Chapter One

Single-Scan Skeletonization Driven by a Neighborhood-Sequence Distance

Aurore Arlicot\textsuperscript{a}, Pierre Évenou\textsuperscript{b} and Nicolas Normand\textsuperscript{c}
École polytechnique de l’Université de Nantes, Rue Christian Pauc, La Chantrerie, 44306 Nantes, France
E-mail: \textsuperscript{a}Aurore.Arlicot@univ-nantes.fr, \textsuperscript{b}Pierre.Evenou@univ-nantes.fr, \textsuperscript{c}Nicolas.Normand@univ-nantes.fr

ABSTRACT

Shape description is an important step in image analysis. Skeletonization methods are widely used in image analysis as they are a powerful tool to describe a shape. Indeed, a skeleton is a one point wide line centered in the shape which keeps the shape’s topology. Commonly, at least two scans of the image are needed for the skeleton computation in the state of art methods of skeletonization. In this work, a single scan is used considering information propagation in order to compute the skeleton. This paper presents also a new single-scan skeletonization using different distances like $d_4$, $d_8$ and $d_{ns}$.

Keywords: Skeleton, single-scan, Neighborhood sequence distance

1.1 INTRODUCTION

The skeletonization is an important method in the image analysis domain. This method is a powerful tool to extract some pertinent informations about a shape. The Skeletonization algorithm is the operation that identifies the skeleton of a shape. The skeleton is a shape subset that has to be one point wide, homotopic and centered in the shape. Moreover, if the distance information is preserved, the
skeleton is reversible. Skeletonization methods have been largely studied in the literature. We are interested in the discrete domain methods because of the discretization of our space due to the pixels. In this domain there are some different methods like thinning [7, 8], distance based methods [1, 5] and other like distance ordered thinning [12, 16] These skeletonization methods use more than two scans of the image in order to produce the skeleton. The goal of this work is to calculate the skeleton with only one scan of the image.

The point of working with a single scan of the image is that the analysis task can be applied during the acquisition task. This amounts to saving time and storage.

In order to understand how this is possible, let us focus on the information propagation in the other methods. In method like that Thinning, the shape is peeled layer by layer until to produce the skeleton. Methods using the thinning from distance transform also peel the shape but they use the distance information like a layer.

These two methods compute the skeleton with a classical raster scan of the image. The dependance between two pixels depends of their distance in the raster scan order. That is to say if the distance map of a pixel is inferior than his neighbor, his neighbor will be peeled after the first one.

In section 1.2, necessary notions to compute a skeleton will be introduced. Section 1.3 explains the main idea of this work in order to compute the distance map and the medial axis with a single scan of the image. In section 1.4, we describe the computation of the skeleton through our reference algorithm and the particular cases are described. The results will be presented in section 1.5.

1.2 PRELIMINARIES

1.2.1 Neighborhood

Definition 1.1. We call \( j \)-neighborhood in \( n \mathbb{D} \), with \( 1 \leq j \leq n \), the set of vectors:

\[
\mathcal{N}_j = \{ \overrightarrow{v} : |v_i| \leq 1 \text{ and } \sum |v_i| \leq j \} .
\]

The point \( q \) is a \( j \)-neighbor of \( p \) if the vector from \( p \) to \( q \) belongs to the \( j \)-neighborhood (\( \overrightarrow{pq} \in \mathcal{N}_j \)).

The two neighborhoods of the 2D square grid, often called cityblock or 4-neighborhood for \( \mathcal{N}_1 \) and chessboard or 8-neighborhood for \( \mathcal{N}_2 \), are depicted on fig. 1.1.

Definition 1.2 (Simple point [13]). Let \( X \) be a part of \( \mathbb{Z}^2 \) with a finite number of \( \alpha \)– connected – components and \( \overline{X} \)– connected – components. A point \( A \) of \( X \) is \( \alpha \)– simple in \( X \) if:

- the set \( X \) and \( X \setminus \{ A \} \) have the same \( \alpha \)– connected – component number;
- the set \( \overline{X} \) and \( \overline{X} \setminus \{ A \} \) have the same \( \overline{\alpha} \)– connected – component number;
1.2.2 Discrete distances

Definition 1.3 (Discrete distance and metric). Consider a function $d : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{N}$ and the following properties $\forall x, y, z \in \mathbb{Z}^n$:

1. **positive definiteness** $d(x, y) \geq 0$ and $d(x, y) = 0 \iff x = y$,
2. **symmetry** $d(x, y) = d(y, x)$,
3. **triangle inequality** $d(x, z) \leq d(x, y) + d(y, z)$,
4. **translation invariance** $d(x + z, y + z) = d(x, y)$.

