



HAL
open science

Distance Transform Computation for Digital Distance Functions

Robin Strand, Nicolas Normand

► **To cite this version:**

Robin Strand, Nicolas Normand. Distance Transform Computation for Digital Distance Functions. Theoretical Computer Science, 2012, 448, pp.80-93. 10.1016/j.tcs.2012.05.010 . hal-00717825

HAL Id: hal-00717825

<https://hal.science/hal-00717825>

Submitted on 14 Jul 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.

Distance Transform Computation for Digital Distance Functions

Robin Strand^a, Nicolas Normand^b

^a*Centre for Image Analysis, Uppsala University,
Box 337, SE-75105 Uppsala, Sweden*

^b*LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597,
Polytech Nantes, La Chantrerie, 44306 Nantes Cedex 3, France*

Abstract

In image processing, the distance transform (DT), in which each object grid point is assigned the distance to the closest background grid point, is a powerful and often used tool. In this paper, distance functions defined as minimal cost-paths are used and a number of algorithms that can be used to compute the DT are presented. We give proofs of the correctness of the algorithms.

Keywords: distance function, distance transform, weighted distances, neighborhood sequences

```
@article{strand2012tcs,  
  Author = {Strand, Robin and Normand, Nicolas},  
  Doi = {10.1016/j.tcs.2012.05.010},  
  Journal = {Theoretical Computer Science},  
  Pages = {80-93},  
  Title = {Distance Transform Computation for Digital Distance Functions},  
  Volume = {448},  
  Year = {2012}}
```

The original publication is available at www.sciencedirect.com

SHERPA/RoMEO: green

<http://www.sherpa.ac.uk/romeo/search.php?jrule=ISSN&search=0304-3975>

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

Preprint submitted to Elsevier

Distance Transform Computation for Digital Distance Functions

Robin Strand^a, Nicolas Normand^b

^a*Centre for Image Analysis, Uppsala University,
Box 337, SE-75105 Uppsala, Sweden*

^b*LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597,
Polytech Nantes, La Chantrerie, 44306 Nantes Cedex 3, France*

Abstract

In image processing, the distance transform (DT), in which each object grid point is assigned the distance to the closest background grid point, is a powerful and often used tool. In this paper, distance functions defined as minimal cost-paths are used and a number of algorithms that can be used to compute the DT are presented. We give proofs of the correctness of the algorithms.

Keywords: distance function, distance transform, weighted distances, neighborhood sequences

1. Introduction

In [1], an algorithm for computing distance transforms (DTs) using the basic city-block (horizontal and vertical steps are allowed) and chessboard (diagonal steps are allowed in conjunction with the horizontal and vertical steps) distance functions was presented in [1]. These distance functions are defined as shortest paths and the corresponding distance maps can be computed efficiently. Since these path-based distance functions are defined by the cost of discrete paths, we call them *digital* distance functions.

There are two commonly used generalizations of the city-block and chessboard distance functions, the *weighted distances* [2, 3, 4], and *distances based on neighborhood sequences* (ns-distances) [5, 6, 7, 8]. The weighted distance

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

039 is defined as the cost of a minimal cost-path and the ns-distance is defined
 040 as a shortest path in which the neighborhood that is allowed in each step
 041 is given by a neighborhood sequence. With weighted distances, a two-scan
 042 algorithm is sufficient for any point-lattice, see [2, 9]. For ns-distances, three
 043 scans are needed for computing correct DTs on a square grid [10].

044 In this paper, we consider the *weighted ns-distance* [11, 12, 13] in which
 045 both weights and a neighborhood sequence are used to define the distance
 046 function. By using “optimal” parameters (weights and neighborhood se-
 047 quence), the asymptotic shape of the discs with this distance function is
 048 a twelve-sided polygon, see [11]. The relative error is thus asymptotically
 049 $(1/\cos(\pi/12) - 1)/((1/\cos(\pi/12) + 1)/2) \approx 3.5\%$ using only 3×3 neigh-
 050 borhoods when computing the DT. In other words, we have a close to ex-
 051 act approximation of the Euclidean distance still using the path-based ap-
 052 proach with connectivities corresponding to small neighborhoods. Some dif-
 053 ferent algorithms for computing the distance transform using the weighted
 054 ns-distance functions are given in this paper.

055 The paper is organized as follows: First, some basic notions are given and
 056 the definition of weighted ns-distances is given. In Section 3, algorithms us-
 057 ing an additional DT holding the *length* of the paths that define the distance
 058 values are presented. The notion of distance propagating path is introduced
 059 to prove that correct DTs are computed. In Section 4, a look-up table that
 060 holds the value that should be propagated in each direction is used to com-
 061 pute the DT. The third approach considered here work for metric distance
 062 functions with periodic neighborhood sequences. A large mask that holds all
 063 distance information corresponding to the first period of the neighborhood
 064 sequence is used.

066 2. Weighted distances based on neighborhood sequences

067 The distance function considered here is defined by a neighborhood se-
 068 quence using two neighborhoods and two weights. The neighborhoods are
 069 defined as follows

$$071 \mathcal{N}_1 = \{(\pm 1, 0), (0, \pm 1)\} \text{ and } \mathcal{N}_2 = \{(\pm 1, \pm 1)\}.$$

072 Two grid points $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{Z}^2$ are strict r -neighbors, $r \in \{1, 2\}$, if $\mathbf{p}_2 - \mathbf{p}_1 \in \mathcal{N}_r$.
 073 Neighbors of higher order can also be defined, but in this paper, we will use
 074 only 1- and 2-neighbors. Let

$$075 \mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2.$$

077 The points $\mathbf{p}_1, \mathbf{p}_2$ are 2-neighbors (or *adjacent*) if $\mathbf{p}_2 - \mathbf{p}_1 \in \mathcal{N}$, i.e., if
078 they are strict r -neighbors for some r . A ns B is a sequence $B = (b(i))_{i=1}^{\infty}$,
079 where each $b(i)$ denotes a neighborhood relation in \mathbb{Z}^2 . If B is periodic, i.e.,
080 if for some finite, strictly positive $l \in \mathbb{Z}_+$, $b(i) = b(i+l)$ is valid for all $i \in \mathbb{N}^*$,
081 then we write $B = (b(1), b(2), \dots, b(l))$.

082 The following notation is used for the number of 1:s and 2:s in the ns B
083 up to position k .

$$084 \quad \mathbf{1}_B^k = |\{i : b(i) = 1, 1 \leq i \leq k\}| \text{ and } \mathbf{2}_B^k = |\{i : b(i) = 2, 1 \leq i \leq k\}|.$$

086 A *path* in a grid, denoted \mathcal{P} , is a sequence $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$ of adjacent grid
087 points. A path is a B -*path* of length $\mathcal{L}(\mathcal{P}) = n$ if, for all $i \in \{1, 2, \dots, n\}$,
088 \mathbf{p}_{i-1} and \mathbf{p}_i are $b(i)$ -neighbors. The number of 1-steps and strict 2-steps in
089 a given path \mathcal{P} is denoted $\mathbf{1}_{\mathcal{P}}$ and $\mathbf{2}_{\mathcal{P}}$, respectively.

090 **Definition 1.** Given the ns B , the ns-distance $d(\mathbf{p}_0, \mathbf{p}_n; B)$ between the
091 points \mathbf{p}_0 and \mathbf{p}_n is the length of a shortest B -path between the points.
092

093 Let the real numbers α and β (the *weights*) and a B -path \mathcal{P} of length
094 n , where exactly l ($l \leq n$) pairs of adjacent grid points in the path are
095 strict 2-neighbors be given. The *cost of the (α, β) -weighted B -path \mathcal{P}* is
096 $\mathcal{C}_{\alpha, \beta}(\mathcal{P}) = (n - l)\alpha + l\beta$. The B -path \mathcal{P} between the points \mathbf{p}_0 and \mathbf{p}_n is
097 a (α, β) -*weighted minimal cost B -path between the points \mathbf{p}_0 and \mathbf{p}_n* if no
098 other (α, β) -weighted B -path between the points has lower cost than the
099 (α, β) -weighted B -path \mathcal{P} .

100 **Definition 2.** Given the ns B and the weights α, β , the weighted ns-distance
101 $d_{\alpha, \beta}(\mathbf{p}_0, \mathbf{p}_n; B)$ is the cost of a (α, β) -weighted minimal cost B -path between
102 the points.
103

104 The following theorem is from [11].

105 **Theorem 1** (Weighted ns-distance in \mathbb{Z}^2). *Let the ns B , the weights α, β
106 s.t. $0 < \alpha \leq \beta \leq 2\alpha$, and the point $(x, y) \in \mathbb{Z}^2$, where $x \geq y \geq 0$, be given.
107 The weighted ns-distance between $\mathbf{0}$ and (x, y) is given by*
108

$$109 \quad d_{\alpha, \beta}(\mathbf{0}, (x, y); B) = (2k - x - y) \cdot \alpha + (x + y - k) \cdot \beta$$

$$110 \quad \text{where } k = \min_l : l \geq \max(x, x + y - \mathbf{2}_B^l).$$

111 Note if $B = (1)$ then $k = x + y$ so $d(0, (x, y); (1)) = (x + y)\alpha$ which
112 is α times the city-block distance whereas if $B = (2)$ then $k = x$ and
113 $d(0, (x, y); (2)) = (x - y)\alpha + y\beta$ which is the (α, β) -weighted distance.
114

115 **3. Computing the distance transform using path-length informa-**
 116 **tion**

117 In this section, the computation of DTs using the distance function de-
 118 fined in the previous section will be considered. Since the size of a digital
 119 image when stored in a computer is finite, we define the *image domain* as
 120 a finite subset of \mathbb{Z}^2 denoted \mathcal{I} . In this paper we use image domains of the
 121 form

$$122 \mathcal{I} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \tag{1}$$

123 **Definition 3.** We call the function $F : \mathcal{I} \rightarrow \mathbb{R}_0^+$ an *image*.
 124

125 Note that real numbers are allowed in the range of F . We denote the
 126 *object* X and the *background* is $\bar{X} = \mathbb{Z}^2 \setminus X$. We denote the distance trans-
 127 form for path-based distances with $DT_{\mathcal{C}}$, where the subscript \mathcal{C} indicates that
 128 costs of paths are computed.
 129

