Asymmetric parallel 3D thinning scheme and algorithms based on isthmuses

Michel Couprie, Gilles Bertrand

Asymmetric Parallel 3D Thinning Scheme and Algorithms Based on Isthmuses

Michel Couprie[1,**], Gilles Bertrand[1]

[a] *Université Paris-Est, LIGM, Équipe A3SI, ESIEE Paris, France*

ABSTRACT

Critical kernels constitute a general framework settled in the context of abstract complexes for the study of parallel thinning in any dimension. We take advantage of the properties of this framework, to propose a generic thinning scheme for obtaining "thin" skeletons from objects made of voxels. From this scheme, we derive algorithms that produce curve or surface skeletons, based on the notion of 1D or 2D isthmus. We compare our new curve thinning algorithm with all the published algorithms of the same kind, based on quantitative criteria. Our experiments show that our algorithm largely outperforms the other ones with respect to noise sensitivity. Furthermore, we show how to slightly modify our algorithms to include a filtering parameter that controls effectively the pruning of skeletons, based on the notion of isthmus persistence.

1. Introduction

The usefulness of skeletons in many applications of pattern recognition, computer vision, shape understanding etc. is mostly due to their property of topology preservation, and preservation of meaningful geometrical features. Here, we are interested in the skeletonization of objects that are made of voxels (unit cubes) in a regular 3D grid, *i.e.*, in a binary 3D image. In this context, topology preservation is usually obtained through the iteration of thinning steps, provided that each step does not alter the topological characteristics. In sequential thinning algorithms, each step consists of detecting and choosing a so-called simple voxel, that may be characterized locally (see Kong and Rosenfeld (1989); Saha et al. (1994); Couprie and Bertrand (2009)), and removing it. Such a process usually involves many choices, and the final result may depend, sometimes heavily, on any of these choices. This is why parallel thinning algorithms are generally preferred to sequential ones. However, removing a set of simple voxels at each thinning step, in parallel, may alter topology. The framework of critical kernels, introduced by one of the authors in Bertrand (2007), provides a condition under which we have the guarantee that a subset of voxels can be removed without changing topology. This condition is, to our knowledge, the most general one among the related works. Furthermore, critical kernels indeed provide a method to design new parallel thinning algorithms, in which the property of topology preservation is built-in, and in which any kind of constraint may be imposed (see Bertrand and Couprie (2008, 2014)).

Among the different parallel thinning algorithms that have been proposed in the literature, we can distinguish between symmetric and asymmetric algorithms. Symmetric algorithms (see Manzanera et al. (2002); Lohou and Bertrand (2007); Palágyi (2008)) produce skeletons that are invariant under 90 degrees rotations. They consist of the iteration of thinning steps that are made of 1) the identification and selection of a set of voxels that satisfy certain conditions, independently of orientation or position in space, and 2) the removal, in parallel, of all selected voxels from the object. Symmetric algorithms, on the positive side, produce a result that is uniquely defined: no choice is needed. On the negative side, they generally produce thick skeletons, see Fig. 1.

Asymmetric skeletons, on the opposite, are preferred when thinner skeletons are required. The price to pay is a certain amount of choices to be made. Most asymmetric parallel thinning algorithms fall into three main classes:
i) In the so-called directional algorithms (see Tsao and Fu (1981, 1982); Gong and Bertrand (1990); Palágyi and Kuba (1998); Palágyi and Kuba (1999a,b); Lohou and Bertrand (2004, 2005); Raynal and Couprie (2011); Németh et al. (2011); Németh and Palágyi (2012); Palágyi et al. (2012)), each thinning step is divided into a certain number of substeps, which

[**] Corresponding author: Tel.: +33 1 45 92 66 88; fax: +33 1 45 92 66 99;
e-mail: michel.couprie@esiee.fr (Michel Couprie)

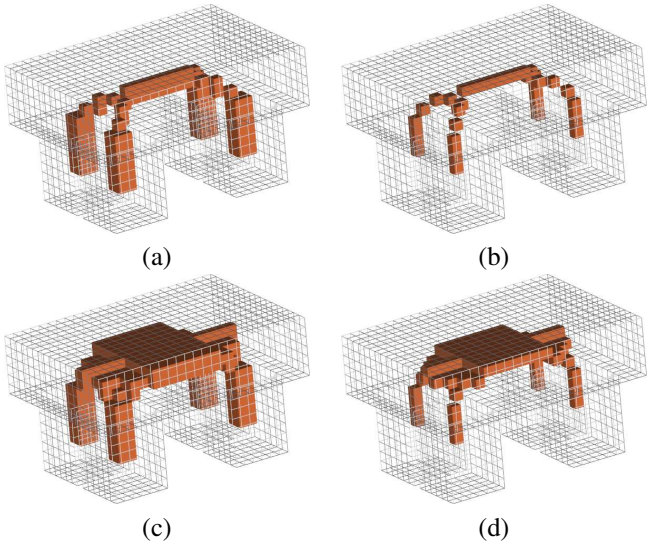

Fig. 1. Different types of skeletons. (a): Curve skeleton, symmetric. (b): Curve skeleton, asymmetric. (c): Surface skeleton, symmetric. (d): Surface skeleton, asymmetric.

are each devoted to the detection and the deletion of voxels belonging to one "side" of the object: all the voxels considered during the substep have, for example, their south neighbor inside the object and their north neighbor outside the object. The order in which these directional substeps are executed is set beforehand, arbitrarily.

ii) Subgrid (or subfield) algorithms (see Bertrand and Aktouf (1995); Saha et al. (1997); Ma et al. (2002a,b); Németh et al. (2010a,b); Németh and Palágyi (2012); Palágyi et al. (2012)) form a second category of asymmetric parallel thinning algorithms. There, each substep is devoted to the detection and the deletion of voxels that belong to a certain subgrid, for example, all voxels that have even coordinates. Considered subgrids must form a partition of the grid. Again, the order in which subgrids are considered is arbitrary. Subgrid algorithms are not often used in practice because they produce artifacts, that is, waving skeleton branches where the original object is smooth or straight.

iii) In a third class of algorithms, known as fully parallel algorithms (see Ma (1995); Ma and Sonka (1996); Németh and Palágyi (2012); Palágyi et al. (2012)), the thinning step is not divided into substeps, and the same detection condition is applied to all voxels in parallel. Notice that among those, Ma (1995) and Ma and Sonka (1996) do not preserve topology (see Lohou and Dehos (2010b,a)).

Most of these algorithms are implemented through sets of masks. A set of masks is used to characterize voxels that must be kept during a given step or substep, in order to 1) preserve topology, and 2) prevent curves or surfaces to disappear. Thus, topological conditions and geometrical conditions cannot be easily distinguished, and the slightest modification of any mask involves the need to make a new proof of the topological correctness.

Our approach is radically different. Instead of considering single voxels, we consider cliques. A clique is a set of mutually adjacent voxels. Then, we identify the critical kernel of the object, according to some definitions, which is a union of cliques. The main theorem of the critical kernels framework (see Bertrand (2007), see also Bertrand and Couprie (2014)) states that we can remove in parallel any subset of the object, provided that we keep at least one voxel of every clique that is part of the critical kernel, and this guarantees topology preservation. Here, as we try to obtain thin skeletons, our goal is to keep, whenever possible, exactly one voxel in every such clique. This leads us to propose a generic parallel asymmetric thinning scheme, that may be enriched by adding any sort of geometrical constraint. From our generic scheme, we easily derive, by adding such geometrical constraints, specific algorithms that produce curve or surface skeletons. To this aim, we define in this paper the notions of 1D and 2D isthmuses that permit to detect skeleton points that are important for shape reconstructibility: a 1D (resp. 2D) isthmus is a voxel whose neighborhood is "like a piece of curve" (resp. surface).

Our article is organized as follows. The first three sections contain a minimal set of basic notions about voxel complexes, simple voxels and critical kernels, respectively, which are necessary to make the article self-contained. In section 5, we introduce our new generic asymmetric thinning scheme, and we provide some examples of ultimate skeletons obtained by using it. Section 6 is devoted to introducing and illustrating our new isthmus-based parallel algorithms for computing curve and surface skeletons. Then in section 7, we describe the experiments that we made for comparing our curve thinning algorithm with all existing parallel curve thinning methods of the same kind. We show that our method ranks first in our quantitive evaluation. Finally, we show in section 8 how to use the notion of isthmus persistence in order to effectively filter the spurious skeleton parts due to noise. Persistence is a criterion, easy to compute in our framework, that allows us to dynamically detect or ignore certain isthmuses.