$d$ is called a distance if it verifies conditions 1 and 2, a metric with conditions 1 to 3 and a translation-invariant metric with the all 4 conditions.

Practically, path-based distances are distance functions that associate to each couple of points $(p, q)$, the minimal cost of a path from $p$ to $q$. A path $P$ is a sequence of points $(p_0 = p, p_1, \ldots, p_n = q)$ such that two successive points are neighbors: $p_{r-1} \leftarrow p_r \in \mathcal{N}$. The cost (or length) of a path $P = (p_0 = p, p_1, \ldots, p_n = q)$ is the number of its displacements, $n$.

Rosenfeld and Pfaltz proposed the first path-based distances, namely $d_4$ and $d_8$, that are respectively generated by neighborhoods of types 1 and 2 [15]. Later, they defined the octagonal distance, a better approximation of the Euclidean distance, using alternated neighborhoods [14]. By allowing longer periodic sequences of neighborhoods, Das et al. introduced the concept of neighborhood-sequence (NS) distances that was afterwards extended to non-periodic sequences [3, 4].

Let $B(r), r \in \mathbb{N}^*$ be a sequence of integers; each $B(r)$ represents the type of neighborhood allowed for the r-th displacement. i.e., a path $(p_0 = p, p_1, \ldots, p_n = q)$ is valid if any two successive points $p_{r-1}$ and $p_r$ are $B(r)$-neighbors: $p_{r-1} \leftarrow p_r \in \mathcal{N}_{B(r)}$. In the 2D case, $B(r) \in \{1, 2\}$. We denote by $J_B(r)$, with $1 \leq j \leq n$, the number of times the $j$-neighborhood appears in sequence $B$ till index $r$:

$$J_B(r) = |\{r : 1 \leq i \leq r, B(i) = j\}| .$$

For any real number $x$, let $\lfloor x \rfloor$ (resp. $\lceil x \rceil$) denote the largest integer less (resp. greater) than or equal to $x$. A Beatty sequence with parameter $\alpha$ where $\alpha$ is a positive irrational number ($\alpha \in \mathbb{R} \setminus \mathbb{Q}$), is the sequence of the positive multiples of $\alpha$ rounded to the largest smaller integers: $b(i) = \lfloor \alpha i \rfloor$. András and Lajos Hajdu suggested to base the neighborhood sequence on a Beatty sequence, taking $B(r) = |\alpha r| - |\alpha (r - 1)| + 1$ [6]. Clearly, if $\alpha \in [0, 1]$ then $\forall r \in \mathbb{N}^*, B(r) \in \{1, 2\}$.

![Figure 1.1. 1-neighborhood, $\mathcal{N}_1$, and 2-neighborhood, $\mathcal{N}_2$, in the 2D square grid](image-url)
and $a$ represents the asymptotic rate of ‘2s’ in the sequence. Then (even for a rational $a$):

$$1_B(r) = r - \lfloor ar \rfloor = \lceil (1 - a) r \rceil ,$$

$$2_B(r) = \lceil ar \rceil .$$

**Definition 1.4 (Disk).** For a given distance $d$, the closed disk $D\leq$ and open disk $D<$ of center $c$ and radius $r$ are the sets of points of $\mathbb{Z}^n$:

$$D\leq(c, r) = \{ p : d(c, p) \leq r \} ,$$

$$D<(c, r) = \{ p : d(c, p) < r \} .$$

Equation (1.1)

Disks of a neighborhood-sequence distance are built by a series of Minkowski sums:

$$D<(c, r + 1) = D<(c, r) \oplus N_B(r) .$$

Equation (1.2)

**1.2.3 Distance map and medial axis**

**Definition 1.5 (Distance transform).** The distance transform $DT_X$ of the binary image $X$ is a function that maps each point $p$ to its distance from the closest background point:

$$DT_X : \mathbb{Z}^n \rightarrow \mathbb{N}$$

$$DT_X(p) = \min \{ d(q, p) : q \in \mathbb{Z}^n \setminus X \} .$$

Equation (1.3)

Alternatively, since all points at a distance less than $DT_X(p)$ to $p$ belong to $X$ ($D<(p, DT_X(p)) \subset X$) and at least one point at a distance to $p$ equal to $DT_X(p)$ is not in $X$ ($D<(p, DT_X(p) + 1) \not\subset X$) then:

$$DT_X(p) \geq r \iff \hat{D}<\subset(p, r) \subset X ,$$

Equation (1.4)

where $D<(c, r)$ and $\hat{D}<\subset(c, r)$ are symmetric to each other with regards to their center $c$, $\hat{D}<\subset(c, r) = \{ p : d(p, c) < r \}$. The DT is usually defined as the distance to the background which is equivalent to the distance from the background by symmetry. The equivalence is lost with asymmetric distances, and this definition better reflects the fact that DT algorithms always propagate paths from the background points.

