2D Subquadratic Separable Distance Transformation for Path-Based Norms[★]

David Coeurjolly

CNRS, LIRIS, UMR5205, F-69621, France

Abstract. In many applications, separable algorithms have demonstrated their efficiency to perform high performance and parallel volumetric computations, 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 detail a subquadratic algorithm to compute such volumetric transformation in dimension 2. More precisely, we describe a $O(\log^2 m \cdot N^2)$ algorithm for shapes in a $N \times N$ domain with chamfer norm of size m.

Keywords: Digital Geometry, Distance Transformation, Path-based Norms.

1 Introduction

Since early works on digital geometry, distance transformation has been playing an important role in many applications [16,15]. 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. Since such characterization is parametrized by a distance function, many authors address 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 [15,2,6] or sequences of chamfer masks [16,11,17,13]; the vector displacement based Euclidean distance [5,14]; Voronoi diagram based Euclidean distance [3,9] or the square of the Euclidean distance [7,10]. For the Euclidean metric, separable volumetric computations have demonstrated to be very efficient: optimal $O(n \cdot N^n)$ time algorithms for shapes in N^n domains, optimal multi-thread/GPU implementation... (please refer to [4] for a discussion). For path-based metrics (chamfer mask, -weighted- neighborhood sequences,...), two main techniques exist to compute the distance transformation. 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 [19]. 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 [16,2]. 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.

Beside specific applications which use the anisotropic nature of the chamfer mask, rotational dependency is usually ensured increasing the mask size m together with optimizing weights. In this context and for arbitrarily large N, both Dijsktra-like and raster scan approaches have a quadratic computational cost with respect to N and m. In practical situations, $m \ll N$ but m still needs to be in $O(N^n)$ to have accurate asymptotic DT.

Please note that Dijkstra's graph approach allows us to defined constrained distance transformation (*i.e.* geodesic metric), both separable approaches and raster-scan for path-based metrics are only dedicated to compact convex domains (usually hyper-rectangular) distance transformation.

Contributions. In this article, we first describe the generic framework for separable distance transformation and metric conditions to be consistent with this model. Then, we describe subquadratic and parallel algorithms in dimension 2 to compute error-free distance transformation and Voronoi map for chamfer norms and other path-based metrics. Overall computational costs can be summarized as follows (see 3.2 for predicate definitions):

Metric	CLOSEST	HIDDENBY	Sep. Voronoi Map	Reference
L_2	$O(1)$	$O(1)$	$\Theta(n \cdot N^n)$	[7]
L_∞	$O(1)$	$O(1)$	$\Theta(n \cdot N^n)$	[10]
L_1	$O(1)$	$O(1)$	$\Theta(n \cdot N^n)$	[10]
L_p (exact pred.)	$O(\log p)$	$O(\log p \cdot \log N)$	$O(n \cdot N^n \cdot \log p \cdot \log N)$	Lem. 1, [1]
L_p (inexact pred.)	$O(1)$	$O(\log N)$	$O(n \cdot N^n \cdot \log N)$	Lem. 1, [1]
2D Chamfer norm	$O(\log m)$	$O(\log^2 m)$	$O(\log^2 m \cdot N^2)$	Theorem 1
2D Neig. seq. norm	$O(\log f)$	$O(\log^2 f)$	$O(\log^2 f \cdot N^2)$	Th. 1 and [13]

2   Preliminaries

Definition 1 (Norm and metric induced by a norm). *Given a vector space* EV, *a norm is a map* g *from* EV *to a sub-group* F *of* $\mathbb{R}$ *such that* $\forall \boldsymbol{x}, \boldsymbol{y} \in EV$,

$$(\text{non-negative}) \quad g(\boldsymbol{x}) \geq 0 \tag{1}$$

$$(\text{identity of indiscernibles}) \quad g(\boldsymbol{x}) = 0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{0} \tag{2}$$

$$(\text{triangular inequality}) \quad g(\boldsymbol{x} + \boldsymbol{y}) \leq g(\boldsymbol{x}) + g(\boldsymbol{y}) \tag{3}$$

$$(\text{homogeneity}) \quad \forall \lambda \in \mathbb{R}, \quad g(\lambda \cdot \boldsymbol{x}) = |\lambda| \cdot g(\cdot \boldsymbol{x}) \tag{4}$$

$d(a, b) := g(b - a)$ *is the metric induced by the norm* g. (E, F, d) *is called a metric space if* $d : E \rightarrow F$ *(with* E *such that for* $a, b \in E$, $(b - a) \in EV$ *).*

