Some Weighted Distance Transforms in Four Dimensions

Gunilla Borgefors

Centre for Image Analysis, Swedish University of Agricultural Sciences
Lägerhyddvägen 17, SE-752 37 Uppsala, Sweden
gunilla@cb.uu.se

Abstract. In a digital distance transform, each picture element in the shape (background) has a value measuring the distance to the background (shape). In a weighted distance transform, the distance between two points is defined by path consisting of a number of steps between neighbouring picture elements, where each type of possible step is given a length-value, or a weight. In 4D, using $3\times3\times3\times3$ neighbourhoods, there are four different weights. In this paper, optimal real and integer weights are computed for one type of 4D weighted distance transforms. The most useful integer transform is probably $\langle 3,4,5,6\rangle$, but there are a number of other ones listed. Two integer distance transform are illustrated by their associated balls.

1 Introduction

Results regarding the 4D digital space, $\mathbf{Z}^4$, are being found more and more in literature, both regarding theory and emerging applications. Examples where 4D is used are: when processing 3D grey-level images, just as some 2D problems are solved using temporary 3D images; for volume data sequences, as ultrasound volume images of a beating heart; or for the discretisation of the parameter space of a robot or robot arm. Some examples are [6,7,8,9].

In a Distance Transform (denoted DT), each element in the shape (background) has a value measuring the distance to the background (shape). DTs have proven to be an excellent tool for many different image operations. Therefore, distance transforms (DT) in 4D are moving from being a theoretical curiosity, [1], to becoming a useful tool.

The basic idea, utilised for most DTs, is to approximate the global Euclidean distance by propagation of local distances, i.e., distances between neighbouring pixels. This idea was probably first presented by Rosenfeld and Pfaltz in 1966, [10]. This approach is motivated by ease of computation. In sequential computation only a small area of the image is available at the same time. In massively parallel computation (if such an approach still exists) each pixel has access only to its immediate neighbours.

Weighted or chamfer distance transforms, denoted WDT. The local steps between neighbouring pixels are given different weights. In 2D, the most common WDTs are $\langle 2,3\rangle$ and $\langle 3,4\rangle$, were the first number is the local distance between

G. Borgefors, I. Nyström, and G. Sanniti di Baja (Eds.): DGCI 2000, LNCS 1953, pp. 325–336, 2000.

edge-neighbours and the second number is the local distance between point-neighbours, [2]. Weighted DTs can be computed in arbitrary dimensions by two raster scans through the image, where, at each point, the image values in a small neighbourhood of the point are used to compute the new point value [1,2]. A first, not very good, effort of discovering WDTs in higher dimensions is found in [1].

Important theoretical results on general DTs in higher dimensions have been published [8] a few years ago. In this paper, necessary conditions for an nD DT to be a metric are presented. In [3], WDTs in 3D, fulfilling these criteria were exhaustively investigated for $3 \times 3 \times 3$ neighbourhoods. There proved to be two types of such DTs, the “obvious” one and one less intuitive. In 4D, the situation is even more complex, and there are at least eight different cases of WDTs. In this paper, the most “natural” case will be investigated, and optimal real and integer weights for this case will be presented.

In Section 2, the geometry and general equations are developed. In Section 3, optimal weights are computed, where optimality is defined as minimising the maximum difference from the Euclidean distance in an $M \times M \times M \times M$ image. In Section 4, the optimal real and integer WDTs are listed and two integer DTs are illustrated by their associated balls.

2 Geometry and Equations

Denote a digital shape on a hypercubic grid F, and the complement of the shape $\overline{F}$, where the sets F and $\overline{F}$ are not necessarily connected. A distance transformation converts the binary image to a distance image, or Distance Transform (DT). In the DT each *hyxel* (hypervolume picture element) has a value measuring the distance to the closest hyxel in $\overline{F}$.

A good underlying concept for all digital distances is the one proposed by Yamashita and Ibaraki, [11]:

Definition 1 *The distance between two points* $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *is the length of the shortest path connecting* $\boldsymbol{x}$ *to* $\boldsymbol{y}$ *in an appropriate graph.*

They proved that any distance is definable in the above manner, by choosing an appropriate neighbourhood relation and an appropriate definition of path length.

In 4D hypercubic space, each hyxel has four types of neighbours: 8 volume neighbours, 24 face neighbours, 32 edge neighbours, and 16 point neighbours. A path between two hyxels in the 4D image can thus include steps in 80 directions, if only steps between immediate neighbours are allowed.

