Exact Medial Axis based on Flake Digital Circles

Édouard THIEL[1] and Rita ZROUR[2]

[1] LIS, Aix-Marseille University, CNRS, Marseille, France
edouard.thiel@univ-amu.fr

[2] XLIM, ASALI, University of Poitiers, CNRS, Poitiers, France
rita.zrour@univ-poitiers.fr

Abstract. Medial Axis, also known as Centres of Maximal Disks, is a useful representation of a shape that interested many researchers since the 60s. The k-flake digital circles introduced in [16] extends most common notions of digital circles and offers a control over the topology of the digital circle leading to a k-tunnel free circle. This paper addresses the fundamental problem of computing an exact Medial Axis based on flake digital circles. For this purpose we define a family of flake distances, then we adapt the algorithm proposed for the Euclidean distance by Remy and Thiel in [10], which computes the look-up table and the neighbourhood to be tested, to the case of the flake distances, and we compare the results.

Keywords: Medial axis; Flake circles; Look-up tables; Maximal disks; Euclidean distance transform

1 Introduction

The Medial Axis (MA) is a useful representation of a shape for image description, analysis and compression. It has been introduced by Blum in the 60s [1] as a set of points where different fire fronts collide. The MA consists in detecting the centres of maximal disks in a binary shape. A disk is maximal if it is not included in any other disk in the shape [2]. The MA is the set of centres and radii of maximal disks. In discrete space is often disconnected, not thin and sensitive to small contour perturbations.

In the literature, many researcher worked on computing MA approximately [3,4,5,6] or exactly [7,8,9,10]. Detecting MA is usually done based on a Distance Transform (DT), where each pixel is labelled with its distance to the background, that is the radius of the largest disk included in the shape centred on the pixel. Saito and Toriwaki proposed in [12] an efficient algorithm, separable in dimension, for computing the exact Squared Euclidean Distance Transform (SEDT). Later on, Hirata [13] and Meijster et al. [14] have optimised this algorithm to linear time in the number of pixels.

In this work we are interested in proposing an exact approach for computing the MA based on flake digital circles. Flake digital circles are disks that can be characterized analytically [15,16]. Our approach extracts the MA based on a look-up table (LUT) and a LUT Mask ($\mathcal{M}_{\text{Lut}}$). The idea of the LUT was introduced in [7,8]. Its principle is that it gives for each radius value read in the DT, the minimum value of the neighbours that forbids a point to be in MA. The problem is to systematically compute the LUT associated with a distance function, for any radius, and also to compute the test neighbourhood $\mathcal{M}_{\text{Lut}}$. An efficient algorithm for computing the LUT and $\mathcal{M}_{\text{Lut}}$ for the exact MA extraction is presented in any dimension for chamfer norms in [9] and for the Euclidean distance in [10].

In this paper, two major contributions are proposed. First, we present a new family of discrete distances whose disks coincide to the flake digital circles. Second, we present an exact method to compute the MA based on flake digital circles which is an adaptation of [10]. Our algorithm computes the LUT columns and the test neighbourhood $\mathcal{M}_{\text{Lut}}$, and certifies that this neighbourhood is sufficient up to a given radius.

The article is divided as follows. We recall in Sect. 2 some basic notions and the definitions of adjacency norms and k-flakes. We study the k-flake digital hyperspheres in Sect. 3 and we present the family of k-flake distances in $\mathbb{Z}^n$. In Sect. 4 we justify the validity of the method thanks to the definition of the k-flake discrete distances in $G(\mathbb{Z}^n)$, and we present the adapted algorithms for the computation of the LUT columns and $\mathcal{M}_{\text{Lut}}$. Results are given in Sect. 5 in the 2D case, and we finally conclude in Sect. 6 .

2 Norms, adjacency and flakes

We consider $\mathbb{R}^n$ as the Euclidean vector space, $\mathbb{Z}^n$ as an n-dimensional $\mathbb{Z}$-module (i.e., a discrete vector space), and both of them as their associated affine space.

The classical ℓ_p norms, denoted $||\cdot||_p$, are defined by: $\forall \vec{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $||\vec{x}||_p = (|x_1|^p + \ldots + |x_n|^p)^{\frac{1}{p}}$. The Manhattan distance d_1 is induced by $||\vec{x}||_1 = |x_1| + \ldots + |x_n|$, the Euclidean distance d_{E} by $||\vec{x}||_2 = \sqrt{|x_1|^2 + \ldots + |x_n|^2}$, and the Tchebychev distance d_∞ by $||\vec{x}||_\infty = \max(|x_1|, \dots, |x_n|)$.

While in $\mathbb{Z}^n$, the values provided by d_1 and d_∞ are integers, those of d_{E} are real. Since d_{E}^2 values are integers, many algorithms compute and store d_{E}^2, with the drawback that the distance d_{E}^2 is not a metric (it does not respect the triangular inequality).

A *voxel* P, where $P = (P_1, \dots, P_n) \in \mathbb{Z}^n$, is the axis-aligned closed unit cube centred on P in $\mathbb{R}^n$, that is $\left\{ Q \in \mathbb{R}^n \,:\, ||Q - P||_\infty \leqslant \frac{1}{2} \right\}$. A *k-face* is a face of dimension k. Two voxels are said *k-adjacent* if they share at least a k-face. Formally, let $P, Q \in \mathbb{Z}^n$ and $0 \leqslant k < n$; P and Q are k-adjacent iff $|Q_i - P_i| \leqslant 1 \,\forall i$ and $\sum_{i=1}^{n} |Q_i - P_i| \leqslant n - k$ (that is, P and Q share at least k equal coordinates).

This notion of adjacency can also be expressed in terms of norms [16]: the *k-adjacency norm*, denoted $[\,\cdot\,]_k$, is defined in $\mathbb{R}^n$ for any $0 \leqslant k < n$ as $\forall \vec{x} \in \mathbb{R}^n$, $[\vec{x}]_k = \max\{||\vec{x}||_\infty\,,\, ||\vec{x}||_1/(n-k)\}$. Let P and $Q \in \mathbb{Z}^n$, then P and Q are k-adjacent iff $[Q - P]_k \leqslant 1$. Note that by the well-known inequality $\ell_\infty \leqslant \ell_2 \leqslant \ell_1 \leqslant n \cdot \ell_\infty$ we have $[\,\cdot\,]_0 = ||\cdot||_\infty$ and $[\,\cdot\,]_{n-1} = ||\cdot||_1$.