Note that the above definition can be extended from vector spaces to *modules* on a commutative ring ($\mathbb{Z}^n$ being a module on $\mathbb{Z}$ but not a vector space) [18]. Path-based approaches (chamfer masks, -weighted- neighborhood sequences...) aim at defining *digital* metrics induced by norms in metric spaces $(\mathbb{Z}^n, \mathbb{Z}, d)$. Note that (weighted, with $w_i \geq 0$) L_p metrics

$$d_{L_p}(a, b) = \left(\sum_{k=1}^{n} w_k |a_k - b_k|^p \right)^{\frac{1}{p}}, \tag{5}$$

define metric spaces $(\mathbb{Z}^n, \mathbb{R}, d_{L_p})$ which are not *digital*. However, rounding up the distance function $(\mathbb{Z}^n, \mathbb{Z}, \lceil d_{L_p} \rceil)$ is a digital metric space [8].

Definition 2 (Distance Transformation and Voronoi Map). *The distance transform* DT_X *associated with a digital metric space* $(\mathbb{Z}^n, \mathbb{Z}, d)$ *is a map* $X \rightarrow \mathbb{Z}$ *such that, for* $a \in X$ $DT_X(a) = \min_{b \in \mathbb{Z}^n \setminus X} \{d(a, b)\}$. *The Voronoi map is the map* $X \rightarrow \mathbb{Z}^n$: $\Pi_X(a) = \arg\min_{b \in \mathbb{Z}^n \setminus X} \{d(a, b)\}$.

Voronoi map Π_X corresponds to the intersection between the continuous Voronoi diagram for the metric d of points $\mathbb{Z}^n \setminus X$ and the lattice $\mathbb{Z}^n$. If a digital point a belongs to a Voronoi diagram $d-$facet $(0 \leq d < n)$, a is equidistant to 2 or more points in $\mathbb{Z}^n \setminus X$ but only one is considered in $\Pi_X(a)$ this choice has no influence on DT_X.

Definition 3 (Chamfer Mask). *A weighted vector is a pair* $(\boldsymbol{v}, w)$ *with* $\boldsymbol{v} \in \mathbb{Z}^n$ *and* $w \in \mathbb{N}^*$. *A chamfer mask* $\mathcal{M}$ *is a central-symmetric set of weighted vectors with no null vectors and containing at least a basis of* $\mathbb{Z}^n$.

Many authors have proposed algorithmic and/or analytic approaches to construct chamfer masks approximating the Euclidean metric. In the following, we focus on such *chamfer norms* which are chamfer metric induced by a norm. To evaluate distances between two digital points for a given chamfer metric, direct formulations have been proposed with simple geometrical interpretation:

Definition 4 (Rational ball, minimal H-representation [18,12]). *Given a Chamfer norm* $\mathcal{M}$, *the rational ball associated with* $\mathcal{M}$ *is the polytope*

$$\mathcal{B}_R = conv \left\{ \frac{\boldsymbol{v}_k}{w_k}; \ (\boldsymbol{v}_k, w_k) \in \mathcal{M} \right\}. \tag{6}$$

Rational ball $\mathcal{B}_R$ *can also be described as the H-representation of polytope with minimal parameter [13]:* $P = \{x \in \mathbb{Z}^n; Ax \leq y\}$ *such that* $\forall k \in [1 \ldots f]$, $\exists x \in P$ $A_k x = y_k$.[1] f *is the number of rows in* A *and the number of facets in* $\mathcal{B}_R$, *and is thus related to* $|\mathcal{M}|$.

---

[1] A_k being the kth row of A.

An important result for distance computation can be formalized as follows:

Proposition 1 (Direct Distance Computation [12]). *Given a chamfer mask $\mathcal{M}$ induced by a norm 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*

$$d_{\mathcal{M}}(O, a) = \max_{1 \leq k \leq f} \{A_k a^T\}. \tag{7}$$

Among path-based digital metric, (weighted) neighborhood sequences have been proposed to have better approximation of the Euclidean metric from sequences of elementary chamfer masks [16,11,17,13]. A key result have been demonstrated in [13] stating that for such distance functions, a minimal parameter polytope representation exists and that distances can be obtained from a expression similar to (7):

$$d(O, a) = \max_{1 \leq k \leq f} \{f_k(A_k a^T)\}, \tag{8}$$

f_k being some integer sequence characterizing the neighborhood sequence metric. In the following and for the sake of simplicity, we describe our algorithms focusing on chamfer norms but similar results can be obtained for more generic path-based metric such as neighborhood sequences.

