Distance Transform Based on Weight Sequences

Benedek Nagy[1](✉), Robin Strand[2], and Nicolas Normand[3]

[1] Eastern Mediterranean University,
Famagusta, North Cyprus, via Mersin-10, Turkey
nbenedek.inf@gmail.com

[2] Centre for Image Analysis, Division of Visual Information and Interaction,
Uppsala University, Uppsala, Sweden
Robin.Strand@it.uu.se

[3] Université de Nantes, LS2N UMR CNRS 6004, Nantes, France
Nicolas.Normand@polytech.univ-nantes.fr

Abstract. There is a continuous effort to develop the theory and methods for computing digital distance functions, and to lower the rotational dependency of distance functions. Working on the digital space, e.g., on the square grid, *digital* distance functions are defined by minimal cost-paths, which can be processed (back-tracked etc.) without any errors or approximations. Recently, digital distance functions defined by weight sequences, which is a concept allowing multiple types of weighted steps combined with neighborhood sequences, were developed. With appropriate weight sequences, the distance between points on the perimeter of a square and the center of the square (i.e., for squares of a given size the weight sequence can be easily computed) are exactly the Euclidean distance for these distances based on weight sequences. However, distances based on weight sequences may not fulfill the triangular inequality. In this paper, continuing the research, we provide a sufficient condition for weight sequences to provide metric distance. Further, we present an algorithm to compute the distance transform based on these distances. Optimization results are also shown for the approximation of the Euclidean distance inside the given square.

Keywords: Digital distances · Weight sequences · Distance transforms · Neighborhood sequences · Chamfer distances · Combined distances · Approximation of the Euclidean distance

1 Introduction

In the digital geometry setting, it is very natural to define distance functions by minimal cost paths. Recently a new digital (i.e., path based) distance function has been investigated which provides the perfect Euclidean distance (from the Origin) for all the points having the same fixed number of steps in the corresponding shortest path from the Origin [11,12]. This new distance function,

M. Couprie et al. (Eds.): DGCI 2019, LNCS 11414, pp. 62–74, 2019.
https://doi.org/10.1007/978-3-030-14085-4_6

which is called distance by weight sequence, is obtained as a kind of mixture of the two well known digital distance approaches, namely of the distances based on neighborhood sequences [5,8,10,19] and of the chamfer distances, also known as weighted distances [1].

Discrete and digital distances may have some non-usual properties, e.g., they may not be metric. A function that is positive definite, symmetric and for which the triangular inequality holds is called a metric. The concept of metricity is an important property of distance functions. In our previous paper [12] we presented a very restricted sufficient condition for a weight sequence distance to be a metric. In Sect. 3 in this paper, we present a less restrictive condition for metricity.

Digital distances are well applicable in digital image processing. Given a distance function, a *distance transform* is a transform where each element in a set is assigned the distance to the closest (as given by the distance function) element in a complementary set. The result of a distance transform is called a distance map. This tool is often used in image processing and computer graphics [6]. In Sect. 4, we present a distance transform algorithm, which utilizes an auxiliary data structure to keep track of the number of steps of the minimal cost path at each point.

One of the 'goodness' measures of digital distances used in practice is related to their rotational dependency and how well they approximate the Euclidean distance. The most basic path-based distance functions, such as the city block and chessboard distances, have very high rotational dependency. In our previous paper [12] (which builds on [17]), we presented a formula for weight sequences that gives the exact Euclidean distance values on the border of a square, for arbitrary size of the square. This property is independent of the order of the weights in the weight sequence, and permutation optimization of the weights by a genetic algorithm is presented in Sect. 5.

2 Definitions, Preliminaries

In this section, first, we recall some basic concepts and fix our notations.

We denote the set of integers by $\mathbb{Z}$ and the set of non-negative integers by $\mathbb{N}$. Consequently the points of the digital two dimensional (square) grid are represented by $\mathbb{Z}^2$. The set of positive real numbers is denoted by $\mathbb{R}^+$.

In this paper, we consider grid points with integer coordinates. Of course, in image processing, each grid point is associated with a picture element, pixel. In a city block (resp. chessboard) distance, distinct points with unit difference in at most one (resp. two) of the coordinates have unit distance. Here, we use the notion of 1- and 2-neighbors in the following sense: Two grid points $P_1 = (x_1, y_1), P_2 = (x_2, y_2) \in \mathbb{Z}^2$ are ρ-neighbors, $\rho \in \{1, 2\}$, if

$$\begin{aligned} |x_1 - x_2| + |y_1 - y_2| \le \rho \text{ and} \\ \max\{|x_1 - x_2|, |y_1 - y_2|\} = 1. \end{aligned} \tag{1}$$

The points are *strict* ρ-neighbors if the equality in (1) is attained. Two points are *adjacent* if they are ρ-neighbors for some $\rho \in \{1, 2\}$.