In the literature in $\mathbb{Z}^2$, d_1 and d_∞ are also named d_4 and d_8 after the number of pixels in their unit balls; the 0- and 1-adjacency correspond to the 8- and 4-neighbours, respectively.

A *k-path* is a sequence $P_1, \ldots, P_m$ of distinct points in $\mathbb{Z}^n$ where P_i and P_{i+1} are k-adjacent, $1 \leqslant i < m$. A set E of points in $\mathbb{Z}^n$ is said *k-connected* if for any two points in E there exists a k-path in E joining them. The set E is said *k-separating* $\mathbb{Z}^n$, or *k-tunnel free*, if the complement $\mathbb{Z}^n \setminus E$ admits exactly two maximum k-connected subsets.

Given a distance d in $\mathbb{R}^n$, the (closed) ball of centre $P \in \mathbb{R}^n$ and radius $r \in \mathbb{R}_+$ is $\mathcal{B}_d(P,r) = \{Q \in \mathbb{R}^n : d(P,Q) \leqslant r\}$, and the corresponding disk is $\mathcal{D}_d(P,r) = \{Q \in \mathbb{R}^n : d(P,Q) = r\}$. A discrete ball is written as $\mathcal{B}_d^{\mathbb{Z}} = \mathcal{B}_d \cap \mathbb{Z}^n$. By extension for a norm h in $\mathbb{R}^n$, we denote by $\mathcal{B}_h$ and $\mathcal{D}_h$ the ball and the disk for the distance induced by h. Any ball $\mathcal{B}_h$ is necessarily convex.

Let A and B be two non-empty sets. The Minkowski sum of A and B is defined by $A \oplus B = \{a + b : a \in A, b \in B\}$. This operation is also called dilation of A by the structuring element B.

The adjacency flakes are introduced in [16] as structuring elements, resulting from the intersection of a ball of an adjacency norm and a finite number of straight lines through the origin. Let $\mathcal{V}_k^n(\rho)$ be the set of extremal points of the convex ball $\mathcal{B}_{[\cdot]_k}(O,\rho)$ of radius $\rho \in \mathbb{R}_+$ in $\mathbb{R}^n$; we have

$$\mathcal{V}_k^n(\rho) = \Big\{ \rho P \in \mathbb{R}^n \,:\, P \in \{-1,0,1\}^n \,,\, \textstyle\sum_{i=1}^n |P_i| = n - k \Big\}. \tag{1}$$

For instance in $\mathbb{R}^2$, $\mathcal{V}_1^2(\rho) = \big\{(0, \pm\rho), (\pm\rho, 0)\big\}$ and $\mathcal{V}_0^2(\rho) = \big\{(\pm\rho, \pm\rho)\big\}$. Remark that $\forall V \in \mathcal{V}_k^n(\rho)$ we have $||\overrightarrow{OV}|| = \rho\sqrt{n-k}$.

Following [17], the *minimal k-adjacency-flake* $\mathcal{F}_k^n(\rho)$ for $[\cdot]_k$ and ρ is then defined in $\mathbb{R}^n$ as

$$\mathcal{F}_k^n(\rho) = \Big\{ P \in \mathcal{V}_k^n(\lambda) \,:\, \lambda \in [0,\rho] \Big\}, \tag{2}$$

which is the set of straight line segments joining the vertices $\mathcal{V}_k^n(\rho)$ to the origin, see Fig. 1 . In the sequel, we will use the term *k-flake* for short.

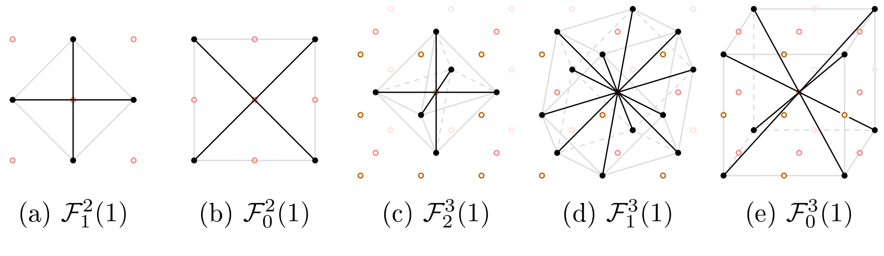

(a) $\mathcal{F}_1^2(1)$ (b) $\mathcal{F}_0^2(1)$ (c) $\mathcal{F}_2^3(1)$ (d) $\mathcal{F}_1^3(1)$ (e) $\mathcal{F}_0^3(1)$

Fig. 1. Flakes of radius 1, (a,b) in $\mathbb{R}^2$ and (c,d,e) in $\mathbb{R}^3$. Integer points are represented by small circles, $\mathcal{V}_k^n(1)$ points by black bullets, flakes $\mathcal{F}_k^n(1)$ by black segments, the k-adjacency norm balls $\mathcal{B}_{[\cdot]_k}(O,1)$ in light grey.

Since the k-flakes are defined from adjacency norms, they also allow to characterize the k-adjacency using a radius of $\frac{1}{2}$: two points $P, Q \in \mathbb{Z}^n$ are k-adjacent iff $(\{P\} \oplus \mathcal{F}^n_k(\frac{1}{2})) \cap (\{Q\} \oplus \mathcal{F}^n_k(\frac{1}{2})) \neq \emptyset$.

