



HAL
open science

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

Nicolas Normand, Robin Strand, Pierre Evenou, Aurore Arlicot

► **To cite this version:**

Nicolas Normand, Robin Strand, Pierre Evenou, Aurore Arlicot. Minimal-Delay Distance Transform for Neighborhood-Sequence Distances in 2D and 3D. *Computer Vision and Image Understanding*, 2013, 117 (4), pp.409-417. 10.1016/j.cviu.2012.08.015 . hal-00765695

HAL Id: hal-00765695

<https://hal.science/hal-00765695v1>

Submitted on 16 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Nicolas Normand^{a,*}, Robin Strand^b, Pierre Evenou^a, Aurore Arlicot^a

^aLUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597,
Polytech Nantes, Rue Christian Pauc, La Chantrerie,
44306 Nantes Cedex 3, France

^bCentre for Image Analysis, Uppsala University, Sweden

^cCentre for Image Analysis, Uppsala University,
Box 337, SE-75105 Uppsala, Sweden

^dLUNAM 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)

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

03
04 Nicolas Normand^{a,*}, Robin Strand^b, Pierre Evenou^a, Aurore Arlicot^a

05 ^a*LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597,*
06 *Polytech Nantes, Rue Christian Pauc, La Chantrerie,*
07 *44306 Nantes Cedex 3, France*

08 ^b*Centre for Image Analysis, Uppsala University, Sweden*

09 **Abstract**

10
11 This paper presents a path-based distance where local displacement costs vary
12 both according to the displacement vector and with the travelled distance. The
13 corresponding distance transform algorithm is similar in its form to classical
14 propagation-based algorithms, but the more variable distance increments are
15 either stored in look-up-tables or computed on-the-fly. These distances and
16 distance transform extend neighborhood-sequence distances, chamfer distances
17 and generalized distances based on Minkowski sums. We introduce algorithms to
18 compute a translated version of a neighborhood sequence distance map both for
19 periodic and aperiodic sequences and a method to derive the centered distance
20 map. A decomposition of the grid neighbors, in \mathbb{Z}^2 and \mathbb{Z}^3 , allows to significantly
21 decrease the number of displacement vectors needed for the distance transform.
22 Overall, the distance transform can be computed with minimal delay, without
23 the need to wait for the whole input image before beginning to provide the result
24 image.

25
26 *Corresponding author

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

01 **1. Introduction**

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

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

21 **2. Preliminaries**

22 *2.1. Lambek-Moser inverse of a integer sequence [6].*

23 Let the function f define a non-decreasing sequence of integers $(f(1), f(2), \dots)$
24 For the sake of simplicity, we call f a sequence. The inverse sequence of f , de-
25 noted by f^\dagger , is a non-decreasing sequence of integers defined by:
26
27

28
$$f(m) < n \Leftrightarrow f^\dagger(n) \not\leq m. \tag{1}$$

29

30 $f^\dagger(n)$ can be seen as the count of values less than n in f , the last index of
31 a value less than n in f , the index that precedes the first value greater than or

01 equal to n :

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

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

08 An interesting property of a sequence f and its inverse f^\dagger is that, by adding
09 the rank of each term to these two sequences, we obtain two complementary
10 sequences $f(m) + m$ and $f^\dagger(n) + n$ [6], as shown in Table 1. This property
11 extends the results given by Ostrowski, Hyslop, and Aitken [7] about Beatty
12 sequences [8]. From [6], we deduce that the inverse of the sequence $f(m) =$
13 $\lfloor \tau m \rfloor$ with a scalar τ , is $f^\dagger(n) = \lceil \frac{n}{\tau} - 1 \rceil$ so $f(m) + m = \lfloor (1 + \tau)m \rfloor$ and
14 $f^\dagger(n) + n = \lceil (1 + \frac{1}{\tau})n - 1 \rceil$ are two complementary sequences. If τ is irrational,
15 these sequences are Beatty sequences and, for any positive n , $\lceil (1 + \frac{1}{\tau})n - 1 \rceil$ is
16 equal to $\lfloor (1 + \frac{1}{\tau})n \rfloor$ as given in [8].

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

23
24 **Proposition 1.** $f^\dagger(f(m) + 1) + 1$ is the rank of the smallest term greater than
25 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.

$$\begin{aligned}
f^\dagger(f(m) + 1) + 1 = m' &\Leftrightarrow \begin{cases} f^\dagger(f(m) + 1) < m' \\ f^\dagger(f(m) + 1) \geq m' - 1 \end{cases} \\
&\Leftrightarrow \begin{cases} f(m') \geq f(m) + 1 \\ f(m' - 1) < f(m) + 1 \end{cases} \\
&\Leftrightarrow f(m') > f(m) \text{ and } f(m' - 1) \leq f(m) .
\end{aligned}$$

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 $d : \mathbb{Z}^n \times \mathbb{Z}^n \rightarrow \mathbb{N}$ 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”.

01 **Definition 2 (Path-based distance).** Let $\mathcal{P}(p, q)$ be the set of paths from
 02 $p \in \mathbb{Z}^n$ to $q \in \mathbb{Z}^n$. $d : \mathbb{Z}^n \times \mathbb{Z}^n \rightarrow \mathbb{N}$ is a *path-based distance* if:

- 03 (i) $\forall(p, q), d(p, q) = \min \{ \mathcal{L}(P), P \in \mathcal{P}(p, q) \}$,
 04 (ii) d is a distance,

05 where $\mathcal{L}(P)$ is the length (or cost) of the path P . A path $P \in \mathcal{P}(p, q)$ is *minimal*
 06 if its length is equal to $d(p, q)$. It is usually not unique between p and q .
 07

