Minimal-Delay Distance Transform for Neighborhood-Sequence Distances in 2D and 3D

Nicolas Normand, Robin Strand, Pierre Evenou, Aurore Arlicot

Minimal-Delay Distance Transform for Neighborhood-Sequence Distances in 2D and 3D

Nicolas Normand[a,*], Robin Strand[b], Pierre Evenou[a], Aurore Arlicot[a]

[a] *LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597, Polytech Nantes, Rue Christian Pauc, La Chantrerie, 44306 Nantes Cedex 3, France*
[b] *Centre for Image Analysis, Uppsala University, Sweden*
[c] *Centre for Image Analysis, Uppsala University, Box 337, SE-75105 Uppsala, Sweden*
[d] *LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597, Polytech Nantes, La Chantrerie, 44306 Nantes Cedex 3, France*

Abstract

This paper presents a path-based distance where local displacement costs vary both according to the displacement vector and with the travelled distance. The corresponding distance transform algorithm is similar in its form to classical propagation-based algorithms, but the more variable distance increments are either stored in look-up-tables or computed on-the-fly. These distances and distance transform extend neighborhood-sequence distances, chamfer distances and generalized distances based on Minkowski sums. We introduce algorithms to compute a translated version of a neighborhood sequence distance map both for periodic and aperiodic sequences and a method to derive the centered distance map. A decomposition of the grid neighbors, in $\mathbb{Z}^2$ and $\mathbb{Z}^3$, allows to significantly decrease the number of displacement vectors needed for the distance transform. Overall, the distance transform can be computed with minimal delay, without the need to wait for the whole input image before beginning to provide the result image.

```
@article{normand2012cviu,
    Author = {Normand, Nicolas and Strand, Robin and Evenou,
              Pierre and Arlicot, Aurore},
    Doi = {10.1016/j.cviu.2012.08.015},
    Journal = {Computer Vision and Image Understanding},
    Title = {Minimal-Delay Distance Transform for
             Neighborhood-Sequence Distances in 2{D} and 3{D}},
    Year = {2013},
    Note = {In Press}
}
```

The original publication is available at ScienceDirect.com:
http://dx.doi.org/10.1016/j.cviu.2012.08.015

SHERPA/RoMEO: green
http://www.sherpa.ac.uk/romeo/issn/1077-3142/

*Corresponding author

Email addresses: Nicolas.Normand@polytech.univ-nantes.fr (Nicolas Normand), robin@cb.uu.se (Robin Strand), Pierre.Evenou@polytech.univ-nantes.fr (Pierre Evenou), Aurore.Arlicot@polytech.univ-nantes.fr (Aurore Arlicot)

Minimal-Delay Distance Transform for Neighborhood-Sequence Distances in 2D and 3D

Nicolas Normand[a,*], Robin Strand[b], Pierre Evenou[a], Aurore Arlicot[a]

[a] *LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597, Polytech Nantes, Rue Christian Pauc, La Chantrerie, 44306 Nantes Cedex 3, France*
[b] *Centre for Image Analysis, Uppsala University, Sweden*

Abstract

This paper presents a path-based distance where local displacement costs vary both according to the displacement vector and with the travelled distance. The corresponding distance transform algorithm is similar in its form to classical propagation-based algorithms, but the more variable distance increments are either stored in look-up-tables or computed on-the-fly. These distances and distance transform extend neighborhood-sequence distances, chamfer distances and generalized distances based on Minkowski sums. We introduce algorithms to compute a translated version of a neighborhood sequence distance map both for periodic and aperiodic sequences and a method to derive the centered distance map. A decomposition of the grid neighbors, in $\mathbb{Z}^2$ and $\mathbb{Z}^3$, allows to significantly decrease the number of displacement vectors needed for the distance transform. Overall, the distance transform can be computed with minimal delay, without the need to wait for the whole input image before beginning to provide the result image.

*Corresponding author
Email addresses: Nicolas.Normand@polytech.univ-nantes.fr (Nicolas Normand), robin@cb.uu.se (Robin Strand), Pierre.Evenou@polytech.univ-nantes.fr (Pierre Evenou), Aurore.Arlicot@polytech.univ-nantes.fr (Aurore Arlicot)

1. Introduction

In [1] discrete distances were introduced along with sequential algorithms to compute the distance transform (DT) of a binary image, where each point is mapped to its distance to the background. These discrete distances are built from adjacency and connected paths (path-based distances): the distance between two points is equal to the cost of the shortest path that joins them. For distance d_4 ("d" in [1]), defined in the square grid $\mathbb{Z}^2$, each point has four neighbors located at its top, left, bottom and right edges. Similarly, for distance d_8 ("d^*" in [1]), each point has four extra diagonally located neighbors. In both cases, d_4 and d_8, the cost of a path is defined as the number of displacements. These simple distances have been extended in different ways, by changing the neighborhood depending on the travelled distance [2, 3], by weighting displacements [3, 4], or even by a mixed approach of weighted neighborhood sequence paths [5].

Section 2 presents definitions of distances, balls and some properties of non-decreasing integer sequences that will be used later. Section 3 introduces a new generalization of path-based distances where displacement costs vary both on the displacement vector and on the travelled distance. An application is presented in Section 4 for the efficient computation of neighborhood-sequence DT in 2D and 3D.

2. Preliminaries

2.1. Lambek-Moser inverse of a integer sequence [6].

Let the function f define a non-decreasing sequence of integers $(f(1), f(2), \dots)$ For the sake of simplicity, we call f a sequence. The inverse sequence of f, denoted by $f^\dagger$, is a non-decreasing sequence of integers defined by:

$$f(m) < n \;\Leftrightarrow\; f^\dagger(n) \not< m\,. \quad (1)$$

$f^\dagger(n)$ can be seen as the count of values less than n in f, the last index of a value less than n in f, the index that precedes the first value greater than or equal to n:

$$\begin{aligned} f^\dagger(n) &= \text{Card}\left(\left\{m : f(m) < n\right\}\right) \\ &= \max\left\{m : f(m) < n\right\} \text{ or } 0 \text{ if } f(1) \geq n \text{ or } \infty \text{ if } \forall m, f(m) < n \\ &= \min\left\{m : f(m) \geq n\right\} - 1 \text{ or } \infty \text{ if } \forall m, f(m) < n \, . \end{aligned} \tag{2}$$

Table 1 shows a non-decreasing sequence f and its inverse $f^\dagger$. $f^\dagger(6) = 3$ because there are exactly 3 values less than 6 in f: $f(1)$, $f(2)$ and $f(3)$.

An interesting property of a sequence f and its inverse $f^\dagger$ is that, by adding the rank of each term to these two sequences, we obtain two complementary sequences $f(m) + m$ and $f^\dagger(n) + n$ [6], as shown in Table 1. This property extends the results given by Ostrowski, Hyslop, and Aitken [7] about Beatty sequences [8]. From [6], we deduce that the inverse of the sequence $f(m) = \lfloor \tau m \rfloor$ with a scalar τ, is $f^\dagger(n) = \lceil \frac{n}{\tau} - 1 \rceil$ so $f(m) + m = \lfloor (1 + \tau)m \rfloor$ and $f^\dagger(n) + n = \lceil (1 + \frac{1}{\tau})n - 1 \rceil$ are two complementary sequences. If τ is irrational, these sequences are Beatty sequences and, for any positive n, $\lceil (1 + \frac{1}{\tau})n - 1 \rceil$ is equal to $\lfloor (1 + \frac{1}{\tau})n \rfloor$ as given in [8].

Hajdu and Hajdu introduced Beatty sequences in the context of neighborhood sequence-distances [9]. Beside their interest in defining neighborhood sequences, Lambek-Moser inverse sequences will be used in following Section 3.2 as a link between the propagation of distance values and the construction of disks. With Proposition 1, we introduce a new use of the Lambek-Moser inverse to iterate over non-decreasing integer sequences.

Proposition 1. $f^\dagger(f(m) + 1) + 1$ *is the rank of the smallest term greater than* m *where* f *increases.*

m or n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$f(n)$	1	3	3	6	7	9	9	12	13	15	15	18	19	21
$f^\dagger(m)$	0	1	1	3	3	3	4	5	5	7	7	7	8	9
$f(n)+n$	2	5	6	10	12	15	16	20	22	25	26	30	32	35
$f^\dagger(m)+m$	1	3	4	7	8	9	11	13	14	17	18	19	21	23
$f^\dagger(f(n)+1)+1$	2	4	4	5	6	8	8	9	10	12	12	13	14	16

Table 1: Example of a non-decreasing sequence f and its Lambek-Moser inverse. f is the cumulative sequence of the periodic sequence $(1, 2, 0, 3)$, $f^\dagger$ its inverse. $f^\dagger(6) = 3$ because there are exactly 3 values less than 6 in f. Each positive integer appears exactly once in the range of $f(n) + n$ or $f^\dagger(m) + m$. $f^\dagger(f(n) + 1) + 1$ locates the rank of the next f increase.