3   Separable Distance Transformation

3.1   Voronoi Map from Separable Approach and Metric Conditions

In [7,3,10,9], 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. Separability means that computations are performed dimension by dimension. In the following, we consider *Voronoi Map* approach as defined in [3]: Let us first define an hyper-rectangular image $I_X : [1..N_1] \times \ldots \times [1..N_n] \rightarrow \{0, 1\}$ 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). In dimension 2, each row of the input image are processed to create independent 1D Voronoi maps along the first dimension for the metric. Then, for each further dimension, the partial Voronoi map Π_X is updated using one dimensional independent processes on image spans along the ith dimension. Algorithm 1 describes the 1D processes to perform on each row, column and higher dimensional image span[2]. In this process, metric information are embedded in the following predicates (see Fig. 1): CLOSEST(a, b, c), given three points $a, b, c \in \mathbb{Z}^n$ this predicate returns true if $d(a, b) < d(a, c)$. HIDDENBY(a, b, c, S), given three points $a, b, c \in \mathbb{Z}^n$ such that $a_i < b_i < c_i$[3] and a 1D image span S, this predicates returns true if there is no $s \in S$ such that

$$d(b, s) < d(a, s) \text{ and } d(b, s) < d(c, s). \tag{9}$$

---

[2] An image span S along the ith direction is a vector of N_i points with same coordinates except at their i^{th} one.

[3] Subscript a_i denotes the ith coordinate of point $a \in \mathbb{Z}^n$.

Algorithm 1. Voronoi map construction on 1D image span S along the ith dimension.

Data: Input binary map I_X if $i = 1$ or partial Voronoi map Π_X obtained for dimensions lower than i, and a 1D span S with points $\{s^1, \ldots, s^{N_i}\}$ sorted by their i^{th} coordinate.

Result: Updated partial Voronoi map Π_X along S.

```
 1  if  i == 1 ;                    // Special case for the first dimension
 2  then
 3  │    k = 0;
 4  │    foreach point s in S do
 5  │    │    if I_X(s) == 1 then
 6  │    │    │    L_S[k] = s;
 7  │    │    │    k + +;

 8  else
 9  │    L_S[0] = Π_X(s¹);
10  │    L_S[1] = Π_X(s²);
11  │    k = 2 , l = 3;
12  │    while l ≤ N_i do
13  │    │    w = Π_X(s^l);
14  │    │    while k ≥ 2 and HIDDENBY(L_S[k − 1], L_S[k], w, S) do
15  │    │    │    k − −;
16  │    │    k + + ; l + +;
17  │    │    L_S[k] = w;

18  foreach point s in S by increasing i^th coordinate do
19  │    while (k < |L_S|) and not(CLOSEST(s, L_S[k], L_S[k + 1])) do
20  │    │    k + +;
21  │    Π_X[s] = L_S[k];
```
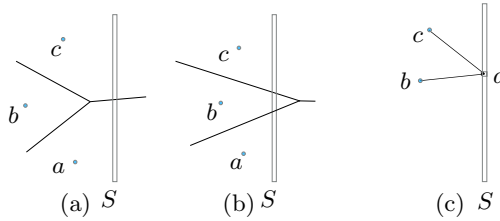

(a) S (b) S (c) S

Fig. 1. Geometrical predicates for Voronoi map construction: 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 $c \in S$.

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. For L_1, L_2 and L_∞ metrics, CLOSEST and HIDDENBY predicates can be computed in $O(1)$ [3,7,10]. Hence, Algorithm 1 is in $O(N_i)$ for dimension i, leading to an overall computational time for the

Voronoi map and distance transformation problem in $\Theta(n \cdot N^n)$ (if we assume that $\forall i \in [1 \dots n], N_i = N$).

In [7] or [9], authors discussed about conditions on the metric d to ensure that Algorithm 1 is correct. The key property can be informally describe as follows: given two points $a, b \in \mathbb{Z}^n$ such that $a_i < b_i$ and a straight line l along the ith 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* in [9] and is related to *quadrangle inequality* in [7]). To sum up these contributions, we have the following sufficient conditions on the metric:

Proposition 2 (Metric conditions [7]). *Let d be a metric induced by a **norm whose unit ball is symmetric with respect to grid axes** and if distance comparison predicate is exact, Algorithm 1 is correct and returns a Voronoi map Π_X.*

When implementing Algorithm 1, the distance comparison predicate is exact if we can compare two distances, *e.g.* CLOSEST predicate, without error.

For L_p norms Algorithm 1 provides exact Voronoi map computation. Indeed, distance comparison predicate can be error-free implemented from integer number comparisons considering the p power of the distance function $\left(d_{L_p}\right)^p$.