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

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

11 where p_i stands for the i^{th} component of p .
 12

13 The k -neighborhood of p , denoted by $\mathcal{N}_k(p)$, is the set of k -neighbors of p and
 14 the k -neighborhood \mathcal{N}_k , in a translation-invariant context, is the set of vectors
 15 from any point p to its k -neighbors. In the 2D square grid, k -neighborhoods are
 16 defined as follows:
 17

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

18 \mathcal{N}_1 and \mathcal{N}_2 are often referred to as 4- and 8-neighborhoods because they contain
 19 respectively four and eight non-zero displacement vectors.
 20

21 For distance d_4 , a path is a sequence of points (p_0, \dots, p_n) where each pair
 22 of successive points (p_{i-1}, p_i) are 1-neighbors and the length of the path is the
 23 number of displacements, n . Whereas the 2-neighborhood is used for distance
 24 d_8 [1, 2]. Distances d_4 and d_8 have a high rotational dependency as noticed by
 25 [Rosenfeld and Pfaltz](#).
 26

27 A neighborhood sequence (NS) $B = (b(i))_{i>0}$ is a sequence where each $b(i)$
 28 denotes a neighborhood relation in \mathbb{Z}^n in the sense of Definition 3. If B is
 29 l -periodic, *i.e.* if for some finite, strictly positive $l \in \mathbb{Z}_+$, $b(i) = b(i + l)$ is
 30
 31

01 valid for all $i \in \mathbb{N}^*$, then we write $B = (\overline{b(1), b(2), \dots, b(l)})$. A B -path is a
 02 sequence of points (p_0, \dots, p_n) where each pair of successive points (p_{i-1}, p_i)
 03 are $B(i)$ -neighbors. Given the NS B , the NS-distance d_B is the path-based
 04 distance whose paths are only B -paths and the length of a path is the number
 05 of displacements. d_4 and d_8 can be seen as NS-distances $d_{(\overline{1})}$ and $d_{(\overline{2})}$ with
 06 1-periodic sequences $B = (\overline{1})$ and $B = (\overline{2})$, but the simplest NS-distance that
 07 combines both neighborhoods is the octagonal distance $d_{(\overline{1,2})}$ with sequence
 08 $B = (\overline{1,2})$ [2].

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

$$11 \quad \mathbf{j}_B(r) = \text{Card}(\{i : b(i) = j, 1 \leq i \leq r\}) .$$

12
 13 A different approach is used for chamfer, or *weighted*, distances where each
 14 displacement vector \vec{v}_k in a neighborhood \mathcal{N} is associated to the *weight* (or *local*
 15 *cost*) w_k [3, 4, 10]. A chamfer mask \mathcal{M} is a central symmetric set of *weightings*
 16 $(\vec{v}_k; w_k)$ with at least a base of \mathbb{Z}^n :

$$17 \quad \mathcal{M} = \{(\vec{v}_k; w_k) \in \mathbb{Z}^n \times \mathbb{N}^*\}_{1 \leq k \leq m} .$$

18
 19 The length of the path (p_0, \dots, p_n) is the sum of the displacements costs:

$$20 \quad \mathcal{L}(p_0, \dots, p_n) = \sum_{i=1}^n w(i), \text{ where } (\overrightarrow{p_{i-1}p_i}; w(i)) \in \mathcal{M} .$$

21
 22 **Definition 4 (Ball).** The disk $D(p, r)$ of center p and radius r and the sym-
 23 metrical disk $\check{D}(p, r)$ are the sets:

$$24 \quad \begin{aligned} 25 \quad D(p, r) &= \{q : d(p, q) \leq r\}, \\ 26 \quad \check{D}(p, r) &= \{q : d(q, p) \leq r\}. \end{aligned} \tag{4}$$

27
 28 By definition, any disk of negative radius is empty and the disk of radius 0 only
 29 contains its center ($D(p, 0) = \{p\}$).

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

of X :

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

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

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

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 $(p_0, \dots, p_n, p_n + \vec{v})$ derives from the cost of the path (p_0, \dots, p_n) :

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

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

$$08 \quad \mathcal{L}(p_0, \dots, p_n) = r \Rightarrow \forall \vec{v}, \mathcal{L}(p_0, \dots, p_n, p_n + \vec{v}) = r + c_{\vec{v}},$$

09 where

$$10 \quad c_{\vec{v}} = \begin{cases} 1 & \text{if } \vec{v} \in \mathcal{N} \\ \infty & \text{otherwise} \end{cases}.$$

11 For a NS-distance characterized by the sequence B :

$$12 \quad \mathcal{L}(p_0, \dots, p_n) = r \Rightarrow \forall \vec{v}, \mathcal{L}(p_0, \dots, p_n, p_n + \vec{v}) = r + c_{\vec{v}}^B(r), \quad (8)$$

13 where the displacement cost $c_{\vec{v}}^B(r)$ is 1 for a displacement vector in the neigh-
 14 borhood $B(r + 1)$ and infinite otherwise:

$$15 \quad c_{\vec{v}}^B(r) = \begin{cases} 1 & \text{if } \vec{v} \in \mathcal{N}_{B(r+1)} \\ \infty & \text{otherwise} \end{cases}.$$

16 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 dis-
 17 tance increment only depends on the displacement vector, but not on the dis-
 18 tance already travelled:

$$19 \quad \mathcal{L}(p_0, \dots, p_n) = r \Rightarrow \forall \vec{v}, \mathcal{L}(p_0, \dots, p_n, p_n + \vec{v}) = r + c_{\vec{v}}, \quad (10)$$

$$20 \quad c_{\vec{v}} = \begin{cases} w & \text{if } (\vec{v}; w) \in \mathcal{M} \\ \infty & \text{otherwise} \end{cases}.$$

21 Briefly, the displacement cost for a vector \vec{v} and the travelled distance r , is
 22 1 or ∞ , independently of r for simple distances, is equal to 1 or ∞ whether
 23 \vec{v} belongs or not to $\mathcal{N}_{B(r)}$ for a NS-distance, is in $\mathbb{N}^* \cup \{\infty\}$ according to the
 24 chamfer mask and independently of r for a weighted distance.