The $\mathrm{DT}(i, j, k, l)$ of a hyxel in F is the minimum length of a path connecting (i, j, k, l) to any hyxel in $\overline{F}$, where steps between volume neighbours have length a, steps between face neighbours have length b, steps between edge neighbours have length c, and steps between point-neighbours have length d, and no other steps are allowed. Due to symmetry, it is enough to consider distances from the origin to a hyxel (x, y, z, w), where $0 \leq w \leq z \leq y \leq x \leq M$ and M

is the maximal dimension of the image when computing optimal a, b, c, and d. The distance to be minimised then becomes $D(x, y, z, w)$. A $3 \times 3 \times 3 \times 3$ WDT will be denoted $\langle a, b, c, d \rangle$.

As the length of any minimal path is defined only by the numbers of steps of different types in it, the *order* of the steps is arbitrary. Therefore, we can always assume a minimal path where the steps are arranged in a number of straight line segments, equal to the number of different directions of steps used.

Not all combinations of local distances a, b, c, d result in useful distance transforms. The DT should have the following property.

Definition 2 *Consider two picture elements that can be connected by a straight line, i.e., by using only one type and direction of local step. If that line defines the distance between the pixels, i.e., is a minimal path, then the resulting DT is* **semi-regular**. *If there are no other minimal paths, then the DT is* **regular**.

From [8] we have the following result.

Theorem 1 *A distance transform in $\mathbf{Z}^n$ that is a metric is semi-regular. A semi-regular distance transform in $\mathbf{Z}^2$ is a metric.*

Thus, all suggested DTs should be semi-regular as this is a necessary but, in higher dimensions, not sufficient condition for being metrics.

As there are four types of steps, there are four types of straight paths possible in the hypercubic grid. To find the conditions for 4D regularity we must investigate all the ways these four straight paths can be approximated by paths using other steps and find the conditions for the straight path being shortest. The result is that a 4D WDT is semi-regular if the following inequalities hold (see [3] for a complete description of the method of computation):

$$a \leq b, \quad b \leq 2a, \quad b \leq c, \quad c \leq \frac{3}{2}b, \quad c \leq d, \quad d \leq \frac{4}{3}c. \tag{1}$$

These inequalities define a hyperpolyhedron in a, b, c, d-parameter space. A cut through this polyhedron at $d = 2$ and with a as a scale factor is shown in Fig. 1.

The conditions in (1) may seem restrictive, but they are *not* sufficient to determine unique expressions for the WDTs. If we compute the distances from the origin, choosing the shortest paths and assuming that the local distances have the properties in (1), we discover (at least) eight different, equally valid cases. For example, the hyxel $(2, 2, 1, 1)$ can be reached either as $(1, 1, 1, 1) + (1, 1, 0, 0) = d + b$ or as $(1, 1, 1, 0) + (1, 1, 0, 1) = 2c$. The inequalities in (1) do not determine which is the shorter path. In 3D there is the same phenomenon, but there are only two cases [3].

In Fig. 1, the eight cases discovered are marked by thin lines. In each of the Cases, expressions could be found for the distance transforms, and the local distances could be optimised. However, the area marked "Case I" is the most interesting and easiest to handle, as there is only one expression valid for all hyxels, and that expression, moreover, is the one we would expect, as it is an extensions of the equation in 2D and in Case I 3D. In the triangular hypercone