130 **Definition 4.** The *distance transform* $DT_{\mathcal{C}}$ of an object $X \subset \mathcal{I}$ is the map-
 131 ping
 132

$$133 DT_{\mathcal{C}} : \mathcal{I} \rightarrow \mathbb{R}_0^+ \text{ defined by}$$

$$134 \mathbf{p} \mapsto d(\mathbf{p}, \bar{X}), \text{ where}$$

$$135 d(\mathbf{p}, \bar{X}) = \min_{\mathbf{q} \in \bar{X}} \{d(\mathbf{p}, \mathbf{q})\}.$$

136 For weighted ns-distances, the size of the neighborhood allowed in each
 137 step is determined by the *length* of the minimal cost-paths (not the cost).
 138 In the first approach to compute the DT, an additional transform, $DT_{\mathcal{L}}$ that
 139 holds the length of the minimal cost path at each point is used.
 140

141 **Definition 5.** The set of transforms $\{DT_{\mathcal{L}}^i\}$ of an object $X \subset \mathbb{Z}^2$ is defined
 142 by all mappings $DT_{\mathcal{L}}^i$ that satisfy
 143

$$144 DT_{\mathcal{L}}^i(\mathbf{p}) = d_{1,1}(\mathbf{p}, \mathbf{q}; B), \text{ where}$$

$$145 \mathbf{q} \text{ is such that } d_{\alpha,\beta}(\mathbf{q}, \mathbf{p}; B) = d_{\alpha,\beta}(\mathbf{p}, \bar{X}; B).$$

146 See Figure 1 for an example showing $DT_{\mathcal{C}}$ and some different $DT_{\mathcal{L}}$ (su-
 147 perscript omitted when it is not explicitly needed) of an object.
 148

149 When $\alpha = \beta = 1$, $DT_{\mathcal{L}}$ is uniquely defined and $DT_{\mathcal{C}} = DT_{\mathcal{L}}$. Example 1
 150 illustrates that $DT_{\mathcal{L}}$ is not always uniquely defined when $\alpha \neq \beta$. We will
 151
 152

see that despite this, the correct distance values are propagated by natural extensions of well-known algorithms when DT_C is used together with $DT_{\mathcal{L}}^i$ for *any* i are used to propagate the distance values.

We now introduce the notion of distance propagating path.

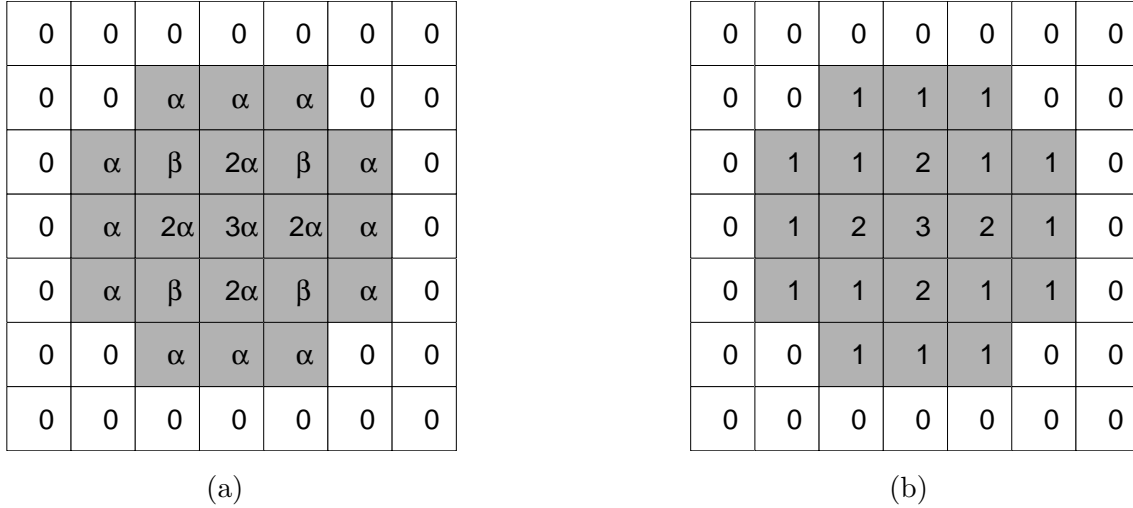


Figure 1: Distance transforms for $B = (2, 1)$ and $\alpha \leq \beta \leq 2\alpha$. The background is shown in white, DT_C is shown in (a) and a $DT_{\mathcal{L}}$ is shown in (b).

Definition 6. Given an object grid point $\mathbf{p} \in X$, a minimal cost B -path $\mathcal{P}_{\mathbf{q}, \mathbf{p}} = \langle \mathbf{q} = \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n = \mathbf{p} \rangle$, where $\mathbf{q} \in \bar{X}$ is a background grid point, is a *distance propagating B -path* if

- (i) $\mathcal{C}_{\alpha, \beta}(\langle \mathbf{p}_0, \dots, \mathbf{p}_i \rangle) = DT_C(\mathbf{p}_i)$ for all i and
- (ii) $\mathbf{p}_i, \mathbf{p}_{i+1}$ are $b(DT_{\mathcal{L}}^j(\mathbf{p}_i) + 1) -$ neighbors for all i ,

for all j .

If property (i) in the definition above is fulfilled, then we say that $\mathcal{P}_{\mathbf{p}, \mathbf{q}}$ is represented by DT_C and if property (ii) is fulfilled, then $\mathcal{P}_{\mathbf{p}, \mathbf{q}}$ is represented by $DT_{\mathcal{L}}^j$. Note that when $\alpha = \beta$, then (i) implies (ii).

If we can guarantee that there is such a path for every object grid point, then the distance transform can be constructed by locally propagating distance information from \bar{X} to any $\mathbf{p} \in X$. Now, a number of definitions will be introduced. Using these definitions, we can show that there is always a distance propagating path when the weighted ns-distance function is used. The following definitions are illustrated in Example 1 and 2.

191 **Definition 7.** Let α, β such that $0 < \alpha \leq \beta \leq 2\alpha$, a ns B , an object X , and
192 a point $\mathbf{p} \in X$ be given. A minimal cost B -path $\mathcal{P}_{\mathbf{q}, \mathbf{p}}$, where $\mathbf{q} \in \overline{X}$, such
193 that $d_{\alpha, \beta}(\mathbf{p}, \mathbf{q}; B) = d_{\alpha, \beta}(\mathbf{p}, \overline{X}; B)$ is a minimal cost B -path with *minimal*
194 *number of 2-steps* if, for all paths $\mathcal{Q}_{\mathbf{q}', \mathbf{p}}$ with $\mathbf{q}' \in \overline{X}$ such that $\mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}}) =$
195 $\mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}})$, we have

$$196 \quad \mathbf{2}_{\mathcal{P}_{\mathbf{q}, \mathbf{p}}} \leq \mathbf{2}_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}}}.$$

197
198 In other words, if there are several paths defining the distance at a point
199 \mathbf{p} , the path with the least number of 2-steps is a minimal cost B -path with
200 *minimal number of 2-steps*. See Example 1 and 2.

201 **Remark 1.** A minimal cost-path with minimal number of 2-steps is a min-
202 imal cost-path of maximal length.
203

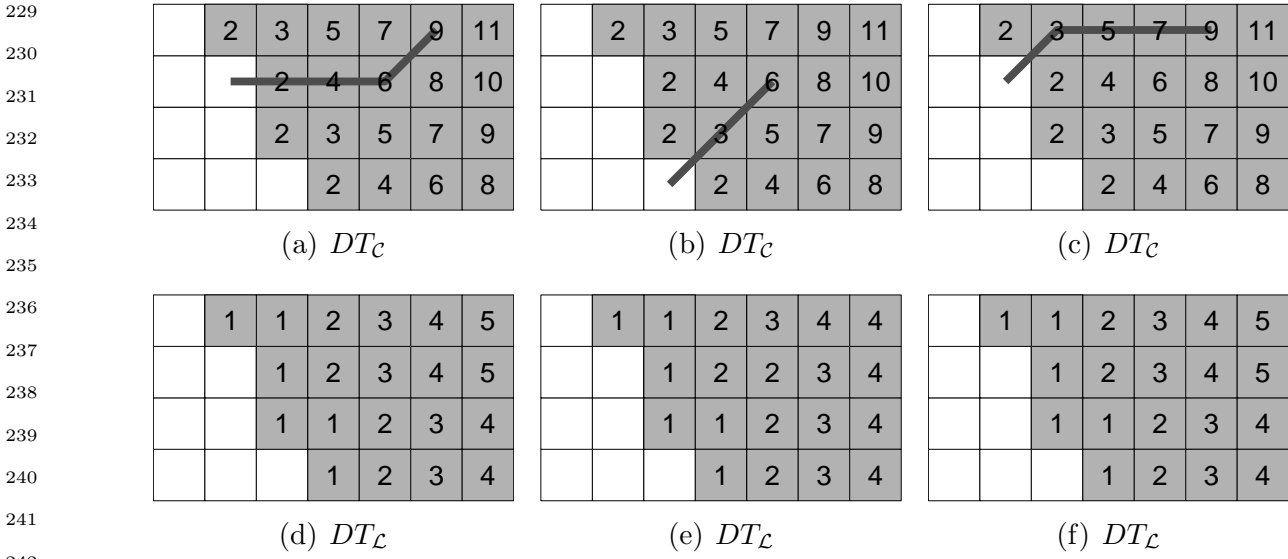
204 **Definition 8.** Let α, β such that $0 < \alpha \leq \beta \leq 2\alpha$, a ns B , and points
205 $\mathbf{p}, \mathbf{q} \in \mathbb{Z}^2$ be given. The minimal cost (α, β) -weighted B -path $\mathcal{P}_{\mathbf{p}, \mathbf{q}} = \langle \mathbf{p} =$
206 $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n = \mathbf{q} \rangle$ is a *fastest* minimal cost (α, β) -weighted B -path between
207 \mathbf{p} and \mathbf{q} if there is an i , $0 \leq i \leq n$ such that
208

$$209 \quad \mathbf{2}_{\langle \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_i \rangle} = \mathbf{2}_B^i \text{ and } \mathbf{2}_{\langle \mathbf{p}_{i+1}, \mathbf{p}_{i+2}, \dots, \mathbf{p}_n \rangle} = 0.$$

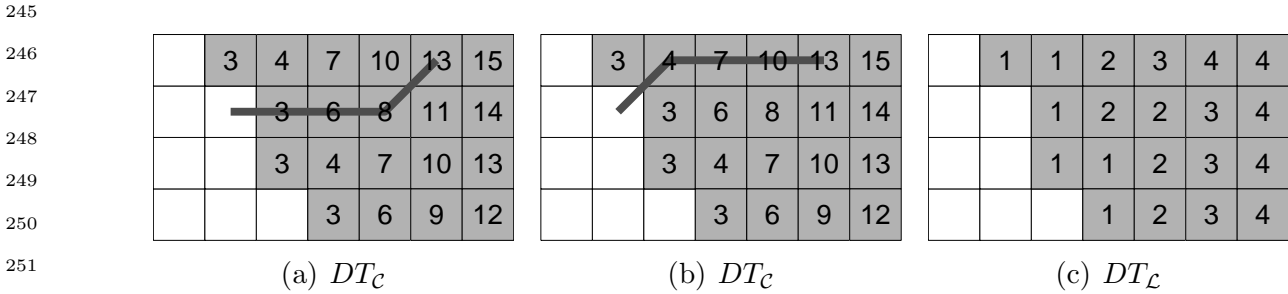
210
211 In other words, the minimal cost path between two points in which the
212 2-steps occur after as few steps as possible is a *fastest* minimal cost (α, β) -
213 weighted B -path. See Example 1 and 2.