**Definition 1.6 (Medial Axis).** We call maximal disk of the set $X$, a disk contained in $X$ but not contained in any other disk also contained in $X$. By definition of $DT_X$, if $D<(p, r)$ is maximal then necessarily $r = DT_X(p)$. The medial axis $MA_X$ of the set $X$ maps each point to the radius of the maximal disk centered in that point if it exists, 0 otherwise:

$$MA_X(p) = \begin{cases} 0 & \text{if } \exists q, r \text{ s.t. } D<(p, DT_X(p)) \subset D<(q, r) \subset X , \\ DT_X(p) & \text{otherwise} . \end{cases}$$

Equation (1.5)
For NS distances, a simple local criterion is sufficient to extract the medial axis from the distance map:

$$\text{MA}_X(p) = \begin{cases} 0 & \text{if } \exists \vec{v} \in \mathcal{N}_B(DT_X(p)), DT_X(p + \vec{v}) > DT_X(p), \\ \text{DT}_X(p) & \text{otherwise.} \end{cases} \quad (1.6)$$

Since the null vector is, by definition 1.1, included in neighborhoods of all types, we can also write:

$$\text{MA}_X(p) = r > 0 \iff \begin{cases} \text{DT}_X(p) \geq r \\ \forall \vec{v} \in \mathcal{N}_B(r), \text{DT}_X(p + \vec{v}) \leq r \end{cases}. \quad (1.7)$$

### 1.3 Distance Map and Medial Axis Computation

Distance from the background to each pixel is the primary order criterion in our skeletonization algorithm. For path-based distances, distance maps are computed by propagating path costs from the background towards the inside of the shape. This process is common to all kinds of algorithms, parallel [15], queue-based [11, 17] or sequential [9, 15]. Classical sequential algorithms require at least two scans (exactly two for simple and chamfer distances) of the image data in reverse order [15]. Wang and Bertrand [18] described a generalized distance transform using one neighborhood or two alternated neighborhoods. When the neighborhoods satisfy the forward scan condition, path costs are only propagated in directions compatible with the raster scan thus a single scan is required. A new generalized distance transform for arbitrary sequences, periodic or not, was presented in [10]. It uses a translated version of neighborhoods $\mathcal{N}_1$ and $\mathcal{N}_2$ as shown in fig. 1.2. Furthermore, it was shown that the regular, centered, distance map can be obtained from the generalized distance map obtained with translated disks. This correcting step requires a random access to pixels in a limited band of image rows, but it follows the main raster scan data flow.

#### 1.3.1 Asymmetric Distance Transform

Let $d$ be a 2D NS distance with sequence $B(r) = \lfloor ar \rfloor$. The main criterion of the thinning order of a shape $X$ is the distance from the background obtained by the distance transform with distance $d$: $DT_X$. Instead of directly computing $DT_X$ (this would require at least three scans of the image), we deduce it from the distance transform $DT'_X$ obtained with another (generalized because asymmetric) distance $d'$. $d'$ is defined with the same sequence $B$ as $d$ but with translated neighborhoods.

For each neighborhood $\mathcal{N}_j$, $j \in \{1, 2\}$, we apply a translation vector $\vec{t}_j$ such that the translated neighborhood $\mathcal{N}'_j = \mathcal{N}_j \oplus \{ \vec{t}_j \}$ is in forward scan condition. In a translation preserved scan order, $\vec{t}_j$ translates the first visited point of $\mathcal{N}_j$ to the
Figure 1.2. Neighborhoods used for the translated NS-distance transform. (a) and (b) are respectively the type 1 and 2 translated neighborhoods, $N'_1$ and $N'_2$. (c) is the whole set of non-null displacements, $N'' = (N'_1 \cup N'_2) \setminus \{O\}$.

origin. Assuming a 2D standard raster scan order:

$$\vec{t}_1 = (0, 1) \quad \vec{t}_2 = (1, 1).$$  \hfill (1.8)

The translated neighborhoods $N'_1$ and $N'_2$ obtained with $\vec{t}_1 = (0, 1)$ and $\vec{t}_2 = (1, 1)$ are depicted in fig. 1.2.a and fig. 1.2.b.