Note: A preliminary version of this work (up to Section 6) was published in the DGCI conference proceedings Couprie and Bertrand (2014). Sections 7 and 8 are new.

2. Voxel Complexes

In this section, we give some basic definitions for voxel complexes, see also Kovalevsky (1989); Kong and Rosenfeld (1989).

Let $\mathbb{Z}$ be the set of integers. We consider the families of sets $\mathbb{F}_0^1$, $\mathbb{F}_1^1$, such that $\mathbb{F}_0^1 = \{\{a\} \mid a \in \mathbb{Z}\}$, $\mathbb{F}_1^1 = \{\{a, a+1\} \mid a \in \mathbb{Z}\}$. A subset f of $\mathbb{Z}^n$, $n \geq 2$, that is the Cartesian product of exactly d elements of $\mathbb{F}_1^1$ and $(n-d)$ elements of $\mathbb{F}_0^1$ is called a *face* or an *d-face* of $\mathbb{Z}^n$, d is the *dimension of* f. In the illustrations of this paper, a 3-face (resp. 2-face, 1-face, 0-face) is depicted by a cube (resp. square, segment, dot), see *e.g.* Fig. 4.

A 3-face of $\mathbb{Z}^3$ is also called a *voxel*. A finite set that is composed solely of voxels is called a *(voxel) complex* (see Fig. 2). We denote by $\mathbb{V}^3$ the collection of all voxel complexes.

We say that two voxels x, y are *adjacent* if $x \cap y \neq \emptyset$. We write $\mathcal{N}(x)$ for the set of all voxels that are adjacent to a voxel x, $\mathcal{N}(x)$ is the *neighborhood of* x. Note that, for each voxel x, we have $x \in \mathcal{N}(x)$. We set $\mathcal{N}^*(x) = \mathcal{N}(x) \setminus \{x\}$.

Let $d \in \{0, 1, 2\}$. We say that two voxels x, y are *d-neighbors* if $x \cap y$ is a d-face. Thus, two distinct voxels x and y are adjacent if and only if they are d-neighbors for some $d \in \{0, 1, 2\}$.

Let $X \in \mathbb{V}^3$. We say that X is *connected* if, for any $x, y \in X$, there exists a sequence $\langle x_0, ..., x_k \rangle$ of voxels in X such that $x_0 = x$, $x_k = y$, and x_i is adjacent to x_{i-1}, $i = 1, ..., k$.

3. Simple Voxels

Intuitively a voxel x of a complex X is called a simple voxel if its removal from X "does not change the topology of X". This notion may be formalized with the help of the following recursive definition introduced in Bertrand and Couprie (2014), see also Kong (1997); Bertrand (1999) for other recursive approaches for simplicity.

Definition 1. Let $X \in \mathbb{V}^3$.
We say that X is *reducible* if either:
i) X is composed of a single voxel; or
ii) there exists $x \in X$ such that $\mathcal{N}^*(x) \cap X$ is reducible and $X \setminus \{x\}$ is reducible.

Definition 2. Let $X \in \mathbb{V}^3$. A voxel $x \in X$ is *simple for* X if $\mathcal{N}^*(x) \cap X$ is reducible. If $x \in X$ is simple for X, we say that $X \setminus \{x\}$ is an *elementary thinning of* X.

Thus, a complex $X \in \mathbb{V}^3$ is reducible if and only if it is possible to reduce X to a single voxel by iteratively removing simple voxels. Observe that a reducible complex is necessarily non-empty and connected.

In Fig. 2 (a), the voxel a is simple for X ($\mathcal{N}^*(a) \cap X$ is made of a single voxel), the voxel d is not simple for X ($\mathcal{N}^*(d) \cap X$ is not connected), the voxel h is simple for X ($\mathcal{N}^*(h) \cap X$ is made of two voxels that are 2-neighbors and is reducible).

In Bertrand and Couprie (2014), it was shown that the above definition of a simple voxel is equivalent to classical characterizations based on connectivity properties of the voxel's neighborhood Bertrand and Malandain (1994); Bertrand (1994); Saha et al. (1994); Kong (1995); Couprie and Bertrand (2009). An equivalence was also established with a definition based on the operation of collapse Whitehead (1939); Giblin (1981), this operation is a discrete analogue of a continuous deformation (a homotopy), see also Kong (1997); Bertrand (2007); Couprie and Bertrand (2009).

The notion of a simple voxel allows one to define thinnings of a complex, see an illustration Fig. 2 (b).

Let $X, Y \in \mathbb{V}^3$. We say that Y *is a thinning of* X or that X is *reducible to* Y, if there exists a sequence $\langle X_0, ..., X_k \rangle$ such that $X_0 = X$, $X_k = Y$, and X_i is an elementary thinning of X_{i-1}, $i = 1, ..., k$.

Thus, a complex X is reducible if and only if it is reducible to a single voxel.

4. Critical Kernels

Let X be a complex in $\mathbb{V}^3$. It is well known that, if we remove simultaneously (in parallel) simple voxels from X, we may "change the topology" of the original object X. For example, the two voxels f and g are simple for the object X depicted

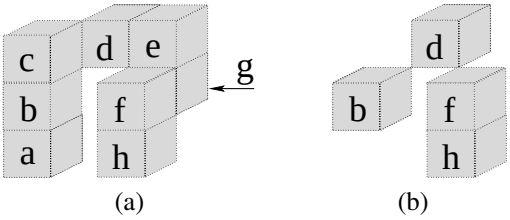

Fig. 2. (a) A complex X which is made of 8 voxels, (b) A complex $Y \subseteq X$, which is a thinning of X.

Fig. 2 (a). Nevertheless $X \setminus \{f, g\}$ has two connected components whereas X is connected.

In this section, we recall a framework for thinning in parallel discrete objects with the warranty that we do not alter the topology of these objects Bertrand (2007); Bertrand and Couprie (2008, 2014). This method is valid for complexes of arbitrary dimension.

Let $d \in \{0, 1, 2, 3\}$ and let $C \in \mathbb{V}^3$. We say that C is a *d-clique* or a *clique* if $\cap\{x \in C\}$ is a d-face. If C is a d-clique, d is the *rank of* C.

If C is made of solely two distinct voxels x and y, we note that C is a d-clique if and only if x and y are d-neighbors, with $d \in \{0, 1, 2\}$.

Let $X \in \mathbb{V}^3$ and let $C \subseteq X$ be a clique. We say that C is *essential for* X if we have $C = D$ whenever D is a clique such that:
i) $C \subseteq D \subseteq X$; and
ii) $\cap\{x \in C\} = \cap\{x \in D\}$.

Observe that any complex C that is made of a single voxel is a clique (a 3-clique). Furthermore any voxel of a complex X constitutes a clique that is essential for X.

In Fig. 2 (a), $\{f, g\}$ is a 2-clique that is essential for X, $\{b, d\}$ is a 0-clique that is not essential for X, $\{b, c, d\}$ is a 0-clique essential for X, $\{e, f, g\}$ is a 1-clique essential for X.

Definition 3. Let $S \in \mathbb{V}^3$. The *$\mathcal{K}$-neighborhood of* S, written $\mathcal{K}(S)$, is the set made of all voxels that are adjacent to each voxel in S. We set $\mathcal{K}^*(S) = \mathcal{K}(S) \setminus S$.

We note that we have $\mathcal{K}(S) = \mathcal{N}(x)$ whenever S is made of a single voxel x. We also observe that we have $S \subseteq \mathcal{K}(S)$ whenever S is a clique.

Definition 4. Let $X \in \mathbb{V}^3$ and let C be a clique that is essential for X. We say that the clique C is *regular for* X if $\mathcal{K}^*(C) \cap X$ is reducible. We say that C is *critical for* X if C is not regular for X.

Thus, if C is a clique that is made of a single voxel x, then C is regular for X if and only if x is simple for X.

In Fig. 2 (a), the cliques $C_1 = \{b, c, d\}$, $C_2 = \{f, g\}$, and $C_3 = \{f, h\}$ are essential for X. We have $\mathcal{K}^*(C_1) \cap X = \emptyset$, $\mathcal{K}^*(C_2) \cap X = \{d, e, h\}$, and $\mathcal{K}^*(C_3) \cap X = \{g\}$. Thus, C_1 and C_2 are critical for X, while C_3 is regular for X.

