A 3D Sequential Thinning Scheme Based on Critical Kernels

Michel Couprie, Gilles Bertrand

A 3D sequential thinning scheme based on critical kernels

Michel Couprie, Gilles Bertrand

Université Paris-Est, LIGM, Équipe A3SI, ESIEE Paris, France[**]
e-mail: michel.couprie@esiee.fr, gilles.bertrand@esiee.fr

Abstract. We propose a new generic sequential thinning scheme based on the critical kernels framework. From this scheme, we derive sequential algorithms for obtaining ultimate skeletons and curve skeletons. We prove some properties of these algorithms, and we provide the results of a quantitative evaluation that compares our algorithm for curve skeletons with both sequential and parallel ones.

1 Introduction

Topology-preserving transformations are used in many applications of 2D and 3D image processing. In discrete grids, they are used in particular to thin objects until obtaining curve or surface skeletons. The notion of simple point [1] allows for efficient implementations of topology-preserving transformations: intuitively, a point of an object X is simple if it may be removed from X without changing its topological characteristics. Thus, a transformation that iterates the detection and the deletion of a single simple point at each step, is topology-preserving. Simple points may be characterized locally in 2D, 3D and even in higher dimensions (see [2]).

In order to preserve the main geometrical features of the object, some simple points must be preserved from deletion, such points will be called *skeletal points* in the sequel. For example, curve extremities can be used as skeletal points if we want to obtain a curve skeleton. In this paper, we consider only algorithms that dynamically detect skeletal points during the thinning, as opposed to those in two passes, that first need to compute skeletal points (sometimes called anchor points) prior to the thinning process.

Furthermore, in order to obtain well-centered skeletons, a sequential thinning algorithm must consider simple points in a specific order. For example, a naive but natural idea consists of considering only points that are simple points for the original object in a first step. During this first step of thinning, new simple points may appear, they are considered in a second step, and so on.

The scheme `SeqNaive` uses this strategy in order to try to obtain centered skeletons. Variants of this scheme may be derived by using different criteria for

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

Algorithm 1: `SeqNaive`(X)

```
   Data: $X$, a set of voxels
   Result: $X$
1  $K := \emptyset$;
2  repeat
3      $K := K \cup \{x$ that is a skeletal voxel for $X\}$;
4      $Y := \{x$ that is a simple voxel for $X\} \setminus K$;
5      foreach $x \in Y$ do
6          if $x$ is simple for $X$ then $X := X \setminus \{x\}$;
7  until stability ;
```

Algorithm 2: `SeqDir`(X)

```
   Data: $X$, a set of voxels
   Result: $X$
1  $K := \emptyset$; $Y := X$;
2  repeat
3      $K := K \cup \{x$ that is a skeletal voxel for $X\}$;
4      foreach $\alpha \in DirSet$ do
5          foreach $x \in Y \setminus K$ that is an $\alpha$-point for $X$ and that is simple for $Y$ do
6              $Y := Y \setminus \{x\}$;
7      $X := Y$;
8  until stability ;
```

defining skeletal points. For obtaining curve skeletons, one can for example use curve extremities as skeletal points (for an object X, a point of X is a curve extremity if it has exactly one neighbor in X).

Except for toy examples or small objects, this scheme yields noisy skeletons, see Fig. 1a. Furthermore, depending on the actual implementation, the centering may be quite bad. Consider for example an horizontal 2-pixel width ribbon in 2D: all its pixels are simple, and depending on the scanning order, it can be seen that the resulting skeleton could be just one pixel located in one of its extremities.

In order to make this order less arbitrary, one can use the so-called directional strategy that has been introduced by Rosenfeld in his seminal work on 2D parallel thinning [3]. Each iteration of the thinning algorithm is divided into several subiterations, and, in each subiteration, only points in a given direction are considered. For example in 2D, we may define a north (resp. south, east, west) point as an object point whose north (resp. south, east, west) neighbor belongs to the background. The scheme `SeqDir` is based on this strategy.