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

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

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 $\vec{0}$ and the positive displacement costs $c_{\vec{v}}$ (with $c_{\vec{0}}(r) = 1$ and $c_{\vec{v} \notin \mathcal{N}}(r) = \infty$). The total cost of the path $P = (p_0, p_1, \dots, p_n)$ is:

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

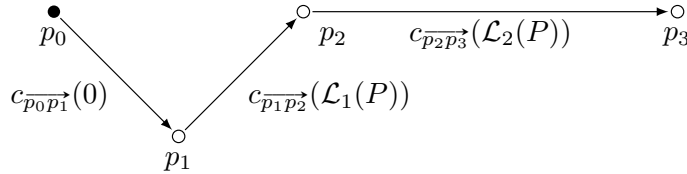


Figure 1: Total cost of a path $P = (p_0, p_1, p_2)$. Costs of displacements $\overrightarrow{p_0 p_1}$, $\overrightarrow{p_1 p_2}$ and $\overrightarrow{p_2 p_3}$ depend on the partial costs $\mathcal{L}_0(P) = 0$, $\mathcal{L}_1(P) = c_{\overrightarrow{p_0 p_1}}(0) + 0$ and $\mathcal{L}_2(P) = c_{\overrightarrow{p_1 p_2}}(\mathcal{L}_1(P)) + \mathcal{L}_1(P)$. The total cost of P is $c_{\overrightarrow{p_2 p_3}}(\mathcal{L}_2(P)) + \mathcal{L}_2(P)$.

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

03 Figure 1 illustrates the individual displacement costs in a path, each one de-
 04 pending on the partial length of the path.

05
 06 **Definition 8 (Absolute and relative costs of displacement).** We use the
 07 notation $C_{\vec{v}_k}(r) = r + c_{\vec{v}_k}(r)$. $c_{\vec{v}_k}(r)$ is the *relative cost* of the displacement \vec{v}_k
 08 when the distance travelled so forth is r . $C_{\vec{v}_k}(r)$ represents the partial cost of
 09 the path after this displacement (the *absolute cost* of this displacement):

$$10 \quad \mathcal{L}_{i+1}(P) = \mathcal{L}_i(P) + c_{\vec{v}_k}(r) = C_{\vec{v}_k}(r) . \quad (14)$$

11
 12 **Definition 9 (Pseudo-distance).** The *pseudo-distance* induced by $(\{\vec{v}_k\}, c_{\vec{v}_k})$
 13 is defined by:

$$14 \quad d(p, q) = 0 \Leftrightarrow p = q$$

$$15 \quad d(p, q) = \min_{P \in \mathcal{P}(p, q)} \{ \mathcal{L}(P) \} .$$

16
 17 **Definition 10 (Minimal cost of displacement).** We call *minimal relative*
 18 (resp. *absolute*) *cost of displacement*, denoted by \hat{c} (resp. \hat{C}), the quantity
 19 $\hat{c}_{\vec{v}}(r) = \min \{ c_{\vec{v}}(s) + s - r, \forall s \geq r \}$ (resp. $\hat{C}_{\vec{v}}(r) = \min \{ C_{\vec{v}}(s), \forall s \geq r \}$).

20
 21 **Proposition 2 (Preservation of cost order by concatenation).**

22 *Appending the same displacement to existing paths preserves the relation order*
 23 *of their costs. Let $P = (p_1, \dots, p_{n_P})$ and $Q = (q_1, \dots, q_{n_Q})$ be two paths*
 24 *with costs $\mathcal{L}(P)$ and $\mathcal{L}(Q)$, \vec{v} a vector and $P' = (p_1, \dots, p_{n_P}, p_{n_P} + \vec{v})$, $Q' =$*
 25 *$(q_1, \dots, q_{n_Q}, q_{n_Q} + \vec{v})$ the extended paths with costs $\mathcal{L}(P')$ and $\mathcal{L}(Q')$ measured*
 26 *with minimal displacement costs. Then:*

$$27 \quad \mathcal{L}(P) \leq \mathcal{L}(Q) \Rightarrow \mathcal{L}(P') \leq \mathcal{L}(Q') . \quad (15)$$

28
 29
 30 **PROOF.** From (14), $\mathcal{L}(P') = \hat{C}_{\vec{v}}(\mathcal{L}(P))$ and $\mathcal{L}(Q') = \hat{C}_{\vec{v}}(\mathcal{L}(Q))$. By definition
 31 of $\hat{C}_{\vec{v}}$, $s \leq r \Rightarrow \hat{C}_{\vec{v}}(s) \leq \hat{C}_{\vec{v}}(r)$, which gives (15).

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

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

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

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 $\mathcal{L}'_{M_{i-1}}(P') = \mathcal{L}_{i-1}(P) + \hat{c}_{\vec{p}_{i-1}p_i}(\mathcal{L}_{i-1}(P))$, and after $m_i - 1$ repeats of p_i , equals $\mathcal{L}'_{M_{i-1}+m_i-1}(P') = \mathcal{L}'_{M_{i-1}}(P') = \mathcal{L}_{i-1}(P) + \hat{c}_{\vec{p}_{i-1}p_i}(\mathcal{L}_{i-1}(P)) + m_i - 1 = \mathcal{L}_{i-1}(P) + c_{\vec{p}_{i-1}p_i}(\mathcal{L}_{i-1}(P)) = \mathcal{L}_i(P)$ 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 $m'_i - 1 + c_{\vec{p}_i p_{i+1}}(\mathcal{L}'_i(P')) + m'_i - 1 = \hat{c}_{\vec{p}_i p_{i+1}}(\mathcal{L}'_i(P'))$. By definition of \hat{c} , $\forall r, \exists s : \hat{c}_{\vec{v}}(r) = c_{\vec{v}}(s) + s - r$, 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 occurrence 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 $\mathcal{L}_{M'_i+m'_i-1}(P) =$