The following result is a consequence of a general theorem that holds for complexes of arbitrary dimension Bertrand (2007); Bertrand and Couprie (2014).

Theorem 5. *Let $X \in \mathbb{V}^3$ and let $Y \subseteq X$.*
The complex Y is a thinning of X if any clique that is critical for X contains at least one voxel of Y.

See an illustration in Fig. 2(a) and (b) where the complexes X and Y satisfy the condition of theorem 5. For example, the voxel d is a non-simple voxel for X, thus $\{d\}$ is a critical 3-clique for X, and d belongs to Y. Also, Y contains voxels in the critical cliques $C_1 = \{b, c, d\}$, $C_2 = \{f, g\}$, and the other ones.

5. A generic 3D parallel and asymmetric thinning scheme

Our goal is to define a subset Y of a voxel complex X that is guaranteed to include at least one voxel of each clique that is critical for X. By theorem 5, this subset Y will be a thinning of X.

Let us consider the complex X depicted Fig. 3 (a). There are precisely three cliques that are critical for X:
- the 0-clique $C_1 = \{b, c\}$ (we have $\mathcal{K}^*(C_1) \cap X = \emptyset$);
- the 2-clique $C_2 = \{a, b\}$ (we have $\mathcal{K}^*(C_2) \cap X = \emptyset$);
- the 3-clique $C_3 = \{b\}$ (the voxel b is not simple).

Suppose that, in order to build a complex Y that fulfills the condition of theorem 5, we select arbitrarily one voxel of each clique that is critical for X. Following such a strategy, we could select c for C_1, a for C_2, and b for C_3. Thus, we would have $Y = X$, no voxel would be removed from X. Now, we observe that the complex $Y' = \{b\}$ satisfies the condition of theorem 5. This complex is obtained by considering first the 3-cliques before selecting a voxel in the 2-, 1-, or 0 cliques.

The complex X of Fig. 3 (b) provides another example of such a situation. There are precisely three cliques that are critical for X:
- the 1-clique $C_1 = \{e, f, g, h\}$ (we have $\mathcal{K}^*(C_1) \cap X = \emptyset$);
- the 1-clique $C_2 = \{e, d, g\}$ (we have $\mathcal{K}^*(C_2) \cap X = \emptyset$);
- the 2-clique $C_3 = \{e, g\}$ ($\mathcal{K}^*(C_3) \cap X$ is not connected).

If we select arbitrarily one voxel of each critical clique, we could obtain the complex $Y = \{f, d, g\}$. On the other hand, if we consider the 2-cliques before the 1-cliques, we obtain either $Y' = \{e\}$ or $Y'' = \{g\}$. In both cases the result is better in the sense that we remove more voxels from X.

This discussion motivates the introduction of the following 3D asymmetric and parallel thinning scheme `AsymThinningScheme` (see also Bertrand and Couprie (2008, 2009, 2014) for other thinning schemes and properties of critical kernels). The main features of this scheme are the following:
- Taking into account the observations made through the two previous examples, critical cliques are considered according to their decreasing ranks (step 4). Thus, each iteration is made of four sub-iterations (steps 4-8). Voxels that have been previously selected are stored in a set Y (step 8). At a given sub-iteration, we consider voxels only in critical cliques included in $X \setminus Y$ (step 6).
- *Select* is a function from $\mathbb{V}^3$ to V^3, the set of all voxels. More precisely, *Select* associates, to each set S of voxels, a unique voxel x of S. We refer to such a function as a *selection function*. This function allows us to select a voxel in a given critical clique (step 7). A possible choice is to take for $Select(S)$, the first pixel of S in the lexicographic order of the voxels coordinates.
- In order to compute curve or surface skeletons, we have to keep other voxels than the ones that are necessary for the preservation of the topology of the object X. In the scheme, the set K corresponds to a set of features that we want to be preserved by a thinning algorithm (thus, we have $K \subseteq X$). This set K, called *constraint set*, is updated dynamically at step 10. $Skel_X$ is a function from X on $\{True, False\}$ that allows us to keep some *skeletal voxels* of X, *e.g.*, some voxels belonging to parts of X that are surfaces or curves. For example, if we want to obtain curve skeletons, a frequently employed solution is to set $Skel_X(x) = True$ whenever x is a so-called *end voxel* of X: an end voxel is a voxel that has exactly one neighbor inside X. Better propositions for such a function will be introduced in section 6.

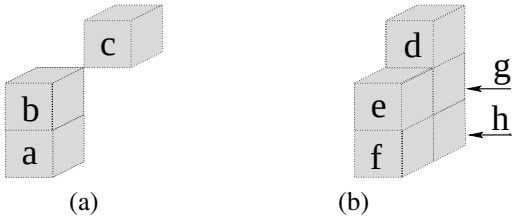

Fig. 3. Two complexes.

By construction, at each iteration, the complex Y at step 9 satisfies the condition of theorem 5. Thus, the result of the scheme is a thinning of the original complex X. Observe also that, except step 4, each step of the scheme may be computed in parallel.

Algorithm 1: `AsymThinningScheme`$(X, Skel_X)$

Data: $X \in \mathbb{V}^3$, $Skel_X$ is a function from X on $\{True, False\}$
Result: X

1 $K := \emptyset$;
2 **repeat**
3 $Y := K$;
4 **for** $d \leftarrow 3$ **downto** 0 **do**
5 $Z := \emptyset$;
6 **foreach** *d-clique* $C \subseteq X \setminus Y$ *that is critical for* X **do**
7 $Z := Z \cup \{Select(C)\}$;
8 $Y := Y \cup Z$;
9 $X := Y$;
10 **foreach** *voxel* $x \in X \setminus K$ *such that* $Skel_X(x) = True$ **do**
11 $K := K \cup \{x\}$;
12 **until** *stability* ;

Fig. 4 provides an illustration of the scheme `AsymThinningScheme`. Let us consider the complex X depicted in (a). We suppose in this example that we do not keep any skeletal voxel, *i.e.*, for any $x \in X$, we set $Skel_X(x) = False$. The traces of the cliques that are critical for X are represented in (b), the *trace of a clique* C is the face $f = \cap\{x \in C\}$. Thus, the set of the cliques that are critical for X is precisely composed of six 0-cliques, two 1-cliques, three 2-cliques, and one 3-clique. In (c) the four different sub-iterations of the first iteration of the scheme are illustrated (steps 4-8):
- when $d = 3$, only one clique is considered, the dark grey

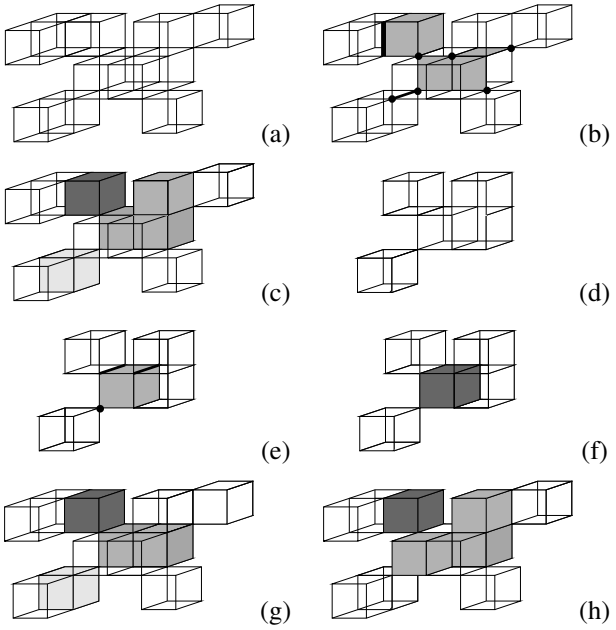

Fig. 4. (a): A complex X made of 12 voxels. (b): The traces of the cliques that are critical for X. (c): Voxels that have been selected by the algorithm. (d): The result Y of the first iteration. (e): The traces of the 4 cliques that are critical for Y. (f): The result of the second iteration. (g) and (h): Two other possible selections at the first iteration.

voxel is selected whatever the selection function;
- when $d = 2$, all the three 2-cliques are considered since none of these cliques contains the above voxel. Voxels that could be selected by a selection function are depicted in medium grey;
- when $d = 1$, only one clique is considered, a voxel that could be selected is depicted in light grey;
- when $d = 0$, no clique is considered since each of the critical 0-cliques contains at least one voxel that has been previously selected.
After these sub-iterations, we obtain the complex depicted in (d). The figures (e) and (f) illustrate the second iteration, at the end of this iteration the complex is reduced to a single voxel. In (g) and (h) two other possible selections at the first iteration are given.