$$b \leq 2a, \quad c \leq d, \quad a + c \leq 2b, \quad b + d \leq 2c. \tag{2}$$

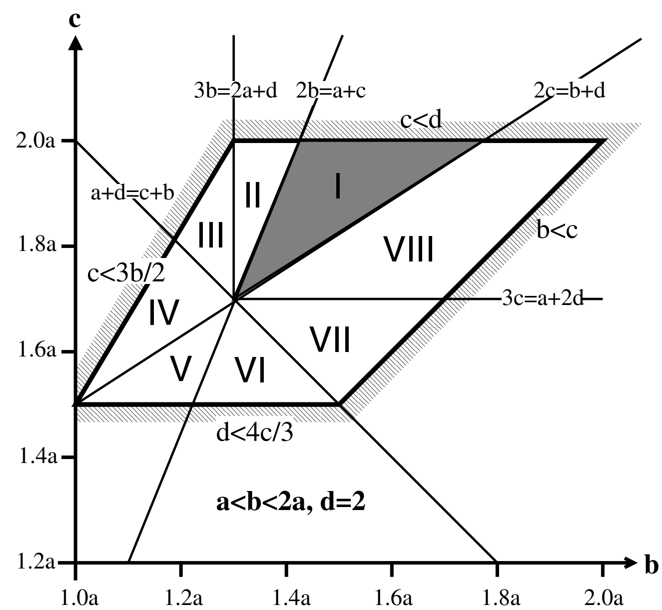

Fig. 1. The hyperpolyhedron in a, b, c, d-space that results in semi-regular 4D weighted distances (thick lines). The thin lines separates different cases. The grey area is Case I, which will be covered in this paper

the distance between the origin and (x, y, z, w) is

$$\text{Case I: } D = wd + (z-w)c + (y-z)b + (x-y)a, \ \text{ for } \ 0 \le w \le z \le y \le x \quad (3)$$

This equation, without the limitation of parameter space (2), is found in [1].

The 4D DT analogous to the chessboard DT in 2D is $\langle 1, 1, 1, 1 \rangle$, or D^{80}, where the distance to all 80 neighbours is set to 1. The 4D DT analogous to the city block DT in 2D is $\langle 1, 2, 3, 4 \rangle$, or D^{8} (not to be confused with chessboard DT in 2D), where the distance to the eight volume neighbours is set to 1. Both D^{8} and D^{80} are semi-regular (but not regular) according to the inequalities in (1). The equations for the distances can be expressed as in (3), still with $0 \le w \le z \le y \le x$:

$$D^{8} = x + y + z + w, \quad (4)$$
$$D^{80} = x. \quad (5)$$

3 Optimality Computations

In this Section, the optimal local distances for the Case I 4D WDT will be computed. Optimality is defined as minimising the maximal difference between the WDT and the Euclidean distance in an image of size $M \times M \times M \times M$. The choice of optimality criterion is somewhat arbitrary. However, this one has the advantage that is does not depend on any non-digital structure, such as an imbedded Euclidean sphere, which would be necessary to, e.g., minimise the average error.

The maximum of a particular type of function will often have to be computed. The following Lemma is used.

Lemma 2 *The function* $f(\xi) = \alpha\xi + \beta - \lambda\sqrt{\gamma + k\xi^2}$*, where* $|\alpha| < \sqrt{k}|\lambda|$ *and* $|\gamma| > 1$ *has the maximum value*

$$f_{\max} = \beta - \sqrt{\lambda^2 - \frac{\alpha^2}{k}} \cdot \sqrt{\gamma} \;\; \textit{for} \;\; \xi = \frac{\alpha\sqrt{\gamma}}{\sqrt{k^2\lambda^2 - k\alpha^2}}$$

Proof: The extremal value is found by setting the derivative of $f'(\xi)$ to zero, solving for ξ and simplifying the resulting expressions. □

The difference between the computed distance, see (3), and the Euclidean distance is

$$E(x, y, z, w) = (d - c)w + (c - b)z + (b - a)y + xa - \sqrt{x^2 + y^2 + z^2 + w^2}, \quad (6)$$

where $0 \leq w \leq z \leq y \leq x \leq M$. This difference is to be minimised in an $M \times M \times M \times M$ image. Put the origin in a corner of the image. We can then assume that the maximum difference, denoted maxdiff, occurs for $x = M$. As $0 \leq w \leq z$, the maximum of E(w) occurs for $w = 0$, $\partial/\partial w E(y, z, w) = 0$, or $w = z$. The difference in these three cases are found by simple insertion or by using Lemma 2 with $\xi = w, \alpha = (d-c), \beta = (c-b)z + (b-a)y + Ma, \lambda = k = 1$, and $\gamma = M^2 + y^2 + z^2$. We get:

$$\begin{aligned}
E_1(y, z) &= (c - b)z + (b - a)y + aM - \sqrt{M^2 + y^2 + z^2} \qquad \text{for } (M, y, z, 0),\\
E_2(y, z) &= (c - b)z + (b - a)y + aM - \sqrt{1 - (d - c)^2}\sqrt{M^2 + y^2 + z^2}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } (M, y, z, w_{\max}),\\
E_3(y, z) &= (d - b)z + (b - a)y + aM - \sqrt{M^2 + y^2 + 2z^2} \qquad \text{for } (M, y, z, z).
\end{aligned}$$