We also recall the notion of path. A *path* in a grid is a sequence of adjacent grid points. A path $\Pi = (P_0, P_1, \ldots, P_n)$ is a path of n steps where for all $i \in \{1, 2, \ldots, n\}$, P_{i-1} and P_i are adjacent. The path Π connects P_0 and P_n. We may also say that the path starts from P_0 and arrives to P_n, i.e., they are its start point and endpoint, respectively.

In this paper, we use the general concept of weight (or chamfer) sequences (from [11,12]), instead of neighborhood sequences [5,8–10] or chamfer distances [1]. This general description is as follows:

Definition 1 (weight-sequence distance). *Let $m \in \mathbb{N}$, $m \geq 0$ be the number of the used weights, and let $S = \{1, \infty\} \cup \{w_3, \ldots, w_m\}$ be the weight set including 1, the sign ∞ and the used weights ($w_i \in \mathbb{R}$, $w_i > 1$ for all $i \in \{3, \ldots, m\}$).*

A weight sequence is $W = (c(i))_{i=1}^{\infty}$, where $c(i) \in S$, for all $i \in \mathbb{N}$.

Let $P, Q \in \mathbb{Z}^2$, then the weight of the path $\Pi = (P = P_0, P_1, \ldots, P_m = Q)$ is the sum of the weights of its steps, where the weight of the j-th step is specified as
$$\begin{cases} c(j), & \text{if the } j\text{-th step is a step to a strict 2-neighbor;} \\ 1, & \text{otherwise.} \end{cases}$$

The W-distance $d(P, Q; W)$ of P and Q, is then, defined as the weight of the minimal weighted path connecting P and Q.

In fact, all city block paths are valid for $d(P, Q; W)$ and have the same weight as in the city block distance. This implies that $d(P, Q; W)$ is upper-bounded by the city block distance. When the ∞ sign used in W for some step i ($c(i) = \infty$), it denotes an arbitrary large weight that prevents paths with a strict 2-neighbor at step i to be minimal. When computing a W-distance one can vary the features of neighborhood sequences and chamfer distances: the cost of a move to a 1-neighbor is 1 in every step, and the cost of a move to a strict 2-neighbor (diagonal step) is given by the actual element $c(j)$ of the weight sequence.

In [12] it was shown that for any weight sequence W, if Π is a minimal W-path between $P, Q \in \mathbb{Z}^2$, then it does not contain any steps to strict 2-neighbors by any weight $w_i > 2$.

The way the set S is defined, m different neighborhood relations (possible steps by various weights) are allowed in our paths:

- a traditional 1-step is a step between 1-neighbors with unit weights, the sign ∞ denotes these steps in W (practically, strict 2-steps are not allowed);
- a traditional 2-step is a step between 2-neighbors with unit weights, they are denoted by value 1 in W;

and if $m > 2$, then, by the used weights $w_3, \ldots, w_m$ the further steps are as follows:

- weighted 2-steps: the steps between 1-neighbors with unit weights, and between strict 2-neighbors with weight w_k (where $3 \leq k \leq m$) with $c(i) = w_k$ for some i in W.

Note that the weight sequence W can contain m values, "weights", according to a predefined set S, i.e. $W = (c(i))_{i=1}^{\infty}$ where $c(i) \in S$.

When written in a formal way, the sum of the weights along the path is:

$$\sum_{i=1}^{n} \delta_i, \text{ where } \delta_i = \begin{cases} c(i), & \text{if } P_{i-1} \text{ and } P_i \text{ are strict 2-neighbors;} \\ 1, & \text{otherwise.} \end{cases}$$

When the weight sequence W is fixed, the term W-path is used for paths having finite cost as defined by the weight sequence W.

We recall that greedy algorithms cannot be used to provide shortest paths. If a smaller weight appears after a larger weight in W, it may be needed in the shortest path instead of the previous larger one, depending on both the weight sequence and on the difference of the coordinate values of the points.

Now we recall the formula for computing the distance between any two grid points. The formula is used for finding optimal parameters in Sect. 5. Before the theorem we define a technical notation which will also be helpful later on.