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

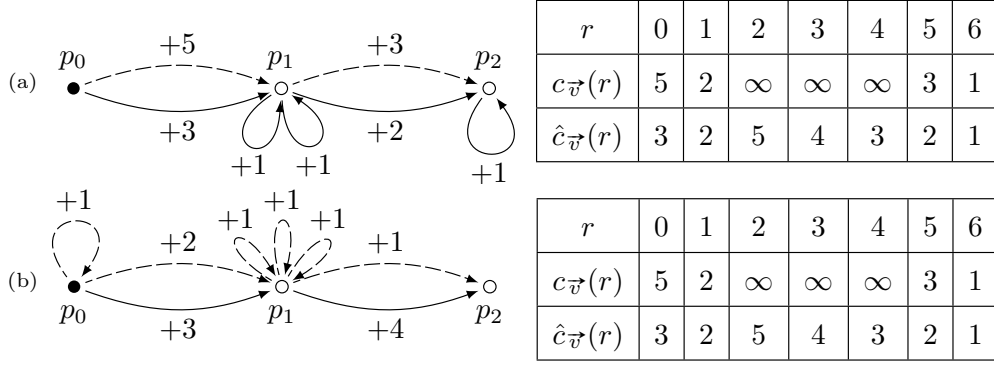


Figure 2: (a) Given $P = (p_0, p_1, p_2)$, shown with dashed lines, has a total cost $\mathcal{L}(P) = 8$ measured with displacement costs $c_{\vec{v}}$. $P' = (p_0, p_1, p_1, p_1, p_2, p_2)$, solid lines, is built in such a way that its cost $\mathcal{L}'(P')$ measured with minimal displacement costs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

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

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 $r \geq s$.

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

$$04 \quad q \in D(p, r) \Leftrightarrow \exists P \in \mathcal{P}(p, q), \mathcal{L}(P) = r. \quad (17)$$

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

$$06 \quad \forall r > 0, D(p, r) = \bigcup_{\vec{v} \in \mathcal{N}} \{q : \exists P \in \mathcal{P}(p, q - \vec{v}) \text{ and } C_{\vec{v}}(\mathcal{L}(P)) = r\}$$

$$07 \quad = \bigcup_{\substack{\vec{v} \in \mathcal{N} \\ s : C_{\vec{v}}(s) = r}} D(p + \vec{v}, s). \quad (18)$$

08 3.2. Iterative Construction of Shapes

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

$$10 \quad D(O, r) = \begin{cases} \emptyset & \text{if } r < 0 \\ \{O\} & \text{if } r = 0 \\ \bigcup_{\vec{v} \in \mathcal{N}} D(O + \vec{v}, S_{\vec{v}}(r) - 1) & \text{otherwise} \end{cases} \quad (19)$$

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

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

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

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

01 3.3. Sequences of Minkowski Sums

02 **Proposition 5.** *If all finite values of absolute costs for radii less than r_1 do*
 03 *not exceed r_1 ($\forall 0 \leq r < r_1, C(r) \leq r_1$ or $C(r) = \infty$) then all balls of ra-*
 04 *dius r_2 greater than or equal to r_1 are the Minkowski sum of $D(p, r_1)$ with the*
 05 *disk $D'(O, r_2 - r_1)$ produced by the displacement costs $c'_{\vec{v}}(r) = c_{\vec{v}}(r + r_1)$ (i.e.*
 06 $C'_{\vec{v}}(r) = C_{\vec{v}}(r + r_1) - r_1$):

07
$$\forall r < r_1, C(r) \leq r_1 \Rightarrow \forall r_2 \geq r_1, D(p, r_2) = D(p, r_1) \oplus D'(O, r_2 - r_1) . \quad (21)$$

08
 09
 10
 11 PROOF. (21) holds for $r_2 = r_1$: $D(p, r_1) = D(p, r_1) \oplus \{O\}$. Suppose (21) holds
 12 in the interval $[r_1, r_2]$. Only values of s in the interval $[r_1, r_2]$ can be such that
 13 $C_{\vec{v}}(s) = r_2 + 1$ and (21) applies to $D(p + \vec{v}, s)$, so (18) can be written as:

14
$$D(p, r_2 + 1) = \bigcup_{\substack{\vec{v} \in \mathcal{N} \\ s : C_{\vec{v}}(s) = r_2 + 1}} D(p + \vec{v}, r_1) \oplus D'(O, s - r_1)$$

15

16
$$= D(p + \vec{v}, r_1) \oplus \bigcup_{\substack{\vec{v} \in \mathcal{N} \\ s : C_{\vec{v}}(s) = r_2 + 1}} D'(O, s - r_1)$$

17

18
$$= D(p + \vec{v}, r_1) \oplus \bigcup_{\substack{\vec{v} \in \mathcal{N} \\ s : C'_{\vec{v}}(s) = r_2 - r_1 + 1}} D'(O, s)$$

19

20

21
$$= D(p + \vec{v}, r_1) \oplus D'(O, r_2 - r_1 + 1) .$$

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

23
 24 Proposition 5 provides a sufficient (but not necessary) condition that enables
 25 to build sequences of Minkowski sums. In particular, when relative displacement
 26 costs are either 1 or ∞ as in (9), then Proposition 5 applies to all positive radius
 27 values so that the NS-distance balls are produced by a sequence of Minkowski
 28 sums with the sequence of neighborhoods, as it is well known. In the follow-
 29 ing Section 4.4, Proposition 5 will be used along with (19) and (20) to build
 30 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 $C_{\vec{v}}$ is such that:*