$\text{DT}'_X$ can be computed in a single raster scan using [10]:

$$\text{DT}'_X(p) = \begin{cases} 0 & \text{if } p \not\in X \\ \min \{ \text{LUT}_{\vec{v}}(\text{DT}'_X(p - \vec{v})), \vec{v} \in N'' \} & \text{otherwise} \end{cases}$$ \hfill (1.9)

where $\text{LUT}_{\vec{v}}(r)$ is the final cost of a path of initial cost $r$ after an extra displacement $\vec{v}$:

$$\text{LUT}_{\vec{v}}(r) = \begin{cases} \text{LUT}_1(r) = \left\lceil \frac{(1 - a)r}{1 - \alpha} \right\rceil + 1 & \text{if } \vec{v} \in N'_1 \text{ and } \vec{v} \not\in N'_2 \\ \text{LUT}_2(r) = \left\lceil \frac{ar}{\alpha} + 1 \right\rceil & \text{if } \vec{v} \not\in N'_1 \text{ and } \vec{v} \in N'_2 \\ \text{LUT}_{12}(r) = r + 1 & \text{if } \vec{v} \in N'_1 \text{ and } \vec{v} \in N'_2 \end{cases}$$

1.3.2 Equivalence with the Centered Distance Transform

Disks generated by neighborhoods $N'_1$, $N'_2$ and the sequence $B(r)$ are such that:

$$D'_\leq(p, r) = D_\leq(p + \vec{t}(r), r) \quad \hat{D}'_\leq(p, r) = \hat{D}_\leq(p - \vec{t}(r), r)$$ \hfill (1.10)

where the translation vector $\vec{t}(r)$ is:

$$\vec{t}(r) = \vec{t}(r - 1) + \vec{B}(r)$$

$$= \vec{1}(r) \vec{t}_1 + \vec{2}(r) \vec{t}_2$$

$$= (\vec{2}(r), \vec{1}(r) + \vec{2}(r))$$

$$= \vec{2}(r), r.$$.

$\text{DT}'_X$ has equivalence with values of $\text{DT}_X$:

$$\text{DT}_X(p) \geq r \iff D_\leq(p, r - 1) \subseteq X$$

$$\iff D'(p + \vec{t}(r - 1), r - 1) \subseteq X$$

$$\iff \text{DT}'_X(p + \vec{t}(r - 1)) \geq r.$$ \hfill (1.11)
Consequently:

\[
\text{DT}_X(p' - \vec{t}(r-1)) = r \iff \text{DT}_X(p' - \vec{t}(r-1)) \geq r \text{ and } \text{DT}_X(p' - \vec{t}(r-1)) < r + 1
\]

\[
\iff \text{DT}_X(p' + \vec{t}_{B(r)}) \leq r \leq \text{DT}_X(p'). \quad (1.12)
\]

Knowing \(\text{DT}_X(p')\) and \(\text{DT}_X(p' + \vec{t})\) for a given vector \(\vec{t}\), we can deduce \(\text{DT}_X(p' - \vec{t}(r-1))\) for all values of \(r\) between \(\text{DT}_X(p' + \vec{t})\) and \(\text{DT}_X(p')\) for which \(\vec{t} = \vec{t}_{B(r)}\).

In [10], an algorithm is given to fully recover the centered distance transform from the asymmetric one. However, for the distance ordered skeletonization process, values of the centered distance map \(\text{DT}\) are not required to be stored as an image. Instead, we compute these values from \(\text{DT}'\), on the fly.

1.3.3 Medial Axis

Let \(\text{MA}_X(p)\) be the medial axis of \(X\) with distance \(d'\). Similarly to (1.7):

\[
\text{MA}_X(p) = r > 0 \iff \begin{cases} 
\text{DT}_X(p) \geq r \\
\forall \vec{d} \in \mathcal{N}_B(r), \text{DT}_X(p + \vec{d}) \leq r
\end{cases}
\]

using (1.11) and by definition of \(\mathcal{N}_B^r\):

\[
\text{MA}_X(p) = r > 0 \iff \begin{cases} 
\text{DT}_X(p - \vec{t}(r-1)) \geq r \\
\forall \vec{d} \in \mathcal{N}_B(r), \text{DT}_X(p + \vec{d} - \vec{t}(r-1)) \leq r
\end{cases}
\]

\[
\iff \text{MA}_X(p - \vec{t}(r-1)) = r > 0
\]

Like all Minkowski-based distances, the medial axis of the generalized distance \(d'\), \(\text{MA}_X\), can be computed by a local maximum criterion. \(\text{MA}_X\) is equivalent, up to a translation of each maximal disk, to the medial axis of the regular distance \(d\), \(\text{MA}_X\).