Proof.

$$f^\dagger(f(m)+1)+1 = m' \Leftrightarrow \begin{cases} f^\dagger(f(m)+1) < m' \\ f^\dagger(f(m)+1) \geq m'-1 \end{cases}$$

$$\Leftrightarrow \begin{cases} f(m') \geq f(m)+1 \\ f(m'-1) < f(m)+1 \end{cases}$$

$$\Leftrightarrow f(m') > f(m) \text{ and } f(m'-1) \leq f(m)\,.$$

For example, in Table 1, $f(6) = 9$, $f^\dagger(f(6)+1)+1 = 8$ is the rank of appearance of the first value greater than 9, which is 12 in this case. If we extend f with $f(0) = 0$, and define g by $g(0) = 0$, $g(n+1) = f^\dagger(f(g(n))+1)+1$, then $f(g(n))$ takes, in increasing order, all the values of f, each one appearing once.

2.2. Path-based distances

Definition 1 (Discrete distance). A function $d : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{N}$ is a *translation-invariant distance* if the following conditions holds $\forall x, y, z \in \mathbb{Z}^n$, $\forall \lambda \in \mathbb{Z}$:

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

In the following sections, we will drop definiteness and symmetry to define "asymmetric pseudo-distances".

Definition 2 (Path-based distance). Let $\mathcal{P}(p,q)$ be the set of paths from $p \in \mathbb{Z}^n$ to $q \in \mathbb{Z}^n$. $d : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{N}$ is a *path-based distance* if:

(i) $\forall (p,q), d(p,q) = \min \left\{ \mathcal{L}(P), P \in \mathcal{P}(p,q) \right\}$,

(ii) d is a distance,

where $\mathcal{L}(P)$ is the length (or cost) of the path P. A path $P \in \mathcal{P}(p,q)$ is *minimal* if its length is equal to $d(p,q)$. It it usually not unique between p and q.

Definition 3 (k-neighbor). In the nD square, cubic or hypercubic grid, two points p and q are k-neighbors, $0 < k \leq n$, if their cubic cells share a face of dimension at least $n-k$, *i.e.*:

$$\begin{aligned} \sum_{i=1\ldots n} |p_i - q_i| \leq k \ , \\ \max_{i=1\ldots n} \left\{ |p_i - q_i| \right\} \leq 1 \ , \end{aligned} \tag{3}$$

where p_i stands for the i^{th} component of p.

The k-neighborhood of p, denoted by $\mathcal{N}_k(p)$, is the set of k-neighbors of p and the k-neighborhood $\mathcal{N}_k$, in a translation-invariant context, is the set of vectors from any point p to its k-neighbors. In the 2D square grid, k-neighborhoods are defined as follows:

$$\mathcal{N}_1 = \{(0,0),(\pm 1,0),(0,\pm 1)\} \text{ and } \mathcal{N}_2 = \{(0,0),(\pm 1,0),(0,\pm 1),(\pm 1,\pm 1)\} \ .$$

$\mathcal{N}_1$ and $\mathcal{N}_2$ are often referred to as 4- and 8-neighborhoods because they contain respectively four and eight non-zero displacement vectors.

For distance d_4, a path is a sequence of points $(p_0, \ldots, p_n)$ where each pair of successive points (p_{i-1}, p_i) are 1-neighbors and the length of the path is the number of displacements, n. Whereas the 2-neighborhood is used for distance d_8 [1, 2]. Distances d_4 and d_8 have a high rotational dependancy as noticed by Rosenfeld and Pfaltz.

A neighborhood sequence (NS) $B = (b(i))_{i>0}$ is a sequence where each $b(i)$ denotes a neighborhood relation in $\mathbb{Z}^n$ in the sense of Definition 3. If B is l-periodic, *i.e.* if for some finite, strictly positive $l \in \mathbb{Z}_+$, $b(i) = b(i+l)$ is valid for all $i \in \mathbb{N}^*$, then we write $B = \left(\overline{b(1), b(2), \ldots, b(l)}\right)$. A B-path is a sequence of points $(p_0, \ldots, p_n)$ where each pair of successive points (p_{i-1}, p_i) are $B(i)$-neighbors. Given the NS B, the NS-distance d_B is the path-based distance whose paths are only B-paths and the length of a path is the number of displacements. d_4 and d_8 can be seen as NS-distances $d_{(\overline{1})}$ and $d_{(\overline{2})}$ with 1-periodic sequences $B = (\overline{1})$ and $B = (\overline{2})$, but the simplest NS-distance that combines both neighborhoods is the octagonal distance $d_{(\overline{1,2})}$ with sequence $B = (\overline{1,2})$ [2].

The notation $\mathbf{1}_B(r)$ for $\mathcal{N}_1$, more generally, $\mathbf{j}_B(r)$ for $\mathcal{N}_j$, is used to count the occurrences of the neighborhood in B up to position r:

$$\mathbf{j}_B(r) = \mathrm{Card}\left(\{i : b(i) = j, 1 \leq i \leq r\}\right) .$$

A different approach is used for chamfer, or *weighted*, distances where each displacement vector $\overrightarrow{v_k}$ in a neighborhood $\mathcal{N}$ is associated to the *weight* (or *local cost*) w_k [3, 4, 10]. A chamfer mask $\mathcal{M}$ is a central symmetric set of *weightings* $(\overrightarrow{v_k}; w_k)$ with at least a base of $\mathbb{Z}^n$:

$$\mathcal{M} = \{(\overrightarrow{v_k}; w_k) \in \mathbb{Z}^n \times \mathbb{N}^*\}_{1 \leq k \leq m} .$$

The length of the path $(p_0, \ldots, p_n)$ is the sum of the displacements costs:

$$\mathcal{L}(p_0, \ldots, p_n) = \sum_{i=1}^{n} w(i), \text{ where } (\overrightarrow{p_{i-1}p_i}; w(i)) \in \mathcal{M} .$$

Definition 4 (Ball). The disk $D(p, r)$ of center p and radius r and the symmetrical disk $\check{D}(p, r)$ are the sets:

$$\begin{aligned} D(p, r) &= \{q \,:\, d(p, q) \leq r\} , \\ \check{D}(p, r) &= \{q \,:\, d(q, p) \leq r\} . \end{aligned} \tag{4}$$

By definition, any disk of negative radius is empty and the disk of radius 0 only contains its center ($D(p, 0) = \{p\}$).

Definition 5 (Distance transform). The *distance transform* DT_X of the set X is a function that maps each point p to its distance from the complement of X:

$$\begin{aligned} \mathrm{DT}_X : \mathbb{Z}^n &\rightarrow \mathbb{N} \\ \mathrm{DT}_X(p) &= \min\left\{d(q,p) \,:\, q \in \mathbb{Z}^n \setminus X\right\} . \end{aligned} \tag{5}$$

Alternatively, since all points at a distance less than $\mathrm{DT}_X(p)$ to p belong to X, because $\check{D}(p, \mathrm{DT}_X(p) - 1) \subset X$, and at least one point at a distance to p equal to $\mathrm{DT}_X(p)$ is not in X, because $\check{D}(p, \mathrm{DT}_X(p)) \not\subset X$, then:

$$\mathrm{DT}_X(p) \geq r \;\Leftrightarrow\; \check{D}(p, r-1) \subset X . \tag{6}$$

The DT is usually defined as the distance *to* the background which is equivalent to the distance *from* the background by symmetry. The equivalence is lost with asymmetric distances, and this definition better reflects the fact that DT algorithms always propagate paths from the background points.

Efficient algorithms exist to compute the DT of path-based distances based on the propagation of values from the neighbor points with the addition of local costs. They require two scans in reverse orders for the simple d_4 and d_8 distances [1], two scans for chamfer distances [3, 4]. NS-distances have an extra complexity because the cost of a path is not invariant to the order of its displacement vectors. NS-DT algorithms are known with four scans [11] and three scans [12].

2.3. Path-based distances and displacement costs

In the following, we show that path-based distances presented in Section 2.2, despite having different definitions of paths and path lengths, can be described with a unique paradigm in which they are only characterized by the local costs of displacement vectors.