Of course, the result of the scheme may depend on the choice of the selection function. This is the price to be paid if we try to obtain thin skeletons. For example, some choices have to be made for reducing a two voxels wide ribbon to a simple curve.

Fig. 5 shows another illustration, on bigger objects, of `AsymThinningScheme`. Here also, for any $x \in X$, we have $Skel_X(x) = False$ (no skeletal voxel). The result is called an ultimate asymmetric skeleton.

6. Isthmus-based asymmetric thinning

In this section, we show how to use our generic scheme `AsymThinningScheme` in order to get a procedure that computes either curve or surface skeletons. This thinning procedure preserves a constraint set K that is made of "isthmuses".

Intuitively, a voxel x of an object X is said to be a 1-isthmus (resp. a 2-isthmus) if the neighborhood of x corresponds - up to a thinning - to the one of a point belonging to a curve (resp. a surface) Bertrand and Couprie (2014).

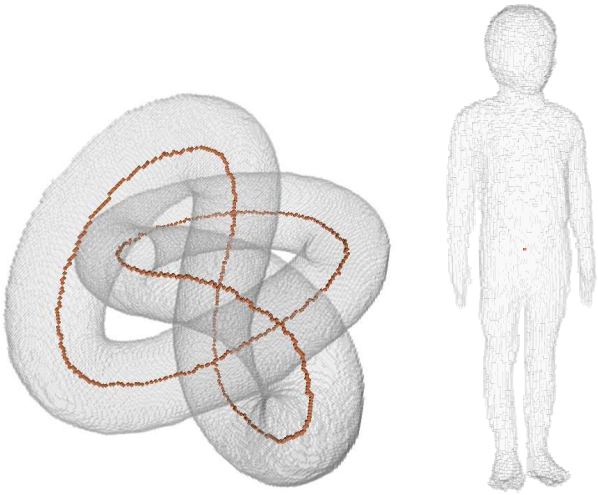

Fig. 5. Ultimate asymmetric skeletons obtained by using `AsymThinningScheme`. On the left, the object (635803 voxels) is a solid cylinder bent to form a knot. Its ultimate skeleton is a discrete curve. On the right, the object (123935 voxels) is connected and without holes and cavities. Its ultimate skeleton is a single voxel.

We say that $X \in \mathbb{V}^3$ is a 0-*surface* if X is precisely made of two voxels x and y such that $x \cap y = \emptyset$.

We say that $X \in \mathbb{V}^3$ is a 1-*surface* (or a *simple closed curve*) if:
i) X is connected; and ii) For each $x \in X$, $\mathcal{N}^*(x) \cap X$ is a 0-surface.

Definition 6. Let $X \in \mathbb{V}^3$, let $x \in X$.
We say that x is a 1-*isthmus for* X if $\mathcal{N}^*(x) \cap X$ is reducible to a 0-surface.
We say that x is a 2-*isthmus for* X if $\mathcal{N}^*(x) \cap X$ is reducible to a 1-surface.
We say that x is a 2^+-*isthmus for* X if x is a 1-isthmus or a 2-isthmus for X.

See Fig. 6 for an illustration of the notion of k-isthmus.

Our aim is to thin an object, while preserving a constraint set K that is made of voxels that are detected as k-isthmuses during the thinning process. We obtain curve skeletons with $k = 1$, and surface skeletons with $k = 2^+$. These two kinds of skeletons may be obtained by using `AsymThinningScheme`, with the function $Skel_X$ defined as follows:

$$Skel_X(x) = \begin{cases} True & \text{if } x \text{ is a } k\text{-isthmus for } X, \\ False & \text{otherwise,} \end{cases}$$

with k being set to 1 or 2^+.

Observe that there is the possibility that a voxel belongs to a k-isthmus at a given step of the algorithm, but not at further steps. This is why previously detected isthmuses are stored (see lines 10-11 of `AsymThinningScheme`).

In Fig. 7, we show a curve skeleton and a surface skeleton obtained by our method from the same object.

A key point, in the implementation of the algorithms proposed in this paper, is the detection of critical cliques and isthmus voxels. In Bertrand and Couprie (2014), we showed that

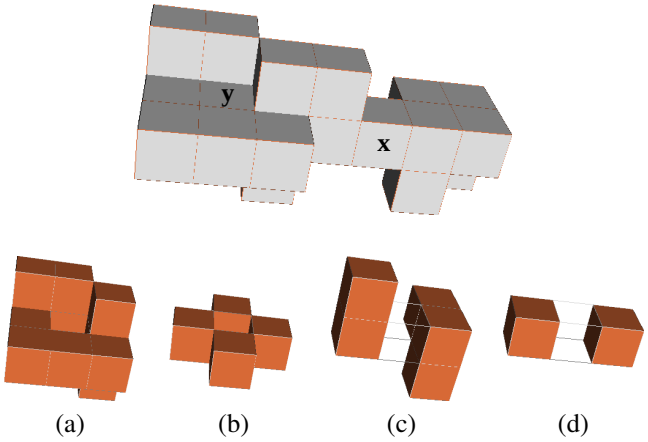

Fig. 6. Top: a voxel complex X. (a): the set $\mathcal{N}^*(y) \cap X$. (b): a 1-surface that is a thinning of $\mathcal{N}^*(y) \cap X$. Hence, y is a 2-isthmus. (c): the set $\mathcal{N}^*(x) \cap X$. (d): a 0-surface that is a thinning of $\mathcal{N}^*(x) \cap X$. Hence, x is a 1-isthmus.

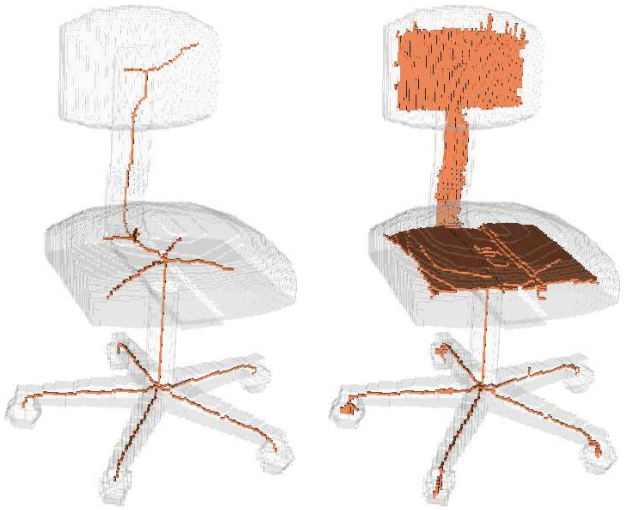

Fig. 7. Asymmetric skeletons obtained by using `AsymThinningScheme`. Left: curve skeleton. The function $Skel_X$ is based on 1-isthmuses. Right: surface skeleton. The function $Skel_X$ is based on 2^+-isthmuses. Of course, these skeletons need some filtering, see section 8 and Fig. 12.

it is possible to detect critical cliques thanks to a set of masks, in linear time. Note also that the configurations of 1D and 2D isthmuses may be pre-computed by a linear-time algorithm and stored in lookup tables. Finally, based on a breadth-first strategy, the whole method can be implemented to run in $O(n)$ time, where n is the number of voxels of the input 3D image.

7. Experiments, results and discussion

In the experiments described below, due to space limitations, we consider only parallel asymmetric thinning methods that produce curve skeletons of voxel objects, and that have no parameter. In particular, we do not consider the variants of the algorithms of Németh et al. (2010b) that involve the checking of extremity voxel neighborhoods of increasing size, as this neighborhood size is indeed a parameter.

Skeletons are notoriously sensitive to noise, and this is major problem for many applications. Even in the continuous case, the slightest perturbation of a smooth contour shape may provoke the appearance of an arbitrarily long skeleton branch, that we will refer to as a spurious branch. A desirable property of discrete skeletonization methods is to generate as few spurious branches as possible, in response to the so-called discretization (or voxelization) noise that is inherent to any discretization process. We will compare the different methods with respect to their ability to produce skeletons that are free of spurious branches. In the following, we compare how different methods behave with respect to this property.