214 **Example 1.** This example illustrates that a path that is not a fastest path
215 is not necessarily represented by $DT_{\mathcal{L}}^i$ for some i . Consider the (part of
216 the) object showed in Figure 2(a)–(f). The parameters $(\alpha, \beta) = (2, 3)$ and
217 $B = (2, 2, 1)$ are used. In (d)–(f), some $DT_{\mathcal{L}}^i$:s are shown. The path in
218 (a) has minimal number of 2-steps, but it is not a fastest path and is not
219 represented by the $DT_{\mathcal{L}}$ in (e). The path in (b) has not minimal number of
220 2-steps. The path in (c) is a fastest path with minimal number of 2-steps
221 and is also represented by all $DT_{\mathcal{L}}$ in (d)–(f). The paths shown in (b) and
222 (c) are distance propagating paths, and the path in (a) is not a distance
223 propagating path for $DT_{\mathcal{L}}$ in (e).
224

225
226 **Example 2.** In Figure 3(a)–(c), $B = (2, 2, 1)$ and $(\alpha, \beta) = (3, 4)$ are used.
227 In (a), the $DT_{\mathcal{L}}$ of an object and a minimal cost (α, β) -weighted B -path
228 with minimal number of 2-steps that is not distance propagating is shown.



243 Figure 2: Distance transform using $(\alpha, \beta) = (2, 3)$ and $B = (2, 2, 1)$ for a part of an object
244 in \mathbb{Z}^2 , see Example 1.



253 Figure 3: Distance transform using $(\alpha, \beta) = (3, 4)$ and $B = (2, 2, 1)$ for a part of an object
254 in \mathbb{Z}^2 , see Example 2.

256 A distance propagating *fastest* minimal cost (α, β) -weighted B -path with
257 minimal number of 2-steps is shown in (b). The $DT_{\mathcal{L}}$ that corresponds to
258 $DT_{\mathcal{C}}$ in (a)–(b) is shown in (c).

259 The following theorem says that a path satisfying Definition 7 and 8 is a
260 distance propagating path as defined in Definition 6. The theorem is proved
261 in Lemma 2 and Lemma 3 below.

263 **Theorem 2.** *If the B -path $\mathcal{P}_{\mathbf{q}, \mathbf{p}}$ ($\mathbf{p} \in X$, $\mathbf{q} \in \bar{X}$ such that $d_{\alpha, \beta}(\mathbf{p}, \bar{X}) =$
264 $\mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}})$) is a fastest minimal cost B -path with minimal number of 2-steps
265 then $\mathcal{P}_{\mathbf{q}, \mathbf{p}}$ is a distance propagating B -path.*

Intuitively, we want the path to be of maximal length (*B-path with minimal number of 2-steps, see Remark 1*) and among the paths with this property, the path such that the 2-steps appear after as few steps as possible (*fastest B-path*). The algorithms we present will always be able to propagate correct distance values along such paths.

Lemma 1 will be used in the proofs of Lemma 2 and Lemma 3. It is a direct consequence of Theorem 1. In Lemma 1, $B(k) = (b(i))_{i=k}^{\infty}$

Lemma 1. *Given α, β such that $0 < \alpha \leq \beta \leq 2\alpha$, the ns B , the points \mathbf{p}, \mathbf{q} , and an integer $k \geq 1$, we have*

$$d_{\alpha, \beta}(\mathbf{p}, \mathbf{q}; B) \leq d_{\alpha, \beta}(\mathbf{p}, \mathbf{q}; B(k)) + (2\alpha - \beta)(k - 1).$$

We consider the case $\alpha < \beta$. Lemma 2 and Lemma 3 gives the proof of Theorem 2 for weighted ns-distances.

Lemma 2. *Let the weights α, β such that $0 < \alpha < \beta \leq 2\alpha$, the ns B , and the point $\mathbf{p} \in X$ be given. Any fastest minimal cost (α, β) -weighted B -path $\mathcal{P}_{\mathbf{q}, \mathbf{p}} = \langle \mathbf{q} = \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n = \mathbf{p} \rangle$ (for some $\mathbf{q} \in \bar{X}$ such that $d_{\alpha, \beta}(\mathbf{p}, \bar{X}) = \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}})$) satisfies (i) in Definition 6, i.e.,*

$$d_{\alpha, \beta}(\mathbf{p}_i, \bar{X}; B) = \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{p}_0, \mathbf{p}_i}) \quad \forall i : 0 \leq i \leq n. \quad (2)$$

Proof. First we note that there always exists a $\mathbf{q} \in \bar{X}$ such that there is a fastest minimal cost (α, β) -weighted B -path $\mathcal{P}_{\mathbf{q}, \mathbf{p}} = \langle \mathbf{q} = \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n = \mathbf{p} \rangle$. To prove (2), assume that there is a $\mathbf{q}' \in \bar{X}$ and a path $\mathcal{Q}_{\mathbf{q}', \mathbf{p}} = \langle \mathbf{q}' = \mathbf{p}'_0, \mathbf{p}'_1, \dots, \mathbf{p}'_k = \mathbf{p}_i, \mathbf{p}'_{k+1}, \dots, \mathbf{p}'_m = \mathbf{p} \rangle$ for some i such that

$$\mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}) > \mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i})$$

Case i: $\mathcal{L}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}) < \mathcal{L}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_i})$

Case i(a): $\mathbf{2}_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}} > \mathbf{2}_{\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}}$

Since $\mathcal{P}_{\mathbf{q}, \mathbf{p}}$ is a fastest path, $\mathbf{2}_{\mathcal{P}_{\mathbf{p}_i, \mathbf{p}}} = 0$. This implies that $\mathcal{P}_{\mathbf{p}_i, \mathbf{p}}$ is a minimal cost (α, β) -weighted B -path for $B = (1)$ and since, by Theorem 1, any ns generates distances less than (or equal to) $B = (1)$, $\mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{p}_i, \mathbf{p}}) \leq \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{p}_i, \mathbf{p}})$. Thus, $\mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}}) = \mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}) + \mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{p}_i, \mathbf{p}}) < \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}) + \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{p}_i, \mathbf{p}}) = \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}})$. This contradicts that $\mathcal{P}_{\mathbf{q}, \mathbf{p}}$ is a minimal cost-path, so this case can not occur.

Case i(b): $\mathbf{2}_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}} \leq \mathbf{2}_{\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}}$

Since $\mathcal{L}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}) = \mathcal{L}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}) - L$ for some positive integer L , we have

$$\mathbf{1}_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}} + \mathbf{2}_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}} = \mathcal{L}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}) = \mathcal{L}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}) - L = \mathbf{1}_{\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}} + \mathbf{2}_{\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}} - L.$$

Using $2_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}} \leq 2_{\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}}$ and $\alpha \leq \beta$, we get

$$\mathbf{1}_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}} \alpha + 2_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}} \beta \leq \mathbf{1}_{\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}} \alpha + 2_{\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}} \beta - L\alpha.$$

Thus, $\mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}) \leq \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}) - L\alpha$. By Lemma 1, $\mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{p}_i, \mathbf{p}}) \leq \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{p}_i, \mathbf{p}}) + (2\alpha - \beta)L$.

We use these results and get $\mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}}) = \mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}) + \mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{p}_i, \mathbf{p}}) \leq \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}) - L\alpha + \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{p}_i, \mathbf{p}}) + (2\alpha - \beta)L = \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}) + \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{p}_i, \mathbf{p}}) + (\alpha - \beta)L < \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}})$.

This contradicts that $\mathcal{P}_{\mathbf{q}, \mathbf{p}}$ is a minimal cost-path.

Case ii: $\mathcal{L}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}) \geq \mathcal{L}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_i})$

Now, $2_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}_i}} < 2_{\mathcal{P}_{\mathbf{q}, \mathbf{p}_i}} \leq 2_B^i$.

Construct the path (*not* a B -path) $\mathcal{Q}'_{\mathbf{q}', \mathbf{p}} = \langle \mathbf{q}' = \mathbf{p}'_0, \mathbf{p}'_1, \dots, \mathbf{p}'_k = \mathbf{p}_i, \mathbf{p}_{i+1}, \dots, \mathbf{p}_n = \mathbf{p} \rangle$ of length $n' = \mathcal{L}(\mathcal{Q}'_{\mathbf{q}', \mathbf{p}_i}) + \mathcal{L}(\mathcal{P}_{\mathbf{p}_i, \mathbf{p}_n}) \geq \mathcal{L}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_n}) = n$. We have $2_{\mathcal{Q}'_{\mathbf{q}', \mathbf{p}_n}} = 2_{\mathcal{Q}'_{\mathbf{q}', \mathbf{p}_i}} + 2_{\mathcal{P}_{\mathbf{p}_i, \mathbf{p}_n}} < 2_B^i + 2_{B^{(i+1)}}^{n-i} = 2_B^n \leq 2_B^{n'}$. This means that there is a B -path $\mathcal{Q}''_{\mathbf{q}', \mathbf{p}_n}$ (obtained by permutation of the positions of the 1-steps and 2-steps in $\mathcal{Q}'_{\mathbf{q}', \mathbf{p}_n}$) of length n' such that $\mathcal{C}_{\alpha, \beta}(\mathcal{Q}''_{\mathbf{q}', \mathbf{p}_n}) < \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_n})$, which contradicts that $\mathcal{P}_{\mathbf{q}, \mathbf{p}_n}$ is a minimal cost path.