For a simple distance, a path is a sequence of points where the difference between two successive points is a displacement vector taken in a fixed neighborhood $\mathcal{N}$, and the cost (or length) of a path is the number of its displacements. The cost of the path $(p_0, \ldots, p_n, p_n + \vec{v})$ derives from the cost of the path $(p_0, \ldots, p_n)$:

$$\mathcal{L}(p_0, \ldots, p_n) = r \;\Rightarrow\; \forall \vec{v} \in \mathcal{N}, \mathcal{L}(p_0, \ldots, p_n, p_n + \vec{v}) = r + 1 . \tag{7}$$

Rosenfeld and Pfaltz specifically forbid paths where a point appears more than once [1]. This restriction has no effect on the distance because a path where a point appears more than once can not be minimal. In a similar manner, they exclude the null vector from the neighborhood, forbidding a point to appear several times consecutively. As before, it has no effect on the distance. Notice that, in terms of distance, forbidding a path is equivalent to giving it an infinite cost, so that it can not be minimal. Equation (7) can be rewritten as:

$$\mathcal{L}(p_0, \ldots, p_n) = r \Rightarrow \forall \vec{v}, \mathcal{L}(p_0, \ldots, p_n, p_n + \vec{v}) = r + c_{\vec{v}} \; ,$$

where

$$c_{\vec{v}} = \begin{cases} 1 & \text{if } \vec{v} \in \mathcal{N} \\ \infty & \text{otherwise} \end{cases} \; .$$

For a NS-distance characterized by the sequence B:

$$\mathcal{L}(p_0, \ldots, p_n) = r \Rightarrow \forall \vec{v}, \mathcal{L}(p_0, \ldots, p_n, p_n + \vec{v}) = r + c^B_{\vec{v}}(r) \; , \tag{8}$$

where the displacement cost $c^B_{\vec{v}}(r)$ is 1 for a displacement vector in the neighborhood $B(r+1)$ and infinite otherwise:

$$c^B_{\vec{v}}(r) = \begin{cases} 1 & \text{if } \vec{v} \in \mathcal{N}_{B(r+1)} \\ \infty & \text{otherwise} \end{cases} \; . \tag{9}$$

For a weighted distance with mask $\mathcal{M} = \{(\vec{v_k}; w_k) \in \mathbb{Z}^n \times \mathbb{N}^*\}_{1 \leq k \leq m}$, the distance increment only depends on the displacement vector, but not on the distance already travelled:

$$\mathcal{L}(p_0, \ldots, p_n) = r \Rightarrow \forall \vec{v}, \mathcal{L}(p_0, \ldots, p_n, p_n + \vec{v}) = r + c_{\vec{v}} \; , \tag{10}$$

$$c_{\vec{v}} = \begin{cases} w & \text{if } (\vec{v}; w) \in \mathcal{M} \\ \infty & \text{otherwise} \end{cases} \; . \tag{11}$$

Briefly, the displacement cost for a vector $\vec{v}$ and the travelled distance r, is 1 or ∞, independently of r for simple distances, is equal to 1 or ∞ whether $\vec{v}$ belongs or not to $\mathcal{N}_{B(r)}$ for a NS-distance, is in $\mathbb{N}^* \cup \{\infty\}$ according to the chamfer mask and independently of r for a weighted distance.

In the following, we propose to use a displacement cost, denoted by $c_{\overrightarrow{v}}(r)$, with values in $\mathbb{N}^* \cup \{\infty\}$, that depends both on the displacement vector $\overrightarrow{v}$ and on the travelled distance r. According to the previous remarks, the cost associated to the null displacement will always be unitary:

$$\forall r \in \mathbb{N}, c_{\overrightarrow{0}}(r) = 1 \,. \tag{12}$$

3. Path-based Distance with Varying Weights

3.1. Definition and Properties

Definition 6 (Path). We call *path from p to q*, any finite sequence of points $P = (p = p_0, p_1, \dots, p_n = q)$ with at least one point, and denote by $\mathcal{P}(p, q)$, the set of these paths.

Notice that this definition of a path is not related to any adjacency relation. The sequence $P = (p)$ is allowed as a path from p to itself. It is distinct from $P = (p, p)$, the path from p to itself with a null displacement.

Definition 7 (Partial and total costs of a path). Let $\mathcal{N}$ be a set of vectors containing the null vector $\overrightarrow{0}$ and the positive displacement costs $c_{\overrightarrow{v}}$ (with $c_{\overrightarrow{0}}(r) = 1$ and $c_{\overrightarrow{v} \notin \mathcal{N}}(r) = \infty$). The total cost of the path $P = (p_0, p_1, \dots, p_n)$ is:

$$\begin{aligned} \mathcal{L}(P) &= \mathcal{L}_n(P)\,, \\ \mathcal{L}_0(P) &= \mathcal{L}(p_0) = 0\,, \\ \mathcal{L}_{i+1}(P) &= \mathcal{L}(p_0, \dots, p_{i+1}) = \mathcal{L}_i(P) + c_{\overrightarrow{p_i p_{i+1}}}(\mathcal{L}_i(P))\,, \end{aligned} \tag{13}$$

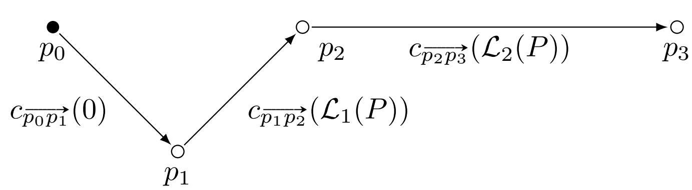


Figure 1: Total cost of a path $P = (p_0, p_1, p_2)$. Costs of displacements $\overrightarrow{p_0p_1}$, $\overrightarrow{p_1p_2}$ and $\overrightarrow{p_2p_3}$ depend on the partial costs $\mathcal{L}_0(P) = 0$, $\mathcal{L}_1(P) = c_{\overrightarrow{p_0p_1}}(0) + 0$ and $\mathcal{L}_2(P) = c_{\overrightarrow{p_1p_2}}(\mathcal{L}_1(P)) + \mathcal{L}_1(P)$. The total cost of P is $c_{\overrightarrow{p_2p_3}}(\mathcal{L}_2(P)) + \mathcal{L}_2(P)$.

where $\mathcal{L}_i(P)$ is the partial cost of the path truncated to its $i+1$ first points (*i.e.* to its i first displacements).

Figure 1 illustrates the individual displacement costs in a path, each one depending on the partial length of the path.

Definition 8 (Absolute and relative costs of displacement). We use the notation $C_{\overrightarrow{v_k}}(r) = r + c_{\overrightarrow{v_k}}(r)$. $c_{\overrightarrow{v_k}}(r)$ is the *relative cost* of the displacement $\overrightarrow{v_k}$ when the distance travelled so forth is r. $C_{\overrightarrow{v_k}}(r)$ represents the partial cost of the path after this displacement (the *absolute cost* of this displacement):

$$\mathcal{L}_{i+1}(P) = \mathcal{L}_i(P) + c_{\overrightarrow{p_i p_{i+1}}}(\mathcal{L}_i(P)) = C_{\overrightarrow{p_i p_{i+1}}}(\mathcal{L}_i(P)) \; . \tag{14}$$

Definition 9 (Pseudo-distance). The *pseudo-distance* induced by $\left(\{\overrightarrow{v_k}\}, c_{\overrightarrow{v_k}}\right)$ is defined by:

$$d(p,q) = 0 \;\Leftrightarrow\; p = q$$
$$d(p,q) = \min_{P \in \mathcal{P}(p,q)} \left\{\mathcal{L}(P)\right\} \; .$$

Definition 10 (Minimal cost of displacement). We call *minimal relative* (resp. *absolute*) *cost of displacement*, denoted by $\hat{c}$ (resp. $\hat{C}$), the quantity $\hat{c}_{\overrightarrow{v}}(r) = \min\left\{c_{\overrightarrow{v}}(s) + s - r, \forall s \geq r\right\}$ (resp. $\hat{C}_{\overrightarrow{v}}(r) = \min\left\{C_{\overrightarrow{v}}(s), \forall s \geq r\right\}$).

Proposition 2 (Preservation of cost order by concatenation).
Appending the same displacement to existing paths preserves the relation order of their costs. Let $P = (p_1, \cdots, p_{n_P})$ and $Q = (q_1, \cdots, q_{n_Q})$ be two paths with costs $\mathcal{L}(P)$ and $\mathcal{L}(Q)$, $\overrightarrow{v}$ a vector and $P' = (p_1, \cdots, p_{n_P}, p_{n_P} + \overrightarrow{v})$, $Q' = (q_1, \cdots, q_{n_Q}, q_{n_Q} + \overrightarrow{v})$ the extended paths with costs $\mathcal{L}(P')$ and $\mathcal{L}(Q')$ measured with minimal displacement costs. Then:

$$\mathcal{L}(P) \leq \mathcal{L}(Q) \;\Rightarrow\; \mathcal{L}(P') \leq \mathcal{L}(Q') \; . \tag{15}$$

Proof. *From (14), $\mathcal{L}(P') = \hat{C}_{\overrightarrow{v}}(\mathcal{L}(P))$ and $\mathcal{L}(Q') = \hat{C}_{\overrightarrow{v}}(\mathcal{L}(Q))$. By definition of $\hat{C}_{\overrightarrow{v}}$, $s \leq r \;\Rightarrow\; \hat{C}_{\overrightarrow{v}}(s) \leq \hat{C}_{\overrightarrow{v}}(r)$, which gives (15).*

Proposition 3. *Let $\mathcal{N} = \{\overrightarrow{v_k}\}$ be a set of vectors and, $c_{\overrightarrow{v}}(r)$, the displacement costs for these vectors. There exists a path P from p to q of cost $\mathcal{L}(P) = r$ measured with costs $c_{\overrightarrow{v}}(r)$ if and only if there exists a path P' from p to q of cost $\mathcal{L}'(P') = r$ measured with the minimal displacement costs $\hat{c}_{\overrightarrow{v}}(r)$.*