Proposition 2 also implies that most chamfer norms and neighborhood sequence based norms can also be considered in separable Algorithm 1 (see Fig. 2). We just need algorithmic tools to efficiently implement both CLOSEST and HIDDENBY predicates.
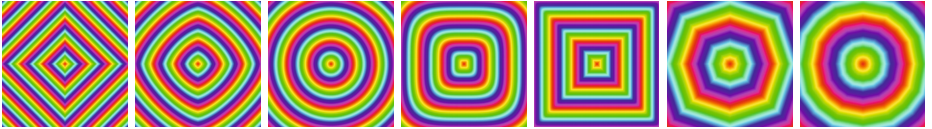

Fig. 2. Balls for different metrics satisfying Proposition 2: *(from left to right)* L_1, $L_{1.4}$, L_2, L_4, $L_{43.1}$, $\mathcal{M}_{3-4}$ and $\mathcal{M}_{5-7-11}$

3.2   A First Generic Adapter

We first detail the overall computational cost of Algorithm 1:

Lemma 1. *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 1 is in $O(n \cdot N^n \cdot (C + H))$.*

From [3,10], $C = H = O(1)$ for L_1, L_2 and L_∞ norms. For given norm d, we first define generic Algorithms 2, 3 and 4: From some evaluations of d, HIDDENBY

predicate is obtained by a dichotomic search on the 1D image span S to localize the abscissa of Voronoi edges of sites $\{a, b\}$ and $\{b, c\}$ (see Fig. 3).

Algorithm 2. Generic CLOSEST($a, b, c \in \mathbb{Z}^n$).

---

1 **return** $d(a, b) < d(a, c)$;

Algorithm 3. Generic VORONOIEDGE($a, b, s^i, s^j \in \mathbb{Z}^n$), $a_i < b_i$.

---

1 **if** $(j - i = 1)$ **then**
2   **if** $i = 1$ *and* CLOSEST(s^i, b, a) **then**
3   **return** $-\infty$;
4   **if** $i = N_i$ *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 4. Generic HIDDENBY($a, b, c \in \mathbb{Z}^n$; S in the ith direction) $a_i < b_i < c_i$.

---

1 $v_{ab} = $ VORONOIEDGE(a, b, s^1, s^{N_i});
2 $v_{bc} = $ VORONOIEDGE(b, c, s^1, s^{N_i});
3 **return** $(v_{ab} > v_{bc})$;

Lemma 2. *Let* $(\mathbb{Z}^n, F, d)$ *be a metric space induced by a norm with axis symmetric unit ball, from Algorithms 2 and 4, we have* $H = O(C \cdot \log N)$.

Proof. First, the generic VORONOIEDGE is dichotomic with $O(\log N)$ steps and each step is in $O(C)$. VORONOIEDGE is in $O(C \cdot \log N)$. To prove the correctness of VORONOIEDGE (and thus Alg. 1), we use the convexity of the metric and the quadrangle property: since $a_i < b_i$, all grid points closer to a than b in S (if exist) are *lower* than all pixels on S closer to b than to a (if exist too). Thanks to the test in line 8, recursive call maintain this invariant. Note that tests on lines $2 - 5$ handle the fact that the edge may not belong to S. □

If we consider a chamfer norm with a rational ball of f facets, Eq. (7) suggests that $C = O(f)$. Hence, we have the following corollary:

Corollary 1. *Let* $\mathcal{M}$ *be a chamfer norm whose rational ball has* f *facets, separable exact Voronoi map* Π_X *can be obtained in* $O(n \cdot N^n \cdot f \cdot \log N)$.

Please remember that naive implementation of chamfer mask distance transformation using raster scan approach would lead to a $O(f \cdot N^n)$ computational cost. In the following sections, we use the convex structure of $\mathcal{B}_R$ to design a parallel subquadratic algorithm for chamfer norms.

3.3   Subquadratic Algorithm in Dimension 2

Let us consider a 2D chamfer norm $\mathcal{M}$ with m weighted vectors (note that $f := |\mathcal{B}_R| = m$ in 2D). We suppose that vectors $\{\boldsymbol{v}^k\}_{k=1\dots m}$ are sorted counter-clockwise. We define a wedge as a pair $(\boldsymbol{v}^k, \boldsymbol{v}^{k+1})$ of vectors. To each wedge is associated a row A_k in the minimal H-representation of A (A_k can also be seen as a –non-unitary– normal vector to $\mathcal{B}_R$ facets [12]). Using similar notations, [18,17] demonstrate that the distance evaluation of point a can be obtained in two steps: First, we compute the wedge $(\boldsymbol{v}^k, \boldsymbol{v}^{k+1})$ a belongs to. Then,

$$d_{\mathcal{M}}(O, a) = A_k \cdot a^T \tag{10}$$

Lemma 3. *Given a chamfer norm $\mathcal{M}$ in dimension 2 with m vectors, the distance computation and thus the* Closest *predicate are in $O(\log m)$.*