In order to get ground truth skeletons, we discretized a set of six simple 3D shapes for which the skeletons are known: a Euclidean ball, a torus, a thickened straight segment, a thickened spiral, and two bent cylinders with no ends, see *e.g.* the one of Fig. 8. For the latter shape, a discrete curve skeleton should ideally be a simple closed discrete curve. Any extra branch of the skeleton must undoubtedly be considered as spurious. Thus, a simple and effective criterion for assessing the quality of a skeletonization method is to count the number of extra branches, or equivalently in our case, the number of extra curve extremities.

In addition, we used a database of 30 three-dimensional "real world" voxel objects. These objects were obtained by converting into voxel sets some 3D models freely available on the internet (mainly from the NTU 3D database, see `http://3d.csie.ntu.edu.tw/~dynamic/benchmark`). Our test set can be downloaded at `http://www.esiee.fr/~info/ck/3DSkAsymTestSet.tgz`. We chose these objects because they all may be well described by a curve skeleton, the branches of which can be intuitively related to object parts (for example, the skeleton of a coarse human body has typically 5 branches, one for the head and one for each limb). For each object, we manually indicated an "ideal" number of branches, having in mind an application of the type shape matching/pattern recognition.

In order to compare methods, we mainly use the indicator $S(X, M) = |c(X, M) - c_i(X)|$, where $c(X, M)$ stands for the number of curve extremities for the result obtained from X after application of method M, and $c_i(X)$ stands for the ideal number of curve extremities to expect with the object X. In other words, $S(X, M)$ counts the number of spurious branches produced by method M for object X, a result of 0 being the best one. Note that, for all objects in our database and all tested methods, the difference was positive, in other words all the methods produced more skeleton branches than expected, or just the right number. We define $S(M)$ as the average, for all objects of the database, of $S(X, M)$. We call $S(M)$ the *spuriousness factor* of method M.

Another useful indicator is the reconstruction error, which can be measured as the mean distance between the original object X and the reconstruction from A in X (union of open balls using the voxels of A as centers and the values of the distance map of X as radii), where A is the skeleton computed from X.

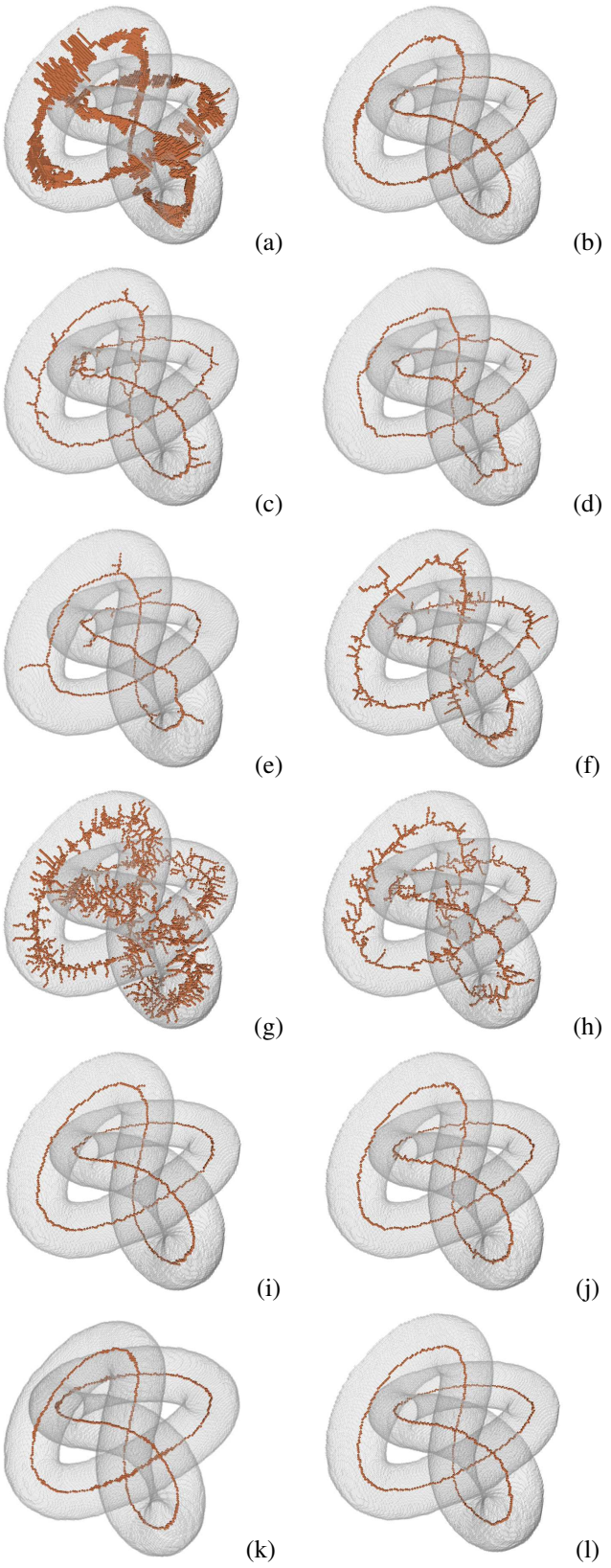

Fig. 8. Curve skeletons of a same object obtained through different methods: (a) Tsao and Fu (1981), (b) Tsao and Fu (1982), (c) Palágyi and Kuba (1998), (d) Palágyi and Kuba (1999a), (e) Palágyi and Kuba (1999b), (f) Ma and Wan (2000), (g) Ma et al. (2002b), (h) Ma et al. (2002a), (i) Lohou and Bertrand (2005), (j) Németh et al. (2010b), 8 subgrids, (k) Németh et al. (2011), (l) Our new method based on isthmuses.

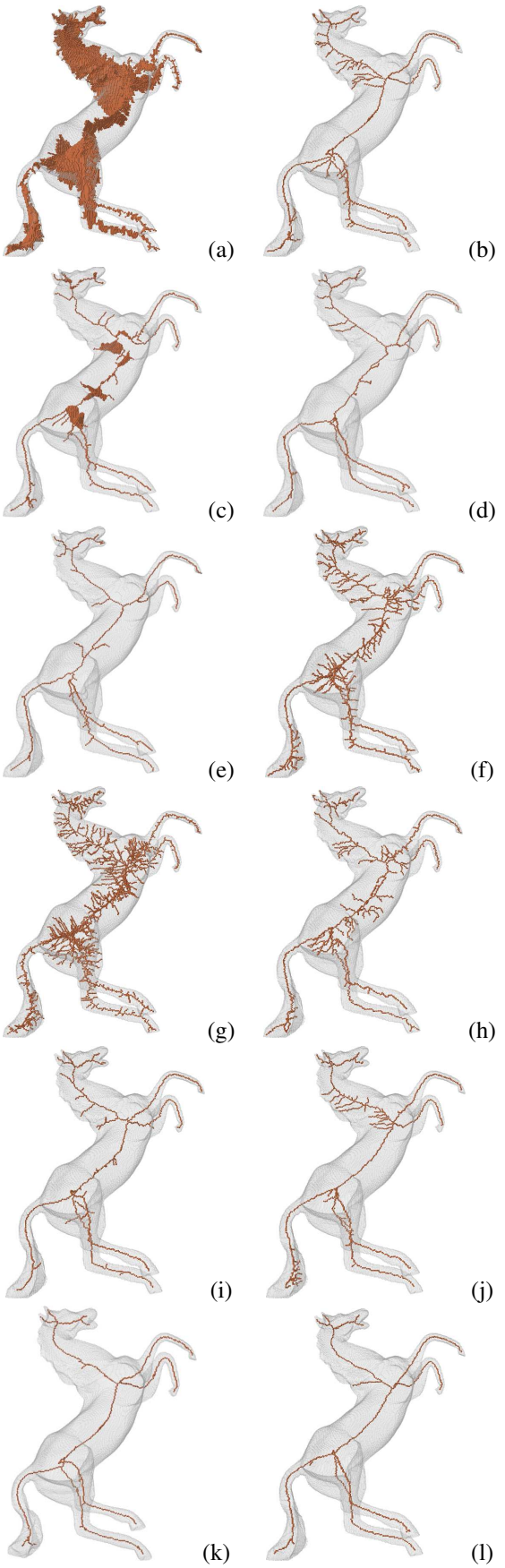

Fig. 9. Idem Fig. 8.