3 Flake digital hyperspheres, balls and distance

We are interested in the morphological digitization scheme using a Minkowski sum with a k-flake of radius $\frac{1}{2}$ as a structuring element [18]. Let S be a subset of $\mathbb{R}^n$, the *flake-digitization* of S is defined by

$$\mathbb{F}^n_k(S) = \Big\{ P \in \mathbb{Z}^n \ :\ \big(\{P\} \oplus \mathcal{F}^n_k(\tfrac{1}{2})\big) \cap S \neq \emptyset \Big\} . \tag{3}$$

Since $\mathcal{F}^n_k(\frac{1}{2})$ is central-symmetric we can equivalently write

$$\mathbb{F}^n_k(S) = \big(S \oplus \mathcal{F}^n_k(\tfrac{1}{2}) \big) \cap \mathbb{Z}^n , \tag{4}$$

and by commutativity we have

$$\mathbb{F}^n_k(S) = \big(\mathcal{F}^n_k(\tfrac{1}{2}) \oplus S \big) \cap \mathbb{Z}^n . \tag{5}$$

The flake-digitization $\mathbb{F}^n_k$ of a Euclidean disk $\mathcal{C}(P, r) = \mathcal{D}_{d_\mathrm{E}}(P, r)$ of centre $P \in \mathbb{R}^n$ and radius $r \in \mathbb{R}_+$ is called a *k-flake digital hypersphere* (or *circle* for $n = 2$). A graphical construction using (5) of a k-flake digital hypersphere $\mathbb{F}^n_k(\mathcal{C}(P, r))$ is shown in Fig. 2 in $\mathbb{Z}^2$; the selected points are delimited between the coloured circles, centred over the corresponding coloured k-flake vertices.

This digitization has an important topological property, which motivated our study: the k-flake digital hypersphere $\mathbb{F}^n_k(\mathcal{C}(P, r))$ is k-tunnel free if $r > \frac{\sqrt{n-k}}{2}$, see [17,19]. The property of k-tunnel freeness can be observed in Fig. 2 .

Let us characterize the k-flake digital hypersphere points. Given $P \in \mathbb{R}^n$ and $r \in \mathbb{R}_+$, by (3) we have

$$\begin{aligned} \mathbb{F}^n_k(\mathcal{C}(P, r)) &= \Big\{ Q \in \mathbb{Z}^n \ :\ \big\{ Q + T \ :\ T \in \mathcal{F}^n_k(\tfrac{1}{2}) \big\} \cap \mathcal{C}(P, r) \neq \emptyset \Big\} , \\ &= \Big\{ Q \in \mathbb{Z}^n \ :\ \exists T \in \mathcal{F}^n_k(\tfrac{1}{2}) \, , \, d_\mathrm{E}(P, Q + T) = r \Big\} . \end{aligned} \tag{6}$$

Since $\mathcal{F}^n_k(\frac{1}{2})$ is constituted by segments, we can bound the selected points by

$$\mathbb{F}^n_k(\mathcal{C}(P, r)) = \left\{ Q \in \mathbb{Z}^n \ :\ \begin{array}{c} \min \big\{ d_\mathrm{E}(P, Q + T) \ :\ T \in \mathcal{F}^n_k(\tfrac{1}{2}) \big\} \leqslant r \ , \\ \max \big\{ d_\mathrm{E}(P, Q + T) \ :\ T \in \mathcal{F}^n_k(\tfrac{1}{2}) \big\} \geqslant r \end{array} \right\} , \tag{7}$$

or equivalently, by translation and symmetry,

$$\mathbb{F}^n_k(\mathcal{C}(P, r)) = \left\{ Q \in \mathbb{Z}^n \ :\ \begin{array}{c} \min \big\{ d_\mathrm{E}(P + T, Q) \ :\ T \in \mathcal{F}^n_k(\tfrac{1}{2}) \big\} \leqslant r \ , \\ \max \big\{ d_\mathrm{E}(P + T, Q) \ :\ T \in \mathcal{F}^n_k(\tfrac{1}{2}) \big\} \geqslant r \end{array} \right\} . \tag{8}$$

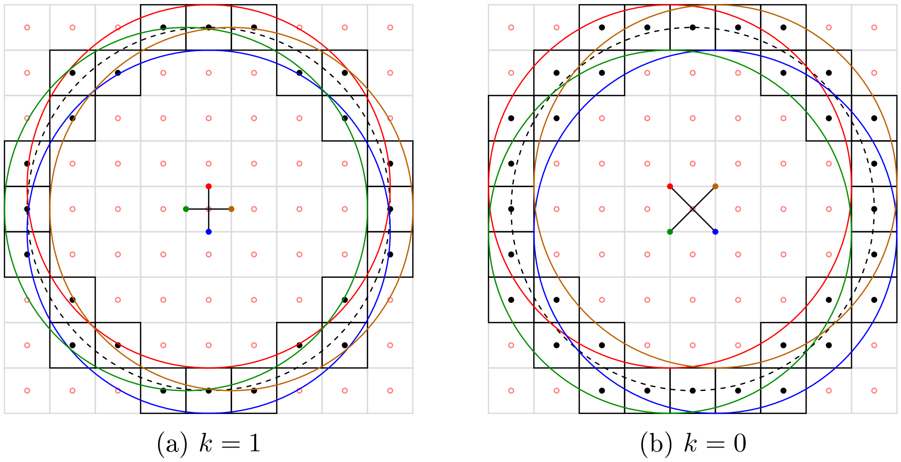


Fig. 2. Flake-digitizations $\mathbb{F}^2_k$ of the Euclidean circle (dotted line) $\mathcal{C}(O, 4)$. Integer points are represented by small circles, corresponding pixels by squares, flakes $\mathcal{F}^2_k(\frac{1}{2})$ by black segments, integer points and pixels belonging to $\mathbb{F}^2_k(\mathcal{C}(O, 4))$ in black.

We can then split (8) as

$$\begin{aligned}\mathbb{F}^n_k(\mathcal{C}(P, r)) = &\Big\{Q \in \mathbb{Z}^n \;:\; \min\big\{d_{\mathrm{E}}(P + T, Q) \;:\; T \in \mathcal{F}^n_k(\tfrac{1}{2})\big\} \leqslant r\Big\} \;\setminus \\ &\Big\{Q \in \mathbb{Z}^n \;:\; \max\big\{d_{\mathrm{E}}(P + T, Q) \;:\; T \in \mathcal{F}^n_k(\tfrac{1}{2})\big\} < r\Big\} \;,\end{aligned} \tag{9}$$

where the first term (with the min) is the discrete ball relative to the k-flake digital hypersphere, called *k-flake digital ball*, while the second term (with the max) stands for the hypersphere interior.