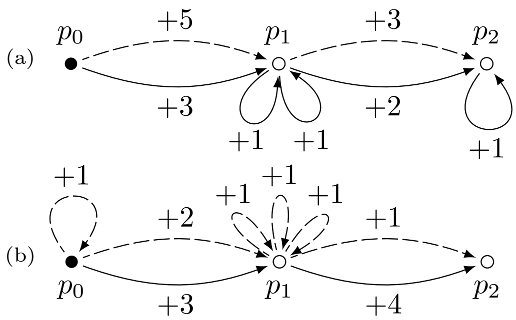
PROOF. *Consider the cost of P after i displacements, $\mathcal{L}_i(P) = \mathcal{L}_i(p_0, p_1, \ldots, p_i)$, we note $m_0 = 1, m_{0<i\leq n} = 1 + \mathcal{L}_i(P) - \mathcal{L}_{i-1}(P) - \hat{c}_{\overrightarrow{p_{i-1}p_i}}(\mathcal{L}_{i-1}(P)) = 1 + c_{\overrightarrow{p_{i-1}p_i}}(\mathcal{L}_{i-1}(P)) - \hat{c}_{\overrightarrow{p_{i-1}p_i}}(\mathcal{L}_{i-1}(P))$ and $M_i = \sum_{j=0}^{i} m_i$ the cumulated sum of m_i. Clearly, if $\mathcal{L}(P)$ is finite then each m_i is finite and positive because $\hat{c}_{\overrightarrow{v}}(r)$ is less than or equal to $c_{\overrightarrow{v}}(r)$ by construction. Let P' be the (finite) path obtained by m_i occurrences of each point p_i:*

$$P' = (p_0, \underbrace{p_1 \ldots p_1}_{m_1}, \ldots, \underbrace{p_i \ldots p_i}_{m_i}, \ldots, \underbrace{p_n \ldots p_n}_{m_n}) \,.$$

We take as an induction hypothesis that the partial cost of P' after m_i occurrences of p_i, $\mathcal{L}'_{M_i-1}(P')$, is equal to $\mathcal{L}_i(P)$. It holds for $i = 0$ because $\mathcal{L}'_{M_0-1}(P') = \mathcal{L}'_{m_0-1}(P') = \mathcal{L}'_0(P') = 0 = \mathcal{L}_0(P)$. If the hypothesis holds for $i-1$, then the partial cost of P' after the first occurrence of p_i is $\mathcal{L}'_{M_{i-1}}(P') = \mathcal{L}_{i-1}(P) + \hat{c}_{\overrightarrow{p_{i-1}p_i}}(\mathcal{L}_{i-1}(P))$, and after $m_i - 1$ repeats of p_i, equals $\mathcal{L}'_{M_{i-1}+m_i-1}(P') = \mathcal{L}'_{M_i-1}(P') = \mathcal{L}_{i-1}(P) + \hat{c}_{\overrightarrow{p_{i-1}p_i}}(\mathcal{L}_{i-1}(P)) + m_i - 1 = \mathcal{L}_{i-1}(P) + c_{\overrightarrow{p_{i-1}p_i}}(\mathcal{L}_{i-1}(P)) = \mathcal{L}_i(P)$ and the hypothesis is true at rank i. Therefore, for every path of finite cost r measured with $\mathcal{L}$, there exists a path with the same cost measured with $\mathcal{L}'$. This is shown in Fig. 2a.

Conversely, let P' be a path with finite cost measured by $\mathcal{L}'$. We build a path P where each point of P' appears m'_i times consecutively with m'_i such that $m'_i - 1 + c_{\overrightarrow{p_ip_{i+1}}}(\mathcal{L}'_i(P') + m'_i - 1) = \hat{c}_{\overrightarrow{p_ip_{i+1}}}(\mathcal{L}'_i(P'))$. By definition of $\hat{c}$, $\forall r, \exists s : \hat{c}_{\overrightarrow{v}}(r) = c_{\overrightarrow{v}}(s) + s - r$, so m'_i exists. Let $M'_0 = 0$ and $M'_{0<i\leq n} = \sum_{j=0}^{i-1} m_j$, be the cumulated sum of the previous terms of m'_i.

The induction hypothesis is that the partial cost of P, measured with $\mathcal{L}$, at the first occurence of p_i, $\mathcal{L}_{M'_i}(P)$, is equal to $\mathcal{L}'_i(P')$. It holds for $i = 0$ with a null partial cost $\mathcal{L}_{M'_0}(P) = \mathcal{L}_0(P) = 0 = \mathcal{L}'_0(P')$. If the hypothesis holds at rank i, the partial cost of P, after $m'_i - 1$ repetitions of p_i, if $\mathcal{L}_{M'_i+m'_i-1}(P) =$



r	0	1	2	3	4	5	6
$c_{\vec{v}}(r)$	5	2	∞	∞	∞	3	1
$\hat{c}_{\vec{v}}(r)$	3	2	5	4	3	2	1

r	0	1	2	3	4	5	6
$c_{\vec{v}}(r)$	5	2	∞	∞	∞	3	1
$\hat{c}_{\vec{v}}(r)$	3	2	5	4	3	2	1

Figure 2: (a) Given $P = (p_0, p_1, p_2)$, shown with dashed lines, has a total cost $\mathcal{L}(P) = 8$ measured with displacement costs $c_{\vec{v}}$. $P' = (p_0, p_1, p_1, p_1, p_2, p_2)$, solid lines, is built in such a way that its cost $\mathcal{L}'(P')$ measured with minimal displacement costs $\hat{c}_{\vec{v}}$, is equal to $\mathcal{L}(P) = 8$. (b) Given $P' = (p_0, p_1, p_2)$, shown with solid lines, has a total cost $\mathcal{L}'(P') = 7$ measured with displacement costs $\hat{c}_{\vec{v}}$. $P = (p_0, p_0, p_1, p_1, p_1, p_1, p_2)$, dashed lines, is built in such a way that $\mathcal{L}(P) = \mathcal{L}'(P') = 7$.

$\mathcal{L}_{M'_i}(P) + m'_i - 1 = \mathcal{L}'_i(P') + m'_i - 1$, and at the first occurence of p_{i+1}, equals $\mathcal{L}'_i(P') + m'_i - 1 + c_{\overrightarrow{p_i p_{i+1}}}(\mathcal{L}'_i(P') + m'_i - 1) = \mathcal{L}'_i(P') + \hat{c}_{\overrightarrow{p_i p_{i+1}}}(\mathcal{L}'_i(P')) = \mathcal{L}'_{i+1}(P')$ and the hypothesis also holds at rank $i+1$. An example of such a path is shown on Fig. 2b.

Corollary 1. *Displacement costs $c_{\vec{v}}$ and $\hat{c}_{\vec{v}}$ induce the same pseudo-distance.*

According to (12), any path from p to q of cost less than r can be extended with null displacements to reach cost r:

$$\mathcal{L}(p_0, \ldots, p_n = q) = s < r \;\Rightarrow\; \mathcal{L}(p_0, \ldots, \underbrace{p_n = q, \ldots, q}_{1+r-s}) = r\ . \tag{16}$$

Proposition 4. *There exists a path of cost r from p to q if and only if $d(p, q) \leq r$.*

Proof. *If a path of cost r from p to q exists then by definition of the distance, $d(p, q) = r$ if P cost is minimal, $d(p, q) < r$ otherwise. Conversely, if $d(p, q) = s$ then there exists a path of cost s from p to q that, according to (16), can be extended to cost $r \geq s$.*

Corollary 2. *For any value of r greater than or equal to $d(p,q)$, there exists a path from p to q which cost is exactly r. The closed disk centered in p with radius r is the set of points for which a path from p of cost equal to r exists:*

$$q \in D(p,r) \Leftrightarrow \exists P \in \mathcal{P}(p,q), \mathcal{L}(P) = r\,. \tag{17}$$

An iterative construction rule of balls is deduced from (17):

$$\begin{aligned} \forall r > 0, D(p,r) &= \bigcup_{\vec{v} \in \mathcal{N}} \big\{ q : \exists P \in \mathcal{P}(p, q - \vec{v}) \text{ and } C_{\vec{v}}(\mathcal{L}(P)) = r \big\} \\ &= \bigcup_{\substack{\vec{v} \in \mathcal{N} \\ s\,:\,C_{\vec{v}}(s) = r}} D(p + \vec{v}, s)\;. \end{aligned} \tag{18}$$

3.2. Iterative Construction of Shapes

Let $\big(D(O,r)\big)$ be a sequence of balls built iteratively using:

$$D(O,r) = \begin{cases} \emptyset & \text{if } r < 0 \\ \big\{O\big\} & \text{if } r = 0 \\ \bigcup\limits_{\vec{v} \in \mathcal{N}} D(O + \vec{v}, S_{\vec{v}}(r) - 1) & \text{otherwise} \end{cases} \tag{19}$$

where the *construction values* $S_{\vec{v}}$ are non decreasing sequences of natural integers, in particular $\forall r > 0, S_{\vec{0}}(r) = r$. A generalized distance that produces these balls was shown in [13], along with a method to decompose any convex polygon into a sequence of balls with a few neighbors $\vec{v}$. Examples of such decompositions are provided in Fig. 5 and Tables 2 and 3 for the 2D case, in Fig. 6 and Table 4 for the 3D case.