Definition 2. *Let a weight sequence $W = (c(i))_{i=1}^{\infty}$ (based on a weight set $S = \{1, \infty, w_3, \ldots, w_m\}$) be given. Let $m, n \in \mathbb{N}$ such that $n \geq m$. Then $I(n, m)$ contains the indices of the smallest m weight values among the first n elements of W, i.e., among $(c(1), \ldots, c(n))$.*

Theorem 1. *Let the weight sequence $W = (c(i))_{i=1}^{\infty}$ (based on a weight set $S = \{1, \infty, w_3, \ldots, w_m\}$) and the point $P(x, y) \in \mathbb{Z}^2$, where $x \geq y \geq 0$, be given. Then the W-distance of P from the Origin $\mathbf{0}$ is given by*

$$d(\mathbf{0}, (x, y); W) = \min_{f \in \{0..y\}} \left\{ x + y - 2f + \sum_{i \in I(x+y-f, f)} c(i) \right\}. \tag{2}$$

Since the roles of the x- and y-coordinates are similar, and our distance function is translation invariant, one can compute the W-distance of any pair of points of $\mathbb{Z}^2$ by the previous formula.

Remark 1. As a special case, the weight sequence $W = (1)_{i=1}^{\infty}$ defines the chessboard distance, since steps to 2-neighbors are always allowed with unit cost. The weight sequence $W = (2)_{i=1}^{\infty}$ defines the city block distance, since, in this case, in any path one needs to pay the weight of two city block steps for a diagonal steps if the path contains any diagonal steps. Equivalently, minimal cost paths can be obtained using only steps to 1-neighbors.

3 Metricity Properties

One important property of digital distances is that of metricity. A metric is a distance function which satisfies the metric properties: positive definiteness, symmetry, and the triangular inequality. Formally:

A distance function d is a metric if it satisfies the following three properties:

- $d(P,Q) \geq 0$ for any point pair P,Q, moreover $d(P,Q) = 0$ if and only if $P = Q$. (positive definiteness)
- $d(P,Q) = d(Q,P)$ for any point pair P,Q. (symmetry)
- $d(P,Q) + d(Q,R) \geq d(P,R)$ for any three points P,Q,R. (triangular inequality)

It is easy to find a weight sequence such that the distance generated by the sequence does not satisfy the metric conditions. The first two properties, namely, the positive definiteness and the symmetry are always satisfied for distances defined by weight sequences on the square grid (it is clear by the definitions and the formula 2 shown in the previous section). However, the triangular inequality is problematic in some cases.

In [12] it was proven that if the weight sequence W contains the weights in a non-increasing order, then the generated distance function is a metric. However, with this restriction the number of the possible weight sequences is very limited. Now, we show a more general sufficient condition for a weight sequence to define a metric. First, we define the shifted sequences of a weight sequence (in a kind of analogous way as similar concept was defined and used for the neighborhood sequences in, e.g., [8]).

Definition 3. *The k-shifted sequence of a weight sequence $W = (c(i))_{i=1}^{\infty}$ is the sequence that obtained from W by starting from its k-th element, i.e., $W^k = (c(i))_{i=k}^{\infty}$.*

It is easy to see that $W = W^1$, while for $k > 1$, W^k is obtained from W by erasing its first $k-1$ elements.

Theorem 2. *Let the weight sequence $W = (c(i))_{i=1}^{\infty}$ be given. If W has the property that for every $n, m, k \in \mathbb{N}$ $(m \leq n)$*

$$\sum_{i \in I(n,m)} c(i) \geq \sum_{i \in I^k(n,m)} c(i) \tag{3}$$

(where, according to Definition 2, $I(n,m)$ and $I^k(n,m)$ are the sets containing the indices of the m smallest weights/elements among the first n elements of W and W^k, respectively, thus $|I(n,m)| = |I^k(n,m)| = m$ is their cardinality), then W defines a metric.