We aim to express a k-flake digital ball as a distance ball. Let $P, Q \in \mathbb{R}^n$ be two points, we define the *k-flake pseudo-distance* between P and Q as

$$d_{\mathcal{F}^n_k}(P, Q) = \min\big\{d_{\mathrm{E}}(P + T, Q) \;:\; T \in \mathcal{F}^n_k(\tfrac{1}{2})\big\} \,. \tag{10}$$

This function is symmetric but not positive definite, because $\forall A \in \mathcal{F}^n_k(\frac{1}{2}) \setminus \{O\}$ we have $d_{\mathcal{F}^n_k}(O, A) = 0$; so $d_{\mathcal{F}^n_k}$ is only a pseudo-distance. However, it allows to express the k-flake digital ball of (9) as the distance ball $\mathcal{B}^{\mathbb{Z}}_{d_{\mathcal{F}^n_k}}(P, r)$.

Now in the discrete case where $P, Q \in \mathbb{Z}^n$, to express $d_{\mathcal{F}^n_k}$ it is sufficient to consider the k-flake vertices, as well as the origin O for the special case $P = Q$:

$$d_{\mathcal{F}^n_k}(P, Q) = \min\big\{d_{\mathrm{E}}(P + T, Q) \;:\; T \in \mathcal{V}^n_k(\tfrac{1}{2}) \cup \{O\}\big\} \,. \tag{11}$$

It is plain that $d_{\mathcal{F}^n_k}$ is symmetric yet positive definite, thus is a distance in $\mathbb{Z}^n$. However, $d_{\mathcal{F}^n_k}$ is not a metric, since we can find counter-examples to the triangular inequality: take for instance $A = (1, 0)$, $B = (2, 0)$ and $C = (3, 0)$, we have $d_{\mathcal{F}^2_1}(A, B) = d_{\mathcal{F}^2_1}(B, C) = \frac{1}{2}$ and $d_{\mathcal{F}^2_1}(A, C) = \frac{3}{2}$, so $d_{\mathcal{F}^2_1}(A, C) \nleqslant d_{\mathcal{F}^2_1}(A, B) + d_{\mathcal{F}^2_1}(B, C)$.

4 Exact medial axis for k-flake digital balls

We present an adaptation of the LUT-mask method in [10] to compute the exact medial axis for k-flake digital balls in $\mathbb{Z}^n$. For simplicity, we consider images of size L^n, and algorithms and examples are given in dimension 2.

4.1 Discrete distance and G-symmetry

The LUT-mask method, originally given for d_{E}^2 in $\mathbb{Z}^n$, is very general and is valid for any distance balls family, respecting certain conditions.

(i) The first condition is to provide integer distance values. If we square $d_{\mathcal{F}_k^n}$ in (11) we will get multiples of $(\frac{1}{2})^2$, that is why we define the discrete distance

$$d'_{\mathcal{F}_k^n}(P,Q) = 4\big(d_{\mathcal{F}_k^n}(P,Q)\big)^2 = \min\big\{4\, d_{\mathrm{E}}^2(P+T,Q) \;:\; T \in \mathcal{V}_k^n(\tfrac{1}{2}) \cup \{O\}\big\} \quad (12)$$

and denote its balls by $\mathcal{B}'_{\mathcal{F}_k^n}(P,r) = \mathcal{B}^{\mathbb{Z}}_{d'_{\mathcal{F}_k^n}}(P,r)$.

(ii) The second condition is to have G-symmetric balls. Let we recall briefly this notion. We denote by $\mathcal{S}_G(n)$ the group of axial and diagonal symmetries of the rectilinear grid of $\mathbb{Z}^n$; its cardinal is $2^n n!$ (that is 8, 48 for $n = 2, 3$ resp.). A subset $X \in \mathbb{Z}^n$ is said *G-symmetric* if $\forall \sigma \in \mathcal{S}_G(n)$ we have $\sigma(X) = X$. We call *generator* of X the subset

$$G(X) = \big\{(x_1,\ldots,x_n) \in X \;:\; x_1 \geqslant \ldots \geqslant x_n \geqslant 0\big\}\,, \quad (13)$$

that corresponds to the first octant in $\mathbb{Z}^2$. When X is G-symmetric, the subset $G(X)$ is sufficient to reconstruct X with the G-symmetries, hence its study can be limited to $G(\mathbb{Z}^n)$ or $G(\mathbb{R}^n)$.

By construction, the k-flakes $\mathcal{F}_k^n(\frac{1}{2})$ are G-symmetric, and the same applies to k-flake digital balls whose centre is in $\mathbb{Z}^n$. Moreover, $G(\mathcal{V}_k^n(\frac{1}{2}))$ is restricted to one point Ω_k^n, where

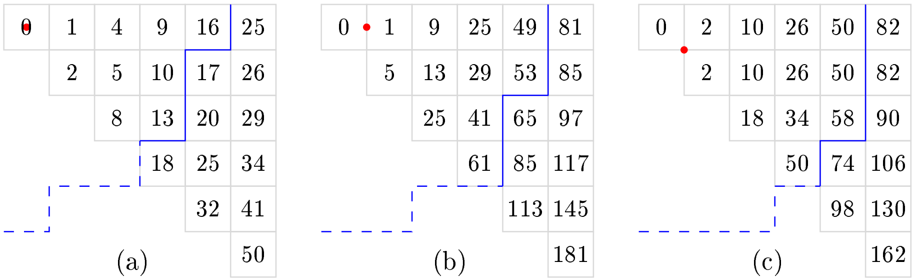
$$\Omega_k^n = \big\{\, (\,\overbrace{\tfrac{1}{2},\ldots,\tfrac{1}{2}}^{n-k},\overbrace{0,\ldots,0}^{k}\,)\,\big\}\,. \quad (14)$$

The points $\Omega_1^2 = (\frac{1}{2},0)$ and $\Omega_0^2 = (\frac{1}{2},\frac{1}{2})$ are shown in Fig. 3 (b,c) as red bullets.