Using the Lambek-Moser inverse and notations used in this paper, we can reformulate the expression of the displacement costs from the construction rules as:

$$\forall r \geq 0, C_{\vec{v}}(r) = S^{\dagger}_{\vec{v}}(r+1) + 1\,. \tag{20}$$

Note that a finite sequence of balls with maximal radius l is practically equivalent to an infinite sequence of balls where all balls with radii greater than or equal to l are equal. In this case, all values of $S_{\vec{v}}$ at index l and beyond are equal and don't exceed the value l. As a consequence, $C^{\dagger}$ is infinite at index l and beyond.

Proposition 5. *If all finite values of absolute costs for radii less than r_1 do not exceed r_1 ($\forall 0 \leq r < r_1, C(r) \leq r_1$ or $C(r) = \infty$) then all balls of radius r_2 greater than or equal to r_1 are the Minkowski sum of $D(p, r_1)$ with the disk $D'(O, r_2 - r_1)$ produced by the displacement costs $c'_{\vec{v}}(r) = c_{\vec{v}}(r + r_1)$ (i.e. $C'_{\vec{v}}(r) = C_{\vec{v}}(r + r_1) - r_1$):*

$$\forall r < r_1, C(r) \leq r_1 \;\Rightarrow\; \forall r_2 \geq r_1, D(p, r_2) = D(p, r_1) \oplus D'(O, r_2 - r_1)\ . \quad (21)$$

Proof. *(21) holds for $r_2 = r_1$: $D(p, r_1) = D(p, r_1) \oplus \{O\}$. Suppose (21) holds in the interval $[r_1, r_2]$. Only values of s in the interval $[r_1, r_2]$ can be such that $C_{\vec{v}}(s) = r_2 + 1$ and (21) applies to $D(p + \vec{v}, s)$, so (18) can be written as:*

$$\begin{aligned} D(p, r_2 + 1) &= \bigcup_{\substack{\vec{v} \in \mathcal{N} \\ s\,:\, C_{\vec{v}}(s) = r_2 + 1}} D(p + \vec{v}, r_1) \oplus D'(O, s - r_1) \\ &= D(p + \vec{v}, r_1) \oplus \bigcup_{\substack{\vec{v} \in \mathcal{N} \\ s\,:\, C_{\vec{v}}(s) = r_2 + 1}} D'(O, s - r_1) \\ &= D(p + \vec{v}, r_1) \oplus \bigcup_{\substack{\vec{v} \in \mathcal{N} \\ s\,:\, C'_{\vec{v}}(s) = r_2 - r_1 + 1}} D'(O, s) \\ &= D(p + \vec{v}, r_1) \oplus D'(O, r_2 - r_1 + 1)\ . \end{aligned}$$

Hence, (21) holds in the interval $[r_1, r_2 + 1]$.

Proposition 5 provides a sufficient (but not necessary) condition that enables to build sequences of Minkowski sums. In particular, when relative displacement costs are either 1 or ∞ as in (9), then Proposition 5 applies to all positive radius values so that the NS-distance balls are produced by a sequence of Minkowski sums with the sequence of neighborhoods, as it is well known. In the following Section 4.4, Proposition 5 will be used along with (19) and (20) to build sequences of balls in which Minkowski sums are decomposed into several steps.

4. Minimal Delay NS-Distance Transform

In [14], Wang and Bertrand, proposed a single scan asymmetric generalized DT based on a neighborhood for which there exists a scanning order such that when a point p in the image is scanned, all neighbors of p have already been scanned (forward scan condition). Then, they extended this result to a sequence where two neighborhoods with forward scan condition are alternated (*i.e.* $B = (1, 2)$) [15]. In the following we propose a method to compute, using a single raster scan, an asymmetric generalized DT based on any number of neighborhoods having forward scan condition used in an arbitrary order defined by a sequence B, either periodic or not. For our purpose, we will use translated versions of regular NS-distances neighborhoods, in order to meet the forward scan condition for each of them. The resulting translated distance map can easily be transformed back into a regular, symmetrical, NS-distance map.

4.1. Generalized Distance Transform

Proposition 6. *The DT of an image X with the distance induced by the neighborhood $\mathcal{N}$ and the displacement costs $C_{\overrightarrow{v}}$ is such that:*

$$\mathrm{DT}_X(p) = \begin{cases} 0 & \text{if } p \notin X \\ \min\left\{\hat{C}_{\overrightarrow{v}}(\mathrm{DT}_X(p - \overrightarrow{v})), \overrightarrow{v} \in \mathcal{N}^*\right\} & \text{otherwise} \end{cases} . \tag{22}$$

where $\hat{C}_{\overrightarrow{v}}$ represents the minimal absolute displacement costs corresponding to $C_{\overrightarrow{v}}$ (Definition 10).

Proof. *Case $p \notin X$ directly results from Definitions 5 and 9. Suppose now that $p \in X$ so any path from $q \notin X$ to p has at least one displacement. Proposition 3 states that distances induced by $\left(\{\overrightarrow{v_k}\}, C_{\overrightarrow{v_k}}\right)$ and $\left(\{\overrightarrow{v_k}\}, \hat{C}_{\overrightarrow{v_k}}\right)$ are equal so we consider the latter cost increments for which Proposition 2 holds. According to Proposition 2, if $P = (q = p_0, \ldots, p_n = p - \overrightarrow{v})$ is a minimal path from q to $p - \overrightarrow{v}$ then $P' = (q = p_0, \ldots, p_n, p + \overrightarrow{v})$ has a minimal cost — among paths from q to p with second last point $p - \overrightarrow{v}$ — equal to $\hat{C}_{\overrightarrow{v}}(\mathcal{L}(P))$. So $\hat{C}_{\overrightarrow{v}}(\mathrm{DT}_X(p - \overrightarrow{v}))$ is the shortest distance from a point $q \notin X$ to p via $p - \overrightarrow{v}$. Since all paths which* *last displacement $\vec{v}$ does not belong to $\mathcal{N}$ have an infinite cost and can not be minimal, (22) holds.*

When all vectors in $\mathcal{N}^*$ are directed forward relatively to the scan order, (22) propagates paths from background pixels in a single scan. As a consequence, a generalized DT using any number of neighborhoods $\mathcal{N}_1 \ldots \mathcal{N}_n$, selected by a sequence $B, B(i) \in [1, n]$, derives directly from (9) and (22) and minimal costs given by:

$$\hat{C}_{\vec{v}}(r) = \min \left\{ s \,:\, s > r \text{ and } \vec{v} \in \mathcal{N}_{B(s)} \right\} . \tag{23}$$

Let $\chi_{\vec{v}}(r)$ denote the characteristic function of the set $\mathcal{N}_{B(r)}$ (*i.e.* $\chi_{\vec{v}}(r) = 1$ if $\vec{v} \in \mathcal{N}_{B(r)}; 0$ otherwise) and $\chi^{\Sigma}_{\vec{v}}(r)$ its cumulative sum ($\chi^{\Sigma}_{\vec{v}}(r) = \sum\limits_{s \leq r} \chi_{\vec{v}}(r)$). Then according to Proposition 1:

$$\hat{C}_{\vec{v}}(r) = [\chi^{\Sigma}_{\vec{v}}]^{\dagger}(\chi^{\Sigma}_{\vec{v}}(r) + 1) + 1 . \tag{24}$$

Algorithm 1 produces a generalized DT using any sequence of neighborhoods ($\mathcal{N}$ represents their union) in forward scan condition, using displacement costs given by (24). A similar algorithm was already presented for the decomposition of convex structuring polygons [13].

