Efficient Distance Transformation for Path-based Metrics

David Coeurjolly[a], Isabelle Sivignon[b]

[a] *Université de Lyon, CNRS, LIRIS, Lyon, France*
[b] *Univ. Grenoble Alpes, CNRS, Grenoble INP, Gipsa-lab, 38000 Grenoble, France*

Abstract

In many applications, separable algorithms have demonstrated their efficiency to perform high performance volumetric processing of shape, such as distance transformation or medial axis extraction. In the literature, several authors have discussed about conditions on the metric to be considered in a separable approach. In this article, we present generic separable algorithms to efficiently compute Voronoi maps and distance transformations for a large class of metrics. Focusing on path-based norms (chamfer masks, neighborhood sequences...), we propose efficient algorithms to compute such volumetric transformation in dimension n. We describe a new $O(n \cdot N^n \cdot \log N \cdot (n + \log f))$ algorithm for shapes in a N^n domain for chamfer norms with a rational ball of f facets (compared to $O(f^{\lfloor \frac{n}{2} \rfloor} \cdot N^n)$ with previous approaches). Last we further investigate a more elaborate algorithm with the same worst-case complexity, but reaching a complexity of $O(n \cdot N^n \cdot \log f \cdot (n + \log f))$ experimentally, under assumption of regularity distribution of the mask vectors.

1. Introduction

Volumetric analysis of digital shapes is crucial in many geometry processing applications, for instance to measure distances between two points in $\mathbb{Z}^n$, or to measure the width of a shape or the proximity between two shapes. Since early works on digital geometry, distance transformation has been widely investigated (*e.g.* Rosenfeld and Pfaltz (1968)). Given a finite input shape $X \subset \mathbb{Z}^n$, the distance transformation labels each point in X with the distance to its closest point in $\mathbb{Z}^n \setminus X$. Labeling each point by the closest background point leads to Voronoi maps (*e.g.* the restriction to $\mathbb{Z}^n$ of Voronoi diagrams from computational geometry (de Berg et al., 2000)). Since such characterization is parametrized by a distance function, many authors have addressed this distance transformation problem with trade-offs between algorithmic performances and the *accuracy* of the digital distance function with respect to the Euclidean one. Hence, authors have considered: distances based on chamfer masks (Rosenfeld and Pfaltz, 1968; Borgefors, 1986; Fouard and Malandain, 2005) or sequences of chamfer masks (Rosenfeld and Pfaltz, 1966; Mukherjee et al., 2000; Strand, 2008; Normand et al., 2013a); vector displacement based Euclidean distance (Danielsson, 1980; Ragnemalm, 1993); Voronoi diagram based Euclidean distance (Breu et al., 1995; Maurer et al., 2003) or square of the Euclidean distance (Hirata, 1996; Meijster et al., 2000). For the Euclidean metric, separable volumetric computations have demonstrated to be very efficient with the design of optimal $O(n \cdot N^n)$ time algorithms for shapes in $[1,N]^n$ domains, optimal multithread/GPU implementation or extensions to toric domains (please refer to Coeurjolly (2012) for a discussion).

Path-based approaches (*e.g.* chamfer mask or –weighted– neighborhood sequences) approximate the Euclidean distance as the length of shortest paths defined from sequences of displacement vectors on the grid (from a finite set of possible moves). Aside distance infor-

*Corresponding author
Email addresses: david.coeurjolly@liris.cnrs.fr (David Coeurjolly), isabelle.sivignon@gipsa-lab.grenoble-inp.fr (Isabelle Sivignon)

mation, path-based approaches provide an explicit notion of discrete path that is not accessible for the Euclidean norm. Furthermore, the discrete and combinatorial nature of the distance function has been used to define efficient algorithms to extract discrete medial axis (Borgefors and Nyström, 1997; Remy and Thiel, 2002; Saha et al., 2016) as local maxima of the distance map. Normand et al. (2013b, 2014) further exploits the combinatorial structure of path-based distances to compute distance transformation in *on-the-fly* streaming context. In terms of distance transform computation, two main techniques exist. The first one considers a weighted graph formulation of the problem and Dijkstra-like algorithms on weighted graphs to compute distances. If m denotes the size of the chamfer mask, computational cost could be in $O(m \cdot N^n)$ using a cyclic bucket data structure as suggested by Verwer et al. (1989). Another approach consists in a raster scan of the domain: first the chamfer mask is decomposed into disjoint sub-masks; then the domain grid points are scanned in a given order (consistent with the sub-mask construction) and a local computation is performed before being propagated (Rosenfeld and Pfaltz, 1966; Borgefors, 1986). Scanning the domain several times (one per sub-mask) leads to the distance transformation values. Again, we end up with a $O(m \cdot N^n)$ computational cost. Besides specific applications which use the anisotropic nature of the chamfer mask, rotational dependency is usually enforced by increasing the mask size m (its number of vectors, see below) leading to expensive computational costs.

Contributions The goal of this work is to demonstrate that the linear factor in the mask size can be lowered down to a logarithmic one in any dimension for path-based metrics. This is achieved by first detailing and analyzing the separable distance transformation algorithm and briefly recalling the preliminary analysis of Coeurjolly (2014) for the 2D case, before extending it to higher dimensional distance transformation problems. More precisely, we describe efficient and parallel algorithms in arbitrary dimension n to compute error-free distance transformation and Voronoi map for chamfer norms and other path-based metrics. Overall computational costs are summarized in Table 1 (see 3.1 for the predicate definitions).

The article is organized as follows: First, we recall basic definitions and properties of path-based norms (Section 2). In Section 3 we clarify the separable n-dimensional Voronoi map extraction. Section 4 is a short discussion about the complexity of this algorithm for L_p metrics. Then Section 5 is dedicated to the design of a fast implementation of the separable algorithm for path-based metrics. In Section 6, we present and analyse the proposed $n-$dimensional algorithm for path-based metrics.

2. Preliminaries

2.1. Metric space and distance transformation

A metric space (E,F,d) is a set E together with a metric $d : E \times E \to F$ on the set. When E is equal to $\mathbb{Z}^n$ and d is an integer-valued metric, also called *digital metric*, *i.e.* $d : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{Z}$, we say that $(\mathbb{Z}^n, \mathbb{Z}, d)$ is a *digital metric space*. A *digital shape* is a finite subset of $\mathbb{Z}^n$.

Definition 1 (Voronoi Map and Distance Transformation) *For a digital shape $X \subset \mathbb{Z}^n$, the Voronoi map V_X associated with a digital metric space $(\mathbb{Z}^n, \mathbb{Z}, d)$ is the map $X \to \mathcal{P}(\mathbb{Z}^n \backslash X)$ such that $V_X(a) = \arg\min_{b \in \mathbb{Z}^n \backslash X} \{d(a,b)\}$. The distance transformation DT_X is a map $X \to \mathbb{Z}$ such that $DT_X(a) = d(a,b)$ for $b \in V_x(a)$.*

The Voronoi map V_X corresponds to the intersection between the continuous Voronoi diagram for the metric d of points $\mathbb{Z}^n \setminus X$ and the lattice $\mathbb{Z}^n$. Note that $V_X(a)$ may contain several equidistant points to a in $\mathbb{Z}^n \setminus X$. In the following, we consider a *restricted Voronoi map*, denoted Π_X, such that $\Pi_X(a) = b$ with $b \in V_X(a)$ choosen arbitrarily. Π_X are not unique but provide the same distance map DT_X. In the following, we focus on the *restricted Voronoi map* computation and we may omit the word *restricted* for the sake of clarity. Interested readers may refer to (Couprie et al., 2007; Hesselink, 2007) for separable algorithms to compute the complete Voronoi map of Definition 1.

Defining digital metrics spaces, notably in the context of digital image processing, has been the object of many works for the past fourty years. Note that (weighted, with $w_i \geq 0$) L_p metrics

$$d_{L_p}(a,b) = \left(\sum_{k=1}^{n} w_k |a_k - b_k|^p \right)^{\frac{1}{p}}, \qquad (1)$$

define metric spaces $(\mathbb{Z}^n, \mathbb{R}, d_{L_p})$ which are not *digital*. However, rounding up the distance function,

Table 1: Computational cost summary for separable Voronoi map computation on N^n domains (m being the size of the chamfer norm and f the number of row in a H-representation of the mask, see below).