Any point $Q \in G(\mathbb{Z}^n)$ is closer to Ω_k^n than to any other point in $\mathcal{V}_k^n(\frac{1}{2})$:

Lemma 1. *Let $Q \in G(\mathbb{Z}^n)$, then $\forall P \in \mathcal{V}_k^n(\frac{1}{2})$, $d_{\mathrm{E}}^2(\Omega_k^n,Q) \leqslant d_{\mathrm{E}}^2(P,Q)$.*

Proof. If $Q = O$ then $d_{\mathrm{E}}^2(\Omega_k^n,Q) = d_{\mathrm{E}}^2(P,Q) = (n-k)\frac{1}{4}$, $\forall P \in \mathcal{V}_k^n(\frac{1}{2})$. Now suppose $Q \neq O$, $Q_1 \geqslant \ldots \geqslant Q_n \geqslant 0$ and take $P \in \mathcal{V}_k^n(\frac{1}{2})$. Remark that $\forall i$, $(Q_i - |P_i|)^2 \leqslant (Q_i - P_i)^2$, so $d_{\mathrm{E}}^2((|P_1|,\ldots,|P_n|),Q) \leqslant d_{\mathrm{E}}^2((P_1,\ldots,P_n),Q)$. Remark also that $\forall i,j$ s.t. $i<j$, $(Q_i - \frac{1}{2})^2 + (Q_j - 0)^2 \leqslant (Q_i - 0)^2 + (Q_j - \frac{1}{2})^2$, thus by shifting all coordinates $\frac{1}{2}$ to the left we have $d_{\mathrm{E}}^2(\Omega_k^n,Q) \leqslant d_{\mathrm{E}}^2(P,Q)$. □



0	1	4	9	16	25
	2	5	10	17	26
		8	13	20	29
			18	25	34
				32	41
					50

(a)

0	1	9	25	49	81
	5	13	29	53	85
		25	41	65	97
			61	85	117
				113	145
					181

(b)

0	2	10	26	50	82
	2	10	26	50	82
		18	34	58	90
			50	74	106
				98	130
					162

(c)

Fig. 3. Beginning of CT^g in $G(\mathbb{Z}^2)$ for (a) d^2_{E}, (b) $d'_{\mathcal{F}^2_1}$ and (c) $d'_{\mathcal{F}^2_0}$. The red bullets represent the distance origin (a) O, (b,c) Ω^2_k; in blue, the border of the balls $\mathcal{B}_{d^2_{\mathrm{E}}}(O, t^2)$ and $\mathcal{B}'_{\mathcal{F}^n_k}(O, 4\,t^2)$ of radius $t = 4$ pixels.

By lemma 1 and (12), the distance $d'_{\mathcal{F}^n_k}$ from O to $Q \in G(\mathbb{Z}^n)$ is then

$$d'_{\mathcal{F}^n_k}(O,Q) = \begin{cases} 0 & \text{if } Q = O\,, \\ 4\,d^2_{\mathrm{E}}(\Omega^n_k, Q) & \text{if } Q \neq O\,. \end{cases} \tag{15}$$

When $Q \neq O$, since $4d^2_{\mathrm{E}}(\Omega^n_k, Q) = d^2_{\mathrm{E}}(2\Omega^n_k, 2Q)$ and by (14) we can still write

$$d'_{\mathcal{F}^n_k}(O,Q) = (2Q_1 - 1)^2 + \ldots + (2Q_{n-k} - 1)^2 + (2Q_{n-k+1})^2 + \ldots + (2Q_n)^2. \tag{16}$$

We name *Cone distance Transform* the image CT^g where each point from $G(\mathbb{Z}^n)$ is labelled to its distance from O. Fig. 3 compares the distances values obtained in CT^g by (16).

4.2 LUT mask and columns computation

We can now adapt the LUT-mask algorithm [10] to $d'_{\mathcal{F}^n_k}$. The aim is to compute a G-symmetric set $\mathcal{M}_{\mathrm{Lut}}$ of vectors in $\mathbb{Z}^n$, that is necessary and sufficient to detect the MA points on a DT with local tests on the neighbourhood $\mathcal{M}_{\mathrm{Lut}}$, together with a LUT column for each vector of $\mathcal{M}_{\mathrm{Lut}}$, that maps the minimum inclusion radii of the balls in that direction.

Thanks to the G-symmetry, all computations are proceeded in $G(\mathbb{Z}^n)$ for efficiency, including $\mathcal{M}^g_{\mathrm{Lut}} = G(\mathcal{M}_{\mathrm{Lut}})$. Given a vector $\vec{v} \in \mathbb{Z}^n$, we denote by $\vec{v}^g$ its G-symmetric in $G(\mathbb{Z}^n)$.

Let us recall the principle of the algorithm. At the beginning, we compute CT^g once; then, we examine each distance ball centred in O, of growing radius, obtained by a simple threshold on CT^g. For each ball, we extract MA using the current $\mathcal{M}^g_{\mathrm{Lut}}$; if a point different from O is detected, its direction is added to $\mathcal{M}^g_{\mathrm{Lut}}$ and the corresponding LUT column is computed on CT^g.

The main function `CompLutMask` is given Fig. 4; it is very similar to the original version, except lines 4,5. It can compute the whole $\mathcal{M}^g_{\mathrm{Lut}}$ up to a radius R_{target}, by giving the parameters $\mathcal{M}^g_{\mathrm{Lut}} = \emptyset$ and $R_{known} = 0$, or be resumed from an already computed $\mathcal{M}^g_{\mathrm{Lut}}$ and radius R_{known}, to a larger radius R_{target}.

```
function CompLutMask ($L$, $k$, $\mathcal{M}^g_{\mathrm{Lut}}$, $R_{known}$, $R_{target}$, $Lut$) :
1  CompCTg ($L$, $k$, $CT^g$) ; CompPV ($L$, $CT^g$, $R_{target}$, $PV$) ; init $DT^g$ to 0
2  for each $\vec{v}^g$ in $\mathcal{M}^g_{\mathrm{Lut}}$ do CompLutCol ($CT^g, L, \vec{v}^g, R_{target}, Lut[\vec{v}^g]$)
3  for $R = R_{known} + 1$ to $R_{target}$ do
4      if not $PV[R]$ then continue                                  // value R is not possible
5      CompBallDTg ($L$, $k$, $R$, $CT^g$, $DT^g$)
6      for $x = 0$ to $L - 1$ , for $y = 0$ to $x$ do
7          if $DT^g[x, y] \neq 0$ and IsMAg ($(x, y), \mathcal{M}^g_{\mathrm{Lut}}, Lut, DT^g$) then
8              $\mathcal{M}^g_{\mathrm{Lut}} = \mathcal{M}^g_{\mathrm{Lut}} \cup ((x, y)\ ,\ R)$            // Insert the new vector
9              CompLutCol ($CT^g$, $L$, $(x, y)$, $R_{target}$, $Lut[(x, y)]$)
10             if IsMAg ($(x, y)$, $\mathcal{M}^g_{\mathrm{Lut}}$, $Lut$, $DT^g$) then error
```