The assumption is false, since a contradiction follows for all cases. \square

Left to prove of Theorem 2 is that there is a path fulfilling the previous lemma that is represented by $DT_{\mathcal{L}}$. This is necessary for the path to be propagated correctly by an algorithm.

Lemma 3. *Let the weights α, β such that $0 < \alpha < \beta \leq 2\alpha$, the n s B , and the point $\mathbf{p} \in X$ be given. Any fastest minimal cost (α, β) -weighted B -path with minimal number of 2-steps $\mathcal{P}_{\mathbf{q}, \mathbf{p}} = \langle \mathbf{q} = \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n = \mathbf{p} \rangle$ (for some $\mathbf{q} \in \overline{X}$ such that $d_{\alpha, \beta}(\mathbf{p}, \overline{X}) = \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}})$) satisfies (ii) in Definition 6, i.e.,*

$$\mathbf{p}_i, \mathbf{p}_{i+1} \text{ are } b(DT_{\mathcal{L}}(\mathbf{p}_i) + 1) - \text{neighbors for all } i.$$

Proof. Given a $\mathbf{p} \in X$, assume that there is a fastest minimal cost (α, β) -weighted B -path $\mathcal{P}_{\mathbf{q}, \mathbf{p}}$ with minimal number of 2-steps such that $\mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}}) = d_{\alpha, \beta}(\mathbf{p}, \overline{X})$ and a $K < n$ such that $\mathcal{P}_{\mathbf{q}, \mathbf{p}_K}$ is *not* represented by $DT_{\mathcal{L}}$, i.e., that $\mathbf{p}_{K-1}, \mathbf{p}_K$ are not $b(DT_{\mathcal{L}}(\mathbf{p}_{K-1}) + 1)$ -neighbors. (Otherwise the value $\mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_K})$ is propagated from $DT_{\mathcal{L}}(\mathbf{p}_{K-1})$.) It follows that the values of $DT_{\mathcal{L}}(\mathbf{p}_{K-1})$ and $DT_{\mathcal{L}}(\mathbf{p}_{K-1})$ are given by a path $\mathcal{Q}_{\mathbf{q}', \mathbf{p}} = \langle \mathbf{q}' = \mathbf{p}'_0, \mathbf{p}'_1, \dots, \mathbf{p}'_k = \mathbf{p}_i, \mathbf{p}'_{k+1}, \dots, \mathbf{p}'_n = \mathbf{p} \rangle$ for some i and some $\mathbf{q}' \in \overline{X}$ such that

$$\mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_{K-1}}) = d_{\alpha, \beta}(\mathbf{p}_{K-1}, \mathbf{q}; B) = d_{\alpha, \beta}(\mathbf{p}_{K-1}, \mathbf{q}'; B) = \mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_{K-1}})$$

and

$$\mathcal{L}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_{K-1}}) \neq \mathcal{L}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_{K-1}}).$$

We follow the cases in the proof of Lemma 2:

Case i: $\mathcal{L}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_{K-1}}) < \mathcal{L}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_{K-1}})$

Case i(a): $2_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}_{K-1}}} > 2_{\mathcal{P}_{\mathbf{q}, \mathbf{p}_{K-1}}}$

We get $\mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}}) = \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}})$ from the proof of Lemma 2 and $\mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_{K-1}}) = \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_{K-1}})$ by construction. Since $\mathcal{P}_{\mathbf{q}, \mathbf{p}}$ is a fastest path, $\mathbf{p}_{K-1}, \mathbf{p}_K$ is a 1-step, so it is also a $b(DT_{\mathcal{L}}(\mathbf{p}_{K-1}) + 1)$ -step. Therefore, $\mathbf{p}_{K-1}, \mathbf{p}_K$ are $b(DT_{\mathcal{L}}(\mathbf{p}_{K-1}) + 1)$ -neighbors. Contradiction.

Case i(b): $2_{\mathcal{Q}_{\mathbf{q}', \mathbf{p}_{K-1}}} \leq 2_{\mathcal{P}_{\mathbf{q}, \mathbf{p}_{K-1}}}$

Following the proof of Lemma 2, we get $\mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}}) < \mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{q}, \mathbf{p}})$.

Case ii: $\mathcal{L}(\mathcal{Q}_{\mathbf{q}', \mathbf{p}_{K-1}}) > \mathcal{L}(\mathcal{P}_{\mathbf{q}, \mathbf{p}_{K-1}})$

This leads to a longer B -path $\mathcal{Q}_{\mathbf{q}', \mathbf{p}}''$ with lower (or equal) cost by the construction in the proof of Lemma 2, so this case leads to a contradiction since $\mathcal{P}_{\mathbf{q}, \mathbf{p}}$ is a B -path of maximal length (see Remark 1) by assumption. \square

3.1. Algorithms

In this section, algorithms for computing DTs using the additional transform $DT_{\mathcal{L}}$ are presented. First, we focus on a wavefront propagation algorithm. By Theorem 2, there is a distance propagating path for each $\mathbf{p} \in X$. This proves the correctness of Algorithm 1.

381 **Algorithm 1:** Computing $DT_{\mathcal{C}}$ and $DT_{\mathcal{L}}$ for weighted ns-distances by
 382 wave-front propagation.

383 **Input:** B , α, β , neighborhoods \mathcal{N}_1 and \mathcal{N}_2 , and an object $X \subset \mathbb{Z}^2$.

384 **Output:** The distance transforms $DT_{\mathcal{C}}$ and $DT_{\mathcal{L}}$.

385 **Initialization:** Set $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and
 386 $DT_{\mathcal{L}}(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$. Set $DT_{\mathcal{L}} = DT_{\mathcal{C}}$. For all grid
 387 points $\mathbf{p} \in \overline{X}$ adjacent to X : push $(\mathbf{p}, DT_{\mathcal{C}}(\mathbf{p}))$ to the list L of
 388 ordered pairs sorted by increasing $DT_{\mathcal{C}}(\mathbf{p})$.

389 **Notation:** $\omega_{\mathbf{v}}$ is α if $\mathbf{v} \in \mathcal{N}_1$ and β if $\mathbf{v} \in \mathcal{N}_2$.

390 **while** L is not empty **do**

391 **foreach** \mathbf{p} in L with smallest $DT_{\mathcal{C}}(\mathbf{p})$ **do**

392 Pop $(\mathbf{p}, DT_{\mathcal{C}}(\mathbf{p}))$ from L ;

393 **foreach** \mathbf{q} : \mathbf{q}, \mathbf{p} are $b(DT_{\mathcal{L}}(\mathbf{p}) + 1)$ -neighbors **do**

394 **if** $DT_{\mathcal{C}}(\mathbf{q}) > DT_{\mathcal{C}}(\mathbf{p}) + \omega_{\mathbf{p}-\mathbf{q}}$ **then**

395 $DT_{\mathcal{C}}(\mathbf{q}) \leftarrow DT_{\mathcal{C}}(\mathbf{p}) + \omega_{\mathbf{p}-\mathbf{q}}$;

396 $DT_{\mathcal{L}}(\mathbf{q}) \leftarrow DT_{\mathcal{L}}(\mathbf{p}) + 1$;

397 Push $(\mathbf{q}, DT_{\mathcal{C}}(\mathbf{q}))$ to L ;

398 **end**

399 **end**

400 **end**

401 **end**

403
 404 Now, the focus is on the raster-scanning algorithm. We will see that the
 405 DT can be computed correctly in three scans. Since a fixed number of scans
 406 is used and the time complexity is bounded by a constant for each visited
 407 grid point, the time complexity is linear in the number of grid points in the
 408 image domain.

409 We recall the following lemma from [14]:

410 **Lemma 4.** *When $\alpha < \beta \leq 2\alpha$, any minimal cost-path between $(0, 0)$ and
 411 (x, y) , where $x \geq y \geq 0$, consists only of the steps $(1, 0)$, $(1, 1)$, and $(0, 1)$.*

413 Since we consider only “rectangular” image domains, the following lemma
 414 holds.

415 **Lemma 5.** *Given two points \mathbf{p}, \mathbf{q} in \mathcal{I} , a ns B and weights $\beta > \alpha$. Any
 416 point in any minimal cost (α, β) -weighted B -path between \mathbf{p} and \mathbf{q} is in the
 417 image domain.*

419 *Proof.* Consider the point $\mathbf{p} = (x, y)$, where $x \geq y \geq 0$. By Lemma 4, any
 420 (α, β) -weighted B -path of minimal cost from $\mathbf{0}$ to \mathbf{p} consists only of the local
 421 steps $(0, 1), (1, 1), (1, 0)$. The theorem follows from this result. \square

422 Let $\mathcal{N}^1 = \{(1, 0), (1, 1)\}, \mathcal{N}^2 = \{(1, 1), (0, 1)\}, \dots, \mathcal{N}^8 = \{(1, -1), (1, 0)\}$.
 423 In other words, the set \mathcal{N} is divided into set according to which octant they
 424 belong.
 425

426 **Lemma 6.** *Let γ be any permutation of $1, 2, \dots, 8$. Between any two points,*
 427 *there is a distance propagating B -path $\mathcal{P}_{\mathbf{q}, \mathbf{p}} = \langle \mathbf{q} = \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n = \mathbf{p} \rangle$ and*
 428 *integers $0 \leq K_1 \leq K_2 \leq \dots \leq K_8 = n$ such that*

$$\begin{aligned}
 429 & \mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^{\gamma(1)} && \text{if } i \leq K_1 \\
 430 & \mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^{\gamma(2)} && \text{if } i > K_1 \text{ and } i \leq K_2 \\
 431 & && \vdots \\
 432 & && \vdots \\
 433 & \mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^{\gamma(8)} && \text{if } i > K_7 \text{ and } i \leq K_8. \\
 434 & &&
 \end{aligned}$$

435 *Proof.* Consider $\mathbf{q} = \mathbf{0}$ and $\mathbf{p} = (x, y)$ such that $x \geq y \geq 0$. Any minimal
 436 minimal cost B -path consists only of the local steps $(1, 0), (1, 1), (0, 1)$ by
 437 Lemma 4. Reordering the 1-steps does not affect the cost of the path. The
 438 path obtained by reordering the 1-steps in a fastest minimal cost (α, β) -
 439 weighted B -path with minimal number of 2-steps is still a fastest minimal
 440 cost (α, β) -weighted B -path with minimal number of 2-steps. Therefore,
 441 there are distance propagating B -path such that
 442

$$\begin{aligned}
 443 & \mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^1 && \text{if } i \leq K_1 \\
 444 & \mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^2 && \text{if } i > K_1 \text{ and } i \leq K_2 \\
 445 & &&
 \end{aligned}$$

446 for some integers $0 \leq K_1 \leq K_2 = n$ and

$$\begin{aligned}
 447 & \mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^2 && \text{if } i \leq K_1 \\
 448 & \mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^1 && \text{if } i > K_1 \text{ and } i \leq K_2 \\
 449 & &&
 \end{aligned}$$

450 for some integers $0 \leq K_1 \leq K_2 = n$. The general case follows from transla-
 451 tion and rotation invariance. \square

452 **Definition 9.** A *scanning mask* is a subset $\mathcal{M} \subset \mathcal{N}$.