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

where $\hat{C}_{\vec{v}}$ represents the minimal absolute displacement costs corresponding to $C_{\vec{v}}$ (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 $(\{\vec{v}_k\}, C_{\vec{v}_k})$ and $(\{\vec{v}_k\}, \hat{C}_{\vec{v}_k})$ are equal so we consider the latter cost increments for which Proposition 2 holds. According to Proposition 2, if $P = (q = p_0, \dots, p_n = p - \vec{v})$ is a minimal path from q to $p - \vec{v}$ then $P' = (q = p_0, \dots, p_n, p + \vec{v})$ has a minimal cost — among paths from q to p with second last point $p - \vec{v}$ — equal to $\hat{C}_{\vec{v}}(\mathcal{L}(P))$. So $\hat{C}_{\vec{v}}(\text{DT}_X(p - \vec{v}))$ is the shortest distance from a point $q \notin X$ to p via $p - \vec{v}$. Since all paths which

01 last displacement \vec{v} does not belong to \mathcal{N} have an infinite cost and can not be
 02 minimal, (22) holds.

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

$$08 \quad \hat{C}_{\vec{v}}(r) = \min \{s : s > r \text{ and } \vec{v} \in \mathcal{N}_{B(s)}\}. \quad (23)$$

09 Let $\chi_{\vec{v}}(r)$ denote the characteristic function of the set $\mathcal{N}_{B(r)}$ (i.e. $\chi_{\vec{v}}(r) =$
 10 1 if $\vec{v} \in \mathcal{N}_{B(r)}$; 0 otherwise) and $\chi_{\vec{v}}^{\Sigma}(r)$ its cumulative sum ($\chi_{\vec{v}}^{\Sigma}(r) = \sum_{s \leq r} \chi_{\vec{v}}(s)$).
 11 Then according to Proposition 1:

$$12 \quad \hat{C}_{\vec{v}}(r) = [\chi_{\vec{v}}^{\Sigma}]^{\dagger}(\chi_{\vec{v}}^{\Sigma}(r) + 1) + 1. \quad (24)$$

13 Algorithm 1 produces a generalized DT using any sequence of neighborhoods
 14 (\mathcal{N} represents their union) in forward scan condition, using displacement costs
 15 given by (24). A similar algorithm was already presented for the decomposition
 16 of convex structuring polygons [13].
 17
 18

19 4.2. Translated NS-distance transform

20 The sequence of balls for a NS-distance induced by a sequence B is produced
 21 by iterative Minkowski sums of neighborhoods:
 22

$$23 \quad D(p, 0) = \{p\}, \quad D(p, r) = D(p, r - 1) \oplus \mathcal{N}_{B(r)}.$$

24 For each neighborhood \mathcal{N}_j , we apply a translation vector \vec{t}_j such that the trans-
 25 lated neighborhood $\mathcal{N}'_j = \mathcal{N}_j \oplus \{\vec{t}_j\}$ is in forward scan condition. In a transla-
 26 tion preserved scan order, \vec{t}_j translates the first visited point in \mathcal{N}_j to the origin.
 27 Assuming a n D standard raster scan order:
 28

$$29 \quad \vec{t}_j = (\underbrace{0, \dots, 0}_{n-j}, \underbrace{1, \dots, 1}_j). \quad (25)$$