Fig. 4. Full $\mathcal{M}^g_{\mathrm{Lut}}$ and Lut Computation. Input: L the side length, $k \in \{0, 1\}$, $\mathcal{M}^g_{\mathrm{Lut}}$, R_{known} and R_{target}. Output: Lut and $\mathcal{M}^g_{\mathrm{Lut}}$.

```
function CompCTg ($L$, $k$, $CT^g$) :
    $CT^g[0, 0] = 0$
    for $x = 1$ to $L - 1$, for $y = 0$ to $x$ do
        $dx = 2 * x - 1$ ; $dy = 2 * y - 1 + k$
        $CT^g[x, y] = dx^2 + dy^2$
```

Fig. 5. Cone distance Transform. Input: L the side length, $k \in \{0, 1\}$. Output: CT^g the $L \times L$ distance image to O for $d'_{\mathcal{F}^2_k}$.

```
function CompPV ($L$, $CT^g$, $rmax$, $PV$) :
    for $r = 1$ to $rmax$ do $PV[r] =$ false
    for $x = 1$ to $L - 1$, for $y = 0$ to $x$ do
        $r = CT^g[x, y]$
        if $r \leqslant rmax$ then $PV[r] =$ true
```

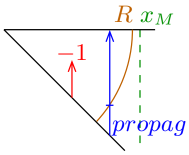
Fig. 6. Possible distance Values. Input: L, CT^g, $rmax$. Output: PV is filled with true for possible distance values.

During the first step (line 1) we compute CT^g using CompCTg. The implementation of CompCTg in Fig. 5 is straightforward by (15) and (16). Since the resulting image CT^g contains all the possible distance values, it is then used in CompPV, shown in Fig. 6, to create an array PV of Possible Values, in order to speed up the computations (line 4 in Fig. 4); it might optionally be used to save memory space when storing the LUT columns.

The second step (line 2) is the computation of the LUT columns in the case where $\mathcal{M}^g_{\mathrm{Lut}}$ is not empty. For each vector $\vec{v}^g \in \mathcal{M}^g_{\mathrm{Lut}}$, the function CompLutCol is called to fill the column $Lut[\vec{v}^g]$, such that for any radius r read in a DT, $Lut[\vec{v}^g][r]$ is the minimum value of a neighbour in direction $\vec{v}^g$ (and its G-symmetries) that forbids a point to be in MA. The function CompLutCol is the original function presented in [10, Fig. 8].

The next part (lines 3–10) checks each possible radius. The DT on the ball of radius R is computed for the k-flake discrete distance $d'_{\mathcal{F}^n_k}$ in line 5 by CompBallDTg, described in Sect. 4.3 . The fundamental idea is that the MA of a ball should be its sole centre, so if another MA point is detected (line 7) in its DT using the already known $\mathcal{M}^g_{\mathrm{Lut}}$, then this point is a direction that should be inserted (lines 8,9) in $\mathcal{M}^g_{\mathrm{Lut}}$ to remove it as a MA point. The function IsMAg, used to detect if a point is in MA, is the original function given in [10, Fig. 10]. The consistency test line 10 validates the whole method.

```
function CompBallDTg ($L$, $k$, $R$, $CT^g$, $DT^g$) :
1   for $x_M = 0$ to $L - 1$ do if $CT^g[0, x_M] > R$ then break // search bound $x_M$
2   if $x_M \geqslant L$ then error
3   // First scan on columns, decreasing order
4   for $x = 0$ to $x_M$ do
5       $i = 0$ ; $propag$ = false
6       for $y = x$ downto 0 do
7           if $CT^g[x, y] > R$ then // outside the ball : background
8               $propag$ = true
9           else if $propag$ then // inside the ball, mark to distance from bg
10              $i = i + 1$ ; $DT^g[x, y] = (2 * i - 1 + k)^2$
11          else $DT^g[x, y] = -1$ // inside the ball, no distance propagated
12  // Final scan on lines
13  for $y = 0$ to $x_M$, for $x = y$ to $x_M$ do
14      if $DT^g[x, y] == 0$ then continue // outside the ball
15      $d_{min} = 8 * L^2$
16      for $j = 0$ to $x_M - x$ do
17          $t_1 = DT^g[x + j, y]$
18          if $t_1 == -1$ then continue // no distance propagated
19          if $k == 0$ and $t_1 == 0$ then $t_1 = 1$
20          $u_2 = (2 * j - 1)^2$ ; $t_2 = t_1 + u_2$
21          if $t_2 < d_{min}$ then $d_{min} = t_2$ else if $u_2 > d_{min}$ then break
22      $DT^g[x, y] = d_{min}$
```



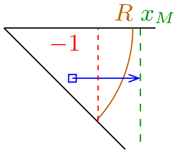


Fig. 7. Ball Distance Transform. Input: L the side length, $k \in \{0, 1\}$, R the radius, CT^g. Output: DT^g is the DT of $G(\mathcal{B}'_{\mathcal{F}^n_k}(O, R))$.

4.3 DT on a ball in the generator

The function CompBallDTg is given in Fig. 7 . It computes in DT^g the DT on $B^g(R) = G(\mathcal{B}'_{\mathcal{F}^n_k}(O, R))$. The ball is obtained by thresholding CT^g with R, any value $> R$ being considered as background. The function is called numerous times in CompLutMask (Fig. 4 , line 5), but it is sufficient to initialize one time DT^g to 0 at the start of CompLutMask (line 1) since R and the ball are growing.

We start (Fig. 7 , lines 1,2) by searching on the first row the bound x_M such that $B^g(R)$ is completely included in the subspace $0 \leqslant x < x_M$.

The next part of the function (lines 3-22) is inspired from the separable in dimension algorithm of Saito and Toriwaki [12] and ideas from [10]. The complexity is in $O({x_M}^{n+1})$. The parts specific to $d'_{\mathcal{F}^n_k}$ are lines 10,20 (to get back to d^2_{E}, replace the expressions by i^2 and j^2) and line 19 for $k = 0$, see below.