Proof. Since vectors are sorted counter-clockwise, $(\boldsymbol{v}^k, \boldsymbol{v}^{k+1})$ wedge can be obtained by a dichotomic search with $O(\log m)$ steps. At each step, we compute the local orientation of point a w.r.t. a direction which is in $O(1)$. Once the wedge has been obtained, Eq. (10) returns the distance value in $O(1)$.   □

Please note that in practical implementations, we can use symmetries in $\mathcal{M}$ to only work on restrictions of chamfer mask directions, so called *generator* in the literature. To optimize the Hiddenby predicate, we focus on the VoronoiEdge function. Given two points a and b ($a_i < b_i$) and a 1D image span S along the ith dimension, we have to find the lowest abscissa e_i of the point e on S such that $d(a, e) < d(b, e)$ and $d(a, e') \geq d(b, e')$ for any e' with $e'_i > e_i$. Let us first suppose that we do not know e but we know the wedge $(\boldsymbol{v}^k, \boldsymbol{v}^{k+1})$ (resp. $(\boldsymbol{v}^j, \boldsymbol{v}^{j+1})$) to which the vector $(e - a)^T$ (resp. $(e - b)^T$) belongs to (see Fig. 3−(c)). In this situation, we know that e is the solution of

$$A_k \cdot (e - a)^T = A_j \cdot (e - b)^T . \tag{11}$$

(since $e \in S$, we have one linear equation with only one unknown e_i). As a consequence, if we know the two wedges the Voronoi edge belongs to, we have the abscissa in $O(1)$ (see Algorithm 5 and Fig. 3−(c)). To obtain both wedges, we use a dichotomic search similar to Algorithm 3: Algorithm 6 returns the wedge associated with a containing the Voronoi edge with respect to b. Applying this algorithm to obtain the wedge associated with b with respect to a defines Algorithm 5. The dichotomic search shrinks the set of vectors

---

Algorithm 5. 2D chamfer norm VoronoiEdge($a, b, s^i, s^j \in \mathbb{Z}^2$).

1 $(\boldsymbol{v}^k, \boldsymbol{v}^{k+1}) = $ VoronoiEdgeWedge($a, b, \boldsymbol{v}^1, \boldsymbol{v}^m, S$);
2 $(\boldsymbol{v}^j, \boldsymbol{v}^{j+1}) = $ VoronoiEdgeWedge($b, a, \boldsymbol{v}^1, \boldsymbol{v}^m, S$);
3 Compute the abscissa e_i of the point e such that $A_k \cdot (e - a)^T = A_j \cdot (e - b)^T$;
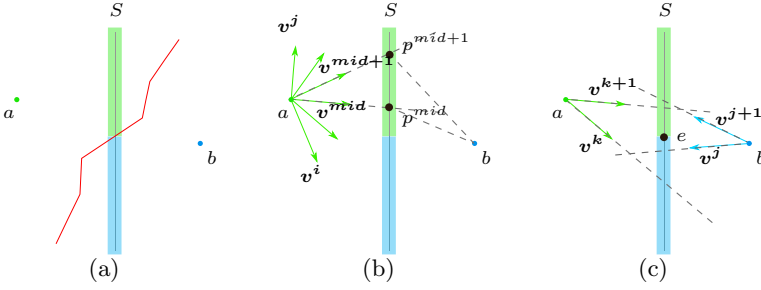**4 return** e_i;

Fig. 3. VORONOIEDGEWEDGE and VORONOIEDGE: (a) initial problem, we want to compute the intersection between S and the Voronoi edge of a and b (in red). (b) an internal step of VORONOIEDGEWEDGE to reduce the set of directions of $\mathcal{M}$ at a (here the next recursive call will be on $(\boldsymbol{v}^i, \boldsymbol{v}^{mid})$).$(c)$ final step of VORONOIEDGE where both wedges have been obtained and thus e can be computed.

$\{\boldsymbol{v}^i, \ldots, \boldsymbol{v}^j\}$ to end up with a wedge $(\boldsymbol{v}^k, \boldsymbol{v}^{k+1})$ such that the intersection point between the straight line $(a + \boldsymbol{v}^k)$ and S is in the Voronoi cell of a and such that the intersection between $(a + \boldsymbol{v}^{k+1})$ and S is in the Voronoi cell of b (see Fig. $3-(c)$). Algorithm 6 thus first computes the intersection points associated with a wedge $(\boldsymbol{v}^{i+(j-i)/2}, \boldsymbol{v}^{i+(j-i)/2+1})$ (lines $5-6$); evaluates the distances at these points (lines $7-8$) and then decides which set $\{\boldsymbol{v}^i, \ldots, \boldsymbol{v}^{i+(j-i)/2}\}$ or $\{\boldsymbol{v}^{i+(j-i)/2}, \ldots, \boldsymbol{v}^j\}$ has to be considered for the recursive call (lines $14-20$ and Fig. $3-(b)$).