For each of these three expressions the maximum can occur for $z = 0$, $\partial/\partial z E(y, z) = 0$, or $z = y$, as $0 \leq z \leq y$. We get the following nine difference expressions, using insertion and Lemma 2.

$$
\begin{aligned}
&E_{11}(y) = (b-a)y + aM - \sqrt{M^2+y^2} && \text{for } (M, y, 0, 0),\\
&E_{12}(y) = (b-a)y + aM - \sqrt{1-(c-b)^2}\sqrt{M^2+y^2} && \text{for } (M, y, z_{\max}, 0),\\
&E_{13}(y) = (c-a)y + aM - \sqrt{M^2+2y^2} && \text{for } (M, y, y, 0),\\
&E_{21}(y) = \emptyset, \quad \text{as } 0 = z < w = w_{\max}, &&\\
&E_{22}(y) = (b-a)y + aM - \sqrt{1-(d-c)^2-(c-b)^2}\sqrt{M^2+y^2} &&\\
& && \text{for } (M, y, z_{\max}, w_{\max}).\\
&E_{23}(y) = (c-a)y + aM - \sqrt{1-(d-c)^2}\sqrt{M^2+2y^2} && \text{for } (M, y, y, w_{\max}),\\
&E_{31}(y) \equiv E_{11}(y), &&\\
&E_{32}(y) = (b-a)y + aM - \sqrt{1-\tfrac{1}{2}(d-b)^2}\sqrt{M^2+y^2} && \text{for } (M, y, z_{\max}, z_{\max}),\\
&E_{33}(y) = (d-a)y + aM - \sqrt{M^2+3y^2} && \text{for } (M, y, y, y).
\end{aligned}
\tag{7}
$$

For each of these seven expressions the maximum can occur for $y = 0$, $\partial/\partial y E(y) = 0$, or $y = M$, as $0 \le y \le x = M$. We get the following 21 expressions, again using insertion and Lemma 2.

$$
\begin{aligned}
&E_{111} = (a-1)M && \text{for } (M, 0, 0, 0),\\
&E_{112} = (a - \sqrt{1-(b-a)^2})M && \text{for } (M, y_{\max}, 0, 0),\\
&E_{113} = (b - \sqrt{2})M && \text{for } (M, M, 0, 0),\\
&E_{121} = \emptyset, \quad \text{as } 0 = y < z = z_{\max}, &&\\
&E_{122} = (a - \sqrt{1-(c-b)^2-(b-a)^2})M && \text{for } (M, y_{\max}, z_{\max}, 0),\\
&E_{123} = (b - \sqrt{2}\sqrt{1-(c-b)^2})M && \text{for } (M, M, z_{\max}, 0),\\
&E_{131} \equiv E_{111}, &&\\
&E_{132} = (a\sqrt{1-\tfrac{1}{2}(c-a)^2})M && \text{for } (M, y_{\max}, y_{\max}, 0),\\
&E_{133} = (c - \sqrt{3})M && \text{for } (M, M, M, 0),\\
&E_{221} = \emptyset, \quad \text{as } 0 = y < z = z_{\max} &&\\
&E_{222} = (a - \sqrt{1-(d-c)^2-(c-b)^2-(b-a)^2})M &&\\
& && \text{for } (M, y_{\max}, z_{\max}, w_{\max}),\\
&E_{223} = (b - \sqrt{2}\sqrt{1-(d-c)^2-(c-b)^2})M && \text{for } (M, M, z_{\max}, w_{\max}),\\
&E_{231} = \emptyset, \quad \text{as } 0 = y = z < w = w_{\max} &&\\
&E_{232} = (a - \sqrt{1-(d-c)^2-\tfrac{1}{2}(c-a)^2})M && \text{for } (M, y_{\max}, y_{\max}, w_{\max}),\\
&E_{233} = (c - \sqrt{3}\sqrt{1-(d-c)^2})M && \text{for } (M, M, M, w_{\max}),\\
&E_{321} = \emptyset, \quad \text{as } 0 = y < z = z_{\max}, &&\\
&E_{322} = a - \sqrt{1-\tfrac{1}{2}(d-b)^2-(b-a)^2})M && \text{for } (M, y_{\max}, z_{\max}, z_{\max}),\\
&E_{323} = (b - \sqrt{2}\sqrt{1-\tfrac{1}{2}(d-b)^2})M && \text{for } (M, M, z_{\max}, z_{\max}),
\end{aligned}
$$

$$\begin{aligned}
E_{331} &\equiv E_{111}, \\
E_{332} &= (a - 2\sqrt{1-(d-a)^2})M \text{ for } (M, y_{\max}, y_{\max}, y_{\max}), \\
E_{333} &= (d-2)M \qquad\qquad\qquad\quad \text{for } (M, M, M, M).
\end{aligned}$$