The first scan is made lines 4–11 on columns in decreasing order, using a flag *propag* which indicates if a distance information can be propagated, that is, if there was a background pixel during the scan. If not, a value of -1 is stored.

The second scan is made on rows, lines 13–22. There is no need to make a copy of the current row to compute the min (lines 16–21) for a pixel of abscissa

x, since the candidates $x+j$ for the min are after the current pixel during a j scan. The `else` part in line 21 is a classical optimisation of the Saito and Toriwaki algorithm.

The function `CompBallDTg` computes for each point x, y the min

$$h(x,y) = \min \left\{ \, (2j-1)^2 + (2i-1+k)^2 \, : \, (x+j, y+i) \in \overline{B^g(R)} \, \right\} \tag{17}$$

over $0 \leqslant x+j, y+i \leqslant x_M < L$. The k-flake distance formula that is evaluated is obtained using (16) by taking (x,y) as the origin. The first scan computes

$$g(x,y) = \min_i \left\{ \, (2i-1+k)^2 \, : \, (x, y+i) \in \overline{B^g(R)} \, \right\} \tag{18}$$

except for non propagated values (set to -1 in DT^g) and background pixels (which must stat at 0 in DT^g); the final scan calculates

$$h(x,y) = \min_j \left\{ \, (2j-1)^2 + g(x+j, y) \, \right\}. \tag{19}$$

The line 19 is a correction for $k=0$ in the case where $(x+j,y)$ is a background pixel (the min over i, not stored in DT^g, is then $(-1)^2$ and is added to t_1).

5 Results and experiments

In this section we present the results that we have obtained for the k-flake discrete distances in 2D. The full source code in C++, together with some more examples, are given in an electronic annex, available online in [20].

5.1 Appearance radii in $\mathcal{M}_{\text{Lut}}$

The beginning of $\mathcal{M}^g_{\text{Lut}}$ in $\mathbb{Z}^2$ is shown in Fig. 8 (left) for d^2_{E}, $d'_{\mathcal{F}^2_1}$ and $d'_{\mathcal{F}^2_0}$. As for d^2_{E}, all these vectors are visible points, that is, their coordinates are mutually prime. They do not appear in the same order, but their number grows slowly with R in a similar way.

On a larger scale, the behavior of the number m of vectors in $\mathcal{M}^g_{\text{Lut}}$ according to the appearance radius in pixels up to 1000 is compared in Fig. 8 (right) for the three distances; the three curves remain fairly close.

It has been shown in $\mathbb{Z}^n$ for d^2_{E} in [11] that $\mathcal{M}^g_{\text{Lut}}$ tends to the set of visible point when R tends to infinity. Our experiments show that we can reasonably conjecture the same property for $d'_{\mathcal{F}^2_k}$.

5.2 DT, RDT and MA examples

In order to compute the MA we need a Distance Transform (DT) algorithm, that labels each shape point with the distance to the closest background point. The original shape can then be reconstructed from MA using a Reverse Distance Transform (RDT).

	(a)			(b)		(c)	
i	$x\,,y$	R	$4R$	$x\,,y$	R	$x\,,y$	R
1	1 , 0	1	4	1 , 0	1	1 , 0	2
2	1 , 1	2	8	1 , 1	5	1 , 1	2
3	2 , 1	101	404	2 , 1	369	2 , 1	442
4	3 , 1	146	584	3 , 1	1233	3 , 1	746
5	3 , 2	424	1696	3 , 2	1413	4 , 1	2826
6	4 , 1	848	3392	4 , 1	4765	3 , 2	3538
7	5 , 1	1370	5480	5 , 1	5337	5 , 1	8298
8	6 , 1	2404	9616	5 , 2	9601	4 , 3	9810
9	4 , 3	3049	12196	4 , 3	10229	6 , 1	13698
10	7 , 1	3250	13000	5 , 4	14965	5 , 3	15850
11	5 , 2	3257	13028	5 , 3	16417	5 , 2	17194
12	7 , 5	3700	14800	7 , 1	20457	7 , 1	28594
13	5 , 3	4709	18836	6 , 1	24085	8 , 1	29258
14	7 , 3	5954	23816	7 , 2	26821	7 , 5	30266
15	5 , 4	9805	39220	8 , 1	44525	5 , 4	38666

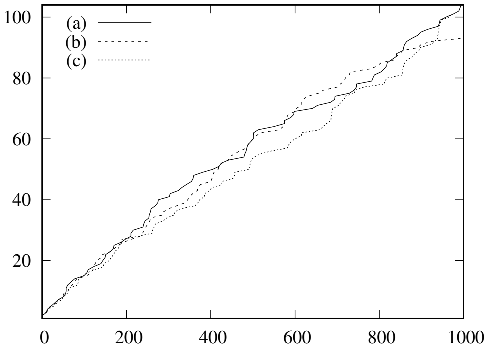
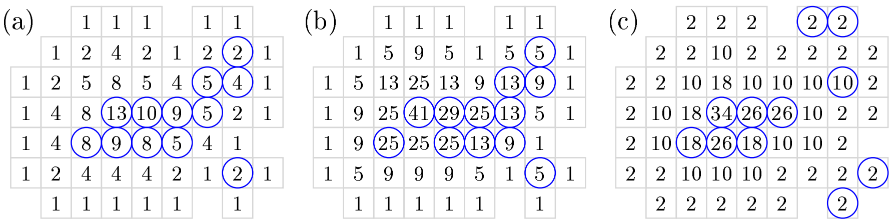


Fig. 8. Beginning of $\mathcal{M}^g_{\mathrm{Lut}}$ in $\mathbb{Z}^2$ for (a) d^2_{E}, (b) $d'_{\mathcal{F}^2_1}$ and (c) $d'_{\mathcal{F}^2_0}$. Left: appearance rank i, coordinates, appearance radius R (and $4R$ for comparison). Right: diagram of the number m of vectors in $\mathcal{M}^g_{\mathrm{Lut}}$ according to the appearance radius in pixels up to 1000 ($\sqrt{R}$ for (a), $\sqrt{R}/2$ for (b,c)).