The directional scheme also leads to several variants, depending on the set of directions that are considered (*DirSet*), their order within the iteration, and the different criteria for defining skeletal points.

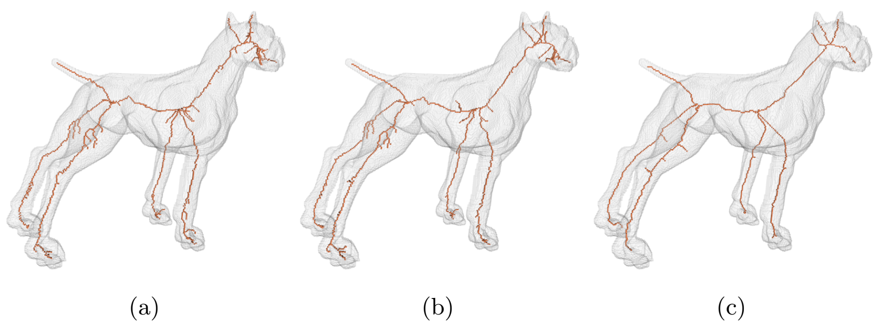


Fig. 1. (a): a skeleton produced by scheme `SeqNaive`. (b): a skeleton produced by scheme `SeqDir`. (c): a skeleton produced by the parallel thinning algorithm of [4]

It is important to note that the skeletons produced using either scheme `SeqNaive` or scheme `SeqDir` are thin, in the sense that they hold the following minimality property: the obtained skeleton contains no simple point, outside of those that have been detected as skeletal points.

While being better centered and a little less noisy than those produced using `SeqNaive`, the skeletons obtained using the directional strategy may also contain many spurious branches, see Fig. 1b.

This fact is a strong argument in favor of parallel thinning algorithms, which are known to produce more robust skeletons than sequential algorithms (see Fig. 1c). However, almost all of them fail to guarantee the minimality property, in other words, there may exist simple points that are not skeletal points in the obtained skeletons. Only some parallel thinning algorithms based on subgrids guarantee this property, but they are subject to some geometric artifacts, and they are not very good in terms of robustness to noise (see [5]).

The motivation of this work is to provide sequential thinning algorithms, that hold the minimality property, and that are as robust to noise as the best parallel algorithms. To achieve this goal, we use the framework of critical kernels.

Critical kernels constitute a framework that has been introduced by one of the authors [6] in order to study the topological properties of parallel thinning algorithms. It also allows one to design new algorithms, in which the guarantee of topology preservation is built in, and in which any kind of constraint may be imposed (see [7, 8]). Recently, we showed in [5] that our parallel algorithm for computing thin 3D curve skeletons, based on critical kernels, ranks first in a quantitative evaluation of the robustness of the thinning algorithms of the same class proposed in the literature.

In the classical approach called digital topology [1], topological notions like connectivity are retrieved thanks to the use of two graphs or adjacency relations, one for the object and another one for the background. In our approach, instead of considering only individual points (or voxels) linked by an adjacency relation, we consider that a digital object is made of *cliques*, a clique being a set of mutually adjacent voxels. In Fig. 2, we show an object made of 14 voxels. Among the 48 different cliques that are subsets of this object, three are highlighted: the cliques C_1, C_2, and C_3 made of respectively 1, 2, and 3 voxels. These three particular cliques are said to be *critical* (this term is defined in section 4), intuitively the notion of critical clique is a kind of generalization of the one of non-simple voxel. We see that, in the example of Fig. 2, removing any one of them from the object would alter its topological characteristics: it would disconnect this object. The main theorem of [6], that holds in any dimension, implies that preserving at least one voxel of each critical clique during the thinning, is sufficient to preserve topology.