Proof. It is clear that all W-distances are symmetric and positive definite, thus we work only with the triangular inequality. Let us assume, by contrary, that there are three points $P,Q,R \in \mathbb{Z}^2$ such that $d(P,Q;W) + d(Q,R;W) < d(P,R;W)$. Let the path $\Pi_{P,Q}$ be the path from P to Q with the least weight, i.e., the path which is defining the distance $d(P,Q;W)$. Further, let $\Pi_{Q,R}$ be the path from Q to R that defines the distance $d(Q, R; W)$. Then let us consider the path $\Pi_{P,R}$ from P to R which is exactly the concatenation of the path $\Pi_{P,Q}$ and the steps of the path $\Pi_{Q,R}$. We notice that since the weights may not be the same in the prefix of W and in its subsequence used from Q to R, it is generally, not true that the obtained path $\Pi_{P,R}$ has the sum of the weight $d(P, Q; W) + d(Q, R; W)$. However, if the condition of the theorem is fulfilled, let us choose the numbers n, m and k as follows. Let n be the number of the steps of $\Pi_{Q,R}$. Set m as the number of its diagonal steps, finally put k as 1 more than the number of the steps of the path $\Pi_{P,Q}$. Considering the steps of $\Pi_{Q,R}$, in the left hand side of the inequality $d(P, Q; W) + d(Q, R; W) < d(P, R; W)$ the W-distances are used, while, considering the right hand side, we have a W-path, such that its first part (between P and Q) has exactly the weight $d(P, Q; W)$, and its second part (from Q to R) has the weight that is based on the weight sequence W^k. Then, we state that, by the condition $\sum\limits_{i \in I(n,m)} c(i) \geq \sum\limits_{i \in I^k(n,m)} c(i)$, there is a path $\Pi'_{Q,R}$ such that its W^k weight is not more than $d(Q, R; W)$. However, in this case, concatenating $\Pi_{P,Q}$ and $\Pi'_{Q,R}$ we have a W-path from P to Q such that its weight is not more than $d(P, Q; W) + d(Q, R; W)$. That is contradicting to our indirect assumption, thus, the triangular inequality must hold, if the condition of the theorem is fulfilled by the sequence W.

What is remained to show, that if $\sum\limits_{i \in I(n,m)} c(i) \geq \sum\limits_{i \in I^k(n,m)} c(i)$, then there is a path $\Pi'_{Q,R}$ such that its W^k weight is not more than $d(Q, R; W)$. Let us assume, by contrary, that $d(Q, R; W^k) > d(Q, R; W)$, i.e., there exist no such path $\Pi'_{Q,R}$. Then applying the formula for the distance, actually, $d(Q, R; W) = n - m + \sum\limits_{i \in I(n,m)} c(i)$. By the condition of the theorem there is also a W^k-path with n steps between Q and R having m diagonal steps with cost $n - m + \sum\limits_{i \in I^k(n,m)} c(i)$. However, since $\sum\limits_{i \in I(n,m)} c(i) \geq \sum\limits_{i \in I^k(n,m)} c(i)$, the cost of this path is not more than $d(Q, R; W)$. Therefore, $d(P, Q; W^k)$ cannot be larger than $d(P, Q; W)$, path $\Pi'_{Q,R}$ must exist with the desired property. □

The following examples illustrate metricity of distance functions based on some specific weight sequences.

Example 1. Let the weight sequence contain additional weights $\{1.5, 1.8\}$ and consider the periodic weight sequence $W = (1.8, 1.5, 1.8, 1.5...)$. One can easily see that W fulfills the condition of Theorem 2, thus it defines a metric.

Example 2. Let the weight set be $\{1.5, 1.55, 1.6, 1.75\}$. Further, let $W = (1.75, 1.5, 1.6, 1.5, 1.55, 1.5, 1.5, 1.5, ...)$ with $c(i) = 1.5$ for all $i > 6$. Again, it is easy to check that the triangular inequality holds and that the given W-distance is a metric.

Notice that in both of our examples the weights in the sequence are not ordered in non increasing way, thus the condition presented in [12] cannot be applied to prove the metricity of the corresponding distances.

Finally, to show that the triangular inequality may fail in some cases, let us consider the next example:

Example 3. Let $W = (1.2, 1.5, 1, ...)$. Then the W-distances of the points $(0, 0)$, $(1, 1)$ and $(2, 2)$ are computed:

$$d((0,0),(1,1);W) = 1.2$$
$$d((1,1),(2,2);W) = 1.2$$
$$d((0,0),(2,2);W) = 1.2 + 1.5 = 2.7.$$

Therefore, the triangular inequality fails for this W-distance.

4 Distance Transform (DT)

The DT is a mapping from the image domain, a subset of the grid, to the range of the distance function. In a DT, each object grid point is assigned the distance from its closest background grid point. A modified version of a wavefront propagation algorithm can be used.

Now the formal definition of an image is given.

Definition 4. *The image domain is a finite subset of $\mathbb{Z}^2$ denoted by $\mathcal{I}$. We call the function $F : \mathcal{I} \longrightarrow \mathcal{R}_d$ an image, where $\mathcal{R}_d$ is the range of the distance function d.*

An *object* is a subset $X \subset \mathcal{I}$ and the *background* is $\overline{X} = \mathcal{I} \setminus X$. We assume that $X, \overline{X} \neq \emptyset$. We denote the distance map for path-based distances with DM_d, where the subscript d indicates what distance function is used.

