and Fig. 12); until now the proposed families were reduced to a single template (up to symmetries) (see Section 7.1, the example in Section 4.2, and our previous study [21,24]).

Therefore, this methodology seems to be interesting to conceive algorithms “more powerful” than some others, in the sense that the new algorithms delete more points than the initial ones from which they are derived. Furthermore, our algorithms are guaranteed to preserve topology; and no proof is required in contrast to most of the already proposed thinning algorithms [26,27,31,33,35].

Our actual works consist in proposing new fully parallel thinning algorithms for 3D binary images, based on \( P \)-simple or \( P^x \)-simple points.

Acknowledgements

We are thankful to Laurent Perroton (assistant professor at ESIEE, Noisy-le-Grand) and to Julien Dehos (student at IUT, Le Puy-en-Velay) for rendering our results (resp. screenshots of synthetic images—Figs. 14 and 15, images of vertebra—Figs. 16 and 17).
References