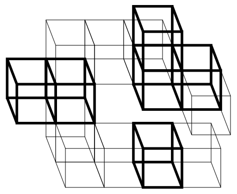


Fig. 2. An object made of 14 voxels, and in which one counts 48 different cliques, some overlapping some others. Three particular cliques, called critical cliques (see text), are highlighted.

The sequel of this paper is organized as follows. Sections 2, 3 and 4 provide all the necessary notions and results relative to, respectively, voxel complexes, simple voxels and critical kernels. We introduce our new generic sequential thinning scheme in section 5, and we prove in section 6 that it guarantees both the preservation of topology and the minimality property. Finally, we describe in section 7 an experimental study that shows that our new 3D curve thinning algorithm outperforms the other sequential thinning methods and even the parallel ones in terms of robustness.

2 Voxel Complexes

In this section, we give some basic definitions for voxel complexes, see also [9, 1]. Let $\mathbb{Z}$ be the set of integers. We consider the families of sets $\mathbb{F}^1_0$, $\mathbb{F}^1_1$, such that $\mathbb{F}^1_0 = \{\{a\} \mid a \in \mathbb{Z}\}$, $\mathbb{F}^1_1 = \{\{a, a+1\} \mid a \in \mathbb{Z}\}$. 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 $\mathbb{F}^1_0$ is called a *face* or an *d-face* of $\mathbb{Z}^n$, d is the *dimension of* f.

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 and Fig. 3). 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 [8], see also [10, 11] 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. 3 (left), 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 [8], 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 [12–15, 2]. An equivalence was also established with a definition based on the operation of collapse [16], this operation is a discrete analogue of a continuous deformation (a homotopy), see [10, 6, 2].

The notion of a simple voxel allows one to define thinnings of a complex, see an illustration Fig. 3 (right).

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

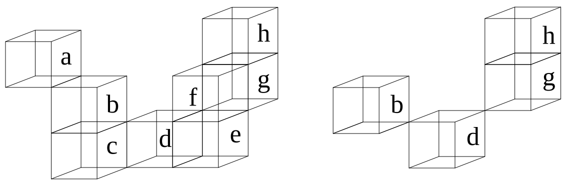


Fig. 3. Left: a complex X which is made of 8 voxels, Right: A complex $Y \subseteq X$, which is a thinning of X.

depicted Fig. 3 (left). 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 [6–8]. 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) $\cap\{x \in C\} = \cap\{x \in D\}$.

In other words, C is essential for X if it is maximal with respect to the inclusion, among all the cliques D in X such that ii) holds.

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. 3 (left), $\{f, g\}$ 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. 3 (left), the cliques $C_1 = \{b, c, d\}$, $C_2 = \{f, g\}$, and $C_3 = \{g, h\}$ are essential for X. We have $\mathcal{K}^*(C_1) \cap X = \emptyset$, $\mathcal{K}^*(C_2) \cap X = \{d, e, h\}$, and $\mathcal{K}^*(C_3) \cap X = \{f\}$. 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 dimensions [6, 8].

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. 3 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 Generic sequential thinning scheme

In this section, we introduce our new generic sequential thinning scheme, see algorithm 3. It is generic in the sense that any notion of skeletal point may be used, for obtaining, *e.g.*, ultimate, curve, or surface skeletons.

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.

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 line 3. $Skel_X$ is a function from X on $\{True, False\}$ that allows us to detect 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. This is the criterion that we will use for this paper.

In the scheme, the set W stores the voxels that are selected to be preserved. At each iteration, W is constructed from scratch by gathering all elements that are selected in all critical cliques (note that a non-simple voxel form a critical 3-clique). Cliques are scanned in decreasing order of rank, and then, according to their orientation α. These orientations correspond to the three axes of the grid, see Fig. 4.

We illustrate our scheme in Fig. 5 on two different objects. For each one, we show an ultimate skeleton, obtained using a function $Skel_X$ that always returns the value *False*, and a curve skeleton, based on a function $Skel_X$ that detects end voxels.