Theorem 1. *Let $\mathcal{M}$ be a 2D chamfer norm with axis symmetric unit ball and m weighted vectors, then we have: (i) Algorithm 5 is in $O(\log^2 m)$; (ii) Algorithm 1 (with predicates from Algorithm 5 and Lemma 3), computes a Voronoi map Π_X and thus the distance transformation of X for metric d_M in $O(\log^2 m \cdot N^2)$.*

Proof. Let us first consider (i). As described above, Algorithm 6 performs $\log m$ recursive calls and each step is in $O(\log m)$. Indeed, p^{mid} and p^{mid+1} are given by the intersections between two rational slope straight lines plus a rounding operations on rational fractions, which are assumed to be $O(1)$. Then, line 8 requires two $O(\log m)$ computations by Lemma 3. Hence, Eq. (11) leads to $O(1)$ computations, $O(\log^2 m)$ is required for Algorithm 5. (ii) is a direct consequence of (i) and Lemma 2 with $n = 2$. $\qquad\square$

4   Implementation Details and Experimental Analysis

In this section, we give some implementation details and experimental results for chamfer norm Voronoi map in dimension 2. First of all, most algorithms presented here are available in the DGTAL library [1]. For L_p metrics, we have implemented several CLOSEST and HIDDENBY predicates: If $p = \{1, 2\}$, exact computations are proposed and all predicates are in $O(1)$ with only integer number computations [7,9,10]; If $p \in \mathbb{R}$, $p \geq 1$, we have approximated computations

---

Algorithm 6. VORONOIEDGEWEDGE($a, b \in \mathbb{Z}^2; \boldsymbol{v}^i, \boldsymbol{v}^j$ in $\mathcal{M}$; S along the ith direction).

---

1 **if** $(j - i = 1)$ **then**
2 $\quad$ **return** $(\boldsymbol{v}^i, \boldsymbol{v}^{i+1})$;
3 **else**
4 $\quad$ $mid = i + (j - i)/2$;
5 $\quad$ Let p^{mid} be the intersection point between $(a + \boldsymbol{v}^{mid})$ and S;
6 $\quad$ Let p^{mid+1} be the intersection point between $(a + \boldsymbol{v}^{mid+1})$ and S;
$\quad$ `// O(1) evaluation of distances w.r.t. a` $d^a_{p^{mid}} = A_{mid} \cdot (p^{mid} - e)^T$;
7 $\quad$ $d^a_{p^{mid+1}} = A_{mid+1} \cdot (p^{mid+1} - e)^T$;
$\quad$ `// O(log m) evaluation of distances w.r.t. b` $d^b_{p^{mid}} = d_{\mathcal{M}}(b, p^{mid})$;
8 $\quad$ $d^b_{p^{mid+1}} = d_{\mathcal{M}}(b, p^{mid+1})$;
9 $\quad$ Let b_{mid} be true if $d^a_{p^{mid}} < d^b_{p^{mid}}$; false otherwise;
10 $\quad$ Let b_{mid+1} be true if $d^a_{p^{mid+1}} < d^b_{p^{mid+1}}$; false otherwise;
11 $\quad$ **if** $b_{mid} \neq b_{mid+1}$; $\qquad\qquad$ `// we have the Voronoi edge wedge`
12 $\quad$ **then**
13 $\quad\quad$ **return** $(\boldsymbol{v}^{mid}, \boldsymbol{v}^{mid+1})$;
14 $\quad$ **if** $b_{mid} = b_{mid+1} = true$; $\qquad\qquad$ `// Both points are in a's cell`
15 $\quad$ **then**
16 $\quad\quad$ **if** $a_i < b_i$ **then**
17 $\quad\quad\quad$ **return** VORONOIEDGEWEDGE($a, b, \boldsymbol{v}^{mid}, \boldsymbol{v}^j, S$);
18 $\quad\quad$ **else**
19 $\quad\quad\quad$ **return** VORONOIEDGEWEDGE($a, b, \boldsymbol{v}^i, \boldsymbol{v}^{mid}, S$);
20 $\quad$ **if** $b_{mid} = b_{mid+1} = false$; $\qquad\qquad$ `// Both points are in b's cell`
21 $\quad$ **then**
22 $\quad\quad$ **if** $a_i < b_i$ **then**
23 $\quad\quad\quad$ **return** VORONOIEDGEWEDGE($a, b, \boldsymbol{v}^i, \boldsymbol{v}^{mid}, S$);
24 $\quad\quad$ **else**
25 $\quad\quad\quad$ **return** VORONOIEDGEWEDGE($a, b, \boldsymbol{v}^{mid}, \boldsymbol{v}^j, S$);