Thus 15 difference expressions E_{ijk} remain. The maximum of these 15 expressions should now be minimised by varying a, b, c, and d. Numerical experimentation show that $\max(E_{ijk})$ is minimal for $E_{222} = -E_{111} = -E_{113} = -E_{133} = -E_{333}$. Solving these equations yields

$$\begin{aligned}
a_{opt} &= 1 - \mathcal{R} && \approx 0.9048, \\
b_{opt} &= \sqrt{2} - \mathcal{R} && \approx 1.3191, \\
c_{opt} &= \sqrt{3} - \mathcal{R} && \approx 1.6369, \\
d_{opt} &= 2 - \mathcal{R} && \approx 1.9048, \\
\text{with} \quad \text{maxdiff} &= \mathcal{R}M && \approx 0.0951M,
\end{aligned} \tag{8}$$

$$\text{where} \quad \mathcal{R} = \frac{1}{2}(1 - \sqrt{2}\sqrt{\sqrt{6} + 2\sqrt{3} + \sqrt{2} - 7}).$$

The optimal solutions for $a \equiv 1$ are needed when computing integer DTs, as then a becomes a scale factor. In this case, we solve $E^*_{222} = -E^*_{113} = -E^*_{133} = -E^*_{333}$, where the star denotes that $a = 1$ has been substituted in the expressions. The solutions are

$$\begin{aligned}
a^*_{opt} & && = 1, \\
b^*_{opt} &= \sqrt{2} - \mathcal{S} && \approx 1.2796, \\
c^*_{opt} &= \sqrt{3} - \mathcal{S} && \approx 1.5975, \\
d^*_{opt} &= 2 - \mathcal{S} && \approx 1.8654, \\
\text{with} \quad \text{maxdiff}^* &= \mathcal{S}M && \approx 0.1346M,
\end{aligned} \tag{9}$$

$$\text{where} \quad \mathcal{S} = \frac{1}{2}\sqrt{2} - \sqrt{\sqrt{6} + 2\sqrt{3} + \sqrt{2} - 7}.$$

In both situations, free a and $a \equiv 1$, it is easy to check that the optimal solutions fulfil the inequalities (2), and thus are in the allowed hypercone in parameter space. Also, in both situations the maximum difference from the Euclidean distance is a fraction of the size of the image, as can be expected.

Using real valued local distances in digital images is generally not desirable. Integer local distances are preferable. Candidate integer approximations of the optimal values, denoted A, B, C, and D, are found by multiplying the optimal local distances by an integer scale factor and rounding to the nearest integer. Then the maximal differences are computed (all expressions are available from the computations of the optimal local distances), to check the approximations. The smallest local distance, a, will act as a scale factor, therefore the resulting WDT will become $\langle 1, B/A, C/A, D/A \rangle$. It is of course important to check that (2) are fulfilled, otherwise the difference expressions are invalid. The best approximation result possible is maxdiff* and the optimal local distances to be multiplied by the scale factor are b^*_{opt}, c^*_{opt}, and d^*_{opt}. Good integer $3 \times 3 \times 3 \times 3$ WDTs are listed in the next Section.

Table 1. Integer $3 \times 3 \times 3 \times 3$ distance transformations

Case	a	b	c	d	maxdiff
D^8	1	-	-	-	2.00000
D^{80}	1	1	1	1	1.00000
real	a_{opt}	b_{opt}	c_{opt}	d_{opt}	0.09515
real	1	b^*_{opt}	c^*_{opt}	b^*_{opt}	0.13456
integer	2	3	4	4	0.29289
integer	3	4	5	6	0.18350
integer	6	8	10	11	0.16667
integer	6	9	10	11	0.16667
integer	7	10	12	13	0.15485
integer	8	11	13	15	0.14304
integer	15	20	24	28	0.13590

4 Results

In this section the results of the optimality computations are summarised and illustrated. Table 1 lists a number of distance transforms. First, the simple D^8 and D^{80} are listed, with their associated maxdiff. These are easily computed from the expressions (4) and (5). Next in Table 1 comes the optimal values for Case I, both for free a and $a \equiv 1$.