453 **Definition 10.** A *scanning order (so)* is an enumeration of the $M = \text{card}(\mathcal{I})$
 454 distinct points in \mathcal{I} , denoted $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M$.

457 **Definition 11.** Let $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M \in \mathcal{I}$ be a scanning order and \mathcal{M} a scan-
 458 ning mask. The scanning mask \mathcal{M} supports the scanning order if

$$459 \quad \forall \mathbf{p}_i, \forall \mathbf{v} \in \mathcal{M}, ((\exists i' > i : \mathbf{p}_{i'} = \mathbf{p}_i + \mathbf{v}) \text{ or } (\mathbf{p}_i + \mathbf{v} \notin \mathcal{I}_{\mathbb{G}})).$$

462 **Algorithm 2:** Computing $DT_{\mathcal{C}}$ and $DT_{\mathcal{L}}$ for weighted ns-distances by
 463 raster scanning.

464 **Input:** B, α, β , scanning masks \mathcal{M}^i , and an object $X \subset \mathbb{Z}^2$.

465 **Output:** The distance transforms $DT_{\mathcal{C}}$ and $DT_{\mathcal{L}}$.

466 **Initialization:** Set $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and
 467 $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$. Set $DT_{\mathcal{L}} = DT_{\mathcal{C}}$.

468 **Comment:** The image domain \mathcal{I} defined by eq. 1 is scanned L
 469 times using scanning orders such that the scanning mask \mathcal{M}^i
 470 supports the scanning order $so_i, i \in \{1, \dots, L\}$.

471 **Notation:** $\omega_{\mathbf{v}}$ is α if $\mathbf{v} \in \mathcal{N}_1$ and β if $\mathbf{v} \in \mathcal{N}_2$.

472 **for** $i = 1 : L$ **do**

```

473   |
474   |   foreach  $\mathbf{p} \in \mathcal{I}$  following  $so_i$  do
475   |   |   if  $DT_{\mathcal{C}}(\mathbf{p}) < \infty$  then
476   |   |   |   foreach  $\mathbf{v} \in \mathcal{M}^i$  do
477   |   |   |   |   if  $\mathbf{p}$  and  $\mathbf{p} + \mathbf{v}$  are  $b(DT_{\mathcal{L}}(\mathbf{p}) + 1)$ -neighbors then
478   |   |   |   |   |   if  $DT_{\mathcal{C}}(\mathbf{p} + \mathbf{v}) > DT_{\mathcal{C}}(\mathbf{p}) + \omega_{\mathbf{v}}$  then
479   |   |   |   |   |   |    $DT_{\mathcal{C}}(\mathbf{p} + \mathbf{v}) \leftarrow DT_{\mathcal{C}}(\mathbf{p}) + \omega_{\mathbf{v}};$ 
480   |   |   |   |   |   |    $DT_{\mathcal{L}}(\mathbf{p} + \mathbf{v}) \leftarrow DT_{\mathcal{L}}(\mathbf{p}) + 1;$ 
481   |   |   |   |   |   |   end
482   |   |   |   |   |   end
483   |   |   |   |   end
484   |   |   |   end
485   |   |   end
486   |   end

```

487 **Theorem 3.** *If*

- 488 • each of the sets $\mathcal{N}^1, \mathcal{N}^2, \dots, \mathcal{N}^8$ is represented by at least one scanning
- 489 mask and
- 490 • the scanning masks support the scanning orders,
- 491
- 492
- 493

494 then Algorithm 2 computes correct distance maps.

495 *Proof.* Any distance propagating path between any pair of grid points in \mathcal{I}
 496 is also in \mathcal{I} by Lemma 5. Since the scanning masks support the scanning
 497 orders, there is, by Lemma 6, a distance propagating path that is propagated
 498 by the scanning masks. \square

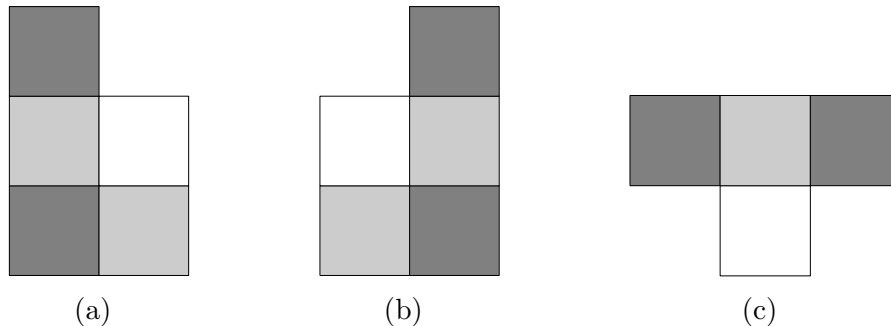
499 **Corollary 1.** *Algorithm 2 with, e.g., the masks*

$$501 \quad \mathcal{M}^1 = \{(-1, 1), (-1, 0), (-1, -1), (0, -1)\},$$

$$502 \quad \mathcal{M}^2 = \{(0, -1), (1, -1), (1, 0), (1, 1)\}, \text{ and}$$

$$503 \quad \mathcal{M}^3 = \{(-1, 1), (0, 1), (1, 1)\},$$

504
 505 see Figure 4, gives correct distance transforms.



517 Figure 4: Masks that can be used with Algorithm 2. The white pixel is the center of the
 518 mask and the two grey levels correspond to the elements in \mathcal{N}_1 and \mathcal{N}_2 , respectively.

519 4. Computing the distance transform using a look-up table

520
 521 The look-up table $LUT_{\mathbf{v}}(k)$ gives the value to be propagated in the di-
 522 rection \mathbf{v} from a grid point with distance value k . We will see that by using
 523 this approach, the additional distance transform $DT_{\mathcal{L}}$ is *not* needed for com-
 524 puting $DT_{\mathcal{L}}$. Thus, we get an efficient algorithm in this way. In this section,
 525 we assume that integer weights are used.

526
 527 The LUT-based approach to compute the distance transform first ap-
 528 peared in [15]. The distance function considered in [15] uses neighborhood
 529 sequences, but is non-symmetric. The non-symmetry allows to compute the
 530 DT in one scan. The same LUT-based approach is used for binary mathe-
 531 matical morphology with convex structuring elements in [16]. This approach
 532

is efficient for, e.g., binary erosion in one scan with a computational per-pixel cost independent of the size of the structuring element.

For the algorithm in [15, 16] the following formula is used in one raster scan:

$$DT_c(\mathbf{p}) = \min_{\mathbf{v} \in \mathcal{N}} (LUT_{\mathbf{v}}(DT_c(\mathbf{p} + \mathbf{v}))),$$

where \mathcal{N} is a non-symmetric neighborhood. In this section, we will extend this approach and allow the (symmetric) weighted ns-distances by allowing more than one scan.

4.1. Construction of the look-up table

Given a distance value k , the look-up table at position k with subscript-vector \mathbf{v} , $LUT_{\mathbf{v}}(k)$, holds information about the *maximal* distance value that can be found in a distance map in direction \mathbf{v} . See Example 3 and 4.

Example 3. For a distance function on \mathbb{Z}^2 defined by $\alpha = 2$, $\beta = 3$ and $B = (1, 2)$, the LUT with $D_{\max} = 10$ is the following:

	j	<u>0</u>	1	<u>2</u>	3	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>
$\mathbf{v} \in \mathcal{N}_1$	$LUT_{\mathbf{v}}(j)$	2	3	4	5	6	7	8	9	10	11	12
$\mathbf{v} \in \mathcal{N}_2$	$LUT_{\mathbf{v}}(j)$	4	4	5	6	7	9	9	10	11	12	14

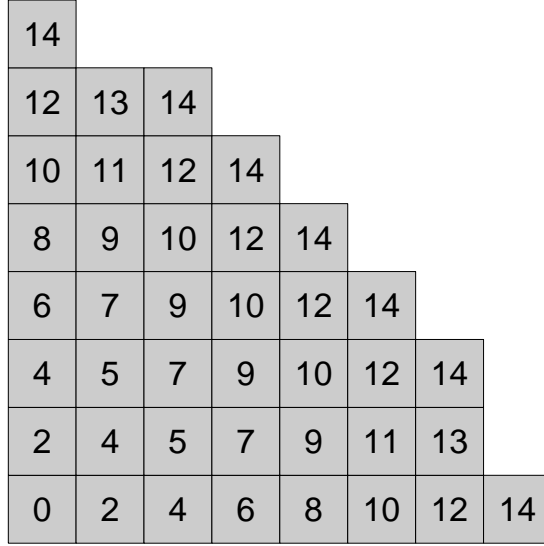
Only the values that are underlined are attained by the distance functions. See also Figure 5. The values in the look-up tables can be extracted from these DTs by, for each distance value 0 to 10, finding the corresponding maximal value in the subscript-direction.

Example 4. For a distance function on \mathbb{Z}^2 defined by $\alpha = 4$, $\beta = 5$ and $B = (1, 2, 1, 2, 2)$, the LUT (only showing values that are attained by the distance function) with $D_{\max} = 23$ is the following:

	j	0	4	8	9	12	13	16	17	18	20	21	22	23
$\mathbf{v} \in \mathcal{N}_1$	$LUT_{\mathbf{v}}(j)$	4	8	12	13	16	17	20	21	22	24	25	26	27
$\mathbf{v} \in \mathcal{N}_2$	$LUT_{\mathbf{v}}(j)$	8	9	13	17	17	18	21	22	23	25	26	27	31

The values in the LUT are given by the formula in Lemma 7.