6 Properties

Consider a single execution of the main loop of the algorithm (lines 3–10). It may be easily seen that, by construction, the set W at line 10 contains at least one

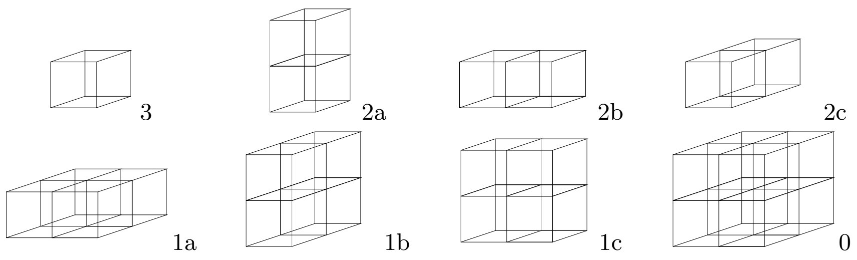


Fig. 4. Masks for cliques with rank $d \in \{3, 2, 1, 0\}$ and orientation $\alpha \in \{a, b, c\}$. An essential clique for a given voxel complex X is a subset of one of those masks. Note that the masks for 3-cliques and the 0-cliques have only one orientation, more precisely, they are invariant by $\pi/2$ rotations.

Algorithm 3: `SeqThinningScheme`$(X, Skel_X)$

```
   Data: $X \in \mathbb{V}^3$, $Skel_X$ is a function from $X$ on $\{True, False\}$
   Result: $X$
1  $K := \emptyset$;
2  repeat
3      $K := K \cup \{x \in X \setminus K$ such that $Skel_X(x) = True\}$;
4      $W := \{x \in X \mid x$ is not simple for $X\} \cup K$;
5      for $d \leftarrow 2, 1, 0$ do
6          for $\alpha \leftarrow a, b, c$ do
7              foreach $d$-clique $C \subseteq X \setminus K$ critical for $X$ with orientation $\alpha$ do
8                  if $C \cap W = \emptyset$ then
9                      Choose $x$ in $C$; $W := W \cup \{x\}$;
10     $X := W$;
11 until stability ;
```

voxel of all cliques that are critical for X and contained in $X \setminus K$. Furthermore, as all voxels of K are preserved during the execution of lines 3–9, the set W at line 10 contains at least one voxel of all cliques that are critical for X. Thus by theorem 5, we have the following property.

Proposition 6. *Let $X \in \mathbb{V}^3$, let $Skel_X$ be a function from X on $\{True, False\}$, let Z =SeqThinningScheme$(X, Skel_X)$. Then, the complex Z is a thinning of X.*

Next, we prove that the produced skeletons hold the minimality property, which is a direct consequence of the following proposition.

Proposition 7. *Consider a single execution of the main loop of the algorithm SeqThinningScheme (lines 3–10). If $X \setminus K$ contains at least one simple voxel, then the steps 3 to 10 of the algorithm remove at least one voxel from X.*

Proof: Since there is at least one simple voxel in $X \setminus K$, we know that W is different from X at the beginning of line 5. Suppose that, at the beginning of