After the real valued WDTs, the best integer approximations, using scale factors ($= A$) up to 20, are listed in Table 1. For practical purposes, $\langle 3, 4, 5, 6\rangle$ is probably the best choice. The maxdiff is reasonably good, with a small scale factor. Note that it is hard to improve on $\langle 15, 20, 24, 28\rangle$. Note also that all integer DTs are on the border of the allowed hypercone defined by (2), except $\langle 7, 10, 12, 13\rangle$, which should thus exhibit the most "typical" traits of Case I DTs in 4D.

It must be remembered that even if Case I is the "natural" one of the Cases for 4D DTs, there is no guarantee that it is the best Case. In 3D, the analogous Case I_3 gave the best maxdiff$_3$ *but* the other case, Case II_3, gave the best maxdiff$^*_3$ (with $a \equiv 1$), so better integer approximations could be found for Case II_3 than for Case I_3, see [3]. The same may well be true in 4D.

A good way to characterise a DT is the shape of its associated ball, defined as all pixels/voxels/hyxels with a distance less than or equal to the radius from a single central element. In 2D, the city block and chessboard distance balls are a diamond and a square, respectively. In 4D, the D^8 and D^{80} balls are a hyperoctahedron and a hypercube, respectively (the tetrahedron, the octahedron, and the cube are the only "Platonic" solids that exist in any dimension, see [4]). In 4D, the Case I $3 \times 3 \times 3 \times 3$ WDT balls are hyperpolyhedra.

Illustrating hyperpolyhedra is, however, not very easy. One way of doing this was presented in [5]. A 4D digital image is created, with a single object hyxel in the middle. The DT is then computed from this object into the background, in the standard way. If this image is thresholded at a suitable level, a ball with the radius of the threshold value is created. The threshold should be as large as possible while the ball created is still completely within the image. We now have a binary 4D image containing the ball we wish to visualise. If we fix the w-level in this image, we will get a 3D image with a "hyperslice" of the 4D ball, which is in itself a polyhedron. This 3D object can be visualised using a simple binary 3D imaging technique. Ideally, the consecutive hyperslices can be shown as a sequence, a "movie," but here we are constrained to show a few sample hyperslices. In Figs. 2 and 3, we show the $\langle 3, 4, 5, 6 \rangle$ and the $\langle 7, 10, 12, 13 \rangle$ balls with radius 46. The six "hyperslices" were chosen so that the different shapes the ball has at different levels are shown. They are *not* equally spaced in 4D, see the Figures for the chosen w-values, where $w = 1$ denotes the slice with the "first" ball hyxel. These two WDTs were chosen, as the $\langle 3, 4, 5, 6 \rangle$ is the one most probable to be used and the $\langle 7, 10, 12, 13 \rangle$ is the only one exhibiting all the faces that a general 4D Case I WDT can have. The "mid-slice", $w = 46$, is a $\langle 3, 4, 5 \rangle$ ball and a $\langle 7, 10, 12 \rangle$ ball, respectively (see [3]).

5 Conclusions

Optimal weighted distance transforms in 4D using $3 \times 3 \times 3 \times 3$ neighbourhoods have been investigated. The best possible such DT, using real-valued weights, has a maximal difference from the Euclidean distance of 9.51%. The best possible integer valued DT is proven to have a maximal difference of 13.46%. The most useful WDT is probably $\langle 3, 4, 5, 6 \rangle$ with a maximal difference of 18.35%. This was, in fact, what was suggested already in [1], but there the motivation was much weaker. A number of other integer WDTs with higher scale factors, but with smaller maximal differences are also listed.

Acknowledgement

Thanks to Stina Svensson for computing and visualising the two balls in Figs 2 and 3.