Metric	CLOSEST	HIDDENBY	Sep. Voronoi Map	Reference
L_2	$O(n)$	$O(n)$	$\Theta(n \cdot N^n)$	Hirata (1996)
L_∞	$O(n)$	$O(n)$	$\Theta(n \cdot N^n)$	Meijster et al. (2000)
L_1	$O(n)$	$O(n)$	$\Theta(n \cdot N^n)$	Meijster et al. (2000)
L_p (exact pred.)	$O(n \cdot \log p)$	$O(n \cdot \log p \cdot \log N)$	$O(n^2 \cdot N^n \cdot \log p \cdot \log N)$	Lemma 1
L_p (inexact pred.)	$O(n)$	$O(n \cdot \log N)$	$O(n^2 \cdot N^n \cdot \log N)$	Lemma 1
2D Chamfer norm	$O(\log m)$	$O(\log^2 m)$	$O(\log^2 m \cdot N^2)$	Coeurjolly (2014)
2D Neig. seq. norm	$O(\log m)$	$O(\log^2 m)$	$O(\log^2 m \cdot N^2)$	Normand et al. (2013a) with Coeurjolly (2014)
nD Chamfer norm	$O(n + \log f)$	$O((n + log f) \cdot \log N)$	$O(n \cdot N^n \cdot \log N \cdot (n + \log f))$	Lemma 4

$(\mathbb{Z}^n, \mathbb{Z}, \lceil d_{L_p} \rceil)$ is a digital metric space (Klette and Rosenfeld, 2004). However, it is not precise enough in many situations, and other approaches have been designed. Among them, the family of *path-based metrics* (chamfer norms, -weighted- neighbourhood sequences...) aim at defining digital metrics induced by norms. In the following and for the sake of simplicity, we focus on chamfer norms but similar results can be obtained for more generic path-based metrics such as neighborhood sequences. Elements to support this claim are provided further in the following section.

2.2. Chamfer norms

Definition 2 (Chamfer Mask) *A weighted vector is a pair $(\vec{v}, w)$ with $\vec{v} \in \mathbb{Z}^n$ and $w \in \mathbb{N} \backslash \{0\}$. A chamfer mask $\mathcal{M}$ is a central-symmetric set of weighted vectors with no null vectors and containing at least a basis of $\mathbb{Z}^n$.*

In most situations, vectors of a chamfer mask exhibit axial symmetries. As examples, see Figure 1(a) and (c), where a subset of vectors (together with their weights) defining chamfer masks by symmetries (called generators) are depicted.

From a chamfer mask, we can define a path between two points a and b as a sequence of k points $\{c_i\}$ such that $c_0 = a$, $c_{k-1} = b$ and $c_i \vec{c_{i+1}} = \vec{v_i} \in \mathcal{M}$ for $i \in \{0 \dots k-2\}$. The length of this path is thus the sum of all weights associated with the vectors $\vec{v_i}$ (*i.e.* $\sum w_i$). As $\mathcal{M}$ contains a basis of $\mathbb{Z}^n$, such path between a and b always exists and we can define the chamfer distance between two points a and b in $\mathbb{Z}^n$ as the length of the shortest path between a and b. Since weights are positive integers (see Def. 2), distance values are *scaled* by the weight of the first vector ($(1,0\dots,0)^T$ by convention). Hence, using masks defined in Fig. 1, $\frac{1}{3} \cdot d_{\mathcal{M}_{3-4}}(a,b)$ and $\frac{1}{5} \cdot d_{\mathcal{M}_{5-7-11}}(a,b)$ are approximations of $d_{L_2}(a,b)$.

For all positive weights, a chamfer mask defines a metric. In many shape processing applications, we usually consider a subset of chamfer masks, the *chamfer norms*, with weights such that the induced metrics have convex unit balls, and thus leading to homogeneous distance functions. Chamfer norms can be characterized by a set linear constraints on the mask weights Borgefors (1986); Strand (2008). In the following, we define the size of a chamfer norm simply by the number of vectors of its associated chamfer mask.

Many authors have proposed algorithmic and/or analytic approaches to construct chamfer norms approximating the Euclidean metric. Following Thiel (2001) and Fouard and Malandain (2005), we briefly recall here the classical construction of chamfer norms from Farey sequences since it will be the base of the study proposed in Section 6.

The Farey sequence $\mathscr{F}_m^n$ of dimension n and order m is defined as follows : $\mathscr{F}_m^n = \{(\frac{x_2}{x_1}, \dots, \frac{x_n}{x_1}), \gcd_{i \in 1..n}(x_i) = 1, 0 \leq x_n \leq x_{n-1} \leq \dots \leq x_1 \leq m\}$. Then a Farey sequence $\mathscr{F}_m^n$ is in bijection with all the points $(x_1, \dots, x_n)$ in $\mathbb{Z}^n$,

$0 \leq x_n \leq \cdots \leq x_1 \leq m$ visible from the origin[1]. The vectors $\vec{v}_k$ of a chamfer norm in dimension n can be defined using a subset of a particular $\mathscr{F}_m^n$: the weights w_k are set so that the rational ball $\mathcal{B}_R$ (see Definition 3 below) is convex. By construction, such chamfer masks have axis symmetric unit balls and thus define chamfer norms.

2.3. Distance computation for chamfer norms

To evaluate distances between two digital points for a given chamfer norm, direct formulations have been proposed with a simple geometrical interpretation (Thiel, 2001; Normand and Évenou, 2009), using the so-called rational ball.

Definition 3 (Rational ball, minimal H-representation) *Given a Chamfer norm $\mathcal{M}$, the rational ball associated with $\mathcal{M}$ is the polytope*

$$\mathcal{B}_R = conv\left\{\frac{\vec{v}_k}{w_k};\ (\vec{v}_k, w_k) \in \mathcal{M}\right\}. \tag{2}$$

where conv *denotes the convex hull of a set of points.*

Rational balls for some 2D and 3D chamfer norms are illustrated in Figure 1. As any convex polytope, the rational ball $\mathcal{B}_R$ can also be described as the intersection of f linear constraints in dimension n, f being the number of $(n-1)-$facets of $\mathcal{B}_R$. This is the H-representation of the polytope which can be written in a matrix form:

$$\begin{aligned}\mathcal{B}_R &= \left\{x \in \mathbb{R}^n;\ x = \sum_{k=1}^{m} \alpha_k \left(\frac{\vec{v}_k}{w_k}\right),\ a_k \geq 0, \sum_{1}^{m} \alpha_k = 1\right\} \\ &= \{x \in \mathbb{R}^n;\ Ax \leq y\},\end{aligned}$$

where A is a $f \times n$ matrix and y a vector of n values, so-called the H-coefficients (Ziegler, 2012). The H-representation of a polytope P is with *minimal parameter* if $P = \{x \in \mathbb{Z}^n; Ax \leq y\}$ with A being such that $\forall k \in [1 \ldots f],\ \exists x \in P \quad A_k x = y_k$ (Normand and Évenou, 2009).[2] In other words, A is the minimal H-representation of $\mathcal{B}_R$ if each linear hyperplane of the H-representation of $\mathcal{B}_R$ contains at least one point in $\mathcal{B}_R \cap \mathbb{Z}^n$.

From Normand and Évenou (2009), an important result for distance computation can be summarized as follows:

Proposition 1 (Direct Distance Computation) *Given a chamfer norm $\mathcal{M}$ and (A, y) its minimal parameter H-representation, then for any $a \in \mathbb{Z}^n$, the chamfer distance of a from the origin is*

$$d_{\mathcal{M}}(O, a) = \max_{1 \leq k \leq f} \{A_k a^T\}. \tag{3}$$

Coming back to general path-based digital metrics, (weighted) neighborhood sequences have been proposed to have a better approximation of the Euclidean metric (Rosenfeld and Pfaltz, 1966; Mukherjee et al., 2000; Strand, 2008; Normand et al., 2013a). The main idea is to combine sequences of elementary chamfer norms. A key result has been demonstrated by Normand et al. (2013a) stating that for such distance functions, a minimal parameter polytope representation exists and that distances can be obtained from an expression similar to (3):

$$d(O, a) = \max_{1 \leq k \leq f} \{\gamma_k(A_k a^T)\}, \tag{4}$$

γ_k being some integer sequence characterizing the neighborhood sequence metric. As we will see in the next sections, direct distance computation is key to design an efficient distance transformation algorithm. Similarity of Equations 3 and 4 makes the algorithms presented in the following sections for chamfer norms easily generalizable to neighborhood sequences.

To conclude this preliminary section, algorithms efficiency is characterized by their asymptotic behavior using the $O(\cdot)$ and $\Theta(\cdot)$ notations[3] as a function of the dimension, the number of vectors defining the chamfer norm and the domain size.

3. Separable distance transformation

3.1. Voronoi map from separable approach and metric conditions

Several authors have described optimal in time and separable techniques to compute error-free Voronoi maps or distance transformations for L_2 and L_p metrics (Breu

[1] A point $p \in \mathbb{Z}^n$ is visible from the origin in $\mathbb{Z}^n$ if there is no point of $\mathbb{Z}^n$ on (Op) between O and p.

[2] A_k being the k^{th} row of A.

[3] In a computational model where arithmetic operations and scalar comparisons are constant time with: $f(x) = O(g(x)) \Leftrightarrow \exists C, x_0 \in \mathbb{R}^+, \forall x > x_0, |f(x)| \leq C \cdot g(x)$, and $f(x) = \Theta(g(x)) \Leftrightarrow \exists C, C', x_0 \in \mathbb{R}^+, \forall x > x_0, C \cdot g(x) \leq |f(x)| \leq C' \cdot g(x)$.