Definition 5. *The* distance map DM_d *generated by the distance function $d(\cdot, \cdot; W)$ of an object $X \subset \mathcal{I}$ is the mapping*

$$\begin{aligned} DM_d : \mathcal{I} \rightarrow \mathcal{R}_d \textit{ defined by} \\ P \mapsto d\left(\overline{X}, P; W\right), \textit{ where} \\ d\left(\overline{X}, P; W\right) = \min_{Q \in \overline{X}} \left\{ d\left(Q, P; W\right) \right\}. \end{aligned}$$

In the case of W-distances with two weights, a minor modification of the Dijkstra algorithm (and with the same time complexity) can be used, see [18] and Theorem 4.1 in [15]. However, for multiple weights, this is not necessarily true. For W-distances, the used weights are also important and they are determined by the *number of steps* of the minimal cost-path (not by the cost). Therefore, in the general case of multiple weights presented here, we need to store this value also when propagating distance information. We define the auxiliary transform DM_s that holds the number of steps of the minimal cost path at each point, see Algorithm 1.

Note that we need to store not only the best distance values at the points, but the best values that are computed by various number of steps. Therefore, for each point P a set $S(P)$ of pairs of values of the form (DM_d, DM_s) are stored, i.e., the weight DM_d of the shortest path among the ones containing exactly DM_s number of steps, with pairwise different DM_s. After the run of the algorithm for each point the minimal DM_d gives the result.

Algorithm 1 shows how the distance map can be computed based on an extended optimal search (Dijkstra algorithm) and using the data structure described above. At the initialization the object points are the only points that are reached and it is done by 0 steps and with 0 cost. Then the border points of the object are listed in an increasing order by the minimal cost path already known for them. Actually every point in the priority queue has 0 cost, but the queue will be updated by involving other points to where paths are already found. The **while** loop chooses (one of the) point(s) with minimal cost from the queue since it is sure that we have the minimal cost path to this point already. Then in the loop the data of all adjacent points of the chosen point are updated by computing the cost of the new paths through the chosen point (having last step from the chosen point to the actual adjacent point). Therefore, the algorithm holds the optimal distance attained at each point (as the usual algorithm), but this is done *for each* path length from the shortest ones till the ones obtained by only steps to 1-neighbors. So, if there are paths of different lengths ending up at the same point, distance information for each of the different path lengths are stored.

The algorithm may look more complex than similar algorithms for the other distances and this is due to the fact that greedy shortest path algorithm does not work in the general case. However, by using the weights in non-decreasing order, the greedy shortest paths algorithms work, and thus, for every point Q of the grid one need to remember only for the value obtained with the path length when the point Q is pushed into the queue.

5 Approximation of the Euclidean Distance

In this section, we apply a genetic algorithm to permute a weight sequence that gives the exact Euclidean distance values on the border of a square. The comparison to the Euclidean distance is carried out with absolute and relative mean and maximum values of the absolute difference between the Euclidean and weight sequence distance.

The following Lemma, which is from our previous paper [12], will be used for defining the fitness function in the genetic algorithm optimization.

Algorithm 1. Computing DM for W-distances given by a weight sequence W.

```
Input: $W$ and an object $X \subset \mathbb{Z}^2$.
Output: The distance map $DM_d$.
Initialization: Let $S(P) \leftarrow \{(0,0)\}$ for grid points $P \in \overline{X}$. Let
$DM_d(P) = \min\{DM_d \mid (DM_d, DM_s) \in S(P)\}$. Initialize $L$, a priority queue of
points $P$ sorted by increasing $DM_d(P)$, with the set of grid points $P \in \overline{X}$
adjacent to $X$.
while L is not empty do
    Pop $P$ from $L$;
    foreach $Q \in \mathcal{I}$: $Q, P$ are strict 2-neighbors do
        foreach pair $(DM_d, DM_s) \in S(P)$ do
            if $c(DM_s + 1) \leq 2$ then
                if there is an element $(DM_d', DM_s + 1) \in S(Q)$ then
                    if $DM_d' > DM_d + c(DM_s + 1)$ then
                        Replace $(DM_d', DM_s + 1)$ by
                        $(DM_d + c(DM_s + 1), DM_s + 1)$ in $S(Q)$
                else
                    Add $(DM_d + c(DM_s + 1),\ DM_s + 1)$ to $S(Q)$
        if $DM_d(Q) > \min\{DM_d' \mid (DM_d', DM_s') \in S(Q)\}$ then
            Let $DM_d(Q) = \min\{DM_d' \mid (DM_d', DM_s') \in S(Q)\}$
            Insert $Q$ in the priority queue $L$ ordered by increasing $DM_d$
    foreach $Q \in \mathcal{I}$: $Q, P$ are 1-neighbors do
        foreach pair $(DM_d, DM_s) \in S(P)$ do
            if there is an element $(DM_d', DM_s + 1) \in S(Q)$ then
                if $DM_d' > DM_d + 1$ then
                    Replace $(DM_d', DM_s + 1)$ by $(DM_d + 1, DM_s + 1)$ in $S(Q)$
            else
                Add $(DM_d + 1, DM_s + 1)$ to $S(Q)$
        if $DM_d(Q) > \min\{DM_d' \mid (DM_d', DM_s') \in S(Q)\}$ then
            Let $DM_d(Q) = \min\{DM_d' \mid (DM_d', DM_s') \in S(Q)\}$
            Insert $Q$ in the priority queue $L$ ordered by increasing $DM_d$
```