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

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  
         $DT_X(p) \leftarrow 0$   
    else  
         $l \leftarrow \infty$   
        foreach  $\vec{v}$  in  $\mathcal{N}$  do  
             $l \leftarrow \min \{l, \hat{C}_{\vec{v}}(DT_X(p - \vec{v}))\}$   
        end  
         $DT_X(p) \leftarrow l$   
    end  
end
```

Algorithm 1: Single scan generalized distance transform

01 The translated neighborhoods \mathcal{N}'_1 and \mathcal{N}'_2 obtained with $\vec{t}_1 = (0, 1)$ and
 02 $\vec{t}_2 = (1, 1)$ are depicted in Fig. 3a and Fig. 3b. Characteristic functions for
 03 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
 04 first differences of $\mathbf{1}_B$, $\mathbf{2}_B$ and the constant value 1 resulting in the following
 05 minimal displacement costs:

$$06 \hat{C}_{\vec{v}}(r) = \begin{cases} \hat{C}_{\vec{v}}^1(r) = \mathbf{1}_B^\dagger(\mathbf{1}_B(r) + 1) + 1 & \text{if } \vec{v} \in \mathcal{N}'_1 \text{ and } \vec{v} \notin \mathcal{N}'_2 \\ \hat{C}_{\vec{v}}^2(r) = \mathbf{2}_B^\dagger(\mathbf{2}_B(r) + 1) + 1 & \text{if } \vec{v} \notin \mathcal{N}'_1 \text{ and } \vec{v} \in \mathcal{N}'_2 \\ \hat{C}_{\vec{v}}^{12}(r) = r + 1 & \text{if } \vec{v} \in \mathcal{N}'_1 \text{ and } \vec{v} \in \mathcal{N}'_2 \end{cases} .$$

10 *Periodic sequence.* When B is a periodic sequence, minimal relative costs $\hat{c}_{\vec{v}}$
 11 are also periodic sequences. Take the periodic sequence of the octagonal distance
 12 $B = (\overline{1, 2})$, then $\mathbf{1}_B(r)_{r \geq 0} = (0, 1, 1, 2, \dots)$, $\mathbf{1}_B^\dagger(r)_{r > 0} = (0, 2, 4, \dots)$,
 13 $\hat{C}_{\vec{v}}^1(r)_{r \geq 0} = (1, 3, 3, 5, \dots)$ and $\hat{c}_{\vec{v}}^1(r)_{r \geq 0} = (1, 2, 1, 2, \dots)$. $\mathbf{1}_B(r)_{r > 0}$ is the cu-
 14 mulative sum of the 2-periodic sequence $(\overline{1, 0})$, whereas $\mathbf{1}_B^\dagger(r)_{r > 0}$ is the cu-
 15 mulative sum with offset -2 of the 1-periodic sequence $(\overline{2})$ as given by Al-
 16 gorithm 2. Similarly, $\mathbf{2}_B(r)_{r \geq 0} = (0, 0, 1, 1, 2, \dots)$, $\mathbf{2}_B^\dagger(r)_{r > 0} = (1, 3, \dots)$,
 17 $\hat{C}_{\vec{v}}^2(r)_{r \geq 0} = (2, 2, 4, \dots)$ and $\hat{c}_{\vec{v}}^2(r)_{r \geq 0} = (2, 1, 2, 1, \dots)$.

18 *Rate-based sequence.* Suppose now that the sequence of neighborhoods is de-
 19 fined as the first difference of a Beatty sequence (as in [9]): $B(r) = \lfloor \tau r \rfloor -$
 20 $\lfloor \tau(r-1) \rfloor$, with $\tau \in [1, 2]$ so that $B(r) \in \{1, 2\}$. $\mathbf{1}_B$ and $\mathbf{2}_B$ are respectively the
 21 cumulative sums of $2 - B(r) = \lceil (2 - \tau)r \rceil - \lceil (2 - \tau)(r-1) \rceil$ and $B(r) - 1 =$
 22 $\lfloor (\tau - 1)r \rfloor - \lfloor (\tau - 1)(r-1) \rfloor$. Then $\mathbf{1}_B(r) = \lceil (2 - \tau)r \rceil$, $\mathbf{2}_B(r) = \lfloor (\tau - 1)r \rfloor$,
 23 $\mathbf{1}_B^\dagger(r) = \lfloor \frac{r-1}{2-\tau} \rfloor$ and $\mathbf{2}_B^\dagger(r) = \lceil \frac{r}{\tau-1} - 1 \rceil$. This allows to compute $\hat{C}_{\vec{v}}^1$ and $\hat{C}_{\vec{v}}^2$
 24 on the fly. For the octagonal distance, $\tau = \frac{3}{2}$, $\mathbf{1}_B(r) = \lceil \frac{r}{2} \rceil$, $\mathbf{2}_B(r) = \lfloor \frac{r}{2} \rfloor$,
 25 $\mathbf{1}_B^\dagger(r) = 2r - 2$ and $\mathbf{2}_B^\dagger(r) = 2r - 1$.

26 An exemple result of Algorithm 1 for the translated octagonal distance (with
 27 displacement costs obtained either from sequence $B = (1, 2)$ either from $\tau = \frac{3}{2}$)
 28 is shown in Fig. 4b.
 29
 30
 31

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

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

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

$g_p \leftarrow \sum_{i=1}^{f_p} f(i);$

$n = f_o + 1;$

$m_0 \leftarrow f_p;$

for $m \leftarrow 1$ **to** f_p **do**

if $f(m) \neq 0$ **then** $m_0 \leftarrow \min \{m_0; m\};$

$g(n) \leftarrow g(n) + 1;$

$n \leftarrow n + f(m);$

end

$g_o \leftarrow m_0 - 1 - g(f_o + 1);$

for $m \leftarrow 1$ **to** $-f_o$ **do** $g_o \leftarrow g_o + g(m + f_o);$ // Adjust g_o if $f_o < 0$

for $m \leftarrow 0$ **down to** $-f_o$ **do** $g_o \leftarrow g_o - g(m + f_o);$ // ...or if $f_o \geq 0$

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

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

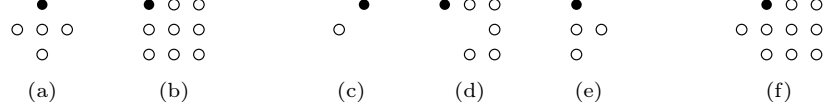


Figure 3: Neighborhoods used for the translated NS-distance transform. (a), and (b) are respectively the type 1 and 2 translated neighborhoods, \mathcal{N}'_1 and \mathcal{N}'_2 . (c) and (d) and (e) are respectively $\mathcal{N}'_1 \setminus \mathcal{N}'_2$, $\mathcal{N}'_2 \setminus \mathcal{N}'_1$ 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.

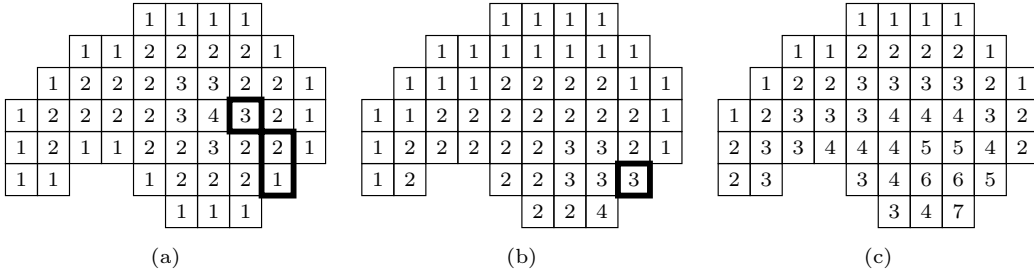


Figure 4: (a) Octagonal DT of a binary image. (b) Translated octagonal DT. Outlined centers of disks (a) are translated to the same location, outlined with value 3 (b). (c) Translated octagonal DT with intermediate disks.

4.3. Symmetric DT from asymmetric DT

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

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

In particular, for the 2D case:

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

01 DT'_X has equivalence with values of DT_X :

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

06 Consequently:

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