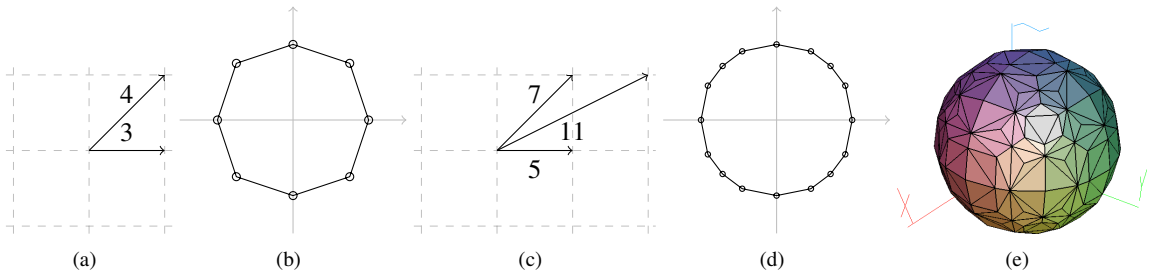

Figure 1: Chamfer masks and rational balls: in dimension 2, generator vectors for the mask $\mathcal{M}_{3-4}$ (a), its rational ball (b). Generator vectors for $\mathcal{M}_{5-7-11}$ (c) and its rational ball (d). In dimension 3, rational ball of a chamfer mask obtained using generator vectors $(x,y,z) \in [\![-3,3]\!]^3$ and weights computed following Fouard and Malandain (2005).

et al., 1995; Hirata, 1996; Meijster et al., 2000; Maurer et al., 2003). Separability means that computations are performed dimension by dimension. In the following, we consider the *Voronoi Map* approach as defined by Breu et al. (1995). Let us first define an hyper-rectangular image $I_X : [1..N_1] \times \ldots \times [1..N_n] \to \{0,1\}$ such that $I_X(a) = 1$ for $a \in [1..N_1] \times \ldots \times [1..N_n]$ iff $a \in X$ ($I_X(a) = 0$ otherwise). The separable algorithm that computes the Voronoi Map for I_x is defined in Algorithm 1 and works on the image *spans* for each dimension. An image span S along dimension q is a vector of N_q points with same coordinates except at their q^{th} one. The q^{th} coordinate of a point $a \in \mathbb{Z}^n$ is denoted by a_q. A given span S in dimension q is denoted by $\{s^i\}_{i=1\ldots N_q}$. In Algorithm 1 the Voronoi map is first initialized by processing each span of the input image along the first dimension in order to create independent 1D Voronoi maps for the metric (lines $5-6$). Then, for each further dimension q, the partial Voronoi map Π_X is updated using one dimensional independent processes on each span along the q^{th} dimension (line 8). Algorithm 2 describes the function VORONOIMAPSPAN. This function is the core of the separable algorithm as it defines the 1D processes to perform on each row, column and higher dimensional image span. In this process, metric information are embedded in the following key predicates (see Fig. 2):

1. CLOSEST(a,b,c): given three points $a,b,c \in \mathbb{Z}^n$ this predicate returns true if $d(a,b) < d(a,c)$;
2. HIDDENBY(a,b,c,S): given a 1D image span S parallel to the q^{th} coordinate axis, and three points $a,b,c \in \mathbb{Z}^n$ such that $a_q < b_q < c_q$, this predicates returns true if there is no $s \in S$ such that

$$d(b,s) < d(a,s) \text{ and } d(b,s) < d(c,s)\,. \qquad (5)$$

Algorithm 1: VORONOIMAP(BINARY MAP I_X)

```
1 Π_X = empty image, same size as I_X;
2 for q in {1...n} do
3     for (x_1,..x_{q-1},x_{q+1},..x_n) in
        [1..N_1] × ..[1..N_{q-1}] × [1..N_{q+1}].. × [1..N_n] do
4         S = {s^i}_{i∈[1..N_q]} where s^i = (x_1..x_{q-1},i,x_{q+1}..x_n);
          // all the coordinates are fixed in S
             except the q^th one
5         if q == 1 then
              // Π_X is initialized span by span
6             Π_X = Π_X ∪ VORONOIMAPSPAN(I_X, q, S);
7         else
              // Π_X is updated along span S
8             Π_X = VORONOIMAPSPAN(Π_X, q, S);
9 return Π_X
```

In other words, HIDDENBY returns true if and only if the Voronoi cells of sites a and c *hide* the Voronoi cell of b along S.

Remark. Note that by construction, for a given span S along dimension q, points a, b, c given as parameters to the HIDDENBY predicate necessarily verify $a_q \neq b_q \neq c_q$. Indeed, these points are defined as the (partial) Voronoi map images of three points s^i, s^j, s^k ($i \neq j \neq k$) of S, therefore having their q^{th} coordinate equal to i, j, and k respectively (see lines 11, 12, 15 of Algorithm 2, and Figure 3 for an illustration).

Algorithm 2: VORONOIMAPSPAN(MAP $\mathcal{M}_X$, DIMENSION q, 1D SPAN S)

Data: q is an integer in $\{1 \dots n\}$;
S is a 1D span along dimension q, with points $\{s^1, \dots, s^{N_q}\}$ sorted by their q^{th} coordinate;
$\mathcal{M}_X$ is either a binary map if $q = 1$ or a partial Voronoi Map.
Result: Partial Voronoi map Π_X updated along S.

1 **if** $q == 1$; // Special case for the first dimension
2 **then**
3 Π_X = empty image, same size as $\mathcal{M}_X$;
4 $k = 0$;
5 **foreach** *point s in S* **do**
6 **if** $\mathcal{M}_X(s) == 0$ **then** // if $s \in \mathbb{Z}^n \backslash X$
7 $L_S[k] = s$;
 // L_S =list of the sites visible on S
8 $k++$;
9 **else**
10 $\Pi_X = \mathcal{M}_X$;
11 $L_S[0] = \mathcal{M}_X(s^1)$;
12 $L_S[1] = \mathcal{M}_X(s^2)$;
13 $k = 2$, $l = 3$;
 // Update the list L_S
14 **while** $l \leq N_q$ **do**
15 $w = \mathcal{M}_X(s^l)$;
16 **while** $k \geq 2$ *and* HIDDENBY$(L_S[k-1], L_S[k], w, S)$ **do**
 // $L_S[k]$ is no longer visible, unstack
17 $k--$;
18 $k++$; $l++$;
19 $L_S[k] = w$;
20 **foreach** *point s in S by increasing* q^{th} *coordinate* **do**
21 **while** $(k < |L_S|)$ *and* CLOSEST$(s, L_S[k+1], L_S[k])$ **do**
 // s is closer to $L_S[k+1]$, look further
22 $k++$;
23 $\Pi_X[s] = L_S[k]$;
24 **return** Π_X

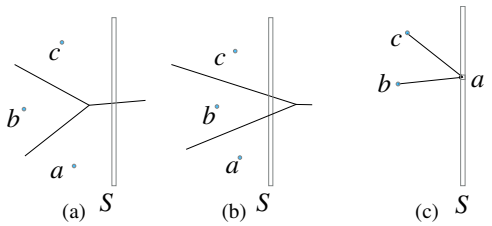

Figure 2: Geometrical predicates for Voronoi map construction Coeurjolly (2014): HIDDENBY(a,b,c,S) returns true in (a) and false in (b) (straight segments correspond to Voronoi diagram edges). (c) illustrates the CLOSEST(a,b,c) predicate for $a \in S$.

For L_1, L_2 and L_∞ metrics, CLOSEST and HIDDENBY predicates can be computed in $O(n)$ in dimension n (Breu et al., 1995; Maurer et al., 2003). Hence, Algorithm 2 is in $O(n \cdot N_q)$ for the dimension q, leading to an overall computational time for the Voronoi Map (Algorihtm 1) and Distance Transformation computations in $\Theta(n^2 \cdot N^n)$ (if we assume that $\forall q \in [1 \dots n], N_q = N$). Note that for L_p metrics, we can derive a $\Theta(n \cdot N^n)$ algorithm as suggested in Hirata (1996); Meijster et al. (2000) using the following observation: when evaluating the CLOSEST predicates in line 21 of Algorithm 2, we compare distances along the 1-D span of dimension q. If we store the partial power p of the distance to the closest site a for each grid point y for previous dimensions (*i.e* the sum $\sum_{i=1}^{q-1}(a_i - y_i)^p$), such distance comparisons can be obtained in $O(1)$. Similar argments can be used for the HIDDENBY predicates of line 16, leading to the overall computational cost in $\Theta(n \cdot N^n)$.

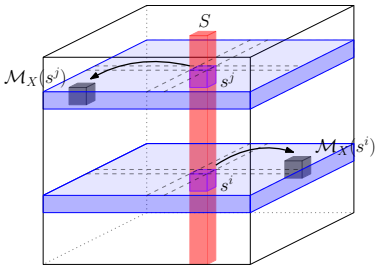

Figure 3: For two points s^i and s^j (in purple) on a span S along dimension q (in red), the partial Voronoi map images $\mathcal{M}_X(s^i)$ and $\mathcal{M}_X(s^j)$ (in black) respectively have i and j as q^{th} coordinate.