Formally, given two voxel sets X, Y and a voxel x, we define $D(x, Y) = \min\{d(x, y) \mid y \in Y\}$, where $d(x, y)$ stands for the Euclidean distance between x and y. The *distance map of* X is the map $\mathcal{D}_X$ on X defined by $\mathcal{D}_X(x) = D(x, \mathbb{V}^3 \setminus X)$. We set $D_X(Y) = \frac{\sum_{x \in X} D(x,Y)}{|X|}$, and $D(X, Y) = \max\{D_X(Y), D_Y(X)\}$ is the *mean distance between sets X and Y*. We define $R(X, M) = D(X, Re(A, X))$, where $A = Sk(X, M)$ is the skeleton obtained from object X using method M, and $Re(A, X)$ stands for the reconstruction from A in X. Note that $Re(A, X) \subseteq X$ always holds. A perfect reconstruction yields $R(X, M) = 0$. We define the *reconstruction error* $R(M)$ as the average, for all objects of the database, of $R(X, M)$. Of course, there is a trade-off between indicators R and S, as a noisy skeleton with many spurious branches will likely yield a low reconstruction error. But for methods with comparable spuriousness factors, a lower reconstruction error indicates a better quality (better centering and/or longer skeleton branches).

The goal of asymmetric thinning is to provide "thin" skeletons. This means in particular that the resulting skeletons should contain no simple voxel, apart from the curve extremities. However, due to their parallel nature, most thinning algorithms considered in this study may leave some extra simple voxels. We define our third indicator as $T(X, M) = 100 \times \frac{|Si(Sk(X,M))|}{|Sk(X,M)|}$, where $Si(A)$ denotes the set of simple voxels of A that are not curve extremities. We define the *thickness factor* $T(M)$ as the average, for all objects of the database, of $T(X, M)$. The lower the value of $T(M)$, the better the method M with respect to thinness.

This evaluation is limited to one particular class of algorithms, and we chose criteria that help us to dicriminate between the methods of this class. The interested reader may find a comprehensive set of criteria in Cornea and Silver (2007). Among those, we do not mention homotopy in this evaluation because all the presented algorithms preserve the homotopy type, and we omit computational complexity because they can all be implemented to run in $O(n)$ time. Note also that all the tested methods can be easily implemented, most of the time with a set of masks. For all of them, computing times can be enhanced by the use of lookup tables. On another hand, criteria like centeredness and rotational invariance are not of primary interest for this comparison, as users that are interested by these property before others would more likely choose other methods, *e.g.* based on Euclidean distance or Voronoi diagrams, which perform much better from this point of view.

First of all, it is interesting to look at the results of different methods for a same object (see Fig. 8 and Fig. 9). For the sake of space and readability, we selected only 12 methods among the 28 that took place in our experiments, see table 1 for the complete quantitative results. We notice in particular that some methods, like Tsao and Fu (1981) and Palágyi and Kuba (1998), are not sufficiently powerful to produce results that may be interpreted as curve skeletons (see also the thickness factor in table 1).

This illustrates the difficulty of designing a method that keeps enough voxels in order to preserve topology, and in the same time, deletes a sufficient number of voxels in order to produce thin curve skeletons. This difficulty is indeed high when these two opposite constraints are not clearly distinguished. One strength of our approach lies in a complete separation of these constraints.

Table 1 gathers the quantitative results of our experiments, that allows us to compare the 27 other existing methods of the same class with our algorithm. We see that our method outperforms all existing methods with respect to the spuriousness factor $S(M)$ on "natural" shapes, and hits the best possible score (0) on artificial ones. On artificial shapes, the only other method that produces no spurious branch is method 19, but we see that this method does not produce thin skeletons for these shapes. On natural shapes, compared with the best methods after ours with respect to $S(M)$, namely methods 8 and 9, our algorithm has also a lower thickness factor $T(M)$ and a lower reconstruction error $R(M)$.

We conclude this section by showing, in Fig. 10, five curve skeletons obtained with our method on shapes from our test database.

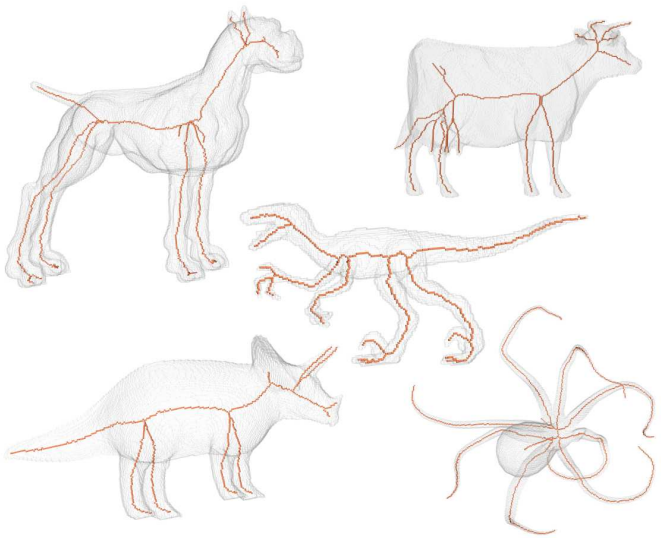

Fig. 10. Curve skeletons computed by `AsymThinningScheme`.

8. Isthmus persistence and skeleton filtering

It is well known that the skeletonization process is highly sensitive to noise, and this is a major issue in practical applications. The origin of this problem lies in the following fact: the transformation that associates its skeleton to a shape is not continuous. In practice, it means that if a small perturbation is applied on the contour of an object, then a big skeleton part may appear or disappear. See for example Attali et al. (2009) for a survey of selected studies on the stability of skeletons.

In consequence, many authors have proposed methods that aim at eliminating, or "pruning", spurious skeleton branches or parts. These methods are essentially based on a criterion that permits to distinguish between points or parts of the skeleton, those that are due to noise from those that are robust to small perturbations.

Table 1. Results of our quantitative comparison (see text). The term "dir" indicates a directional algorithm, "sgr" a subgrid algorithm, "fp" a fully parallel algorithm, and "other" an algorithm that falls in none of these classes. The measures $S_i(M)$, $R_i(M)$, $T_i(M)$ were obtained either from artificial shapes with known skeletons ($i = 1$), or from "natural" shapes ($i = 2$).

Method M	$S_1(M)$	$T_1(M)$	$R_1(M)$	$S_2(M)$	$T_2(M)$	$R_2(M)$
1. Tsao and Fu (1981), 6 dir	139.5	24.8	0.02	177.2	29.1	0.15
2. Tsao and Fu (1982), 6 dir	13.2	26.3	0.08	37.0	5.7	0.58
3. Gong and Bertrand (1990), 6 dir	321.5	22.5	0.01	134.1	28.1	0.12
4. Bertrand and Aktouf (1995), 8 sgr	4.3	0.27	0.12	15.6	0.13	0.82
5. Saha et al. (1997), 8 sgr	86.8	0.43	0.11	117.4	0.29	0.46
6. Palágyi and Kuba (1998), 6 dir	31.5	2.3	0.03	43.2	2.0	0.43
7. Palágyi and Kuba (1998), other	23.8	3.6	0.06	25.7	8.3	0.48
8. Palágyi and Kuba (1999a), 8 dir	3.8	0	1.7	8.97	0.23	1.50
9. Palágyi and Kuba (1999b), 12 dir	3.8	0.15	6.5	9.2	0.72	2.72
10. Ma and Wan (2000), 6 dir	78.2	6.0	0.03	115.9	10.4	0.30
11. Ma et al. (2002b), 4 sgr	349.5	0.24	0.03	380.1	0.18	0.17
12. Ma et al. (2002a), 2 sgr	53.7	0.01	0.07	51.5	0.43	0.42
13. Lohou and Bertrand (2004), 12 dir	19.5	0	3.3	21.0	0.13	1.94
14. Lohou and Bertrand (2005), 6 dir	2.2	0	0.08	11.3	0.003	0.96
15. Németh et al. (2010a), 2 sgr	722.3	42.2	0.004	67.9	38.3	0.11
16. Németh et al. (2010b), 4 sgr	8.0	0	0.08	38.2	0	0.63
17. Németh et al. (2010b), 8 sgr	8.2	0	0.07	31.7	0	0.65
18. Lohou and Dehos (2010a), other	4.3	11.3	3.14	16.6	8.1	1.81
19. Németh et al. (2011), 6 dir	0	28.7	0.11	10.1	5.6	0.82
20. Raynal and Couprie (2011), 6 dir	10.8	0.74	0.04	12.9	1.6	0.71
21. Palágyi et al. (2012), fp	11.3	0	0.14	19.9	0	0.87
22. Palágyi et al. (2012), 6 dir	17.8	0	0.20	36.8	0	0.72
23. Palágyi et al. (2012), 8 sgr	11.3	0	0.09	44.7	0	0.63
24. Németh and Palágyi (2012), fp	10.0	0.20	0.16	17.7	0.18	0.88
25. Németh and Palágyi (2012), 6 dir	15.3	0.52	0.21	32.0	0.33	0.74
26. Németh and Palágyi (2012), 8 sgr	9.0	0.30	0.09	39.8	0.25	0.64
27. Palágyi (2013), 12 dir	15.3	15.8	1.8	27.33	16.6	1.41
28. `AsymThinningScheme` (our method)	0	0	0.16	5.5	0.05	1.08

Among the different criteria that were proposed in the literature, the notion of isthmus persistence introduced in Liu et al. (2010) (see also Chaussard (2010)) yields a simple yet efficient method to filter skeletons during the thinning process. Originally, this method has been formulated in the framework of 3D cubical complexes, *i.e.*, objects made of faces of different dimensions. In this section, we show that it can be adapted to the context of voxel complexes.