Lemma 1. *If the weight sequence W is non-decreasing and all elements in W are smaller than or equal to 2, then the distance value in Eq. (2) is given by*

$$d(\mathbf{0}, (x,y); W) = x - y + \sum_{i=1}^{y} c(i)$$

The following Theorem, from [12], states that the W-distance with r different weights (as given in the Theorem) can give the exact Euclidean distance values on the border points of a square of side length $2r - 1$.

Theorem 3. *Given an integer $x > 0$, the Euclidean distance values from $(0,0)$ to each point of the set $\{(x,y) \in Z^2, 0 \leq y \leq x\}$ is given without errors by the* *weight sequence* $W = (c(i))_{i=1}^{\infty}$ *with* $c(i) = \left(1 + \sqrt{x^2 + i^2} - \sqrt{x^2 + (i-1)^2}\right)$ *for* $1 \leq i \leq x$.

Note that the distance values on the border of the square of side $2r - 1$ are independent of the order of the weights r weights in the sequence given in Theorem 3. In [12], we presented a greedy algorithm for ordering the weights in the weight sequence. Here, we extend the optimization by a global optimization, where the optimal permutation of the r weights are sought in order to minimize

- the absolute mean error,

$$\frac{2}{r \cdot (r+1)} \sum_{r \geq x \geq y \geq 0} |d_E(x,y) - d(x,y;W)| \tag{4}$$

- the relative mean error

$$\frac{2}{r \cdot (r+1)} \sum_{r \geq x \geq y \geq 0} \frac{|d_E(x,y) - d(x,y;W)|}{d_E(x,y)} \tag{5}$$

- the absolute maximum error,

$$\max_{r \geq x \geq y \geq 0} |d_E(x,y) - d(x,y;W)| \tag{6}$$

- the relative maximum error

$$\max_{r \geq x \geq y \geq 0} \frac{|d_E(x,y) - d(x,y;W)|}{d_E(x,y)} \tag{7}$$

In the experiments, a weight sequence of length 40 was calculated by Theorem 3. The weight sequence is $c(i) = \left(1 + \sqrt{40^2 + i^2} - \sqrt{40^2 + (i-1)^2}\right)$, $i = 1..40$. The weights were permuted by a genetic algorithm [7] with the fitness functions (4), (5), (6), and (7). The fitness functions are efficiently computed by Lemma 1. The *crossover* is a flip of a section (with random start and end point) of the weight sequence, the *mutation* swaps to random elements of the weight sequence. The optimization is run with a population size of 60 and the maximum number of generations was set to 1000.

The optimal permutations are found below and illustrations of the difference between the so-obtained distance functions and the Euclidean distance are found in Fig. 1.