Hirata (1996) or Maurer et al. (2003) discussed about conditions on the metric d to ensure that Algorithm 2 is correct. The key property can be informally described as follows: given two points $a, b \in \mathbb{Z}^n$ such that $a_q < b_q$ and a straight line l along the q^{th} direction and if we denote by $v_l(a)$ (resp. $v_l(b)$) the intersection between the Voronoi cell of a (resp. b) and l, then $v_l(a)$ and $v_l(b)$ are simply connected Euclidean segments and $v_l(a)$ appears *before* $v_l(b)$ on l (so called *monotonicity property* by Maurer et al. (2003) and is related to *quadrangle inequality* by Hirata 1996). These contributions are summed up in Definition 4 and Proposition 2.

Definition 4 (Axis symmetric ball norm) *A metric d induced by a norm whose unit ball is symmetric with respect to grid axes is called* axis symmetric ball norm.

Proposition 2 (Metric conditions (Hirata, 1996)) *Algorithm 1 exactly computes the Voronoi Map Π_X of a binary input image I_X for any axis symmetric ball norm.*

Proposition 2 implies that most chamfer norms and neighborhood sequence based norms can also be considered in separable Algorithm 1 (see Fig. 4). However, note that Algorithm 2, and as a by-product Algorithm 1, are exact only if the distance comparison predicate is exact, i.e. if we can compare distances, through the CLOSEST and HIDDENBY predicates, without error.

Furthermore, computational efficiency of the algorithm requires the design of efficient algorithmic tools to implement these predicates, and this the purpose of the next sections.

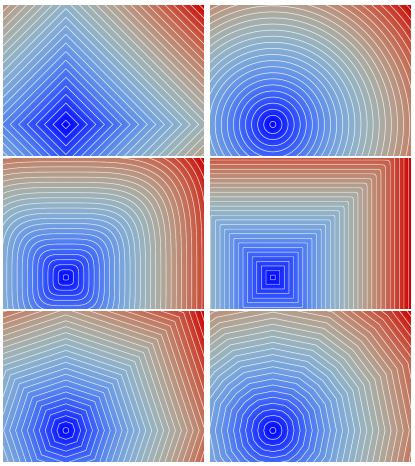

Figure 4: Distance transformation from a single source for different metrics satisfying Definition 4 and thus Proposition 2: *(from left to right)* L_1, L_2, L_4, L_{80}, $\mathcal{M}_{3-4}$ and $\mathcal{M}_{5-7-11}$.

3.2. Generic predicates and complexity analysis for axis symmetric ball norms

We first detail the overall computational cost of Algorithms 2 and 1. We assume in the following that $\forall q \in [1 \dots n], N_q = N$.

Lemma 1 (Maurer et al. 2003; Coeurjolly 2014) *Let $(\mathbb{Z}^n, F, d)$ be a metric space induced by a norm with axis symmetric unit ball. If C denotes the computational cost of* CLOSEST *predicate and H is the computational cost of the* HIDDENBY *predicate, then Algorithm 2 is in $O(N \cdot (C+H))$, leading to a complexity of $O(n \cdot N^n \cdot (C+H))$ for Algorithm 1.*

For a given axis symmetric ball norm d, generic Algorithms 3, 4 and 5 were defined in Coeurjolly (2014). Note that these algorithms are valid for any dimension n. The computational cost of the CLOSEST predicate is simply the one of a distance evaluation. As a first approach, Algorithms 4 and 5 show that the HIDDENBY predicate can be obtained by a binary search on the 1D image span S to localize the abscissa of Voronoi edges of sites $\{a,b\}$ and $\{b,c\}$.

Algorithm 3: Generic CLOSEST$(a,b,c \in \mathbb{Z}^n)$.

```
1 return d(a,b) < d(a,c);
```

Algorithm 4: Generic VORONOIEDGE$(a,b,s^i,s^j \in \mathbb{Z}^n)$ with $i < j$, $a_q < b_q$.

```
1  if (j − i = 1) then
2      if i = 1 and CLOSEST(s^i, b, a) then
3          return −∞;
4      if i = N_q and CLOSEST(s^i, a, b) then
5          return ∞;
6      return i;
7  mid = i + (j − i)/2;
8  if CLOSEST(s^mid, a, b) then
       // s_mid closer to a
9      return VORONOIEDGE(a, b, s^mid, s^j)
10 else
       // s_mid closer to b
11     return VORONOIEDGE(a, b, s^i, s^mid)
```

Algorithm 5: Generic HIDDENBY$(a,b,c \in \mathbb{Z}^n; S$ in the q^{th} direction) with $a_q < b_q < c_q$.

```
1 v_ab = VORONOIEDGE (a, b, s^1, s^{N_q});
2 v_bc = VORONOIEDGE (b, c, s^1, s^{N_q});
3 return (v_ab > v_bc);
```

The complexity H of Algorithm 5 can be expressed as a function of the complexity C of Algorithm 3, leading to the general result below:

Lemma 2 (Coeurjolly 2014) *Let $\mathcal{M}$ be a chamfer norm with axis symmetric unit ball in dimension n whose rational ball has f facets, Algorithm 1 can be implemented with a computational complexity of $O(n \cdot N^n \cdot C \cdot \log N)$, where N^n is the size of the image.*

4. Distance transformation for L_p metrics

As a direct consequence of Lemma 1, we briefly derive computational costs for L_p metrics. For such metrics, as discussed in Section 3.1, the CLOSEST and HIDDENBY predicates are in $O(n)$ for $p = \{1,2,\infty\}$ with exact integer only computations (Maurer et al., 2003; Meijster et al., 2000). We thus have distance transformation algorithms in $\Theta(n^2 \cdot N^n)$. Let us now show that Algorithm 3 and 5 lead to a faster algorithm for any $p \geq 1$.

For $p \in \mathbb{R}$, $p \geq 1$, we can use approximations of the evaluation of distances on IEEE 754 *double* and then consider the Generic HIDDENBY predicate in $O(n \cdot \log N)$ (Alg. 5). As predicates being based on floating point computations, numerical issues may occur but we have an $O(n^2 \cdot N^n \cdot \log N)$ distance transformation algorithm (L_p inexact predicates in Table 1). If $p \in \mathbb{Z}$, $p \geq 3$, we use exact integer number based computations of distances storing sum of power p quantities (which can be computed in $O(n \cdot \log p)$ thanks to exponentiation by squaring). The HIDDENBY predicate is also based on Algorithm 5, leading to an $O(n^2 \cdot N^n \cdot \log p \cdot \log N)$ distance transformation algorithm (L_p exact predicates in Table 1).

5. Distance transformation in higher dimension for chamfer norms

In this section, we consider digital metrics given by a chamfer norms and propose an efficient algorithm to compute the separable distance transformation for such metrics in nD. In Coeurjolly (2014), the structure of the rational ball of a chamfer mask in dimension 2 was used to obtain an $O(\log m)$ algorithm for the CLOSEST predicate, and an $O(\log^2 m)$ one for the HIDDENBY predicate, leading to an overall $O(\log^2 m \cdot N^2)$ algorithm for the separable distance transformation (see Table 1). The following sections extend these results to higher dimensions.

5.1. Definitions and general principle

Let us consider a general chamfer norm in arbitrary dimension n with m weighted chamfer vectors. As explained in Section 2, these vectors define a rational ball $\mathcal{B}_R$ (see Definition 3), the center of which can be any point p. We define a *wedge* (p, f_k) as the conical hull of p (called the apex) and the vertices of a given facet f_k of $\mathcal{B}_R$ ($k \in \{1..f\}$ if $\mathcal{B}_R$ has f facets). Thus, to each wedge is associated a row A_k of the matrix of the minimal H-representation of $\mathcal{B}_R$. Note that A_k can also be seen as a - non-unitary - normal vector to the facet f_k Normand and Évenou (2009). In the following, given a point a and a point p, we denote by $(a, \mathcal{F}^a(p))$ the wedge of apex a containing point p, $\mathcal{F}^a(p)$ being one facet of $\mathcal{B}_R$ (see Figure 5).

Using similar notations, Thiel (2001) and Strand (2008) demonstrated that distance evaluation between a point a and a point p can be obtained in two steps: first, compute the wedge $(a, \mathcal{F}^a(p) = f_k)$; then

$$d_{\mathcal{M}}(a,p) = A_k.(p-a)^T \,. \tag{6}$$

Thus, implementing the CLOSEST predicate comes down to computing the wedge a given point belongs to. In 2D, it was shown in Coeurjolly (2014) that a binary search over the chamfer vectors was enough. The nD case is discussed in section 5.2.