10 Knowing $DT'_X(p)$ and $DT'_X(p + \vec{t})$, we can deduce the values of $DT_X(p -$
11 $\vec{t}(r-1))$ for all values of r between $DT'_X(p + \vec{t})$ and $DT'_X(p)$ for which $\vec{t}(r) =$
12 $\vec{t}(r-1) + \vec{t}$, i.e. $\vec{t} = \overrightarrow{t_{B(r)}}$. Algorithm 3 recovers the values r of the centered
13 DT by selecting all r in the interval $[DT'_X(p + \vec{t}_j), DT'_X(p)]$ such that $B(r) = j$.
14 Iterating through values r with $B(r) = j$ is achieved using Proposition 1. Values
15 of DT'_X become available before the whole image is computed. For instance, in
16 a standard raster scan, as soon as line y is processed, all lines of DT'_X above
17 $y - r_{\max}$ are fully recovered (where r_{\max} denotes the maximal value of DT' in
18 that line).

20 4.4. Translated NS-DT with Intermediate disks.

21 The full set of neighbors used by the previous algorithm is the union of all
22 translated neighbors (excluding the new origin of the neighborhood). In 2D,
23 this set contains nine vectors, compared to the eight neighbors needed by the
24 classical algorithms. This count increases rapidly with the dimension: 32, 107,
25 350 respectively for 3D, 4D and 5D (apparently following, with a constant offset
26 -1 , sequence A126184 in Sloane's On-Line Encyclopedia of Integer Sequences
27 [16]). In this section we will show how we can drastically reduce the count of
28 required vectors, by further decomposing neighborhoods using set unions and
29 translation.

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

01 sets of points \mathcal{N}_j with $j \in [1, n]$. For the sake of simplicity, we assume that
 02 all sub-sequences have the same length l . We build the expanded sequence of
 03 displacement costs for neighbor \vec{v} , $c_{\vec{v}}$, by concatenation of the sub-sequences
 04 $c_{\vec{v}}^1 \dots c_{\vec{v}}^n$ selected by the master sequence B :

$$05 \quad \forall \vec{v}, \forall r > 0, c_{\vec{v}}(r) = c_{\vec{v}}^{B(\lfloor \frac{r}{l} \rfloor + 1)}(\langle r \rangle_l) .$$

06
 07 By Proposition 5, the balls generated by these displacement costs are such that:

$$08 \quad \forall k > 0, D(O, kl) = D(O, (k-1)l) \oplus \mathcal{N}_{B(k)} .$$

09
 10 The i^{th} sub-sequence occupies the radii $(i-1)l$ to $il-1$ in the expanded
 11 sequence and conversely, radius r corresponds to sub-sequence with index $i =$
 12 $\lfloor r/l \rfloor + 1$ starting at radius $r - \langle r \rangle_l$. According to Proposition 6, the distance
 13 transform can be computed by propagation of the minimal absolute displace-
 14 ment costs $\hat{C}_{\vec{v}}$. Minimal displacement costs are either equal to costs in the
 15 corresponding sub-sequences when these costs are finite, or deduced from the
 16 first value of a subsequent sub-sequence. The quantity $\hat{C}_{\vec{v}}(r)$ corresponds either
 17 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
 18 first value in next sub-sequence $\hat{c}_{\vec{v}}^j$ where \vec{v} is used: $\hat{c}_{\vec{v}}^j(0) + l(\mathbf{j}_B^\dagger(\mathbf{j}_B(i) + 1))$,
 19 whichever is minimal according to Definition 10.

$$20 \quad \hat{C}_{\vec{v}}(r) = \min \left\{ \hat{c}_{\vec{v}}^{B(i)}(\langle r \rangle_l) + r; \right. \quad (29a)$$

$$21 \quad \left. \hat{c}_{\vec{v}}^j(0) + l\mathbf{j}_B^\dagger(\mathbf{j}_B(i) + 1), j \in [1, n] \right\} , \quad (29b)$$

22
 23 where $i = \lfloor r/l \rfloor + 1$.

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

$$27 \quad \forall j, \hat{c}_{\vec{v}}^j(0) \neq \infty \Rightarrow \hat{C}_{\vec{v}}(r) = \min \left\{ \hat{c}_{\vec{v}}^{B(i)}(\langle r \rangle_l) + r; c_{\vec{v}}^{B(i+1)}(0) + li, j \in [1, n] \right\} . \quad (30)$$

28
 29 A further simplification holds when the displacement costs are always finite for
 30 a given vector \vec{v} , then we can omit (29b) in (29):

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

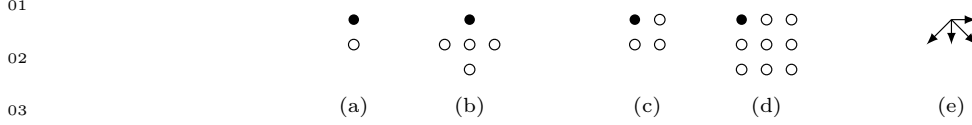


Figure 5: Decomposition of the neighborhoods used for the translated NS-distance transform. (b), and (d) are respectively the type 1 and 2 translated neighborhoods, \mathcal{N}'_1 and \mathcal{N}'_2 . (a) and (c) are intermediate disks. Only four displacement vectors (e) are needed instead of nine without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

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

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

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

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

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

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

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

(a) (b) (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.

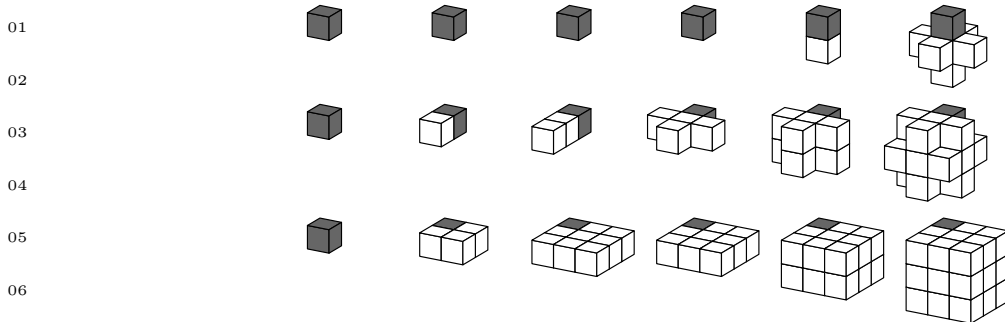


Figure 6: Decomposition of the neighborhoods used for the translated NS-distance transform in 3D. Top, middle and bottom rows correspond to neighborhoods \mathcal{N}'_1 , \mathcal{N}'_2 and \mathcal{N}'_3 . The decomposition steps are represented from left to right. The number of displacements vectors 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-

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31











r		\mathcal{N}_1					\mathcal{N}_2					\mathcal{N}_3				
		1	2	3	4	5	1	2	3	4	5	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.