1.4 ONE PASS SKELETON COMPUTATION

This section presents a single-scan skeletonization algorithm. We start from a multi-scan reversible distance-ordered algorithm where the thinning order is guided by the distance to the background. From this “reference” algorithm, we derive a single-scan skeletonization algorithm that, despite the fact that the thinning order is modified, preserves the dependencies of pixel removals.

1.4.1 Reference algorithm (rlk)

Like DOHT [12] or Svensson’s work [16], we use a distance-controlled thinning method. The reference algorithm proceeds by thinning pixels, layer by layer, according to their distance to the background. Medial axis points are not deleted because it allows the reconstruction of a shape from its skeleton.
Let $p$ and $q$ be two points, $r_p$ and $r_q$ be the distance map value to the point $p$ and $q$ respectively ($DT_X(p)$ and $DT_X(q)$). A lexicographic order (here called $rlk$) is defined such that:

$$p \prec^{rlk} q \iff \begin{cases} r_p < r_q \\ r_p = r_q \text{ and } p \prec q \end{cases}$$ (1.13)

**Algorithm 1:** The $rlk$ algorithm

**Data:** A binary image $X$, $DT_X$, $MAX_X$

**Result:** The skeleton $Sk_X$

for $r \leftarrow 1$ to $\max_{p \in X} \{DT_X(p)\}$ do

for $l \leftarrow 0$ to $L$ do

for $k \leftarrow 0$ to $K$ do

if $DT_X(p) = r$ and $MAX_X(p) \neq r$ and $p$ is simple then

$Sk_X(p) \leftarrow 0$;

end

end

end

end

$1.4.2$ Single-scan algorithm ($l'k'r$)

Now the goal is to switch the order of the loops to produce a skeletonization algorithm in one scan. The order of the loops for the reference algorithm was radii, lines and columns ($rlk$ order scan).

Thanks to the translation of the center of the disks, it is possible to change the order of the loops. Now the order is translated lines ($l'$), translated columns ($k'$) and radii ($l'k'r$ order).

The following equivalences appear:

Let $l'_p = l_p + r_p - 1$, $l'_q = l_q + r_q - 1$ and $k'_p = k_p$, $k'_q = k_q$ for the distance $d_4$ and $k'_p = k_p + r_p - 1$, $k'_q = k_q + r_q - 1$ for the distance $d_8$.

$$p \prec^{l'k'r} q \iff \begin{cases} l'_p < l'_q \iff p \prec^{rlk} q \\ l'_p = l'_q \text{ and } \begin{cases} k'_p < k'_q \\ k'_p = k'_q \text{ and } r_p < r_q \iff p \prec^{rlk} q \end{cases} \end{cases}$$ (1.14)
Algorithm 2: The \( l'k'r \) algorithm

**Data:** A \( K \times L \) binary image \( X \), \( DT_X' \), \( MA_X' \)

**Result:** The skeleton \( Sk_X \)

\[
\text{for } l' \leftarrow 0 \text{ to } L \text{ do}
\]

\[
\text{for } k' \leftarrow 0 \text{ to } K \text{ do}
\]

\[
p' \text{: point } (k', l')
\]

\[
\text{for } r \leftarrow \max \{1, \min \{DT_X'(p' + \vec{t}_1), DT_X'(p' + \vec{t}_2)\}\} \text{ to } DT_X'(p') \text{ do}
\]

\[
p \leftarrow p' - (r - 1)
\]

\[
\text{if } r \geq DT_X'(p' + \vec{t}_{B(r)}) \text{ then}
\]

\[
\text{if } MA_X'(p') \neq r \text{ and } p \text{ is simple then}
\]

\[
Sk_X(p) \leftarrow 0
\]

\[
\text{if Next open disk built with } N_8 \text{ and } k' \geq 0 \text{ and}
\]

\[
MA_X[l][k - 1] = d \text{ and } MA_X' \neq r \text{ and simple then}
\]

\[
Sk_X(k - 1, l) \leftarrow 0
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\text{end}
\]

1.4.3 Particular cases

We note two particulars cases in our algorithm. The first one is a simplicity particular case, due to our special order scan. Figure 1.3. shows this particular case.