571
572
573
574
575
576
577
578
579
580
581
582
583
584



585
586
587
588

Figure 5: Each pixel above is labeled with the distance to the pixel with value 0. The parameters $B = (1, 2)$, $(\alpha, \beta) = (2, 3)$ are used. See also Example 3.

589
590

Lemma 7. *Let α, β such that $0 < \alpha \leq \beta \leq 2\alpha$, the ns B , and the integer value k be given. Then*

591
592
593
594

$$\left\{ \begin{array}{l} \max_{\mathbf{v} \in \mathcal{N}_1} \\ \mathbf{p}: d_{\alpha, \beta}(\mathbf{0}, \mathbf{p}; B) = k \end{array} \right\} (d_{\alpha, \beta}(\mathbf{0}, \mathbf{p} + \mathbf{v}; B) - k) = \alpha \text{ and}$$

595
596
597
598
599

$$\left\{ \begin{array}{l} \max_{\mathbf{v} \in \mathcal{N}_2} \\ \mathbf{p}: d_{\alpha, \beta}(\mathbf{0}, \mathbf{p}; B) = k \end{array} \right\} (d_{\alpha, \beta}(\mathbf{0}, \mathbf{p} + \mathbf{v}; B) - k) = \begin{cases} 2\alpha & \text{if } \exists n : b(n+1) = 1 \\ & \text{and } k = \mathbf{1}_B^n \alpha + \mathbf{2}_B^n \beta \\ \beta & \text{else.} \end{cases}$$

600
601
602
603

Proof. When \mathbf{v} is a 1-step, then the maximum difference between $d_{\alpha, \beta}(\mathbf{0}, \mathbf{p} + \mathbf{v}; B)$ and $d_{\alpha, \beta}(\mathbf{0}, \mathbf{p}; B)$ is α by definition. There is a local step $\mathbf{v} \in \mathcal{N}_1$ that increases the length of the minimal cost B -path (for any B) by 1, so the maximum difference α is always attained.

604
605
606
607
608

When \mathbf{v} is a strict 2-step, $\mathbf{v} \in \mathcal{N}_2$ is the sum of two local steps from \mathcal{N}_1 . Intuitively, if there are “enough” 2s in B , then the maximum difference is β . Otherwise, two 1-steps are used and the maximum difference is 2α . To prove this, let $\mathcal{P}_{\mathbf{0}, \mathbf{p}} = \langle \mathbf{0} = \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n = \mathbf{p} \rangle$ be a minimal cost B -path and let $\mathbf{p} = (x, y)$ be such that $x \geq y \geq 0$. We have the following conditions on B :

609 (i) $b(n + 1) = 1$ and

610
611 (ii) $d_{\alpha,\beta}(\mathbf{0}, \mathbf{p}; B) = \mathbf{1}_B^n \alpha + \mathbf{2}_B^n \beta$.

612 We note that (i) implies that $\mathcal{P}_{\mathbf{0},\mathbf{p}} \cdot \langle \mathbf{p} + \mathbf{w} \rangle$ is a B -path iff $\mathbf{w} \in \mathcal{N}_1$ and a
613 minimal cost B -path if \mathbf{w} is either $(1, 0)$ or $(0, 1)$. Also, (ii) implies that the
614 number of 2:s in B up to position n equals the number 2-steps in $\mathcal{P}_{\mathbf{0},\mathbf{p}}$.

615 If both (i) *and* (ii) are fulfilled, since $b(n + 1) = 1$, the 2-step $\mathbf{v} = (1, 1)$
616 is divided into two 1-steps $(1, 0)$ and $(0, 1)$ giving a minimal cost B -path, so
617 $d_{\alpha,\beta}(\mathbf{0}, \mathbf{p} + \mathbf{v}; B) = d_{\alpha,\beta}(\mathbf{0}, \mathbf{p}; B) + 2\alpha$.

618 If (i) is not fulfilled, then there is a 2-step \mathbf{v} such that $\mathcal{P}_{\mathbf{0},\mathbf{p}} \cdot \langle \mathbf{p} + \mathbf{v} \rangle$ is a
619 minimal cost B -path of cost $d_{\alpha,\beta}(\mathbf{0}, \mathbf{p} + \mathbf{v}; B) = d_{\alpha,\beta}(\mathbf{0}, \mathbf{p}; B) + \beta$.

620 If (i), but not (ii) is fulfilled, then for any minimal cost B -path $\mathcal{Q}_{\mathbf{0},\mathbf{p}}$, we
621 have

$$622 \quad k = \mathbf{1}_{\mathcal{Q}_{\mathbf{0},\mathbf{p}}} \alpha + \mathbf{2}_{\mathcal{Q}_{\mathbf{0},\mathbf{p}}} \beta \neq \mathbf{1}_B^{\mathcal{L}(\mathcal{Q}_{\mathbf{0},\mathbf{p}})} \alpha + \mathbf{2}_B^{\mathcal{L}(\mathcal{Q}_{\mathbf{0},\mathbf{p}})} \beta.$$

623
624 It follows that $\mathbf{2}_{\mathcal{Q}_{\mathbf{0},\mathbf{p}}} < \mathbf{2}_B^{\mathcal{L}(\mathcal{Q}_{\mathbf{0},\mathbf{p}})}$. Therefore, there is a 1-step in $\mathcal{Q}_{\mathbf{0},\mathbf{p}}$ that
625 can be swapped with the 2-step \mathbf{v} giving a minimal cost B -path of cost
626 $d_{\alpha,\beta}(\mathbf{0}, \mathbf{p}; B) + \beta$. \square

627
628 The formula in Lemma 7 gives an efficient way to compute the look-up
629 table, see Algorithm 3. The algorithm gives a correct LUT by Lemma 7. The
630 output of Algorithm 3 for some parameters is shown in Example 3 and 4.

Algorithm 3: Computing the look-up table for weighted ns-distances.

Input: Neighborhoods $\mathcal{N}_1, \mathcal{N}_2$, weights α and β ($0 < \alpha \leq \beta \leq 2\alpha$), a ns B , and the largest distance value D_{\max} .

Output: The look-up table LUT .

```

647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667

```

for $k = 1 : D_{\max}$ **do**
 foreach $\mathbf{v} \in \mathcal{N}_1$ **do**
 | $LUT_{\mathbf{v}}(k) \leftarrow k + \alpha;$
 end
 foreach $\mathbf{v} \in \mathcal{N}_2$ **do**
 | $LUT_{\mathbf{v}}(k) \leftarrow k + \beta;$
 end
end
 $n \leftarrow 0;$
while $\mathbf{1}_B^n \alpha + \mathbf{2}_B^n \beta \leq D_{\max}$ **do**
 if $b(n+1) == 1$ **then**
 | $LUT_{\mathbf{v}}(\mathbf{1}_B^n \alpha + \mathbf{2}_B^n \beta) \leftarrow (\mathbf{1}_B^n + 2) \alpha + \mathbf{2}_B^n \beta;$
 end
 $n \leftarrow n + 1;$
end

Lemma 8 shows that the distance values are propagated correctly along distance propagating paths by using the look-up table.

Lemma 8. *Let $\mathcal{P}_{\mathbf{p}_0, \mathbf{p}_n} = \langle \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n \rangle$ be a distance propagating B -path. Then*

$$\mathcal{C}_{\alpha, \beta}(\langle \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{i+1} \rangle) = LUT_{\mathbf{p}_{i+1} - \mathbf{p}_i}(\mathcal{C}_{\alpha, \beta}(\langle \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_i \rangle)) \forall i < n.$$

Proof. Assume that the lemma is false and let i be the minimal index such that

$$\mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{p}_0, \mathbf{p}_{i+1}}) \neq LUT_{\mathbf{p}_{i+1} - \mathbf{p}_i}(\mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{p}_0, \mathbf{p}_i})).$$

Then there is a path $\mathcal{Q}_{\mathbf{q}_0, \mathbf{q}_j} = \langle \mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_j \rangle$ such that

$$\mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{p}_0, \mathbf{p}_i}) = \mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}_0, \mathbf{q}_j}) \quad \text{and} \quad \mathcal{L}(\mathcal{P}_{\mathbf{p}_0, \mathbf{p}_i}) \neq \mathcal{L}(\mathcal{Q}_{\mathbf{q}_0, \mathbf{q}_j})$$

defining the value in the LUT, i.e.,

$$\mathcal{C}_{\alpha, \beta}(\mathcal{Q}_{\mathbf{q}_0, \mathbf{q}_j} \cdot \langle \mathbf{q}_j + \mathbf{v} \rangle) = LUT_{\mathbf{v}}(\mathcal{C}_{\alpha, \beta}(\mathcal{P}_{\mathbf{p}_0, \mathbf{p}_i})),$$

where $\mathbf{v} = \mathbf{p}_{i+1} - \mathbf{p}_i$. Since the LUT stores the maximal local distances that are attained,

$$\mathcal{C}_{\alpha,\beta} (\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j} \cdot \langle \mathbf{q}_j + \mathbf{v} \rangle) > \mathcal{C}_{\alpha,\beta} (\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_{i+1}}).$$

It follows from Lemma 7 that \mathbf{v} is a strict 2-step and that $LUT_{\mathbf{v}} (\mathcal{C}_{\alpha,\beta} (\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i})) = 2\alpha$ and

$$\mathcal{C}_{\alpha,\beta} (\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_{i+1}}) - \mathcal{C}_{\alpha,\beta} (\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}) = \beta.$$

Since $\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}$ is a distance propagating B -path and \mathbf{v} is a strict 2-step,

$$\mathbf{2}_B^{\mathcal{L}(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i})} = \mathbf{2}_{\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}} \quad (3)$$

and $\mathbf{2}_B^{\mathcal{L}(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_{i+1}})} = \mathbf{2}_{\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_{i+1}}}$.

case i $\mathcal{L}(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}) > \mathcal{L}(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j})$

It follows that $\mathbf{2}_{\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}} < \mathbf{2}_{\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}} \leq \mathbf{2}_B^{\mathcal{L}(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j})} \leq \mathbf{2}_B^{\mathcal{L}(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i})}$ which contradicts (3).

case ii $\mathcal{L}(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}) < \mathcal{L}(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j})$

This implies that $\mathbf{2}_{\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}} > \mathbf{2}_{\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}}$. Then $\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j} \cdot \langle \mathbf{q}_j + \mathbf{v} \rangle$ is not a distance propagating path (there are more elements 2 in B than 2-steps in the path). It follows from Lemma 7 that there is a distance propagating path from \mathbf{q}_0 to $\mathbf{q}_j + \mathbf{v}$ of cost $\mathcal{C}_{\alpha,\beta} (\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}) + \beta$. Since $\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}$ is arbitrary, it follows that $LUT_{\mathbf{v}} (\mathcal{C}_{\alpha,\beta} (\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i})) = LUT_{\mathbf{v}} (\mathcal{C}_{\alpha,\beta} (\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j})) = \beta$. Contradiction. \square

4.2. Algorithms for computing the DT using look-up tables

In this section, we give algorithms that can be used to compute the distance transform using the LUT-approach. By Lemma 8, distance values are propagated correctly along distance propagating paths, so Algorithm 4 produces correct distance maps.