The obtained optimal permutations are (the indices of the weights $c(i)$ are shown for brevity):

- the absolute mean error (4)
  36, 1, 32, 16, 22, 11, 23, 35, 7, 25, 10, 26, 12, 40, 4, 30, 18, 27, 5, 37, 14, 19, 38, 3, 28, 13, 24, 9, 29, 15, 39, 2, 31, 17, 33, 8, 20, 34, 6, 21
- the relative mean error (5)
  19, 7, 37, 28, 2, 32, 22, 16, 10, 31, 14, 27, 4, 39, 21, 12, 26, 6, 40, 15, 30, 3, 35, 17, 20, 8, 34, 13, 24, 29, 1, 36, 18, 11, 38, 9, 25, 23, 5, 33

– the absolute maximum error (6)
14, 37, 18, 6, 32, 26, 1, 21, 38, 16, 25, 3, 30, 27, 11, 15, 29, 8, 40, 10, 35, 12, 28, 17, 5, 24, 36, 2, 20, 23, 33, 4, 19, 39, 7, 34, 13, 31, 9, 22
– the relative maximum error (7)
19, 13, 36, 4, 40, 26, 1, 16, 21, 14, 27, 8, 25, 7, 34, 35, 15, 23, 29, 2, 31, 3, 18, 38, 20, 10, 12, 33, 24, 9, 37, 11, 28, 39, 5, 22, 30, 6, 32, 17

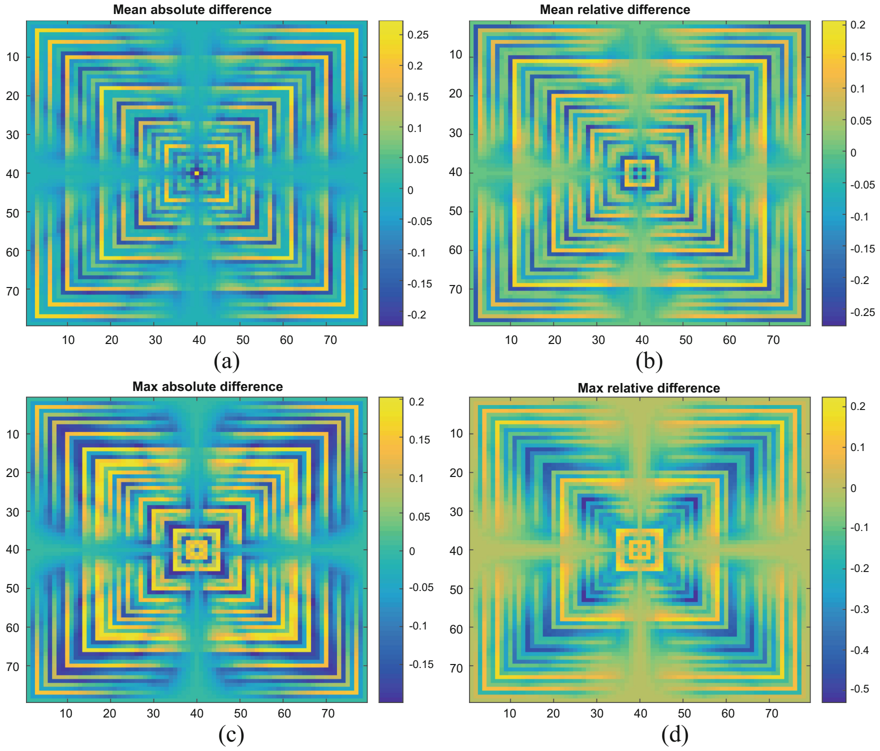

Fig. 1. Illustration of the point-wise difference between the Euclidean distance and the weight sequence distance with a permutation of the weights obtained by a genetic algorithm. The fitness functions used are Absolute mean difference (a), Relative mean difference (b), Absolute max difference (c), and Relative max difference (d). All distances are computed from a single central source point.

6 Conclusions and Discussion

This paper extends the work presented in [12], and further develops the framework for distances based on weight sequences, that is digital path-based distance functions defined as minimal cost paths on the square grid.

By the results in this paper, a digital and completely error-free distance function with very low rotational dependency, together with an algorithm that can be used to compute distance transforms, Algorithm 1, are presented. Efficient algorithms for computing distance transforms are necessary for real-world applications, where computation time often is a bottle-neck. Due to the very low rotational dependency the proposed digital distance function can be used instead of, for example, the Euclidean distance transform. The Euclidean distance transform can be computed efficiently by state-of-the-art methods, but Euclidean distance transform computation has some drawbacks: (i) the vector-propagation method inevitably produces errors for some pixel configurations [4,13], (ii) fast-marching methods are based on coarse approximations of the Eikonal equation, and thus give only approximate distance values [14], (iii) separable algorithms are not suited for constrained distance transform computation [2,3]. See also [15,16].

This paper presents a less restrictive condition for metricity, compared to what was presented in [12]. This less restrictive condition will be important when further developing the theory for these distance functions.