Let us now see how to optimize the HIDDENBY predicate, which comes down to optimizing the VORONOIEDGE function. Given two points a and b ($a_q < b_q$) and a 1D image span S along the q^{th} dimension, we have to find the point e of S ($e \in \mathbb{Z}^n$) with abscissa e_q such that all the points of S of abscissa lower than e_q are in the Voronoi cell of a while all the points with a greater abscissa are in the Voronoi cell of b. To compute e_q let us define $l(S)$ as the one-dimensional flat that contains all the points of S and consider the Euclidean point $\xi \in l(S)$ satisfying:

$$d_{\mathcal{M}}(a,\xi) = d_{\mathcal{M}}(b,\xi)\,. \tag{7}$$

In other words, we are looking for the point in $l(S)$ which is equidistant to a and b. To compute ξ, we first suppose that we know the two wedges $(a, \mathcal{F}^a(\xi) = f_k)$ and $(b, \mathcal{F}^b(\xi) = f_l)$ (see Fig. 6$-(b)$). In this situation, ξ is the solution of

$$A_k \cdot (\xi - a)^T = A_l \cdot (\xi - b)^T \ . \tag{8}$$

Note that since $\xi \in l(S)$, we have one linear equation with only one unknown, ξ_q. As a consequence, if we know the two wedges point ξ belongs to, we have its q^{th} coordinate ξ_q in $O(1)$. As A is the minimal representation of $\mathcal{B}_R$, it has rational components and thus ξ has integer coordinates execpt ξ_q which is rational. Finally, the integer point $e \in S$ is given such that $e_q = \lfloor \xi_q \rfloor$.

The next section is dedicated to the CLOSEST predicate while in section 5.3 we detail how to efficiently compute the wedges $(a, \mathcal{F}^a(\xi) = f_k)$ and $(b, \mathcal{F}^b(\xi) = f_l)$, summarized in Algorithm 7 (see also Fig. 6$-(b)$).

5.2. CLOSEST *predicate and first results*

To begin with, let us discuss about the combinatorics of the chamfer norm rational ball. If m denotes the number of weighted vectors of $\mathcal{M}$, its rational ball $\mathcal{B}_R$ has $O(m^{\lfloor \frac{n}{2} \rfloor})$ $i-$facets (for all $0 \leq i \leq n$) (de Berg et al., 2000). If f denotes the number of $(n-1)-$facets of $\mathcal{B}_R$, then we have:

Lemma 3 *Let $\mathcal{M}$ be a chamfer norm whose rational ball $\mathcal{B}_R$ has f $(n-1)-$facets in dimension n, then distance computation and thus* CLOSEST *predicate are in (amortized) $O(n + \log f)$ with $O\left(\frac{f^{\lfloor \frac{n}{2} \rfloor}}{(\log f)^{\lfloor \frac{n}{2} \rfloor - \delta}}\right)$ space and preprocessing time*[4].

Proof:. Similarly to the 2D case, the distance $d_{\mathcal{M}}(O,a)$ for $a \in \mathbb{Z}^n$ is given by first solving a ray-shooting problem on convex polytopes which consists in first computing the $(n-1)$-facet of $\mathcal{B}_R$ pierced by the ray (O,a) (see Fig. 5). Once the facet is obtained, the associated A_k row is used to evaluate $d_{\mathcal{M}}(O,a) = A_k \cdot a^T$ in $O(n)$. Following Matousek and Schwarzkopf (1993) Theorem 10, such a ray-shooting query on convex polytopes can be solved in $O(\log f)$ thanks to a preprocessing in $O\left(\frac{f^{\lfloor \frac{n}{2} \rfloor}}{(\log f)^{\lfloor \frac{n}{2} \rfloor - \delta}}\right)$. In the case when the ray hits a facet of dimension strictly lower than $n-1$, the algorithm returns one of the adjacents $(n-1)$-facets. Propositions 3 and 4 from Normand and Évenou (2009) ensure that the choice of any $(n-1)$-facet leads to the same distance evaluation. Please note also that the preprocessing time is roughly equivalent to the convex hull computation in higher dimension which is in $O(f^{\lfloor \frac{n}{2} \rfloor})$. Hence, preprocessing for ray-shooting can be done while computing the rational ball $\mathcal{B}_R$ using Eq. (2). □

Algorithm 4 being valid in any dimension, we can merge this result with Lemma 2 to straightforwardly obtain the result below:

Lemma 4 *Let $\mathcal{M}$ be a chamfer norm whose rational ball $\mathcal{B}_R$ has f $(n-1)-$facets in dimension n, separable exact Voronoi Map Π_X can be obtained in $O(n \cdot N^n \cdot \log N \cdot (n + \log f))$, thanks to a preprocessing in $O\left(\frac{f^{\lfloor \frac{n}{2} \rfloor}}{(\log f)^{\lfloor \frac{n}{2} \rfloor - \delta}}\right)$.*

However, we show below that we can still expect faster VORONOIEDGE function even in higher dimension.

5.3. *Improved* HIDDENBY *predicate*

In dimension 2, it was shown in Coeurjolly (2014) that it was possible to reduce the complexity from a logarithmic factor on the size N of the image to a logarithmic factor on the size m of the mask using binary search over chamfer vectors. This process cannot be extended straighforwardly in higher dimensions since chamfer vectors cannot be ordered to perform a binary search anymore. However, it is interesting to notice that, whatever the dimension n, vectors from a given point a to any point of a span S lie in the smallest affine subspace containing a and the one-dimensional flat $l(S)$.

In the general case where a does not lie on S, this is actually always a 2-flat, denoted by $\mathcal{P}$, and the intersection of this 2-flat with the rational ball $\mathcal{B}_R$ is a polygon (see Fig. 5). The vertices of this polygon could be used to define a set of vectors on which a binary search could be performed as in the 2D case Coeurjolly (2014). However, since a 2-flat is actually the intersection of $n-2$ hyperplanes, computing this polygon comes down to intersecting a n-polytope with $n-2$ hyperplanes. By duality, each

[4] δ is an arbitrarily small positive constant.

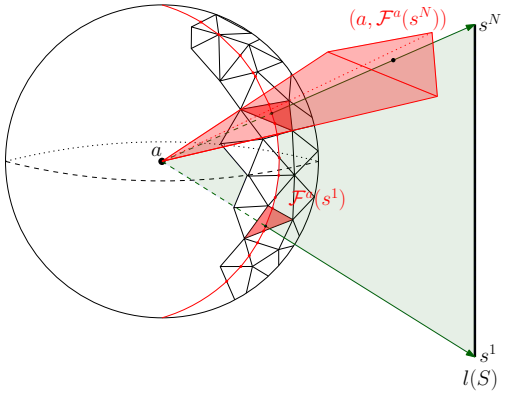

Figure 5: Vectors $s^i - a$ lie in a 2-flat defined by $l(S)$ and a, in light green. The distance $d_{\mathcal{M}}$ between a point on S and a is computed via a ray shooting that returns the $n-1$-facet of $\mathcal{B}_R$ traversed by the ray $s^i - a$: wedge $(a, \mathcal{F}^a(s^N))$ and facet $\mathcal{F}^a(s^1)$ are depicted in light red.

of these operations is equivalent to a convex hull computation, with a complexity of $\mathcal{O}(f^{\lfloor n/2 \rfloor})$ (Chazelle, 1993; Bajaj and Pascucci, 1996). As a consequence, in order to make the approach efficient, we must avoid to compute explicitly this intersection.

This is achieved by rewriting the VORONOIEDGEWEDGE function as presented in Algorithm 6. As before, let $e \in \mathbb{Z}^n$ be the point on S such that all the points of S of abscissa lower than e_q are in the Voronoi of a while all the points with a greater abscissa are in the Voronoi cell of b. Given a, the goal of this function is to find the wedge of $\mathcal{B}_R$ (of apex a) e belongs to.

The algorithm computes two points s^i and s^j such that s^i belongs to the Voronoi cell of a (if $a_q < b_q$, b otherwise), s^j to the Voronoi cell of b (if $a_q < b_q$, a otherwise) and either $j - i = 1$ or $\mathcal{F}^a(s^i) = \mathcal{F}^a(s^j)$. Similarly to generic Algorithm 4, this is done by performing a binary search over the points of S, with a key difference on exit conditions: now, the algorithm does not wait until point e is found, but exits as soon as the two points s^i and s^j belong to the same wedge (line 1). Indeed, by convexity of a wedge, this implies that any point on S between s^i and s^j - and in particular e - also belongs to the same wedge. Correctness of the algorithm is ensured by maintaining two invariants : (i) s^i is lower than s^j on span S ($s^i{}_q < s^j{}_q$) ; (ii) if $a_q < b_q$, s^i is in a's Voronoi cell, s^j in b's Voronoi cell, and conversely if $b_q < a_q$.

Figure 6 illustrates the first step of the binary search in (a), and the situation at the end of the search in (b) (projection in plane $\mathcal{P}$).