References

1. G. Borgefors, Distance transformations in arbitrary dimensions, *Computer Vision, Graphics, and Image Processing* **27**, 1984, pp. 321–345. 325, 326, 328, 333
2. G. Borgefors, Distance transformations in digital images *Computer Vision, Graphics, and Image Processing* **34**, 1986, pp. 344–371. 326
3. G. Borgefors, On digital distance transforms in three dimensions, *Computer Vision and Image Understanding,* Vol. **64**, No. 3, (1996), pp. 368–376. 326, 327, 332, 333
4. H. S. M. Coxeter, Regular polytopes, Dover Publications, Inc., New York, 1973. 332
5. G. Borgefors, H. Guo, Weighted distance transform hyperspheres in four dimensions, Proc. SSAB Symposium on Image Analysis 1997, Stockholm, Sweden, March 1997, pp. 71–76. 332

6. M. Fidrich: Iso-surface extraction in 4D with applications related to scale space, In Miguet, Montanvert, Ubéda, Eds., Discrete Geometry for Computer Imagery, Springer 1996 (LNCS 1176), pp. 257–268. 325
7. P. P. Jonker and O. Vermeij, On skeletonization in 4D images, In Perner, Wang, Rosenfeld, Eds., Advances in Structural and Syntactical Pattern Recognition, Springer 1996 (LNCS 1121), pp. 79–89. 325
8. C. O. Kiselman, Regularity properties of distance transformations in image analysis, *Computer Vision and Image Understanding,* Vol. **64**, No. 3, (1996), pp. 390–398. 325, 326, 327
9. T. Y. Kong, Topology-preserving deletion of 1's from 2-, 3-, and 4-dimensional binary images, In Ahronovitz and Fiorio, Eds., Discrete Geometry for Computer Imagery, Springer 1997 (LNCS 1347), pp. 3–18. 325
10. A. Rosenfeld and J. Pfaltz, Sequential operations in digital picture processing, *Journal of the ACM* **13** (4), 1966, pp. 471–494. 325
11. M. Yamashita and T. Ibaraki: Distance defined by neighborhood sequences, *Pattern Recognition* **19**, 1986, pp. 237–246. 326