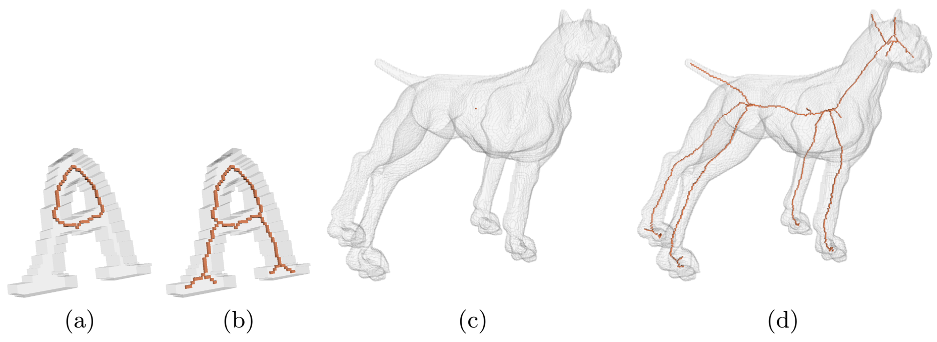


Fig. 5. Skeletons obtained by using `SeqThinningScheme`. (a,c): ultimate skeletons. We set $Skel_X(x) = False$ for all x. Note that the ultimate skeleton of (c) is a single voxel, as the original object is connected and has no holes or cavities. (b,d): curve skeletons. We set $Skel_X(x) = False$ whenever x is a curve extremity.

line 10, we have $W = X$. Let x be the last voxel that has been added to W. This voxel must have been added in line 9, thus the condition line 8: $C \cap W = \emptyset$ was fulfilled. But C is made of at least two voxels (it is not a 3-clique), thus there is another voxel $y \neq x$ in C that is not in W at this moment. This voxel y also has to be added to W since in the end, we have $W = X$. This contradicts the hypothesis that x is the last voxel added to W. Thus, we have $W \neq X$ at the beginning of line 10, and at least one voxel is removed from X. $\square$

Corollary 8. *At the end of* `SeqThinningScheme`*, the resulting complex X does not contain any simple voxel outside of the constraint set K.*

7 Experiments and results

In these experiments, we used a database of 30 three-dimensional 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 human body in coarse resolution has typically 5 branches, one for the head and one for each limb). For each object, we manually indicated an "ideal" number of branches. Unnecessary branches are essentially due to noise. Thus, a simple and effective criterion for assessing the robustness of a skeletonization method is to count the number of extra branches, or equivalently in our case, the number of extra curve extremities.

In order to compare methods, we mainly use the indicator $E(X, M) = |c(X, M) - c_i(X)|$, 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. Note that, for all objects in our database and all tested methods, the difference $c(X, M) - c_i(X)$ was positive, in other words the methods produced more skeleton branches than expected, or just the right number. We define $E(M)$ as the average, for all objects of the database, of $E(X, M)$. The lower the value of $E(M)$, the better the method M with respect to robustness.

The goal of sequential 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, most parallel thinning algorithms may leave some extra simple voxels. We define our second indicator $P(X, M)$ as the percentage of voxels in the result obtained from X after application of method M that are simple voxels but not curve extremities. We define $P(M)$ as the average, for all objects of the database, of $P(X, M)$. The lower the value of $P(M)$, the better the method M with respect to thinness.

Table 1 gathers the results of our quantitative comparison. For a comparison with parallel methods, we chose the ones that produce thin skeletons (asymmetric methods) and that provide the best results, see [5] for a comparative study of all parallel algorithms of this kind. Note that the criterion for defining skeletal voxels is the same in all the tested method, that is, a skeletal voxel is a curve extremity.

Table 1. Results of our experiments

Sequential			Parallel		
Method M	$E(M)$	$P(M)$	Method M	$E(M)$	$P(M)$
`SeqNaive`	29.5	0	Palágyi & Kuba 99 [4]	8.97	0.23
`SeqDir`	24.0	0	Lohou & Bertrand 05 [17]	11.3	0.003
`SeqThinningScheme`	6.53	0	Couprie & Bertrand [5]	6.73	0

In addition to those quantitative results, it is interesting to look at the results of some different methods for a same object (see Fig. 6). The example of the second column of Fig. 6 illustrates very well the sensitivity to contour noise of the methods. The original object is a solid cylinder bent in order to form a knot. Thus, its curve skeleton should ideally be a simple closed curve. Any extra branch of the skeleton must undoubtedly be considered as spurious. As can be seen in the figure, only our method (last row) among these four ones produces a skeleton of this object that is totally free of spurious branches.