(a)

		1	1	1		1	1	
	1	2	4	2	1	2	②	1
1	2	5	8	5	4	⑤	④	1
1	4	8	⑬	⑩	⑨	⑤	2	1
1	4	⑧	⑨	⑧	⑤	4	1	
1	2	4	4	4	2	1	②	1
	1	1	1	1	1		1	

(b)

		1	1	1		1	1	
	1	5	9	5	1	5	⑤	1
1	5	13	25	13	9	⑬	⑨	1
1	9	25	㊶	㉙	㉕	⑬	5	1
1	9	㉕	25	㉕	⑬	⑨	1	
1	5	9	9	9	5	1	⑤	1
	1	1	1	1	1		1	

(c)

		2	2	2		②	②	
	2	2	10	2	2	2	2	2
2	2	10	18	10	10	10	⑩	2
2	10	18	㉞	㉖	㉖	10	2	2
2	10	⑱	㉖	⑱	10	10	2	
2	2	10	10	10	2	2	2	②
	2	2	2	2	2		②	

Fig. 9. Medial Axis over DT for (a) d^2_{E}, (b) $d'_{\mathcal{F}^2_1}$ and (c) $d'_{\mathcal{F}^2_0}$. DT values are in black, MA points are circled in blue.

For this paper we have chosen to adapt in 2D the Saito and Toriwaki algorithms [12] that compute the exact DT and RDT in $O(L^{n+1})$ for an image of size L^n. Due to lack of space we only give the source code in [20]; the adaptations are rather tricky and quite different for $k = 0$ and $k = 1$. An example of DT and MA of a shape is shown in Fig. 9 for the three distances d^2_{E}, $d'_{\mathcal{F}^2_1}$ and $d'_{\mathcal{F}^2_0}$.

6 Conclusion and future work

In this paper we have proposed an exact method for computing the medial axis based on flake digital circles. For that purpose, we have defined a family of flake distances and adapt the algorithm proposed for the Euclidean distance by Remy and Thiel in [10], which computes the look-up table and the neighbourhood to be tested, to the case of the flake distances. Due to the lack of space, DT and RDT are given in annex [20] and are not detailed in this work. Their descriptions are not trivial and should be explained in a further paper. One of the perspectives of this work will be the study of the adaptation to a DT based on Hirata [13] optimised linear time algorithm, see also [3]. This adaptation is not guaranteed, because it depends on a property of uniqueness of intersection of parabolas, which has not yet been studied in the case of k-flakes circles.

References

1. Blum, H.: A transformation for extracting new descriptors of shape. In: Models for the Perception of Speech and Visual Form, MIT Press, Cambridge, 362–380 (1967)
2. Pfaltz, J.L., Rosenfeld, A.: Computer representation of planar regions by their skeletons. Communications of the ACM **10**(2), 119–122 (1967)
3. Coeurjolly, D.: d-Dim. Reverse Euclidean Distance Trans. and Euclidean Medial Axis Extraction in Optimal Time. In: 11th DGCI, LNCS **2886**, 327-337 (2003)
4. Chaussard, J., Couprie, M., Talbot, H.: Robust skeletonization using the discrete λ-medial axis. Pattern Recognition Letters **32**(9), 1384–1394 (2011)
5. Couprie, M., Coeurjolly, D., Zrour, R.: Discrete bisector function and Euclidean skeleton in 2D and 3D. Image and Vision Computing, **25**(10), 1543–1556 (2007)
6. Postolski, M., Couprie, M., Janaszewski, M.: Scale filtered Euclidean medial axis and its hierarchy. Computer Vision and Image Understanding, **129**, 89-102 (2014)
7. Borgefors, G., Ragnemalm, I., Sanniti di Baja, G.: The Euclidean DT: finding the local maxima and reconstructing the shape. In: 7th SCIA, **2**, 974—981 (1991)
8. Borgefors, G.: Centres of maximal disks in the 5–7–11 distance transform. In: Eighth Scandinavian Conference on Image Analysis, Tromsø, Norway, 105—111 (1993)
9. Remy, E., Thiel, E.: Medial axis for chamfer distances: computing look-up tables and neighbourhoods in 2D or 3D. Pattern Recognition Letters **23**(6), 649–661 (2002)
10. Remy, E., Thiel, E.: Exact Medial Axis with Euclidean Distance. Image and Vision Comp. **23**(2), 167–175 (2005) https://pageperso.lis-lab.fr/edouard.thiel/IVC2004/
11. Hulin, J., Thiel, E.: Visible vectors and discrete Euclidean medial axis. Discrete and Computational Geometry **42**(4), 759–773 (2009)
12. Saito, T., Toriwaki, J.I.: New algorithms for Euclidean DT of an n-dimensional digitized picture with applications. Pattern Recognition **27**(11), 1551–1565 (1994)
13. Hirata, T.: A unified linear-time algorithm for computing distance maps. Information Processing Letters, **58**(3), 129-133 (1996)
14. Meijster, A., Roerdink, J., Hesselink, W.: A General Algorithm for Computing Distance Transforms in Linear Time. Mathematical Morphology and its Applications to Image and Signal Processing, Springer US, 331–340 (2000)
15. Fiorio, S., Toutant, J.L.: Arithmetic Discrete Hyperspheres and Separatingness. In: 13th DGCI, LNCS **4245**, 425–436 (2006)
16. Toutant, J.L., Andres, E., Roussillon, T.: Digital circles, spheres and hyperspheres: From morphological models to analytical characterizations and topological properties. Discrete Applied Mathematics **161**, 2662–2677 (2013)
17. Toutant, J.L., Andres, E., Largeteau-Skapin, G., Zrour, R.: Implicit Digital Surfaces in Arbitrary Dimensions. In: 18th DGCI, LNCS **8668**, 332–343 (2014)
18. Andres, E., Largeteau-Skapin, G.: Digital Surfaces of Revolution Made Simple. In: 19th DGCI, LNCS **9647**, 244–255 (2016)
19. Zrour, R., Largeteau-Skapin, G., Andres, E.: Optimal consensus set for digital Flake hyperspheres in nD. J. Comput. Syst. Sci. **95**, 218–231 (2018)
20. Annex with source code: https://pageperso.lis-lab.fr/edouard.thiel/DGMM2024/