4.2. Translated NS-distance transform

The sequence of balls for a NS-distance induced by a sequence B is produced by iterative Minkowski sums of neighborhoods:

$$D(p, 0) = \{p\}, \;\; D(p, r) = D(p, r-1) \oplus \mathcal{N}_{B(r)} .$$

For each neighborhood $\mathcal{N}_j$, we apply a translation vector $\vec{t_j}$ such that the translated neighborhood $\mathcal{N}'_j = \mathcal{N}_j \oplus \{\vec{t_j}\}$ is in forward scan condition. In a translation preserved scan order, $\vec{t_j}$ translates the first visited point in $\mathcal{N}_j$ to the origin. Assuming a nD standard raster scan order:

$$\vec{t_j} = (\underbrace{0, \ldots, 0}_{n-j}, \underbrace{1, \ldots, 1}_{j}) . \tag{25}$$

Data: X: a set of points
Data: $\mathcal{N}$: neighborhood in forward scan condition
Data: $\hat{C}_{\vec{v}}$: minimal absolute displacement costs
Result: DT_X: generalized distance transform of X

foreach p *in* DT *domain, in raster scan* **do**
- **if** $p \notin X$ **then**
  - $\mathrm{DT}_X(p) \leftarrow 0$
- **else**
  - $l \leftarrow \infty$
  - **foreach** $\vec{v}$ *in* $\mathcal{N}$ **do**
    - $l \leftarrow \min\left\{l; \hat{C}_{\vec{v}}(\mathrm{DT}_X(p - \vec{v})\right\}$
  - **end**
  - $\mathrm{DT}_X(p) \leftarrow l$
- **end**

end

Algorithm 1: Single scan generalized distance transform

The translated neighborhoods $\mathcal{N}'_1$ and $\mathcal{N}'_2$ obtained with $\overrightarrow{t_1} = (0,1)$ and $\overrightarrow{t_2} = (1,1)$ are depicted in Fig. 3a and Fig. 3b. Characteristic functions for vectors in $\mathcal{N}'_1 \setminus \mathcal{N}'_2$, $\mathcal{N}'_2 \setminus \mathcal{N}'_1$ and $\mathcal{N}'_1 \cap \mathcal{N}'_2$ (see Fig. 3c-e) are respectively the first differences of $\mathbf{1}_B$, $\mathbf{2}_B$ and the constant value 1 resulting in the following minimal displacement costs:

$$\hat{C}_{\vec{v}}(r) = \begin{cases} \hat{C}^1_{\vec{v}}(r) = \mathbf{1}^\dagger_B(\mathbf{1}_B(r)+1)+1 & \text{if } \vec{v} \in \mathcal{N}'_1 \text{ and } \vec{v} \notin \mathcal{N}'_2 \\ \hat{C}^2_{\vec{v}}(r) = \mathbf{2}^\dagger_B(\mathbf{2}_B(r)+1)+1 & \text{if } \vec{v} \notin \mathcal{N}'_1 \text{ and } \vec{v} \in \mathcal{N}'_2 \\ \hat{C}^{12}_{\vec{v}}(r) = r+1 & \text{if } \vec{v} \in \mathcal{N}'_1 \text{ and } \vec{v} \in \mathcal{N}'_2 \end{cases}.$$

Periodic sequence. When B is a periodic sequence, minimal relative costs $\hat{c}_{\vec{v}}$ are also periodic sequences. Take the periodic sequence of the octagonal distance $B = (\overline{1,2})$, then $\mathbf{1}_B(r)_{r\geq 0} = (0,1,1,2,\dots)$, $\mathbf{1}^\dagger_B(r)_{r>0} = (0,2,4,\dots)$, $\hat{C}^1_{\vec{v}}(r)_{r\geq 0} = (1,3,3,5\dots)$ and $\hat{c}^1_{\vec{v}}(r)_{r\geq 0} = (1,2,1,2\dots)$. $\mathbf{1}_B(r)_{r>0}$ is the cumulative sum of the 2-periodic sequence $(\overline{1,0})$, whereas $\mathbf{1}^\dagger_B(r)_{r>0}$ is the cumulative sum with offset -2 of the 1-periodic sequence $(\overline{2})$ as given by Algorithm 2. Similarly, $\mathbf{2}_B(r)_{r\geq 0} = (0,0,1,1,2,\dots)$, $\mathbf{2}^\dagger_B(r)_{r>0} = (1,3,\dots)$, $\hat{C}^2_{\vec{v}}(r)_{r\geq 0} = (2,2,4\dots)$ and $\hat{c}^2_{\vec{v}}(r)_{r\geq 0} = (2,1,2,1\dots)$.

Rate-based sequence. Suppose now that the sequence of neighborhoods is defined as the first difference of a Beatty sequence (as in [9]): $B(r) = \lfloor \tau r \rfloor - \lfloor \tau(r-1) \rfloor$, with $\tau \in [1,2]$ so that $B(r) \in \{1,2\}$. $\mathbf{1}_B$ and $\mathbf{2}_B$ are respectively the cumulative sums of $2 - B(r) = \lceil (2-\tau)r \rceil - \lceil (2-\tau)(r-1) \rceil$ and $B(r) - 1 = \lfloor (\tau-1)r \rfloor - \lfloor (\tau-1)(r-1) \rfloor$. Then $\mathbf{1}_B(r) = \lceil (2-\tau)r \rceil$, $\mathbf{2}_B(r) = \lfloor (\tau-1)r \rfloor$, $\mathbf{1}^\dagger_B(r) = \lfloor \frac{r-1}{2-\tau} \rfloor$ and $\mathbf{2}^\dagger_B(r) = \lceil \frac{r}{\tau-1} - 1 \rceil$. This allows to compute $\hat{C}^1_{\vec{v}}$ and $\hat{C}^2_{\vec{v}}$ on the fly. For the octagonal distance, $\tau = \frac{3}{2}$, $\mathbf{1}_B(r) = \lceil \frac{r}{2} \rceil$, $\mathbf{2}_B(r) = \lfloor \frac{r}{2} \rfloor$, $\mathbf{1}^\dagger_B(r) = 2r-2$ and $\mathbf{2}^\dagger_B(r) = 2r-1$.

An exemple result of Algorithm 1 for the translated octagonal distance (with displacement costs obtained either from sequence $B = (1,2)$ either from $\tau = \frac{3}{2}$) is shown in Fig. 4b.

Data: f_p, $f(1)\cdots f(f_p)$ and f_o

Result: g_p, $g(1)\cdots g(g_p)$ and g_o

```
g_p ← Σ_{i=1}^{f_p} f(i);
n = f_o + 1;
m_0 ← f_p;
for m ← 1 to f_p do
    if f(m) ≠ 0 then  m_0 ← min{m_0 ; m};
    g(n) ← g(n) + 1;
    n ← n + f(m);
end
g_o ← m_0 − 1 − g(f_o + 1);
for m ← 1 to −f_o do  g_o ← g_o + g(m + f_o);      // Adjust g_o if f_o < 0
for m ← 0 down to −f_o do  g_o ← g_o − g(m + f_o) ; // ...or if f_o ≥ 0
```

Algorithm 2: Computation of $g^{\Sigma} = f^{\Sigma\dagger}$, inverse of the sequence f^{Σ}. f^{Σ} is the cumulative sequence of the f_p-periodical non-negative sequence f with constant offset f_o and negative values clipped to 0: $\forall r > 1, f^{\Sigma}(r) = \max\left\{0\,; f_o + \sum_{s=1}^{r} f(s)\right\}$. Likewise, g^{Σ} is the cumulative sequence of the g_p-périodical non-negative sequence g with offset g_o and negative values clipped to 0.

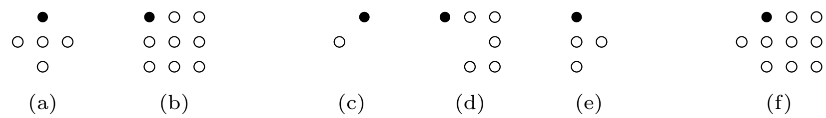


Figure 3: Neighborhoods used for the translated NS-distance transform. (a), and (b) are respectively the type 1 and 2 translated neighborhoods, $\mathcal{N}'_1$ and $\mathcal{N}'_2$. (c) and (d) and (e) are respectively $\mathcal{N}'_1 \setminus \mathcal{N}'_2$, $\mathcal{N}'_2 \setminus \mathcal{N}'_1$ and $\mathcal{N}'_1 \cap \mathcal{N}'_2$, each set associated to a different sequence of displacement costs. (f) is the whole set of neighbors, $\mathcal{N}'_1 \cup \mathcal{N}'_2$, used for the translated NS-DT.

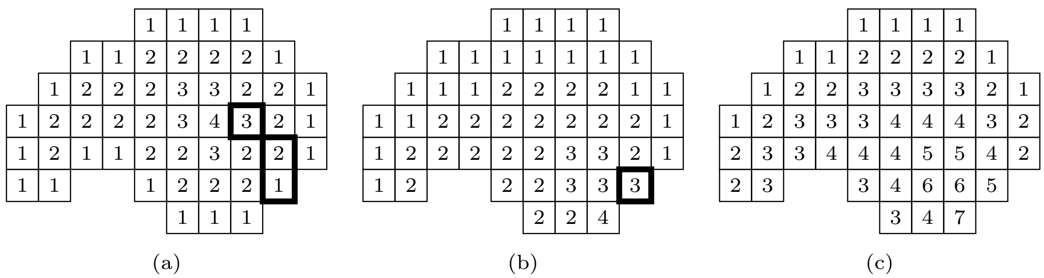


Figure 4: (a) Octagonal DT of a binary image. (b) Translated octagonal DT. Outlined centers of disks (a) are translated to the same location, outlined with value 3 (b). (c) Translated octagonal DT with intermediate disks.