Algorithm 6: VORONOIEDGEWEDGEND($a, b \in \mathbb{Z}^n; i, j \in \mathbb{Z}$, $i < j$; span S along the q^{th} direction; faces $f_{k_i} = \mathcal{F}^a(s^i)$, $f_{k_j} = \mathcal{F}^a(s^j)$)

```
1  if f_{k_i} = f_{k_j} or j − i = 1 then
2      return f_{k_i};
3  else
4      mid = i + (j − i)/2;
       // Check whether s^mid is closest to a or to b
       // O(n + log f) evaluation of distances w.r.t.
          a and b
5      Compute f_k = F^a(s^mid); d(a, s^mid) = A_k · (s^mid − a)^T;
6      Compute f_l = F^b(s^mid); d(b, s^mid) = A_l · (s^mid − b)^T;
7      if d(a, s^mid) < d(b, s^mid) then
8          if a_q < b_q then
9              return VORONOIEDGEWEDGEND(a, b, mid, j, S, f_k, f_{k_j})
10         else
11             return VORONOIEDGEWEDGEND(a, b, i, mid, S, f_{k_i}, f_k)
12     else
13         if a_q < b_q then
14             return VORONOIEDGEWEDGEND(a, b, i, mid, S, f_{k_i}, f_k)
15         else
16             return VORONOIEDGEWEDGEND(a, b, mid, j, S, f_k, f_{k_j})
```

It remains now to use Algorithm 6 to compute point e. Algorithm 7 implements the VORONOIEDGEND function as the nD counterpart of the VORONOIEDGE function of Coeurjolly (2014). First, lines 1 to 11 are dedicated to checking whether the bisector of a and b crosses span S or not. If it does, then wedges $(a, \mathcal{F}^a(e))$ $(b, \mathcal{F}^b(e))$ are computed calling the VORONOIEDGEWEDGEND function (see Figure 6(b) for an illustration).

Proposition 3 *Let $\mathcal{M}$ be a chamfer norm in dimension n whose rational ball $\mathcal{B}_R$ has f $(n-1)-$facets. Let W be the computational time complexity of the* VORONOIEDGEWEDGEND *function. Then, the separable exact Voronoi Map can be obtained in (amortized) $O(n \cdot N^n \cdot (n + \log f + W))$ with a $O\left(\frac{f^{\lfloor \frac{n}{2} \rfloor}}{(\log f)^{\lfloor \frac{n}{2} \rfloor - \delta}}\right)$ space*

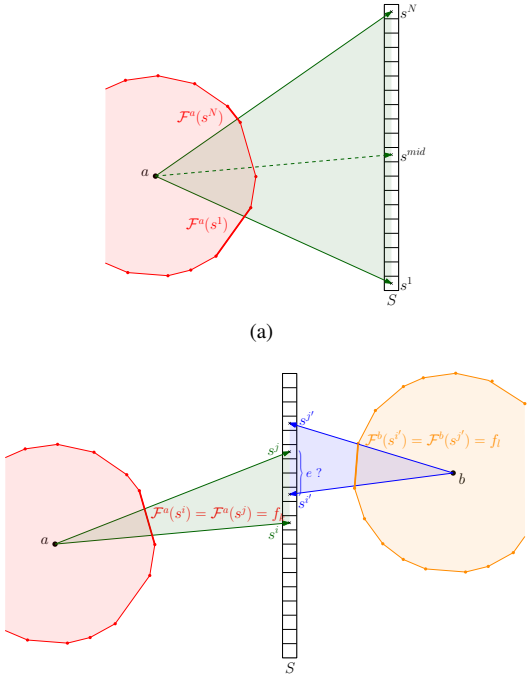

Figure 6: View of the 2-flat $\mathcal{P}$: (a) Binary search initialization to compute the wedge $(a, \mathcal{F}^a(e))$. (b) After completion of two binary searches, both wedges $(a, f_k = \mathcal{F}^a(e))$ and $(a, f_l = \mathcal{F}^b(e))$ are known.

and preprocessing time. More precisely, the worst-case complexity W being $\mathcal{O}((n+\log f)\cdot \log N)$, this leads to a global (amortized) complexity of $O(n\cdot N^n \cdot \log N \cdot (n+\log f))$ (same preprocessing).

Proof:. Following Lemma 1, the generic separable algorithms computes the Voronoi map in $O(n\cdot N^n\cdot(C+H)$. Lemma 3 states that $C = O(n+\log f)$ with a $O\left(\frac{f^{\lfloor \frac{n}{2} \rfloor}}{(\log f)^{\lfloor \frac{n}{2} \rfloor - \delta}}\right)$ space and preprocessing time. Remains to evaluate H, *i.e.* the complexity of the VORONOIEDGEND function. In Algorithm 7, the first eleven lines are in $O(C)$ since only distance computations are involved. Lines 12 and 13 are calls to the VORONOIEDGEWEDGEND function, with a complexity in $O(W)$. In the worst case, we have $W = \mathcal{O}((n+\log f)\cdot \log N)$ thanks to the test $j-i=1$ on line 1 of Algorithm 6. Last, the system to solve in line 14 has only one unkwown ξ_q since ξ belongs to the one-dimensional span S, with a complexity of $O(1)$.

Algorithm 7: VORONOIEDGEND$(a,b\in\mathbb{Z}^n$, span $S)$.

```
   // Check that the bisector of a and b crosses
      span S
1  Compute f_{k_1} = F^a(s^1), f_{l_1} = F^b(s^1);
2  d(a,s^1) = A_{k_1}·(s^1 − a)^T; d(b,s^1) = A_{l_1}·(s^1 − b)^T;
3  if (a_q < b_q and d(b,s^1) < d(a,s^1)) ; // or (b_q < a_q and
     d(a,s^1) < d(b,s^1))
4  then
5  |  Bisector does not cross S.
6  else
7  |  Compute f_{k_N} = F^a(s^N), f_{l_N} = F^b(s^N);
8  |  d(a,s^N) = A_{k_N}·(s^N − a)^T; d(b,s^N) = A_{l_N}·(s^N − b)^T;
9  |  if (a_q < b_q and d(a,s^N) < d(b,s^N)) ; // or (b_q < a_q
   |    and d(b,s^N) < d(a,s^N))
10 |  then
11 |  |  Bisector does not cross S.
   // Compute e
12 f_k =VORONOIEDGEWEDGEND(a,b,1,N,S,f_{k_1},f_{k_N});
13 f_l =VORONOIEDGEWEDGEND(b,a,1,N,S,f_{l_1},f_{l_N});
14 Compute abscissa ξ_q of the point ξ ∈ S such that
     A_k·(ξ − a)^T = A_l·(ξ − b)^T;
15 return⌊ξ_q⌋;
```

□

Note that in the worst-case, this approach does not improve the result presented in Lemma 4 (using the generic VORONOIEDGE of Algorithm 4). However, in Section 6, we give some experimental insights on a finer analysis of the complexity W under distribution hypothesis.

6. Experimental analysis

6.1. Insights on the complexity in dimension n

The complexity W of Algorithm 7 depends on the number of recursion steps done until points s^i and s^j are in the same wedge. Wedges being defined by the $n-1$ dimensional faces of $\mathcal{B}_R$, this complexity depends on the distribution of the chamfer vectors defining $\mathcal{B}_R$.

Let us denote by P the intersection between the $(n-1)$-faces of $\mathcal{B}_R$ and the 2-flat $\mathcal{P}$ (see the red polygon on Fig. 5). Note that $\mathcal{P}$ goes through the center of $\mathcal{B}_R$. If we assume that the vectors defining $\mathcal{B}_R$ are uniformely distributed on the unit sphere $\mathbb{S}^n$ and that the faces of P are also uniformely distributed on $\mathcal{B}_R \cap \mathcal{P}$, then we can expect that $W = \mathcal{O}((n+\log f)\cdot \log |P|)$. Even if studying precisely these questions is out of scope of this work, in

the following we give insights on both the relevance of these assumptions and the behaviour of $|P|$ in the context of chamfer norms.

6.1.1. *Some observations on the distribution hypothesis*

To study the distribution of chamfer vectors, we consider chamfer masks where vectors are defined from a subset of Farey sequences, as presented in Section 2 (see also Thiel (2001) and Fouard and Malandain (2005)). Studying the distribution of such sets of vectors is a field of research in itself, and we simply mention below several results relevant to our context.

First, it is well-known from Marklof (2013); Marklof and Strömbergsson (2015) that n-dimensional lattice points visible from the origin have a constant density in $\mathbb{R}^n$. Moreover, Boca et al. (2000) studied in the 2D case the distribution of the angles of straight lines from the origin through visible points. More precisely, they study the proportion of differences between consecutive angles which are larger than the average: they show that this proportion is smaller than what is expected for a random distribution, and give an explicit formulation of the repartition function. Similar results in higher dimension remain an open question.

These results tend to support the hypothesis of a uniform distribution of the vectors of $\mathcal{B}_R$, but the question of the distribution of the faces of the polygon P has not been investigated to our knowledge.

6.1.2. *Experimental behaviour of $|P|$*

In this part, we investigate the number of faces of P when $\mathcal{B}_R$ is a rational ball defined from Farey Sequences. The results are presented in Figure 7 and we detail below how the rational balls are generated, how the 2-flats $\mathcal{P}$ are selected, and how the intersection between $\mathcal{B}_R$ and $\mathcal{P}$ is performed.

In the four subfigures of Figure 7, rational balls are defined from Farey sequences:

- In (b-d), the vectors of $\mathcal{B}_R$ are all normalized vectors of a Farey sequence of order m (the higher the order, the greater the number of vertices - and $(n-1)$-faces - of $\mathcal{B}_R$). The order of the Farey sequences ranges from 1 to 10 in (b-c), from 1 to 6 in (d);
- in (a), $\mathcal{B}_R$ is computed thanks to the algorithm presented by Fouard and Malandain (2005).[5] Given a (odd) mask size m, and a maximal error ε, the algorithm computes a subset of vectors of $\mathcal{F}_{\frac{m-1}{2}}$ and weights such that the rational ball $\mathcal{B}_R$ is convex and the error with respect to the optimal theoretical error expected (wrt the Euclidean distance) for this mask size is below ε.