```

01   Data:  $DT'_X$ : translated distance map of  $X$ 
02   Result:  $DT_X$ : centered distance map of  $X$ 
03   foreach  $p$  in  $DT'$  domain, in raster scan do
04       if  $DT'(p) = 0$  then
05            $DT(p) \leftarrow 0$ 
06       else
07           foreach  $j$  do
08                $r \leftarrow \max \{1; DT'(p + \vec{t}_j)\}$ 
09               // Get the minimal  $r \geq DT'(p - \vec{t}_j)$  such that  $B(r) = j$ 
10                $r \leftarrow \mathbf{j}_B^\dagger(\mathbf{j}_B(r)) + 1$ 
11               while  $r \leq DT'(p)$  do
12                    $DT(p - \vec{t}(r - 1)) \leftarrow r$ 
13                   // Get the next  $r$  such that  $B(r) = j$ 
14                    $r \leftarrow \mathbf{j}_B^\dagger(\mathbf{j}_B(r) + 1) + 1$ 
15               end
16           end
17       end
18   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

01 non-decreasing integer sequences studied by Lambek and Moser. First, the
02 Lambek-Moser inverse connects the iterative construction of disks with the dis-
03 placement costs propagated in the distance transform. Next, by allowing to iter-
04 ate through values of integer sequences, it permits to compute the displacement
05 costs on-the-fly. An implementation in C language is publicly available at <http://www.irccyn.ec-nantes.fr/~normand/LUTBasedNSDistanceTransform>.
06

07 **References**

- 09 [1] A. Rosenfeld, J. L. Pfaltz, Sequential operations in digital picture process-
10 ing, *Journal of the ACM* 13 (1966) 471–494.
11
- 12 [2] A. Rosenfeld, J. L. Pfaltz, Distances functions on digital pictures, *Pattern*
13 *Recognition* 1 (1968) 33–61.
14
- 15 [3] G. Borgefors, Distance transformations in arbitrary dimensions, *Computer*
16 *Vision, Graphics, and Image Processing* 27 (1984) 321–345.
17
- 18 [4] U. Montanari, A method for obtaining skeletons using a quasi-Euclidean
19 distance, *Journal of the ACM* 15 (1968) 600–624.
20
- 21 [5] R. Strand, Weighted distances based on neighbourhood sequences, *Pattern*
22 *Recognition Letters* 28 (2007) 2029–2036.
23
- 24 [6] J. Lambek, L. Moser, Inverse and complementary sequences of natural
25 numbers, *The American Mathematical Monthly* 61 (1954) 454–458.
26
- 27 [7] A. Ostrowski, J. Hyslop, A. C. Aitken, Solutions to problem 3173, *The*
28 *American Mathematical Monthly* 34 (1927) 159–160.
29
- 30 [8] S. Beatty, Problem 3173, *The American Mathematical Monthly* 33 (1926)
31 159.
- [9] A. Hajdu, L. Hajdu, Approximating the Euclidean distance using non-
periodic neighbourhood sequences, *Discrete Mathematics* 283 (2004) 101–
111.

- 01 [10] É. Thiel, A. Montanvert, Chamfer masks: Discrete distance functions,
02 geometrical properties and optimization, in: International Conference on
03 Pattern Recognition, volume III, pp. 244–247.
- 04 [11] P.-E. Danielsson, Euclidean distance mapping, *Computer Graphics, and*
05 *Image Processing* 14 (1980) 227–248.
- 06 [12] R. Strand, B. Nagy, C. Fouard, G. Borgefors, Generating distance maps
07 with neighbourhood sequences, in: A. Kuba, L. G. Nyúl, K. Palágyi (Eds.),
08 *Discrete Geometry for Computer Imagery 2006*, volume 4245 of *Lecture*
09 *Notes in Computer Science*, Springer Berlin / Heidelberg, Szeged, Hungary,
10 2006, pp. 295–307.
- 11 [13] N. Normand, Convex structuring element decomposition for single scan
12 binary mathematical morphology, in: I. Nyström, G. Sanniti di Baja,
13 S. Svensson (Eds.), *Discrete Geometry for Computer Imagery 2003*, volume
14 2886 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg,
15 Naples, Italy, 2003, pp. 154–163.
- 16 [14] X. Wang, G. Bertrand, An algorithm for a generalized distance transfor-
17 mation based on Minkowski operations, in: International Conference on
18 Pattern Recognition, volume 2, pp. 1164–1168.
- 19 [15] X. Wang, G. Bertrand, Some sequential algorithms for a generalized dis-
20 tance transformation based on Minkowski operations, *IEEE Transactions*
21 *on Pattern Analysis and Machine Intelligence* 14 (1992) 1114–1121.
- 22 [16] E. Deutsch, Number of hex trees with n edges and having no nonroot
23 nodes of outdegree 2, in: N. J. A. Sloane, S. Plouffe (Eds.), *The On-Line*
24 *Encyclopedia of Integer Sequences*.
- 25
26
27
28
29
30
31