In this example the deletion causes a change in the topology of the shape if you remove the point \( p \) (creating a hole). In an other algorithm, the point \( p \) was removed after his right neighbor. Due to our special order scan, \( p \) is seen before its right neighbor and is not (for the moment) simple.

The solution is to re-evaluate the simplicity of the previous point when the current point is deleted. That is to say, in the example figure 1.3., the deletion of the point \( q \) lead the simplicity evaluation of the point \( p \).

\[\begin{array}{cccccc}
3 & 2 & 2 & 2 & 2 & 2 \\
3 & 2 & 1 & 1 & 1 & 1 \\
3 & 2 & 1 & 1 & 1 & 1 \\
3 & 2 & 1 & 1 & 1 & 1 \\
3 & 2 & 1 & 1 & 1 & 1
\end{array}\]

Figure 1.3. Special case in one scan algorithms
The second particular case is due to the translation of the center of the disk $d_4$. Figure 1.4. shows the particular case.

Figure 1.4. $p \prec_{rk} q \iff l'_p = l'_q$ et $k'_p < k'_q$.

In this special case, the point $p$ is computed before the point $q$ (because $l'_p = l'_q$ et $k'_p < k'_q$) whereas it is not the case in the reference algorithm because $p \prec_{rk} q \iff DT_X(p) < DT_X(q)$.

So if distance in top-left corner and left neighbor (untranslated) is bigger than $r$ behave like this point hasn’t been thinned yet.

The simplicity function determines if a point is simple or not. In order to compute the simplicity of a point, we subtract from the count of 1-connected background pixels, the number of corners of $N_2$. The point is simple if the result equals 1. Figure 1.5. shows the patterns used for the simplicity function.

Figure 1.5. (a)+(b)+(c)+(d)-(e)-(f)-(g)-(h) = 1 if the point is simple.

1.5 RESULTS

Table 1.1 presents the execution time of the six algorithms. The reference algorithms has been evaluated with three different distances, along with their single-scan versions. Note that, for all three distances, the single-scan algorithm is faster than the reference algorithm.

In figure 1.6. is shown an example of an original shape. The figure 1.7. shows the images resulting from the algorithms. For each one, the one scan algorithm produced the same, pixel per pixel, skeleton as the reference algorithm.

Figure 1.6. Original shape.
Table 1.1 Execution time results

<table>
<thead>
<tr>
<th>Image</th>
<th>number of pixel</th>
<th>distance used</th>
<th>time in second</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ref</td>
<td>one scan</td>
</tr>
<tr>
<td>main</td>
<td>287523 pixels</td>
<td>$d_4$</td>
<td>0.204439</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_8$</td>
<td>0.407720</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_{ns}$</td>
<td>0.168092</td>
</tr>
<tr>
<td>Bone</td>
<td>93632 pixels</td>
<td>$d_4$</td>
<td>0.021276</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_8$</td>
<td>0.040200</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_{ns}$</td>
<td>0.018037</td>
</tr>
<tr>
<td>Corail</td>
<td>3027888 pixels</td>
<td>$d_4$</td>
<td>0.499546</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_8$</td>
<td>0.903974</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_{ns}$</td>
<td>0.386084</td>
</tr>
<tr>
<td>anguille</td>
<td>5018112 pixels</td>
<td>$d_4$</td>
<td>1.996670</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_8$</td>
<td>5.095797</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_{ns}$</td>
<td>1.735280</td>
</tr>
</tbody>
</table>

(a) with distance $d_4$  (b) with distance $d_8$  (c) with distance $d_{ns}$

Figure 1.7. Image results

Applications that continuously acquire an image could definitely take advantage of the proposed approach while they typically produce unbounded images. In previous works [2], this method extracted the skeleton from a radiography of trabecular bone.

1.6 CONCLUSION

This paper presented an algorithm which computes a distance map, a medial axis and a skeleton with one scan of the image. The algorithm is able to compute a skeleton using any neighborhood-sequence distance. Examples were shown for $d_4$ (cityblock), $d_8$ (chessboard) and the octagonal distance $d_{oct}$.

The proposed algorithm is up to 10-15 times faster than the reference algorithm. Remark that best performance improvements are obtained with large images.

References

A. Arlicot, P. Évenou and N. Normand