Once the sets of vectors defined, we use Qhull (Barber et al., 1996) to compute both the rational ball itself and its intersection with a 2-flat $\mathcal{P}$ that goes through the center of $\mathcal{B}_R$. This intersection is performed by randomly picking the coefficients of $n-2$ $(n-1)$-hyperplanes containing the center of $\mathcal{B}_R$, and iteratively adding each $(n-1)$-hyperplane. The vertices of P are the points lying on all $(n-1)$-hyperplanes.[6]

For each rational ball, a certain number of cuts is performed: from 1000 in dimension 3 to only 6 in dimension 5 for rational balls obtained from Farey sequences of order 5 and 6 (due to precision issues in Qhull). 95% confidence intervals are depicted for each point (*i.e.* for each rational ball) as error bars, but most of the time too small to be visible on the graphs. Note that this remark suggests that the size of $|P|$ does not depend on the position of $\mathcal{P}$, thus supporting the uniform distribution hypothesis discussed in the previous section.

Analysing these results, we see that $|P|$ seems to behave as f^α, with $\alpha < 0$ and decreasing when the dimension increases. This suggests that, in practice, the complexity W of Algorithm 7 is expected to be $O((n+\log f)\cdot \log f)$. Similarly to dimension 2 (Coeurjolly, 2014), this approach is expected to lower down the worst case complexity of the computation of the distance transformation for chamfer norms in dimension n from a logarithmic factor on the size N of the image, to a logarithmic factor on the size f of the rational ball.

6.2. *Distance transformation in dimension 2*

We evaluate the performance of the separable ap-

[5] Code is available on the TC18 website www.tc18.org/code_data_set/code.php

[6] Python code used to generate Farey sequences and to compute these graphs is available on http://www.gipsa-lab.fr/~isabelle.sivignon/recherches_en.html.

proach to compute restricted Voronoi diagrams and distance transformation for chamfer norms in dimension 2. First, we observe that using Algorithm 1 with the nD VORONOIEDGE (Alg. 7), we obtain an overall complexity in $O(\log m \cdot \log N \cdot N^2)$ which is close to the $O(\log^2 m \cdot N^2)$ complexity of the ad-hoc 2D version of the problem (Coeurjolly, 2014). These complexities have to be compared with the $O(m \cdot N^2)$ complexity of the classical raster scan approach for chamfer norms (Borgefors, 1986). In Fig. 8-(a), we first illustrate some restricted Voronoi map results on small domains. In Fig.8-$(b-c)$, we have considered a 2D domain 2048^2 with 2048 random sites. First, we observe that fixing N, the $\log^2 m$ term is clearly visible in the computational cost of the Voronoi map (single thread curve). Bumps in the single thread curve may be due to memory cache issues. Please note that if we consider classical chamfer norm DT from raster scan (and sub-masks), the computational cost is in $O(m \cdot N^2)$ and thus has a linear behavior (green curves in Fig. 8-$(b-c)$). Since we have a separable algorithm, we can trivially implement it in a multi-thread environment. Hence, on a bi-processor and quad-core (hyper-threading) Intel(R) Xeon(R) cpu (16 threads can run in parallel), we observe a speed-up by a factor 10 (orange curves in Fig. 8-$(b-c)$).

Implementation of all separable algorithms are publicly available in the DGtal library (dgt).

7. Conclusion and Discussion

In this article, we have proposed generic algorithms to efficiently solve the restricted Voronoi map and distance transformation problems for a large class of metrics in any dimension. Focusing on chamfer norms, geometrical interpretation of this generic approach allows us to design an algorithm with logarithmic factors in the chamfer mask size compared to a linear one for previous approaches. Thanks to separability, parallel implementation of the distance transformation leads to efficient distance computation for path based metrics.

For the L_2 metric,(additively) weighted voronoi maps, also known as power maps, can be used to solve the reverse distance transformation and medial axis extraction problem using similar separable techniques (Coeurjolly and Montanvert, 2007). A challenging future work would be to extend these results for path-based norms such as chamfer norms.

References

, . DGTAL: Digital geometry tools and algorithms library. http://dgtal.org.

Bajaj, C.L., Pascucci, V., 1996. Splitting a complex of convex polytopes in any dimension, in: Proceedings of the Twelfth Annual Symposium on Computational Geometry, ACM. pp. 88–97.

Barber, C.B., Dobkin, D.P., Huhdanpaa, H., 1996. The quickhull algorithm for convex hulls. ACM Trans. Math. Softw. 22, 469–483. URL: www.qhull.org, doi:10.1145/235815.235821.

Boca, F.P., Cobeli, C., Zaharescu, A., 2000. Distribution of lattice points visible from the origin. Communications in Mathematical Physics 213, 433–470.

Borgefors, G., 1986. Distance transformations in digital images. Computer Vision, Graphics, and Image Processing 34, 344–371.

Borgefors, G., Nyström, I., 1997. Efficient shape representation by minimizing the set of centres of maximal discs/sphere. Pattern Recognition Letters 18, 465–472.

Breu, H., Gil, J., Kirkpatrick, D., Werman, M., 1995. Linear time euclidean distance transform algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 529–533.

Chazelle, B., 1993. An Optimal Convex Hull Algorithm in Any Fixed Dimension. Discrete & Computational Geometry 10, 377–409.

Coeurjolly, D., 2012. Applications of Discrete Geometry and Mathematical Morphology. Springer-Verlag. volume 7346 of *LNCS*. chapter Volumetric Analysis of Digital Objects Using Distance Transformation: Performance Issues and Extensions.

Coeurjolly, D., 2014. 2D subquadratic separable distance transformation for path-based norms, in: Discrete Geometry for Computer Imagery. Springer. volume 8668 of *LNCS*, pp. 75–87.

Coeurjolly, D., Montanvert, A., 2007. Optimal separable algorithms to compute the reverse euclidean distance transformation and discrete medial axis in arbitrary dimension. IEEE Transactions on PAMI 29, 437–448.

Couprie, M., Coeurjolly, D., Zrour, R., 2007. Discrete bisector function and Euclidean skeleton in 2D and 3D. Image and Vision Computing 25, 1543–1556. URL: http://linkinghub.elsevier.com/retrieve/pii/S0262885606003064.

Danielsson, P.E., 1980. Euclidean distance mapping. Computer Graphics and Image Processing 14, 227–248.

de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O., 2000. Computational Geometry. Springer-Verlag.

Fouard, C., Malandain, G., 2005. 3-D chamfer distances and norms in anisotropic grids. Image and Vision Computing 23, 143–158.

Hesselink, W.H., 2007. A linear-time algorithm for euclidean feature transform sets. Information Processing Letters 102, 181–186.

Hirata, T., 1996. A unified linear-time algorithm for computing distance maps. Information Processing Letters 58, 129–133.

Klette, R., Rosenfeld, A., 2004. Digital Geometry: Geometric Methods for Digital Picture Analysis. Series in Computer Graphics and Geometric Modelin, Morgan Kaufmann.

Marklof, J., 2013. Fine-Scale Statistics for the Multidimensional Farey Sequence. Springer Berlin Heidelberg. pp. 49–57.

Marklof, J., Strömbergsson, A., 2015. Visibility and directions in quasicrystals. International Mathematics Research Notices 15, 6588–6617.

Matousek, J., Schwarzkopf, O., 1993. On ray shooting in convex polytopes. Discrete & Computational Geometry 10, 215–232.

Maurer, C.R., Qi, R., Raghavan, V., 2003. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 265–270.

Meijster, A., Roerdink, J.B.T.M., Hesselink, W.H., 2000. A general algorithm for computing distance transforms in linear time, in: Mathematical Morphology and its Applications to Image and Signal Processing, Kluwer. pp. 331–340.

Mukherjee, J., Das, P.P., Kumarb, M.A., Chatterjib, B.N., 2000. On approximating euclidean metrics by digital distances in 2D and 3D. Pattern Recognition Letters 21, 573–582.

Normand, N., Évenou, P., 2009. Medial axis lookup table and test neighborhood computation for 3d chamfer norms. Pattern Recognition 42, 2288–2296.

Normand, N., Strand, R., Évenou, P., 2013a. Digital distances and integer sequences, in: González-Díaz, R., Jiménez, M.J., Medrano, B. (Eds.), DGCI. Springer. volume 7749 of *LNCS*, pp. 169–179.

Normand, N., Strand, R., Évenou, P., Arlicot, A., 2013b. Minimal-delay distance transform for neighborhood-sequence distances in 2d and 3d. Computer Vision and Image Understanding 117, 409–417.

Normand, N., Strand, R., Évenou, P., Arlicot, A., 2014. A streaming distance transform algorithm for neighborhood-sequence distances. IPOL Journal 4, 196–203. URL: https://doi.org/10.5201/ipol.2014.68, doi:10.5201/ipol.2014.68.