---

on real numbers (*double*) and we consider the Generic HIDDENBY predicate in $O(\log N)$ (Alg. 4). Since predicates are based on floating point computations, numerical issues may occur. 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 $O(\log p)$ thanks to exponentiation by squaring). The HIDDENBY predicate is also based on Algorithm 4. Beside these predicates for L_p metrics, DGTAL also contains a generic metric adapter: if the user specifies a distance function (taking two points and returning a value) corresponding to a norm with axis symmetric unit ball, generic CLOSEST and HIDDENBY predicates can be automatically constructed. Please note that since all algorithms are separable, the generic framework provided in DGTAL allow us to have a free multi-thread implementation [4].

To implement efficient predicates leading to subquadratic algorithm in dimension 2 (Alg. 5 and 6), we store the chamfer norm weighted vectors $\mathcal{M}$ in a random access container sorted counterclockwise to be able to get the midvector $\boldsymbol{v}^{mid}$ in $O(1)$. When implementing Algorithms 5 and 6, few special cases have to be taken into account. For instance, we have to handle situations where a, b or c belong to S in Alg. 5 and 6. Furthermore, Eq. (11) has a solution iff $A_k \neq A_j$. Thanks to the geometrical representation of the dichotomic process (Fig. 3), such special cases are easy to handle. Fig. 4-(a) illustrates some results on a small domain.
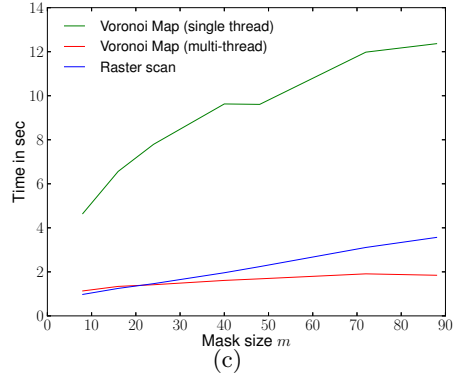
To evaluate experimentally the computational cost given in Theorem 1, we use the following setting: given a mask size m, we generate m distinct random
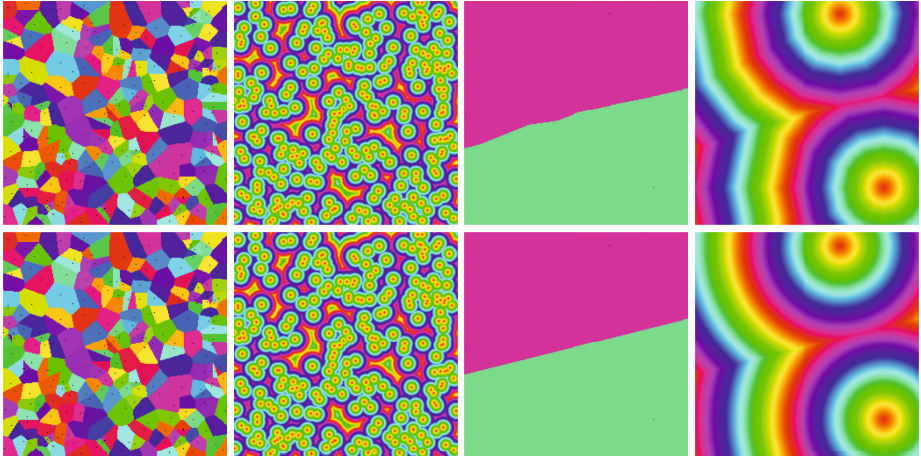
(a)
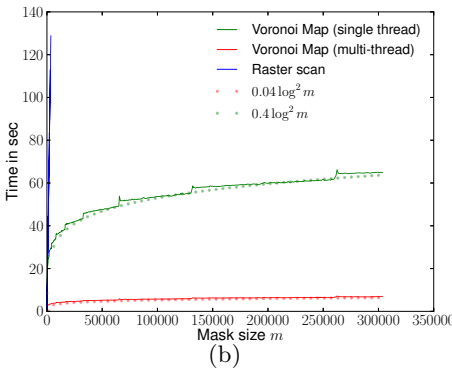
(b)

(c)

Fig. 4. (a) Separable Voronoi map and distance transformation for $\mathcal{M}_{5-7-11}$ and L_2: on a 256^2 domain with 256 and 2 random seeds, the first row corresponds to $\mathcal{M}_{5-7-11}$ and the second one to L_2. (b) and zoom in (c): Experimental evaluation of subquadratic chamfer norm Voronoi map computation.

vectors $(x, y)^T$ with $gcd(x, y) = 1$ (extracted from Farey series for instance). For a general mask of size m, we do not optimize the weights to approximate as best as possible the Euclidean metric. Indeed, weights do not play any role in the computational analysis, we just use trivial ones that ensure that $\mathcal{M}$ is a norm with axis symmetric unit ball. In Fig.4-$(b - c)$, 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 in Fig. 4-(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 (blue curve in Fig. 4-(b)). Please note that on this 2048^2 domain with 2048 sites, Euclidean Voronoi Map (L_2) is obtained in 954.837 milliseconds on a single core and 723.196 msec on 16 cores.