Let x be a voxel in a voxel complex X, that becomes an isthmus for the first time at step i of the parallel thinning. Then, we define the *birth date* of x, denoted by $b(x)$, as $b(x) = i$. Intuitively, $b(x)$ corresponds to the local thickness of the object around the voxel x, see Fig. 11 for an illustration in 2D.

Now, consider an isthmus voxel x that becomes, at step j of the parallel thinning process, a deletable voxel. Then, we define the *death date* of x, denoted by $d(x)$, as $d(x) = j$.

Finally, we define the *persistence* of the voxel x as the difference between the death date ane the birth date, that is, $d(x) - b(x)$. It may be seen that a voxel with a high persistence value is likely to belong to a robust skeleton part, whereas a low persistence characterizes a voxel in a spurious skeleton part (Fig. 11). Therefore, skeleton filtering may be performed by keeping in the constraint set of the thinning algorithm, only the isthmuses that have a persistence greater than a given threshold.

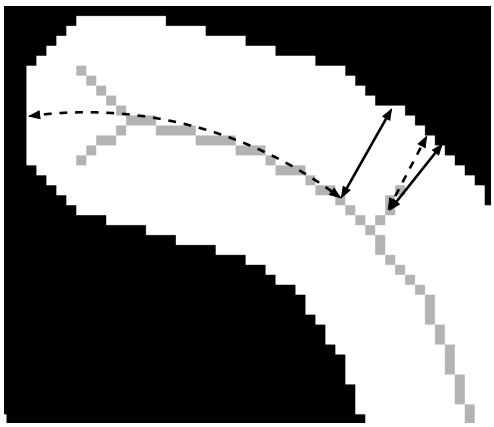

Fig. 11. The lengths depicted with a solid line correspond to the birth dates, the dotted lines to the death dates.

Intuitively, the persistence may be interpreted as a relative measure of the "elongation" of a certain object part that relates

to a given skeleton point. In some methods, like Ogniewicz and Kübler (1995); Reniers et al. (2008); Couprie (2013), the filtering is also based on a kind of measure of elongation, that is the length or area of the boundary portion that "collapses" onto the given point. The latter measure can be considered as a global one, whereas the persistence combines global information (the death date that relates to "pure elongation") and more local information (the birth date that relates to width or thickness). By setting the parameter p, the user decides how much the elongation dimension of an object part must exceed its width, in order to be preserved.

In the following algorithm, k stands for the dimension of the considered isthmuses (1 or 2^+), and p is a parameter that sets the persistence threshold. The function b associates to certain voxels their birth date, and K is a constraint set that is dynamically updated by adding those voxels whose persistence is greater than the threshold p (lines 12-13).

Algorithm 2: `PersistenceAsymThinning`(X, k, p)

```
   Data   : X ∈ 𝕍³, k ∈ {1, 2⁺}, p ∈ ℕ ∪ {+∞}
   Result : X
1  i := 0; K := ∅; foreach x ∈ X do b(x) := 0;
2  repeat
3      i := i + 1;
4      Y := K;
5      for d ← 3 downto 0 do
6          Z := ∅;
7          foreach d-clique C ⊆ X \ Y that is critical for X do
8              Z := Z ∪ {Select(C)};
9          Y := Y ∪ Z;
10     W := {x ∈ X \ K | x is a k-isthmus for X};
11     foreach x ∈ W such that b(x) = 0 do b(x) := i;
12     W' := {x ∈ Y | b(x) > 0 and i + 1 − b(x) ⩾ p};
13     X := Y; K := K ∪ W';
14 until stability ;
```

In line 11, the birth date $b(x)$ of each new isthmus voxel x is recorded. In line 12, the test $b(x) > 0$ implies that the considered voxel x has been recorded as an isthmus voxel. Furthermore, since this voxel x belongs to Y, it is not deletable, thus its death date $d(x)$ is strictly greater than i. The condition $i + 1 - b(x) \geqslant p$ thus implies $d(x) - b(x) \geqslant p$, meaning that the voxel x must be added to the constraint set K (see line 13) because its persistence is greater than p.

Extreme cases for the values of the parameter p are $p = 1$ and $p = +\infty$. Notice that, by the very definitions of isthmus and persistence, the persistence of any isthmus is at least one (since an isthmus is not deletable). If $p = 1$, then all detected isthmuses are added to the constraint set. In this case, we retrieve the behaviour of algorithm `IsthmusAsymThinnning`. If $p = +\infty$, then no voxel is added to the constraint set. In this case, the result is an ultimate asymmetric skeleton of X.

Fig. 13 illustrates the usefulness and the effectiveness of persistence-based filtering. Fig. 13(a) shows a 3D shape and its skeleton obtained by using `AsymThinningScheme`. In Fig. 13(b), we added some random noise to the shape contour.

Fig. 12. Filtered skeletons of the same object as in Fig. 7. Left: curve skeleton, $p = 3$. Right: surface skeleton, $p = 2$.

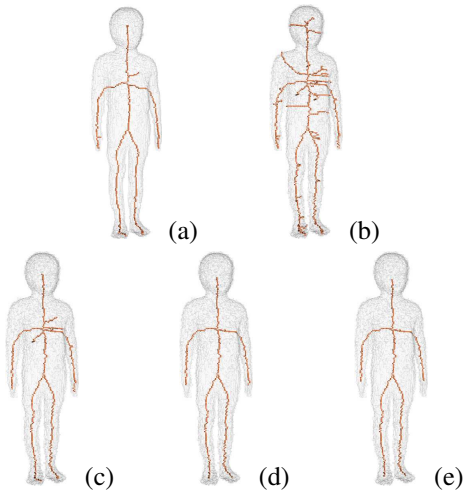

Fig. 13. (a) Original shape and its curve skeleton obtained by using `AsymThinningScheme`. (b) Noisy shape and its curve skeleton. (c,d,e) Filtered skeletons of the noisy shape, obtained by using `PersistenceAsymThinning`, with parameter values $2, 5, 8$ respectively.

We clearly see that, for noisy objects, some filtering is mandatory. We obtain satisfactory results with values of p greater than 5. See also Fig. 12.

9. Conclusion

We introduced an original generic scheme for asymmetric parallel topology-preserving thinning of 3D objects made of voxels, in the framework of critical kernels. We saw that from this scheme, one can easily derive several thinning operators having specific behaviours, simply by changing the definition of skeletal points. In particular, we showed that ultimate, curve, and surface skeletons can be obtained, based on the notion of 1D/2D isthmuses.

We performed some experiments in order to compare our curve skeletonization algorithm with all methods of the same class found in the literature. The results show that our method outperforms the other ones, in the sense that it produces less spurious branches.

Furthermore, we showed that an effective filtering can be easily performed within our framework, thanks to the notion of persistence. In this approach, the filtering is done dynamically, with very little added cost, and is governed by a unique parameter.

Acknowledgments

This work has been partially supported by the "ANR-2010-BLAN-0205 KIDICO" project.

References

Attali, D., Boissonnat, J., Edelsbrunner, H., 2009. Stability and computation of the medial axis — a state-of-the-art report, in: Möller, T., Hamann, B., Russell, B. (Eds.), Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration. Springer-Verlag, pp. 109–125.