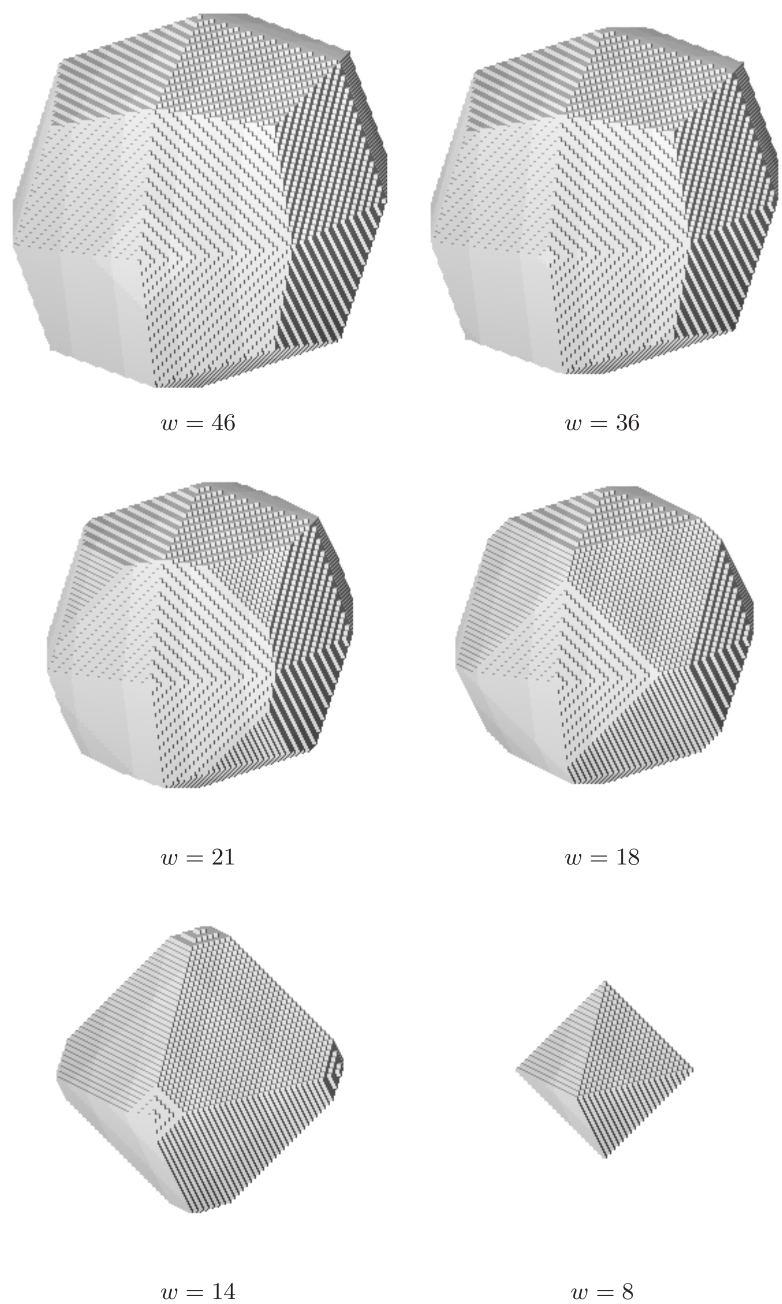

Fig. 2. The ball of the $\langle 3, 4, 5, 6 \rangle$ distance transform, shown as 3D cuts through 4D space at six different levels

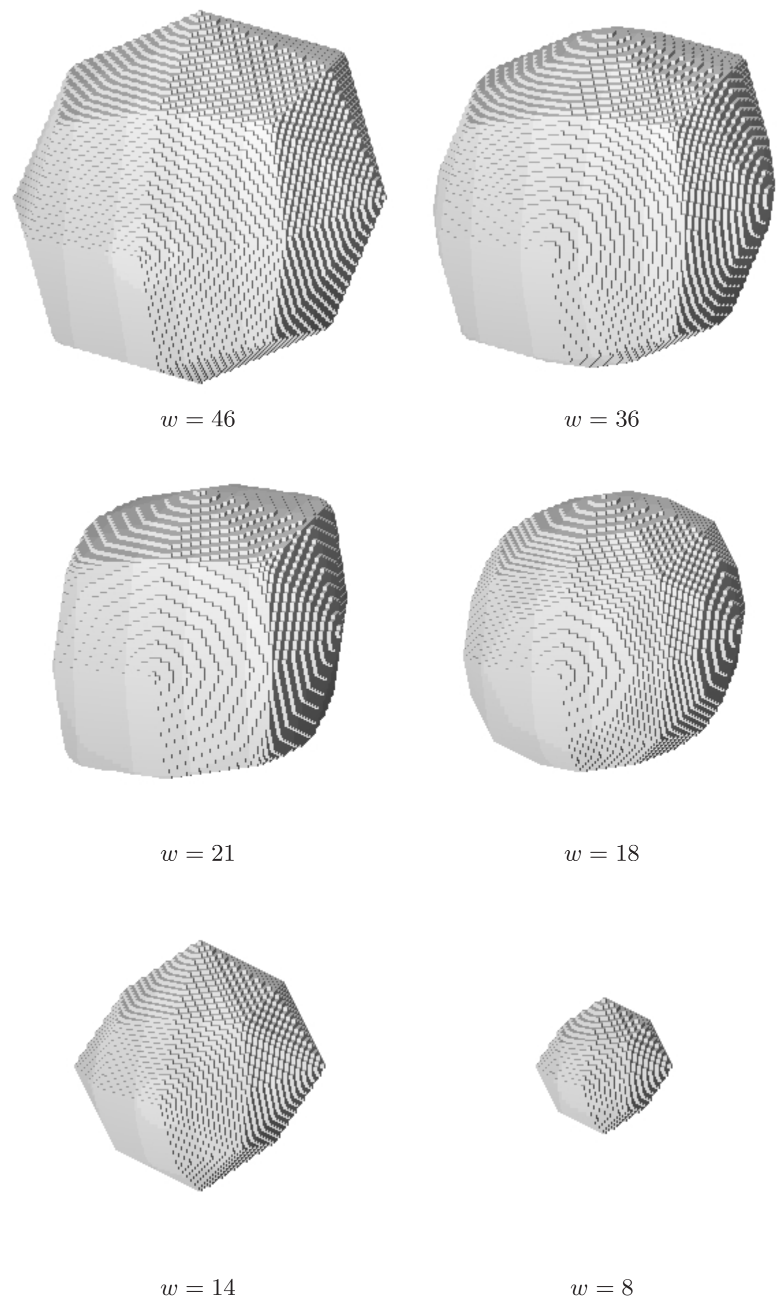

Fig. 3. The ball of the $\langle 7, 10, 12, 13 \rangle$ distance transform, shown as 3D cuts through 4D space at six different levels