5   Conclusion and Discussion

In the literature, several authors discussed about the fact that a large class of metrics can be considered in separable approaches for volumetric analysis. In this article, we have proposed several algorithms to efficiently solve the Voronoi map and distance transformation: given a user-specified distance function (induced by a norm with some properties) a first generic separable algorithm can be used. Focusing on chamfer norms, geometrical interpretation of this generic approach allows us to design a first subquadratic algorithm in dimension 2 to compute the Voronoi map. Thanks to separability, parallel implementation of the distance transformation leads to efficient distance computation.

In higher dimensions, it turns out that most results are still true: distance function can be evaluated in $O(n \cdot \log m)$ and the dichotomic search described in VORONOIEDGEWEDGE can also be extended to n-dimensional chamfer norms.

References

1. DGTAL: Digital geometry tools and algorithms library, `http://dgtal.org`
2. Borgefors, G.: Distance transformations in digital images. Computer Vision, Graphics, and Image Processing 34(3), 344–371 (1986)
3. Breu, H., Gil, J., Kirkpatrick, D., Werman, M.: Linear time euclidean distance transform algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(5), 529–533 (1995)
4. Coeurjolly, D.: Volumetric Analysis of Digital Objects Using Distance Transformation: Performance Issues and Extensions. In: Köthe, U., Montanvert, A., Soille, P. (eds.) WADGMM 2010. LNCS, vol. 7346, pp. 82–92. Springer, Heidelberg (2012)
5. Danielsson, P.E.: Euclidean distance mapping. Computer Graphics and Image Processing 14, 227–248 (1980)

6. Fouard, C., Malandain, G.: 3-D chamfer distances and norms in anisotropic grids. Image and Vision Computing 23, 143–158 (2005)
7. Hirata, T.: A unified linear-time algorithm for computing distance maps. Information Processing Letters 58(3), 129–133 (1996)
8. Klette, R., Rosenfeld, A.: Digital Geometry: Geometric Methods for Digital Picture Analysis. Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann (2004)
9. Maurer, C., Qi, R., Raghavan, V.: A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions. IEEE Trans. Pattern Analysis and Machine Intelligence 25, 265–270 (2003)
10. Meijster, A., Roerdink, J.B.T.M., Hesselink, W.H.: A general algorithm for computing distance transforms in linear time. In: Mathematical Morphology and its Applications to Image and Signal Processing, pp. 331–340. Kluwer (2000)
11. Mukherjee, J., Das, P.P., Kumar, M.A., Chatterji, B.N.: On approximating euclidean metrics by digital distances in 2D and 3D. Pattern Recognition Letters 21(6-7), 573–582 (2000)
12. Normand, N., Évenou, P.: Medial axis lookup table and test neighborhood computation for 3D chamfer norms. Pattern Recognition 42(10), 2288–2296 (2009)
13. Normand, N., Strand, R., Evenou, P.: Digital distances and integer sequences. In: Gonzalez-Diaz, R., Jimenez, M.-J., Medrano, B. (eds.) DGCI 2013. LNCS, vol. 7749, pp. 169–179. Springer, Heidelberg (2013)
14. Ragnemalm, I.: Contour processing distance transforms, pp. 204–211. World Scientific (1990)
15. Rosenfeld, A., Pfaltz, J.: Distance functions on digital pictures. Pattern Recognition 1, 33–61 (1968)
16. Rosenfeld, A., Pfaltz, J.: Sequential operations in digital picture processing. Journal of the ACM (JACM) 13, 471–494 (1966),
http://portal.acm.org/citation.cfm?id=321357
17. Strand, R.: Distance Functions and Image Processing on Point-Lattices With Focus on the 3D Face- and Body-centered Cubic Grids. Phd thesis, Uppsala Universitet (2008)
18. Thiel, E.: Géométrie des distances de chanfrein. Ph.D. thesis, Aix-Marseille 2 (2001)
19. Verwer, B.J.H., Verbeek, P.W., Dekker, S.T.: An efficient uniform cost algorithm applied to distance transforms. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(4), 425–429 (1989)