Bertrand, G., 1994. Simple points, topological numbers and geodesic neighborhoods in cubic grids. Pattern Recognition Letters 15, 1003–1011.

Bertrand, G., 1999. New notions for discrete topology, in: Discrete Geometry for Computer Imagery, Springer. pp. 218–228.

Bertrand, G., 2007. On critical kernels. Comptes Rendus de l'Académie des Sciences, Série Math. I, 363–367.

Bertrand, G., Aktouf, Z., 1995. Three-dimensional thinning algorithm using subfields, in: Vision Geometry III, SPIE. pp. 113–124.

Bertrand, G., Couprie, M., 2008. Two-dimensional thinning algorithms based on critical kernels. Journal of Mathematical Imaging and Vision 31, 35–56.

Bertrand, G., Couprie, M., 2009. On parallel thinning algorithms: Minimal non-simple sets, P-simple points and critical kernels. Journal of Mathematical Imaging and Vision 35, 23–35.

Bertrand, G., Couprie, M., 2014. Powerful Parallel and Symmetric 3D Thinning Schemes Based on Critical Kernels. Journal of Mathematical Imaging and Vision 48, 134–148.

Bertrand, G., Malandain, G., 1994. A new characterization of three-dimensional simple points. Pattern Recognition Letters 15, 169–175.

Chaussard, J., 2010. Topological tools for discrete shape analysis. Ph.D. dissertation. Université Paris-Est.

Cornea, N.D., Silver, D., 2007. Curve-skeleton properties, applications, and algorithms. IEEE Transactions on Visualization and Computer Graphics 13, 530–548.

Couprie, M., 2013. Topological maps and robust hierarchical Euclidean skeletons in cubical complexes. Computer Vision and Image Understanding 117, 355–369.

Couprie, M., Bertrand, G., 2009. New characterizations of simple points in 2D, 3D and 4D discrete spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 637–648.

Couprie, M., Bertrand, G., 2014. Isthmus-based parallel and asymmetric 3d thinning scheme and algorithms, in: Discrete Geometry for Computer Imagery, Springer. pp. 51–62.

Giblin, P., 1981. Graphs, surfaces and homology. Chapman and Hall.

Gong, W., Bertrand, G., 1990. A simple parallel 3d thinning algorithm, in: ICPR90, pp. 188–190.

Kong, T.Y., 1995. On topology preservation in 2-D and 3-D thinning. International Journal on Pattern Recognition and Artificial Intelligence 9, 813–844.

Kong, T.Y., 1997. Topology-preserving deletion of 1's from 2-, 3- and 4-dimensional binary images, in: Discrete Geometry for Computer Imagery, Springer. pp. 3–18.

Kong, T.Y., Rosenfeld, A., 1989. Digital topology: introduction and survey. Comp. Vision, Graphics and Image Proc. 48, 357–393.

Kovalevsky, V., 1989. Finite topology as applied to image analysis. Computer Vision, Graphics and Image Processing 46, 141–161.

Liu, L., Chambers, E.W., Letscher, D., Ju, T., 2010. A simple and robust thinning algorithm on cell complexes. Computer Graphics Forum 29, 2253–2260.

Lohou, C., Bertrand, G., 2004. A 3D 12-subiteration thinning algorithm based on P-simple points. Discrete Applied Mathematics 139, 171–195.

Lohou, C., Bertrand, G., 2005. A 3D 6-subiteration curve thinning algorithm based on P-simple points. Discrete Applied Mathematics 151, 198–228.

Lohou, C., Bertrand, G., 2007. Two symmetrical thinning algorithms for 3D binary images. Pattern Recognition 40, 2301–2314.

Lohou, C., Dehos, J., 2010a. Automatic correction of ma and sonka's thinning algorithm using p-simple points. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1148–1152.

Lohou, C., Dehos, J., 2010b. An automatic correction of ma's thinning algorithm based on p-simple points. Journal of Mathematical Imaging and Vision 36, 54–62.

Ma, C., Wan, S., Chang, H., 2002a. Extracting medial curves on 3D images. Pattern Recognition Letters 23, 895–904.

Ma, C.M., 1995. A 3D fully parallel thinning algorithm for generating medial faces. Pattern Recognition Letters 16, 83–87.

Ma, C.M., Sonka, M., 1996. A 3D fully parallel thinning algorithm and its applications. Computer Vision and Image Understanding 64, 420–433.

Ma, C.M., Wan, S.Y., 2000. Parallel thinning algorithms on 3D (18,6) binary images. Computer Vision and Image Understanding 80, 364–378.

Ma, C.M., Wan, S.Y., Lee, J.D., 2002b. Three-dimensional topology preserving reduction on the 4-subfields. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 1594–1605.

Manzanera, A., Bernard, T., Prêteux, F., Longuet, B., 2002. n-dimensional skeletonization: a unified mathematical framework. Journal of Electronic Imaging 11, 25–37.

Németh, G., Kardos, P., Palágyi, K., 2010a. Topology preserving 2-subfield 3D thinning algorithms, in: Signal Processing, Pattern Recognition and Applications (SPPRA 2010). ACTA Press. volume 678, pp. 311–316.

Németh, G., Kardos, P., Palágyi, K., 2010b. Topology preserving 3D thinning algorithms using four and eight subfields, in: Campilho, A., Kamel, M. (Eds.), Image Analysis and Recognition. Springer Berlin / Heidelberg. Volume 6111 of *Lecture Notes in Computer Science*, pp. 316–325.

Németh, G., Kardos, P., Palágyi, K., 2011. A family of topology-preserving 3D parallel 6-subiteration thinning algorithms, in: Proc. 14th International Workshop on Combinatorial Image Analysis, IWCIA2011, Springer. pp. 17–30.

Németh, G., Palágyi, K., 2012. 3d parallel thinning algorithms based on isthmuses, in: Advanced Concepts for Intelligent Vision Systems. Volume 7517 of *Lecture Notes in Computer Science*, pp. 325–335.

Ogniewicz, R., Kübler, O., 1995. Hierarchic Voronoi skeletons. Pattern Recognition 28, 343–359.

Palágyi, K., 2008. A 3D fully parallel surface-thinning algorithm. Theoretical Computer Science 406, 119–135.

Palágyi, K., 2013. Parallel 3d 12-subiteration thinning algorithms based on isthmuses, in: Advances in Visual Computing. Volume 8033 of *Lecture Notes in Computer Science*, pp. 87–98.

Palágyi, K., Kuba, A., 1998. A 3D 6-subiteration thinning algorithm for extracting medial lines. Pattern Recognition Letters , 613–627.

Palágyi, K., Kuba, A., 1998. A hybrid thinning algorithm for 3d medical images. Journal of Computing and Information Technology 6, 149–164.

Palágyi, K., Kuba, A., 1999a. Directional 3D thinning using 8 subiterations, in: Discrete Geometry for Computer Imagery, Springer. pp. 325–336.

Palágyi, K., Kuba, A., 1999b. A parallel 3D 12-subiteration thinning algorithm. Graphical Models and Image Processing 61, 199–221.

Palágyi, K., Németh, G., Kardos, P., 2012. Topology preserving parallel 3d thinning algorithms, in: Digital Geometry Algorithms. Volume 2 of *Lecture Notes in Computational Vision and Biomechanics*, Springer, pp. 165–188.

Raynal, B., Couprie, M., 2011. Isthmus-based 6-directional parallel thinning algorithms, in: Discrete Geometry for Computer Imagery, Springer. pp. 141–152.

Reniers, D., van Wijk, J., Telea, A., 2008. Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. IEEE Transactions on Visualization and Computer Graphics 14, 355–368.

Saha, P., Chaudhuri, B., Chanda, B., Dutta Majumder, D., 1994. Topology preservation in 3D digital space. Pattern Recognition 27, 295–300.

Saha, P., Chaudhuri, B., Dutta Majumder, D., 1997. A new shape preserving parallel thinning algorithm for 3d digital images. Pattern Recognition 30, 1939–1955.

Tsao, Y., Fu, K., 1981. A parallel thinning algorithm for 3D pictures. Computer Graphics and Image Processing 17, 315–331.

Tsao, Y., Fu, K., 1982. A 3D parallel skeletonwise thinning algorithm pictures, in: Pattern Recognition and Image Processing, pp. 678–683.

Whitehead, J., 1939. Simplicial spaces, nuclei and m-groups. Proceedings of the London Mathematical Society 45, 243–327.