8 Conclusion

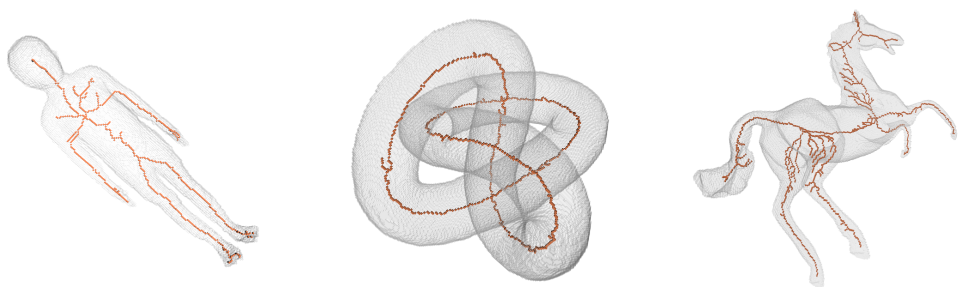
We have presented a new generic sequential 3D thinning scheme. From this scheme, it is possible to derive algorithms for obtaining ultimate, curve or surface skeletons, that hold the minimality property. Any criterion for defining skeletal voxels can be employed to this aim. Here, we used curve extremities as skeletal voxels in order to obtain curve skeletons. In an other work [5], we showed that a criterion based on 1D and 2D isthmuses allows one to obtain robust curve and/or surface skeletons.

We showed experimentally that our new curve thinning algorithm has an excellent robustness to noise. Furthermore, our approach allows us do define a notion of "thinning iteration", as in parallel algorithms, that corresponds intuitively to the removal of one layer of voxels from the object. This feature, together with the use of isthmuses to characterize skeletal points, makes it possible to apply in the same approach, a strategy based on isthmus persistence (see [8, 5]) in order to filter skeletons based on a single parameter.