Ragnemalm, I., 1993. The Euclidean Distance Transform. Ph.D. thesis. Linköping University, Linköping, Sweden.

Remy, E., Thiel, E., 2002. Medial axis for chamfer distances: computing look-up tables and neighbourhoods in 2D or 3D. Pattern Recognition Letters 23, 649–661.

Rosenfeld, A., Pfaltz, J., 1968. Distance functions on digital pictures. Pattern Recognition 1, 33–61.

Rosenfeld, A., Pfaltz, J.L., 1966. Sequential operations in digital picture processing. Journal of the ACM 13, 471–494.

Saha, P.K., Borgefors, G., di Baja, G.S., 2016. A survey on skeletonization algorithms and their applications. Pattern Recognition Letters 76, 3–12.

Strand, R., 2008. Distance Functions and Image Processing on Point-Lattices With Focus on the 3D Face- and Body-centered Cubic Grids. Phd thesis. Uppsala Universitet.

Thiel, E., 2001. Géométrie des distances de chanfrein. Ph.D. thesis. Aix-Marseille 2.

Verwer, B.J.H., Verbeek, P.W., Dekker, S.T., 1989. An efficient uniform cost algorithm applied to distance transforms. IEEE Transactions on Pattern Analysis and Machine Intelligence 11, 425–429.

Ziegler, G.M., 2012. Lectures on polytopes. volume 152. Springer Science & Business Media.

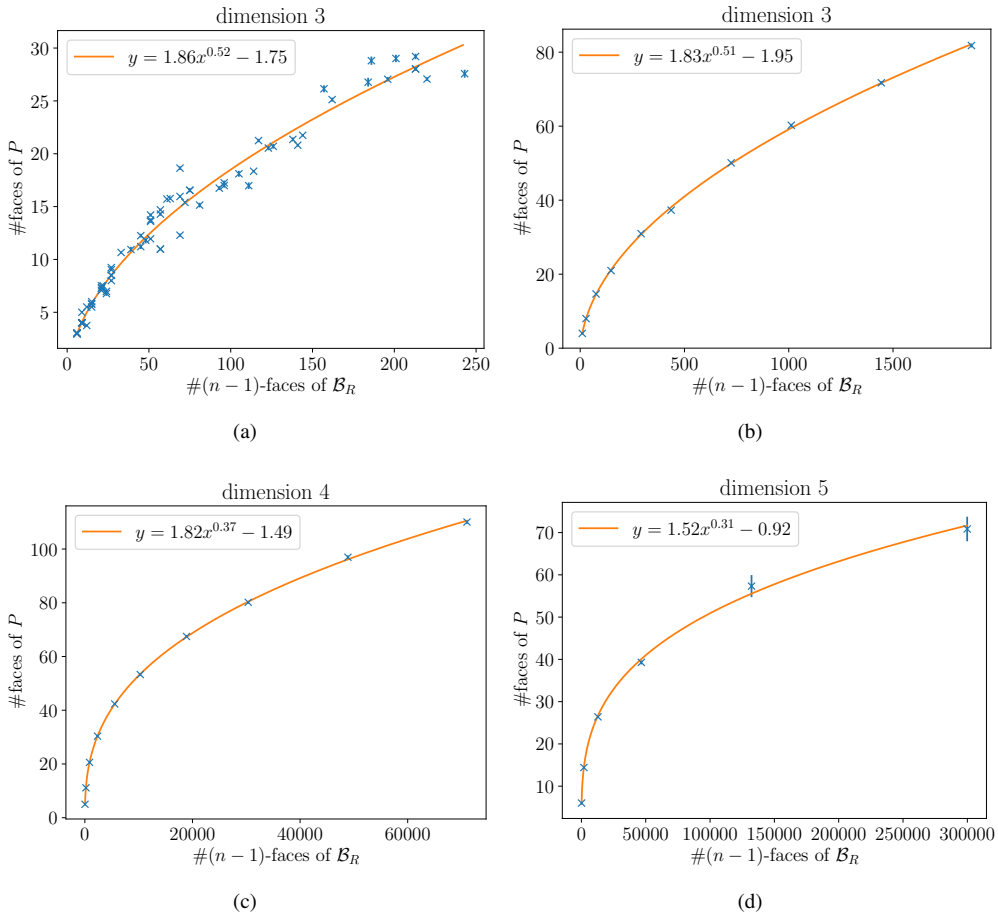

Figure 7: Number of faces of P with respect to the number of $(n-1)$-faces of $\mathcal{B}_R$ in different settings. In (a), $\mathcal{B}_R$ is a rational ball as computed by Fouard and Malandain (2005) in dimension 3. In (b-c), $\mathcal{B}_R$ is defined from a Farey sequence of given order and dimension, taking all the fractions : (b) dimension 3, for orders between 1 and 10, (c) dimension 4 for order between 1 and 10, (d) dimension 5 for orders between 1 and 6. Each point is the mean of a certain number of random cuts (1000 in dimension 3, 500 in dimension 4, 400 in dimension 5 for orders up to 4, and 6 in dimension 5 for orders 5 and 6.).

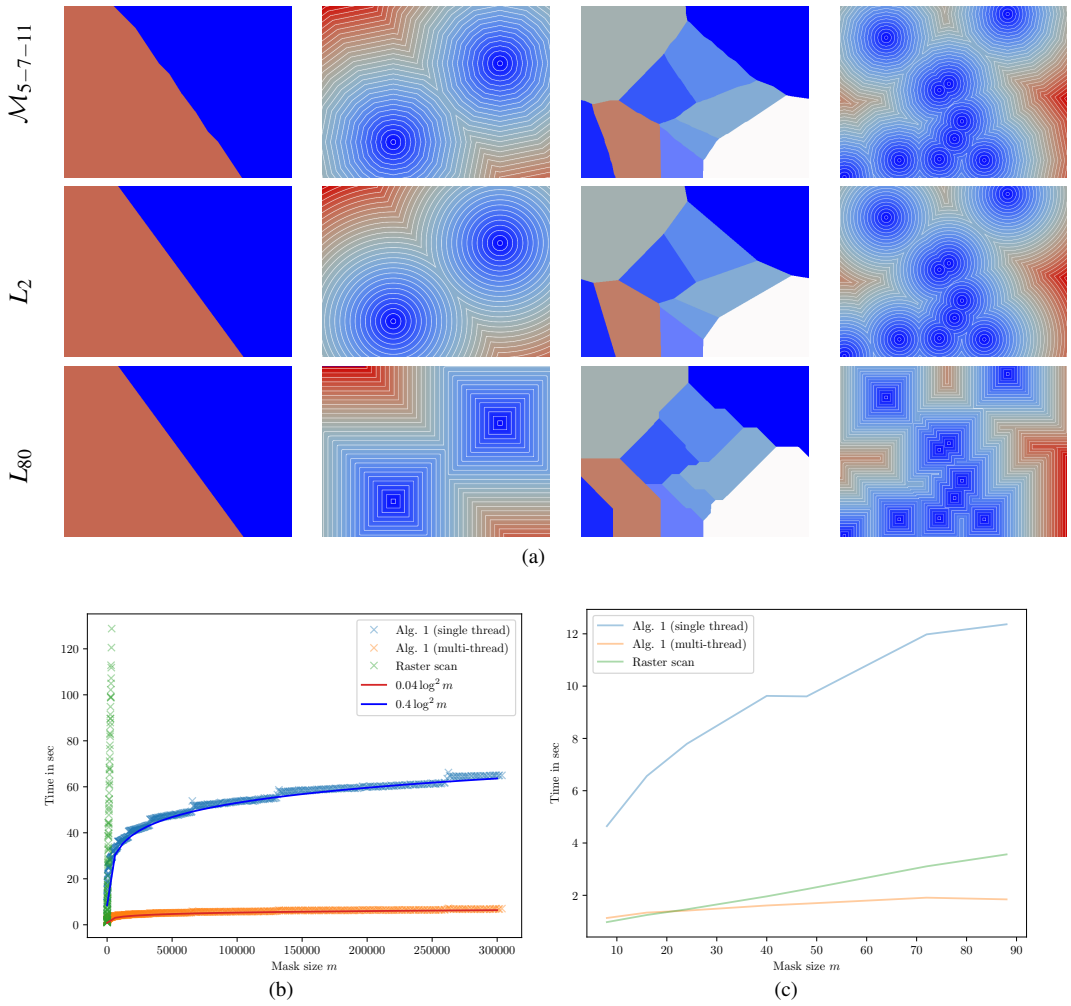

Figure 8: (a) Separable Voronoi map and distance transformation for the chamfer norm $\mathcal{M}_{5-7-11}$ (see Fig. 1), the L_2, and the L_{80} metrics on a 256^2 domain with 2 and 10 random seeds (respectively first two and last two columns). (b) Experimental evaluation of the subquadratic algorithm when increasing the mask size on a 2048^2 and following the chamfer norm construction of Fouard and Malandain (2005) (zoom in (c)). We compare the efficiency of Algorithm 1 in single thread and multi-thread settings, with the classical raster scan approach of Borgefors (1986) (only single thread).