723 **Algorithm 4:** Computing $DT_{\mathcal{L}}$ for weighted ns-distances by wave-front
724 propagation using a look-up table.

725 **Input:** LUT and an object $X \subset \mathbb{Z}^2$.

726 **Output:** The distance transform $DT_{\mathcal{L}}$.

727 **Initialization:** Set $DT_{\mathcal{L}}(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and
728 $DT_{\mathcal{L}}(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$. For all grid points $\mathbf{p} \in \overline{X}$
729 adjacent to X : push $(\mathbf{p}, DT_{\mathcal{L}}(\mathbf{p}))$ to the list L of ordered pairs sorted
730 by increasing $DT_{\mathcal{L}}(\mathbf{p})$.

731 **while** L is not empty **do**
732 **foreach** \mathbf{p} in L with smallest $DT_{\mathcal{L}}(\mathbf{p})$ **do**
733 Pop $(\mathbf{p}, DT_{\mathcal{L}}(\mathbf{p}))$ from L ;
734 **foreach** $\mathbf{v} \in \mathcal{N}$ **do**
735 **if** $DT_{\mathcal{L}}(\mathbf{p} + \mathbf{v}) > LUT_{\mathbf{v}}(DT_{\mathcal{L}}(\mathbf{p}))$ **then**
736 $DT_{\mathcal{L}}(\mathbf{p} + \mathbf{v}) \leftarrow LUT_{\mathbf{v}}(DT_{\mathcal{L}}(\mathbf{p}))$;
737 Push $(\mathbf{p} + \mathbf{v}, DT_{\mathcal{L}}(\mathbf{p} + \mathbf{v}))$ to L ;
738 **end**
739 **end**
740 **end**
741 **end**

743 **Theorem 4.** *If*

- 744 • *each of the sets $\mathcal{N}^1, \mathcal{N}^2, \dots, \mathcal{N}^8$ is represented by at least one scanning*
745 *mask and*
- 746 • *the scanning masks support the scanning orders,*

747 *then Algorithm 5 computes correct distance maps.*

751 *Proof.* Since the same paths are propagated using this technique, the same
752 conditions on the masks, scanning orders, and image domain are needed
753 for Algorithm 5 to produce distance transforms without errors as when the
754 additional distance transform $DT_{\mathcal{L}}$ is used. \square

756 Note that Algorithm 4 and 5 derive from the work in [15, 16], but here,
757 symmetrical distance functions are allowed due to the increased number of
758 scans.

761 **Algorithm 5:** Computing DT_C for weighted ns-distances by raster scan-
762 ning using a look-up table.

763 **Input:** LUT, scanning masks \mathcal{M}^i , scanning orders so_i and an object
764 $X \subset \mathbb{Z}^2$.

765 **Output:** The distance transform DT_C .

766 **Initialization:** Set $DT_C(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and
767 $DT_C(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$.

768 **Comment:** The image domain \mathcal{I} defined by eq. 1 is scanned L
769 times using scanning orders such that the scanning masks \mathcal{M}^i
770 supports the scanning order so_i , $i \in \{1, \dots, L\}$

```

771 for  $i = 1 : L$  do
772     |   foreach  $\mathbf{p} \in \mathcal{I}$  following  $so_i$  do
773     |   |   if  $DT_C(\mathbf{p}) < \infty$  then
774     |   |   |   foreach  $\mathbf{v} \in \mathcal{N}^i$  do
775     |   |   |   |   if  $DT_C(\mathbf{p} + \mathbf{v}) > LUT_{\mathbf{v}}(DT_C(\mathbf{p}))$  then
776     |   |   |   |   |    $DT_C(\mathbf{p} + \mathbf{v}) \leftarrow LUT_{\mathbf{v}}(DT_C(\mathbf{p}))$ ;
777     |   |   |   |   |   end
778     |   |   |   |   end
779     |   |   |   end
780     |   |   end
781     |   end
782     end

```

783 We remark that the computational cost of Algorithm 3 is linear with
784 respect to the maximal radius D_{\max} and that the LUT can be computed on
785 the fly when computing the DT. In other words, if it turns out during the
786 DT computation that the LUT is too short, it can be extended by using
787 Algorithm 3 with the modification that the loop variable starts from the
788 missing value. Note also that for short neighborhood sequences, the LUT can
789 sometimes be replaced by a modulo operator. For example, when $(\alpha, \beta) =$
790 $(4, 5)$ and $B = (1, 2)$, then by propagating distances to 2-neighbors only when
791 $DT_C(\mathbf{p})$ is not divisible by nine gives a very fast algorithm the computes a
792 DT with low rotational dependency.

794 5. Computing the distance transform in two scans using a large 795 mask

797 In [6], it is proved that if the weights and neighborhood sequence are such
798 that the generated distance function is a metric, then the distance function

799 is generated by constant neighborhood using a (large) neighborhood. This
 800 implies that the 2-scan chamfer algorithm can be used to compute the DT
 801 in \mathbb{Z}^2 , see [12].

802 The following theorem is proved in [11]:

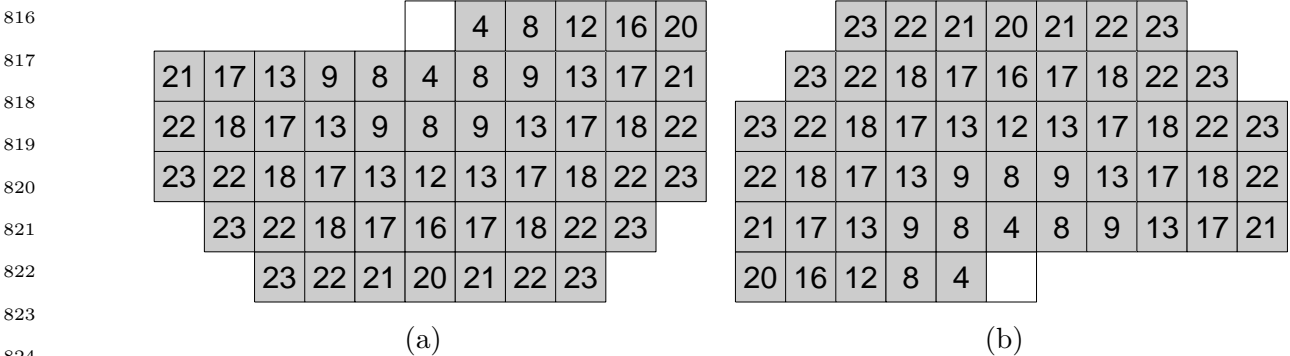
803 **Theorem 5.** *If*

$$805 \sum_{i=1}^N b(i) \leq \sum_{i=j}^{j+N-1} b(i) \quad \forall j, N \geq 1 \quad \text{and} \quad (4)$$

$$808 \quad 0 < \alpha \leq \beta \leq 2\alpha \quad (5)$$

809 then $d_{\alpha,\beta}(\cdot, \cdot; B)$ is a metric.

810 In [11], the distance function generated by $B = (1, 2, 1, 2, 2)$, $(\alpha, \beta) =$
 811 $(4, 5)$ is suggested. In Figure 6, the masks that can be used by a two-scan
 812 algorithm to compute the DT with this distance function are shown.



823 Figure 6: Masks that can be used by a two-scan algorithm to compute a DT using the
 824 weighted ns-distance defined by $B = (1, 2, 1, 2, 2)$, $(\alpha, \beta) = (4, 5)$.

825 In this section we assume that the weights α and β and the ns B are such
 826 that

- 827 • B is periodic and
- 828 • the distance function generated by α , β , and B is a metric.

829 For this family of distance functions, a two-scan chamfer algorithm with
 830 large scanning masks can be used instead of the three-scan algorithm with

837 small scanning masks using DT_{length} or the three-scan algorithm with small
 838 scanning mask using a LUT.

839 Let $\overline{\mathcal{N}}$ be the set of grid points such that the distance value from $\mathbf{0}$ is
 840 defined by the first period of B .

841 We now define two sets that are used in Algorithm 6.

$$842 \quad \mathcal{M}^1 = \overline{\mathcal{N}} \cap \{(x, y) : y < 0 \text{ or } y = 0 \text{ and } x \geq 0\} \text{ and}$$

$$843 \quad \mathcal{M}^2 = \overline{\mathcal{N}} \cap \{(x, y) : y > 0 \text{ or } y = 0 \text{ and } x \leq 0\}.$$

844
 845
 846 The following theorem is proved in [6]:

847 **Theorem 6.** *If $d_{\alpha,\beta}(\cdot, \cdot; B)$ is a metric, then the weighted ns-distance defined*
 848 *by B and (α, β) defines the same distance function as the weighted distance*
 849 *defined by the weighted vectors*

$$850 \quad \left\{ \left(\mathbf{v}, d_{\alpha,\beta}(\mathbf{0}, \mathbf{0} + \mathbf{v}; B) \right) : \mathbf{v} \in \overline{\mathcal{N}} \right\}$$

855 **Algorithm 6:** Computing $DT_{\mathcal{C}}$ for weighted ns-distances by wave-front
 856 propagation using a large weighted mask.

857 **Input:** The mask $\overline{\mathcal{N}}$, and an object $X \subset \mathbb{Z}^2$.

858 **Output:** The distance transform $DT_{\mathcal{C}}$.

859 **Initialization:** Set $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and
 860 $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$. For all grid points $\mathbf{p} \in \overline{X}$
 861 adjacent to X : push $(\mathbf{p}, DT_{\mathcal{C}}(\mathbf{p}))$ to the list L of ordered pairs sorted
 862 by increasing $DT_{\mathcal{C}}(\mathbf{p})$.