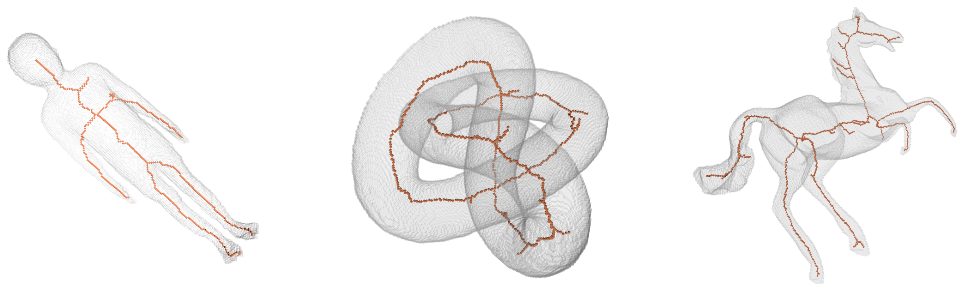
References

1. Kong, T.Y., Rosenfeld, A.: Digital topology: introduction and survey. Comp. Vision, Graphics and Image Proc. **48** (1989) 357–393
2. Couprie, M., Bertrand, G.: New characterizations of simple points in 2D, 3D and 4D discrete spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(4) (2009) 637–648
3. Rosenfeld, A.: A characterization of parallel thinning algorithms. Information and control **29**(3) (1975) 286–291
4. Palágyi, K., Kuba, A.: Directional 3D thinning using 8 subiterations. In: Discrete Geometry for Computer Imagery. Volume 1568 of Lecture Notes in Computer Science., Springer (1999) 325–336
5. Couprie, M., Bertrand, G.: Asymmetric parallel 3d thinning scheme and algorithms based on isthmuses. Technical report (2014) hal-01104691, preprint, submitted for publication.
6. Bertrand, G.: On critical kernels. Comptes Rendus de l'Académie des Sciences, Série Math. **I**(345) (2007) 363–367
7. Bertrand, G., Couprie, M.: Two-dimensional thinning algorithms based on critical kernels. Journal of Mathematical Imaging and Vision **31**(1) (2008) 35–56
8. Bertrand, G., Couprie, M.: Powerful Parallel and Symmetric 3D Thinning Schemes Based on Critical Kernels. Journal of Mathematical Imaging and Vision **48**(1) (2014) 134–148
9. Kovalevsky, V.: Finite topology as applied to image analysis. Computer Vision, Graphics and Image Processing **46** (1989) 141–161
10. Kong, T.Y.: Topology-preserving deletion of 1's from 2-, 3- and 4-dimensional binary images. In: Discrete Geometry for Computer Imagery. Volume 1347 of LNCS., Springer (1997) 3–18
11. Bertrand, G.: New notions for discrete topology. In: Discrete Geometry for Computer Imagery. Volume 1568 of LNCS., Springer (1999) 218–228
12. Bertrand, G., Malandain, G.: A new characterization of three-dimensional simple points. Pattern Recognition Letters **15**(2) (1994) 169–175
13. Bertrand, G.: Simple points, topological numbers and geodesic neighborhoods in cubic grids. Pattern Recognition Letters **15** (1994) 1003–1011
14. Saha, P., Chaudhuri, B., Chanda, B., Dutta Majumder, D.: Topology preservation in 3D digital space. Pattern Recognition **27** (1994) 295–300
15. Kong, T.Y.: On topology preservation in 2-D and 3-D thinning. International Journal on Pattern Recognition and Artificial Intelligence **9** (1995) 813–844
16. Whitehead, J.: Simplicial spaces, nuclei and m-groups. Proceedings of the London Mathematical Society **45**(2) (1939) 243–327

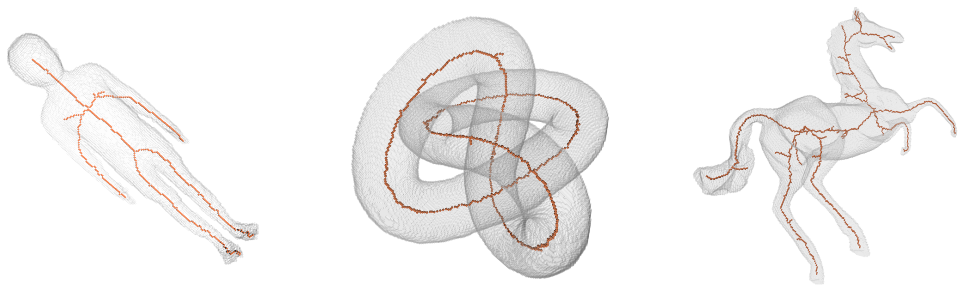
17. Lohou, C., Bertrand, G.: A 3D 6-subiteration curve thinning algorithm based on P-simple points. Discrete Applied Mathematics **151** (2005) 198–228



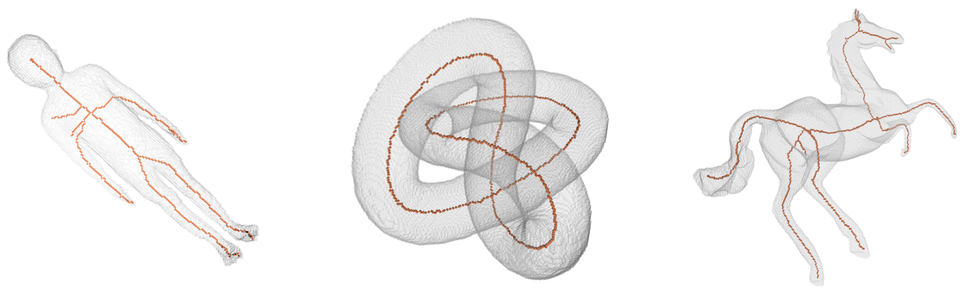
Algorithm `SeqDir`



Algorithm [4]



Algorithm [17]



Our new method `SeqThinningScheme`

Fig. 6. Some results with some selected images.