4.3. Symmetric DT from asymmetric DT

Let $\left\{\overrightarrow{t}(r), r \in \mathbb{N}^*\right\}$ be a sequence of translation vectors such that the translated disks $D'(p,r) = D(p+\overrightarrow{t}(r), r)$ and $\check{D}'(p,r) = \check{D}(p-\overrightarrow{t}(r), r)$ are increasing according to the set inclusion. For a sequence of disks produced by translated neighborhoods defined in (25), the translation vectors are:

$$\begin{aligned}\overrightarrow{t}(r) &= \overrightarrow{t}(r-1) + \overrightarrow{t_{B(r)}} \\ &= \sum_j \mathbf{j}_B(r)\overrightarrow{t_j} \\ &= \left(\sum_{j=n}^{n} \mathbf{j}_B(r), \ldots, \sum_{j=1}^{n} \mathbf{j}_B(r)\right) .\end{aligned}$$

In particular, for the 2D case:

$$\overrightarrow{t}(r) = (\mathbf{2}_B(r), \mathbf{1}_B(r) + \mathbf{2}_B(r)) = (\mathbf{2}_B(r), r) . \qquad (26)$$

DT'_X has equivalence with values of DT_X:

$$\begin{aligned} \mathrm{DT}_X(p) \geq r &\Leftrightarrow \check{D}(p, r-1) \subseteq X \\ &\Leftrightarrow \check{D}'(p + \vec{t}(r-1), r-1) \subseteq X \\ &\Leftrightarrow \mathrm{DT}'_X(p + \vec{t}(r-1)) \geq r \ . \end{aligned} \tag{27}$$

Consequently:

$$\begin{aligned} \mathrm{DT}_X(p) = r &\Leftrightarrow \mathrm{DT}_X(p) \geq r \text{ and } \mathrm{DT}_X(p) < r+1 \\ &\Leftrightarrow \mathrm{DT}'_X(p + \vec{t}(r)) \leq r \leq \mathrm{DT}'_X(p + \vec{t}(r-1)) \ . \end{aligned} \tag{28}$$

Knowing $\mathrm{DT}'_X(p)$ and $\mathrm{DT}'_X(p + \vec{t})$, we can deduce the values of $\mathrm{DT}_X(p - \vec{t}(r-1))$ for all values of r between $\mathrm{DT}'_X(p+\vec{t})$ and $\mathrm{DT}'_X(p)$ for which $\vec{t}(r) = \vec{t}(r-1) + \vec{t}$, *i.e.* $\vec{t} = \overrightarrow{t_{B(r)}}$. Algorithm 3 recovers the values r of the centered DT by selecting all r in the interval $[\mathrm{DT}'_X(p+\vec{t_j}), \mathrm{DT}'_X(p)]$ such that $B(r) = j$. Iterating through values r with $B(r) = j$ is achieved using Proposition 1. Values of DT'_X become available before the whole image is computed. For instance, in a standard raster scan, as soon as line y is processed, all lines of DT'_X above $y - r_{\max}$ are fully recovered (where $r_{\max}$ denotes the maximal value of DT' in that line).

4.4. Translated NS-DT with Intermediate disks.

The full set of neighbors used by the previous algorithm is the union of all translated neighbors (excluding the new origin of the neighborhood). In 2D, this set contains nine vectors, compared to the eight neighbors needed by the classical algorithms. This count increases rapidly with the dimension: 32, 107, 350 respectively for 3D, 4D and 5D (apparently following, with a constant offset -1, sequence A126184 in Sloane's On-Line Encyclopedia of Integer Sequences [16]). In this section we will show how we can drastically reduce the count of required vectors, by further decomposing neighborhoods using set unions and translation.

Let B be a sequence of values in $[1, n]$ and $c^j_{\vec{v}}$ where $j \in [1, n]$ be n relative cost sub-sequences with finite length corresponding to the construction of n

sets of points $\mathcal{N}_j$ with $j \in [1, n]$. For the sake of simplicity, we assume that all sub-sequences have the same length l. We build the expanded sequence of displacement costs for neighbor $\vec{v}$, $c_{\vec{v}}$, by concatenation of the sub-sequences $c^1_{\vec{v}} \dots c^n_{\vec{v}}$ selected by the master sequence B:

$$\forall \vec{v}, \forall r > 0, c_{\vec{v}}(r) = c_{\vec{v}}^{B(\lfloor \frac{r}{l} \rfloor + 1)}(\langle r \rangle_l) \; .$$

By Proposition 5, the balls generated by these displacement costs are such that:

$$\forall k > 0, D(O, kl) = D(O, (k-1)l) \oplus \mathcal{N}_{B(k)} \; .$$

The i^{th} sub-sequence occupies the radii $(i-1)l$ to $il - 1$ in the expanded sequence and conversely, radius r corresponds to sub-sequence with index $i = \lfloor r/l \rfloor + 1$ starting at radius $r - \langle r \rangle_l$. According to Proposition 6, the distance transform can be computed by propagation of the minimal absolute displacement costs $\hat{C}_{\vec{v}}$. Minimal displacement costs are either equal to costs in the corresponding sub-sequences when these costs are finite, or deduced from the first value of a subsequent sub-sequence. The quantity $\hat{C}_{\vec{v}}(r)$ corresponds either to the value at index $\langle r \rangle_l$ in the current i^{th} sub-sequence: $\hat{c}_{\vec{v}}^{B(i)}(\langle r \rangle_l) + r$, or to the first value in next sub-sequence $\hat{c}^j_{\vec{v}}$ where $\vec{v}$ is used: $\hat{c}^j_{\vec{v}}(0) + l(\mathbf{j}_B^{\dagger}(\mathbf{j}_B(i) + 1))$, whichever is minimal according to Definition 10.

$$\hat{C}_{\vec{v}}(r) = \min \Big\{ \hat{c}_{\vec{v}}^{B(i)}(\langle r \rangle_l) + r; \tag{29a}$$

$$\hat{c}^j_{\vec{v}}(0) + l\mathbf{j}_B^{\dagger}(\mathbf{j}_B(i) + 1)), j \in [1, n] \Big\} \; , \tag{29b}$$

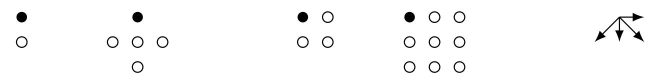
where $i = \lfloor r/l \rfloor + 1$.

Clearly, if a vector $\vec{v}$ is not used in the decomposition of the neighborhood j, then $\hat{c}^j_{\vec{v}}$ is infinite and the vector can be omitted from (29b). On the contrary, when a vector is used in all n neighborhoods, (29) is simplified to:

$$\forall j, \hat{c}^j_{\vec{v}}(0) \neq \infty \Rightarrow \hat{C}_{\vec{v}}(r) = \min \left\{ \hat{c}_{\vec{v}}^{B(i)}(\langle r \rangle_l) + r; c_{\vec{v}}^{B(i+1)}(0) + li, j \in [1, n] \right\} \; . \tag{30}$$

A further simplification holds when the displacement costs are always finite for a given vector $\vec{v}$, then we can omit (29b) in (29):

$$\forall j, \hat{c}^j_{\vec{v}}(l) \neq \infty \Rightarrow \forall r \geq 1, \hat{C}_{\vec{v}}(r) = \hat{c}_{\vec{v}}^{B(i)}(\langle r \rangle_l) + r \; . \tag{31}$$



(a) (b) (c) (d) (e)

Figure 5: Decomposition of the neighborhoods used for the translated NS-distance transform. (b), and (d) are respectively the type 1 and 2 translated neighborhoods, $\mathcal{N}_1'$ and $\mathcal{N}_2'$. (a) and (c) are intermediate disks. Only four displacement vectors (e) are needed instead of nine without this decomposition.

If B is periodic, then so is $\hat{c}_{\vec{v}}$, with a period multiplied by l. Using (29), we can avoid an actual expansion of the master sequence B by concatenation of sub-sequences. On the contrary, $\hat{C}_{\vec{v}}$ can be efficiently computed on the fly, when needed by Algorithm 1, for periodic sequences B with long periods or, *a fortiori*, for aperiodic sequences. However, for short sequences, it can be desirable to precompute $\hat{c}_{\vec{v}}$ once and for all.

4.4.1. 2D case

Consider the following decomposition of the translated 2D neighborhoods:

$$\begin{aligned}
D^1(O,1) &= D^1(O,0) \cup D^1((0,1),0)\ , \\
D^1(O,2) &= D^1(O,1) \cup D^1((0,1),1) \cup D^1((-1,1),0) \cup D^1((1,1),0)\ , \\
D^2(O,1) &= D^2(O,0) \cup D^2((0,1),0) \cup D^2((1,0),0) \cup D^2((1,1),0)\ , \\
D^2(O,2) &= D^2(O,1) \cup D^2((0,1),1) \cup D^2((1,0),1) \cup D^2((1,1),1)\ ,
\end{aligned}$$

where $D^1(O,2)$ and $D^2(O,2)$ are equal to the translated neighborhoods $\mathcal{N}_1'$ and $\mathcal{N}_2'$. The intermediate disks $D^1(O,1)$ and $D^2(O,1)$ are depicted in Fig. 5a and Fig. 5c. This decomposition is summarized by the construction values S^1 and S^2, as defined in (19), and shown in Table 2 (where $S_{(0,0)}$ is omitted). The corresponding displacements costs, deduced from (20) are shown in Table 3. Note that (30) holds for vectors $(1,1)$ and $(0,1)$ as well as (31) for $(0,1)$.