863 **while** L is not empty **do**

864 **foreach** \mathbf{p} in L with smallest $DT_{\mathcal{C}}(\mathbf{p})$ **do**

865 Pop $(\mathbf{p}, DT_{\mathcal{C}}(\mathbf{p}))$ from L ;

866 **foreach** $\mathbf{v} \in \overline{\mathcal{N}}$ **do**

867 **if** $DT_{\mathcal{C}}(\mathbf{p} + \mathbf{v}) > DT_{\mathcal{C}}(\mathbf{p}) + \omega_{\mathbf{v}}$ **then**

868 $DT_{\mathcal{C}}(\mathbf{p} + \mathbf{v}) \leftarrow DT_{\mathcal{C}}(\mathbf{p}) + \omega_{\mathbf{v}}$;

869 Push $(\mathbf{p} + \mathbf{v}, DT_{\mathcal{C}}(\mathbf{p} + \mathbf{v}))$ to L ;

870 **end**

871 **end**

872 **end**

873 **end**

875 **Algorithm 7:** Computing DT_C for weighted ns-distances by two scans
 876 using a large weighted mask.

877 **Input:** Scanning masks \mathcal{M}^i , scanning orders so_i , weights, and an ob-
 878 ject $X \subset \mathbb{Z}^2$.

879 **Output:** The distance transform DT_C .

880 **Initialization:** Set $DT_C(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and
 881 $DT_C(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$.

882 **Comment:** The image domain \mathcal{I} defined by eq. 1 is scanned two
 883 times using scanning orders such that the scanning mask defined by
 884 \mathcal{M}^i supports the scanning order so_i , $i \in \{1, \dots, 2\}$

```

885 for  $i = 1 : 2$  do
886     |
887     |   foreach  $\mathbf{p} \in \mathcal{I}$  following  $so_i$  do
888     |   |   foreach  $\mathbf{v} \in \mathcal{M}^i$  do
889     |   |   |   if  $DT_C(\mathbf{p} + \mathbf{v}) > DT_C(\mathbf{p}) + \omega_{\mathbf{v}}$  then
890     |   |   |   |    $DT_C(\mathbf{p} + \mathbf{v}) \leftarrow DT_C(\mathbf{p}) + \omega_{\mathbf{v}}$ ;
891     |   |   |   end
892     |   |   end
893     |   end
894 end
  
```

895 **Theorem 7.** *If the scanning masks \mathcal{M}^1 and \mathcal{M}^2 support the scanning orders*
 896 *then Algorithm 6 and 7 compute correct distance maps.*

898 *Proof.* Any path consists of steps from \overline{N} and the order of the steps is arbi-
 899 trary. Consider the point $\mathbf{p} = (x, y)$ such that $x \geq y \geq 0$. All points in any
 900 minimal cost path between $\mathbf{0}$ and \mathbf{p} have non-negative coordinates. Also, all
 901 local steps are in \mathcal{M}^2 except $(1, 0)$. Thus, the local steps in any minimal
 902 cost path between $\mathbf{0}$ and \mathbf{p} can be rearranged such that the steps from \mathcal{M}^1
 903 are first and the the steps from \mathcal{M}^2 are last or vice-versa. It follows that a
 904 minimal cost path is propagated from $\mathbf{0}$ to each point \mathbf{p} such that $x \geq y \geq 0$.
 905 The theorem holds by translation and rotation invariance. \square

907 6. Conclusions

908 We have examined the DT computation for weighted ns-distances. Three
 909 different, but related, algorithms have been presented and we have proved
 910 that the resulting DTs are correct.
 911

913 We have shown that using the additional transform $DT_{\mathcal{L}}$ is not needed
914 for computing the DT $DT_{\mathcal{L}}$. This extra information can, however, be useful
915 when extracting medial representations, see [14].

916 We note that when the LUT-approach is used, a fast and efficient al-
917 gorithm is obtained. This approach can also be used for computing the
918 *constrained* DT. When the constrained DT is computed, there are obstacle
919 grid points that are not allowed to intersect with the minimal cost paths that
920 define the distance values. The path-based approach is well-suited for such
921 algorithms. When the Euclidean distance is used, the corresponding algo-
922 rithm must keep track of *visible* point, i.e., points which can be given the
923 distance value by adding the length of the straight line segment between al-
924 ready visited points. Such algorithms, see [17], are slow and computationally
925 heavy compared to the distance functions used in this paper.

926 For short sequences, the LUT can be replaced by a modulo function:
927 consider $B = (1, 2)$ and weights (α, β) , then β is propagated to a two neighbor
928 only from grid points with distance values that are divisible by $\alpha + \beta$. This
929 approach gives a fast and efficient algorithm.

930 The LUT can be computed “on-the-fly” by using Algorithm 3. In other
931 words, if it turns out during the DT computation that the LUT is too short,
932 Algorithm 3 can be used to find the missing values in time that is proportional
933 to the number of added values.

934 For long sequences, the two-scan Algorithm 6 and 7 is not efficient since
935 the size of the masks depend on the length of the sequence. Also, this
936 approach is valid only for metric distance functions.

937 Due to the low rotational dependency and the efficient algorithms pre-
938 sented here, we expect that the weighted ns-distance has the potential of
939 being used in several image processing-applications where the DT is used:
940 matching [18], morphology [19], and more recent applications such as separat-
941 ing arteries and veins in 3-D pulmonary CT, [20] and traffic sign recognition,
942 [21].

944 References

- 945
- 946 [1] A. Rosenfeld, J. L. Pfaltz, Sequential operations in digital picture pro-
947 cessing, *Journal of the ACM* 13 (4) (1966) 471–494.
- 948
- 949 [2] U. Montanari, A method for obtaining skeletons using a quasi-Euclidean
950 distance, *Journal of the ACM* 15 (4) (1968) 600–624.

- 951 [3] G. Borgefors, Distance transformations in digital images, *Computer Vi-*
952 *sion, Graphics, and Image Processing* 34 (1986) 344–371.
- 953 [4] G. Borgefors, On digital distance transforms in three dimensions, *Com-*
954 *puter Vision and Image Understanding* 64 (3) (1996) 368–376.
- 955 [5] A. Rosenfeld, J. L. Pfaltz, Distance functions on digital pictures, *Pattern*
956 *Recognition* 1 (1968) 33–61.
- 957 [6] M. Yamashita, T. Ibaraki, Distances defined by neighbourhood se-
- 958 quences, *Pattern Recognition* 19 (3) (1986) 237–246.
- 959 [7] B. Nagy, Metric and non-metric distances on \mathbb{Z}^n by generalized neigh-
- 960 bourhood sequences, in: *Proceedings of 4th International Symposium on*
961 *Image and Signal Processing and Analysis (ISPA 2005)*, Zagreb, Croatia,
962 2005, pp. 215–220.
- 963 [8] B. Nagy, Distances with neighbourhood sequences in cubic and triangu-
- 964 lar grids, *Pattern Recognition Letters* 28 (1) (2007) 99–109.
- 965 [9] C. Fouard, R. Strand, G. Borgefors, Weighted distance transforms gen-
- 966 eralized to modules and their computation on point lattices, *Pattern*
967 *Recognition* 40 (9) (2007) 2453–2474.
- 968 [10] R. Strand, B. Nagy, C. Fouard, G. Borgefors, Generating distance maps
- 969 with neighbourhood sequences, in: *Proceedings of 13th International*
970 *Conference on Discrete Geometry for Computer Imagery (DGCI 2006)*,
971 Szeged, Hungary, Vol. 4245 of *Lecture Notes in Computer Science*,
972 Springer-Verlag, 2006, pp. 295–307.
- 973 [11] R. Strand, Weighted distances based on neighbourhood sequences, *Pat-*
974 *tern Recognition Letters* 28 (15) (2007) 2029–2036.
- 975 [12] A. Hajdu, L. Hajdu, R. Tijdeman, General neighborhood sequences in
- 976 \mathbb{Z}^n , *Discrete Applied Mathematics* 155 (2007) 2507–2522.
- 977 [13] R. Strand, B. Nagy, G. Borgefors, Digital distance functions on three-
- 978 dimensional grids, *Theoretical Computer Science* 412 (15) (2011) 1350
- 979 – 1363, *theoretical Computer Science Issues in Image Analysis and Pro-*
980 *cessing*.
- 981
- 982
- 983
- 984
- 985
- 986
- 987
- 988

- 989 [14] R. Strand, Shape representation with maximal path-points for path-
990 based distances, in: Proceedings of 5th International Symposium on Im-
991 age and Signal Processing and Analysis (ISPA 2007), Istanbul, Turkey,
992 2007, pp. 397–402.
- 993 [15] N. Normand, C. Viard-Gaudin, A background based adaptive page seg-
994 mentation algorithm, in: Proceedings of the 3rd International Conference
995 on Document Analysis and Recognition (ICDAR 1995), Washington,
996 DC, USA, Vol. 1, 1995, pp. 138–141.
- 997 [16] N. Normand, Convex structuring element decomposition for single scan
998 binary mathematical morphology, in: Proceedings of 11th International
999 Conference on Discrete Geometry for Computer Imagery (DGCI 2003),
1000 Naples, Italy, Vol. 2886 of Lecture Notes in Computer Science, Springer-
1001 Verlag, 2003, pp. 154–163.
- 1002 [17] D. Coeurjolly, S. Miguet, L. Tougne, 2D and 3D visibility in discrete ge-
1003 ometry: an application to discrete geodesic paths, Pattern Recognition
1004 Letters 25 (5) (2004) 561–570.
- 1005 [18] G. Borgefors, Hierarchical chamfer matching: A parametric edge match-
1006 ing algorithm, IEEE Transactions on Pattern Analysis and Machine In-
1007 telligence 10 (6) (1988) 849–865.
- 1008 [19] P. Nacken, Chamfer metrics in mathematical morphology, Journal of
1009 Mathematical Imaging and Vision 4 (1994) 233–253.
- 1010 [20] P. Saha, Z. Gao, S. Alford, M. Sonka, E. Hoffman, Topomorphologic
1011 separation of fused isointensity objects via multiscale opening: Sepa-
1012 rating arteries and veins in 3-D pulmonary CT, IEEE Transactions on
1013 Medical Imaging 29 (3) (2010) 840–851.
- 1014 [21] A. Ruta, Y. Li, X. Liu, Real-time traffic sign recognition from video by
1015 class-specific discriminative features, Pattern Recognition 43 (1) (2010)
1016 416 – 430.
- 1017
1018
1019
1020
1021
1022
1023
1024
1025
1026