In addition, this paper presents an optimization procedure based on a genetic algorithm. Note that the genetic algorithm can be run with several different parameters, and the ones used in the manuscript are standard parameters with slight fine-tuning based on experimental evaluation. The optimization is not deterministic and there are no guarantees that the obtained local optimum is a global one, but the process was reasonably stable when executed several times. The low rotational dependency obtained by the optimization procedure is key in many real-world problems.

In our future work, we will analyze Algorithm 1 in depth, including both proof of correctness and space and time complexity. We also leave it as an open problem whether the condition of Theorem 2 is also a necessary condition to define metric distances.

The proposed distance function has great potential in many image processing algorithms where distance transforms with low rotational dependency is required.

References

1. Borgefors, G.: Distance transformations in digital images. Comput. Vis. Graph. Image Process. **34**, 344–371 (1986). https://doi.org/10.1016/S0734-189X(86)80047-0
2. Breu, H., Kirkpatrick, D., Werman, M.: Linear time Euclidean distance transform algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **17**(5), 529–533 (1995). https://doi.org/10.1109/34.391389
3. Coeurjolly, D., Vacavant, A.: Separable distance transformation and its applications. In: Brimkov, V., Barneva, R. (eds.) Digital Geometry Algorithms. Theoretical Foundations and Applications to Computational Imaging. LNCVB, vol. 2, pp. 189–214. Springer, Dordrecht (2012). https://doi.org/10.1007/978-94-007-4174-4_7
4. Danielsson, P.E.: Euclidean distance mapping. Comput. Graph. Image Process. **14**(3), 227–248 (2008). https://doi.org/10.1016/0146-664X(80)90054-4

5. Das, P.P., Chakrabarti, P.P.: Distance functions in digital geometry. Inf. Sci. **42**, 113–136 (1987). https://doi.org/10.1016/0020-0255(87)90019-3
6. Fabbri, R., Costa, L.D.F., Torelli, J.C., Bruno, O.M.: 2D Euclidean distance transform algorithms: a comparative survey. ACM Comput. Surv. **40**(1), 1–44 (2008). https://doi.org/10.1145/1322432.1322434
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley, Boston (1989)
8. Nagy, B.: Distance functions based on neighbourhood sequences. Publ. Math. Debrecen **63**(3), 483–493 (2003)
9. Nagy, B.: Metric and non-metric distances on $\mathbb{Z}^n$ by generalized neighbourhood sequences. In: IEEE Proceedings of 4^{th} International Symposium on Image and Signal Processing and Analysis (ISPA 2005), Zagreb, Croatia, pp. 215–220 (2005). https://doi.org/10.1109/ISPA.2005.195412
10. Nagy, B.: Distance with generalized neighbourhood sequences in nD and ∞D. Discrete Appl. Math. **156**(12), 2344–2351 (2008). https://doi.org/10.1016/j.dam.2007.10.017
11. Nagy, B., Strand, R., Normand, N.: A weight sequence distance function. In: Hendriks, C.L.L., Borgefors, G., Strand, R. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 292–301. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38294-9_25
12. Nagy, B., Strand, R., Normand, N.: Distance functions based on multiple types of weighted steps combined with neighborhood sequences. J. Math. Imaging Vis. **60**(8), 1209–1219 (2018). https://doi.org/10.1007/s1085
13. Ragnemalm, I.: The Euclidean distance transform in arbitrary dimensions. Pattern Recogn. Lett. **14**(11), 883–888 (1993). https://doi.org/10.1016/0167-8655(93)90152-4
14. Sethian, J.A.: Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, Cambridge (1999)
15. Strand, R.: Distance functions and image processing on point-lattices: with focus on the 3D face- and body-centered cubic grids. Ph.D. thesis, Uppsala University, Sweden (2008). http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-9312
16. Strand, R.: Weighted distances based on neighbourhood sequences. Pattern Recogn. Lett. **28**(15), 2029–2036 (2007). https://doi.org/10.1016/j.patrec.2007.05.016
17. Strand, R., Nagy, B.: A weighted neighborhood sequence distance function with three local steps. In: IEEE Proceedings of 8^{th} International Symposium on Image and Signal Processing and Analysis (ISPA 2011), Dubrovnik, Croatia, pp. 564–568 (2011)
18. Strand, R., Normand, N.: Distance transform computation for digital distance functions. Theor. Comput. Sci. **448**, 80–93 (2012). https://doi.org/10.1016/j.tcs.2012.05.010
19. Yamashita, M., Ibaraki, T.: Distances defined by neighborhood sequences. Pattern Recogn. **19**(3), 237–246 (1986). https://doi.org/10.1016/0031-3203(86)90014-2