Using these two set of sequences, any combination of Minkowski sums of $\mathcal{N}_1$ and $\mathcal{N}_2$ can be obtained. Take the octagonal distance with 2-periodic sequence $B = (\overline{1,2})$. The direct application of (29) gives $\hat{c}_{\vec{v}}$ values shown in Table 3c. An example of DT$''$ is shown in Fig. 4c.

r	1	2
$S^1_{(1,0)}(r)$	0	0
$S^1_{(-1,1)}(r)$	0	1
$S^1_{(0,1)}(r)$	1	2
$S^1_{(1,1)}(r)$	0	1

r	1	2
$S^2_{(1,0)}(r)$	1	2
$S^2_{(-1,1)}(r)$	0	0
$S^2_{(0,1)}(r)$	1	2
$S^2_{(1,1)}(r)$	1	2

Table 2: Construction values for the decomposition of the two neighborhoods in 2D.

r	0	1
$c^1_{(1,0)}(r)$	∞	∞
$c^1_{(-1,1)}(r)$	2	∞
$c^1_{(0,1)}(r)$	1	1
$c^1_{(1,1)}(r)$	2	∞

(a)

r	0	1
$c^2_{(1,0)}(r)$	1	1
$c^2_{(-1,1)}(r)$	∞	∞
$c^2_{(0,1)}(r)$	1	1
$c^2_{(1,1)}(r)$	1	1

(b)

r	0	1	2	3
$\hat{c}_{(1,0)}(r)$	3	2	1	1
$\hat{c}_{(-1,1)}(r)$	2	5	4	3
$\hat{c}_{(0,1)}(r)$	1	1	1	1
$\hat{c}_{(1,1)}(r)$	2	2	1	1

(c)

Table 3: Displacement costs for the decomposition of $\mathcal{N}'_1$ and $\mathcal{N}'_2$, (a) and (b). In (c), minimal relative displacement costs of the alternate concatenation of (a) and (b) providing a translated octagonal distance with intermediate disks.

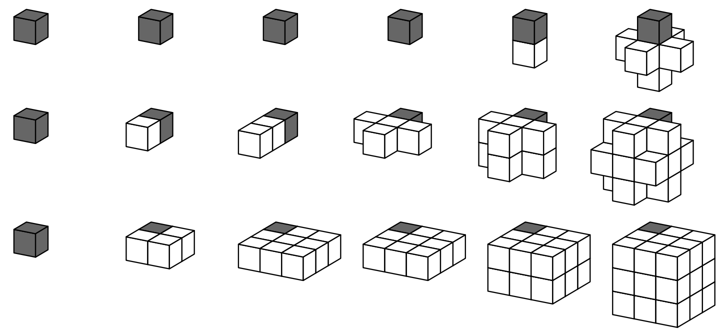


Figure 6: Decomposition of the neighborhoods used for the translated NS-distance transform in 3D. Top, middle and bottom rows correspond to neighborhoods $\mathcal{N}_1'$, $\mathcal{N}_2'$ and $\mathcal{N}_3'$. The decomposition steps are represented from left to right. The number of displacements vectors is reduced from 32 without the decomposition to only nine.

4.4.2. 3D case

The same method can be applied in 3D: from a decomposition of each of the neighborhoods, and a sequence B, an extended sequence is built for each of the vectors concerned. Figure 6 illustrates a possible decomposition of 3D neighborhoods that uses only nine vectors with the corresponding construction values shown in Table 4.

5. Conclusion

In this paper, a path-based pseudo-distance scheme where displacement costs vary both with the displacement vector and with the travelled distance was presented. This scheme is generic enough to describe neighborhood-sequence distances, weighted distances as well as generalized distances produced by Minkowski sums. It was shown that a set of displacement costs can be provided in a minimal form, where each displacement vector is assigned a non-decreasing sequence of costs, without altering the distance function. These non-decreasing sequences are directly applied in the distance transform algorithm to keep track of the costs of minimal paths from the background. An application to a translated neighborhood-sequence distance transform in a single scan was presented along with a method to recover the proper, centered, distance transform. Com-

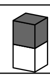
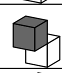
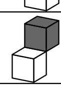
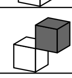
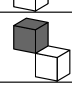
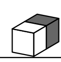
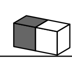
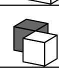
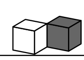
r		$\mathcal{N}_1$ 1	2	3	4	5	$\mathcal{N}_2$ 1	2	3	4	5	$\mathcal{N}_3$ 1	2	3	4	5
$(0,0,1)$					4	5				4	5				4	5
$(0,-1,1)$						4										
$(0,1,1)$						4										
$(-1,0,1)$						4					3					
$(1,0,1)$						4					3					
$(0,1,0)$							1	2				1	2			
$(1,0,0)$												1	2			
$(1,1,0)$									1			1	2			
$(-1,1,0)$									1							

Table 4: Construction values for the decomposition of the 3D neighborhoods. Only strictly positive values are shown.

```
Data: DT'_X: translated distance map of X
Result: DT_X: centered distance map of X
foreach p in DT' domain, in raster scan do
    if DT'(p) = 0 then
        DT(p) ← 0
    else
        foreach j do
            r ← max {1; DT'(p + t⃗_j)}
            // Get the minimal r ≥ DT'(p − t⃗_j) such that B(r) = j
            r ← j†_B(j_B(r)) + 1
            while r ≤ DT'(p) do
                DT(p − t⃗(r − 1)) ← r
                // Get the next r such that B(r) = j
                r ← j†_B(j_B(r) + 1) + 1
            end
        end
    end
end
```

Algorithm 3: Obtention of a regular (centered) DT from a translated DT′.

bined methods provide partial result with a minimal delay, before the input image is fully processed. Their efficiency can benefit all applications where neighborhood-sequence distances are used, particularly in pipelined processing architectures, or when the size of objects in the source image is limited. It was also shown that, by further decomposing the Minkowski sums involved in the neighborhood distance transform, the amount of displacement vector can be reduced. In 3D, instead of using a total of 26 displacement vectors, or even 32 as required by the direct application of the translated distance transform, a full transform can be computed with only nine vectors.

The pseudo-distance presented here is strongly linked to the properties of non-decreasing integer sequences studied by Lambek and Moser. First, the Lambek-Moser inverse connects the iterative construction of disks with the displacement costs propagated in the distance transform. Next, by allowing to iterate through values of integer sequences, it permits to compute the displacement costs on-the-fly. An implementation in C language is publicly available at http://www.irccyn.ec-nantes.fr/~normand/LUTBasedNSDistanceTransform.

References

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

[2] A. Rosenfeld, J. L. Pfaltz, Distances functions on digital pictures, Pattern Recognition 1 (1968) 33–61.

[3] G. Borgefors, Distance transformations in arbitrary dimensions, Computer Vision, Graphics, and Image Processing 27 (1984) 321–345.

[4] U. Montanari, A method for obtaining skeletons using a quasi-Euclidean distance, Journal of the ACM 15 (1968) 600–624.

[5] R. Strand, Weighted distances based on neighbourhood sequences, Pattern Recognition Letters 28 (2007) 2029–2036.

[6] J. Lambek, L. Moser, Inverse and complementary sequences of natural numbers, The American Mathematical Monthly 61 (1954) 454–458.

[7] A. Ostrowski, J. Hyslop, A. C. Aitken, Solutions to problem 3173, The American Mathematical Monthly 34 (1927) 159–160.

[8] S. Beatty, Problem 3173, The American Mathematical Monthly 33 (1926) 159.

[9] A. Hajdu, L. Hajdu, Approximating the Euclidean distance using non-periodic neighbourhood sequences, Discrete Mathematics 283 (2004) 101–111.

[10] É. Thiel, A. Montanvert, Chamfer masks: Discrete distance functions, geometrical properties and optimization, in: International Conference on Pattern Recognition, volume III, pp. 244–247.

[11] P.-E. Danielsson, Euclidean distance mapping, Computer Graphics, and Image Processing 14 (1980) 227–248.

[12] R. Strand, B. Nagy, C. Fouard, G. Borgefors, Generating distance maps with neighbourhood sequences, in: A. Kuba, L. G. Nyúl, K. Palágyi (Eds.), Discrete Geometry for Computer Imagery 2006, volume 4245 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Szeged, Hungary, 2006, pp. 295–307.

[13] N. Normand, Convex structuring element decomposition for single scan binary mathematical morphology, in: I. Nyström, G. Sanniti di Baja, S. Svensson (Eds.), Discrete Geometry for Computer Imagery 2003, volume 2886 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Naples, Italy, 2003, pp. 154–163.

[14] X. Wang, G. Bertrand, An algorithm for a generalized distance transformation based on Minkowski operations, in: International Conference on Pattern Recognition, volume 2, pp. 1164–1168.

[15] X. Wang, G. Bertrand, Some sequential algorithms for a generalized distance transformation based on Minkowski operations, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (1992) 1114–1121.

[16] E. Deutsch, Number of hex trees with n edges and having no nonroot nodes of outdegree 2, in: N. J. A. Sloane, S. Plouffe (Eds.), The On-Line Encyclopedia of Integer Sequences.