

Distance transforms: Academics versus industry

Egon L. van den Broek ^{a,*} and Th. E. Schouten ^b,

^a *Human-Centered Computing Consultancy, <http://www.human-centeredcomputing.com/>, Vienna, Austria; Human Media Interaction (HMI), Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands; Karakter University Center, Radboud University Medical Center (UMC) Nijmegen, Nijmegen, The Netherlands*

^b *Institute for Computing and Information Sciences (ICIS), Faculty of Science, Radboud University Nijmegen, Nijmegen, The Netherlands*

Abstract: In image and video analysis, distance transformations (DT) are frequently used. They provide a distance image (DI) of background pixels to the nearest object pixel. DT touches upon the core of many applications; consequently, not only science but also industry has conducted a significant body of work in this field. However, in a vast majority of the cases this has not been published in major scientific outlets but has been filed as a patent application. This article provides a brief introduction into DT, including a specification of a few of the most prominent algorithms in the field. Next, a few interesting algorithms from the last decade are discussed. A benchmark including eight DT algorithms (i.e., city block, Danielsson's algorithm, chamfer 3-4, hexadecagonal region growing, a recent claimed true Euclidean DT, and three exact Euclidean DT) has been executed, which illustrates the intriguing complexity of DT in terms of precision and computational complexity. Subsequently, a selection of key patent applications are discussed that have emerged in this field, including their scientific merit and areas of application. Finally, this article's findings are summarized and discussed, with an emphasis on both the common ground of scientific articles and patent applications as well as the added value they can have to each other.

Keywords: Distance transforms (DT), distance image, Euclidean distance, computational geometry, patent, chamfer, FEED

1. INTRODUCTION

When comparing academic work with industry's patent applications on distance transforms (DT), there appears to be hardly any overlap between the authors of scientific articles and the inventors of granted patents. Exceptions to this rule of thumb can be found, but are rare. So, the transfer of knowledge between both these research communities seems to be suboptimal, to say the least. This limits progress in both academic settings and in industry, important work from

the other community remains unknown. Consequently, the risk that work is reinvented is high. In particular, for industry this is an undesirable situation, as this increases the risk of patent applications will not hold.

This article aims to bridge the gap in communication between academics and industry in the area of DT. We will go through academics' developments on DT. Subsequently, recent patent applications on DT are discussed and compared with the work conducted in academics. This will be preceded by a brief introduction on DT, with which we will start now.

With the rise of the computer, already more than 50 years ago, processing of discrete (digital) data became more and more important [1,2]. Consequently, a proper understanding of discrete spaces was required. From this, the field of digital geometry (or digital topology) emerged, which provided the means to study

*Corresponding author's address: Human Media Interaction (HMI), Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands; Tel: +31 53 489 3740; Fax: +31 53 489 3503; E-mails: vandenbroek@acm.org and t.schouten@cs.ru.nl

the geometry of discrete point spaces (i.e., that belong to \mathbb{Z}^2).

The transition from classical geometry (i.e., in Euclidean space in which values include an interval of real numbers) to digital geometry proved to be challenging. Some properties of classical geometry do not hold for digital geometry; for example,

- Often there is not a straight line between two points. Consequently, multiple shortest paths can exist between two points;
- On the one hand, when two lines cross, this can occur without them having pixels in common. So, it is possible that two non-parallel lines do not intersect. On the other hand, they can also share several digital points. So, they intersect over more than one point;
- No angle exists between two lines that cross without intersecting and, hence, digital trigonometry does not hold [3]; and
- “Discrete Euclidean Voronoi regions are not always connected” [4].

In Klette and Rosenfeld’s excellent handbook “Digital geometry: Geometric methods for digital picture analysis” [5], in particular in Chapters 3, an exhaustive discussion is provided on the differences between classical and digital geometry. Throughout their survey, Fabri *et al.* [4] also denotes some of these. So, for more information on this topic we refer to [4, 5].

One of the field’s biggest challenges is the calculation of distance transforms (DT) and their resulting distance maps or distance image (DI) [1], in particular when both precision and computational complexity are of importance. This transform and its resulting map is a basic operation, a preprocessing step in the field of image processing. For example, morphological operations (e.g., dilation/dilatation and erosion) rely on the computation of DI [6–8]; see also Fig. (1).

In the areas of computer vision, image and video processing, it is usually necessary to extract information about the shape and the position of the foreground pixels relative to each other using a segmentation procedure [9–15]. Subsequently, many techniques are involved to accomplish this task; one such technique is the DT; see also Fig. (1) [16]. In such a DI, the value of each pixel represents its distance to the set O of object pixels o in the binary image. As such, a Voronoi surface $V(b)$ of the set O can also be considered to be a DT [17–19] because it gives the distance from any background pixel b to the nearest point in the set O . This is also illustrated by Figs. (2) and (3) that present

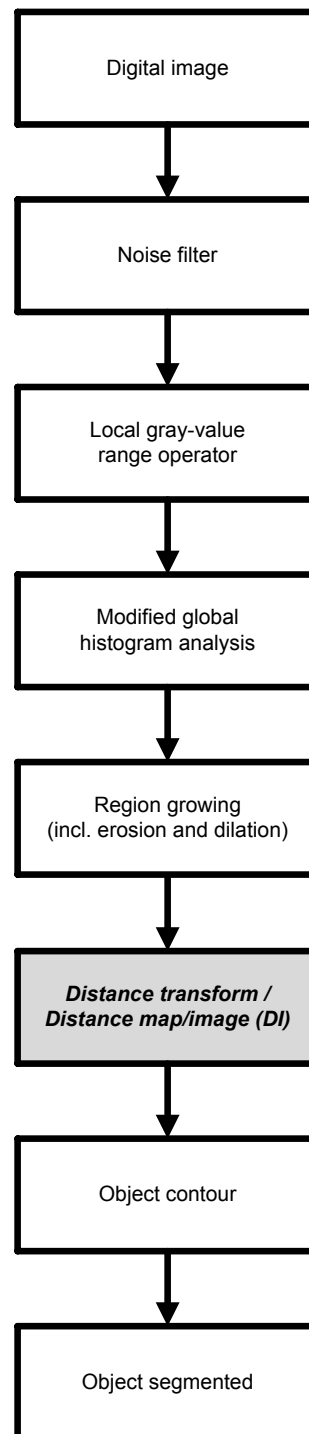


Fig. (1). A processing pipeline of image segmentation, adapted from [16]. This fundamental operation in image processing, most often, requires a distance transformation, as is also indicated in this processing pipeline.

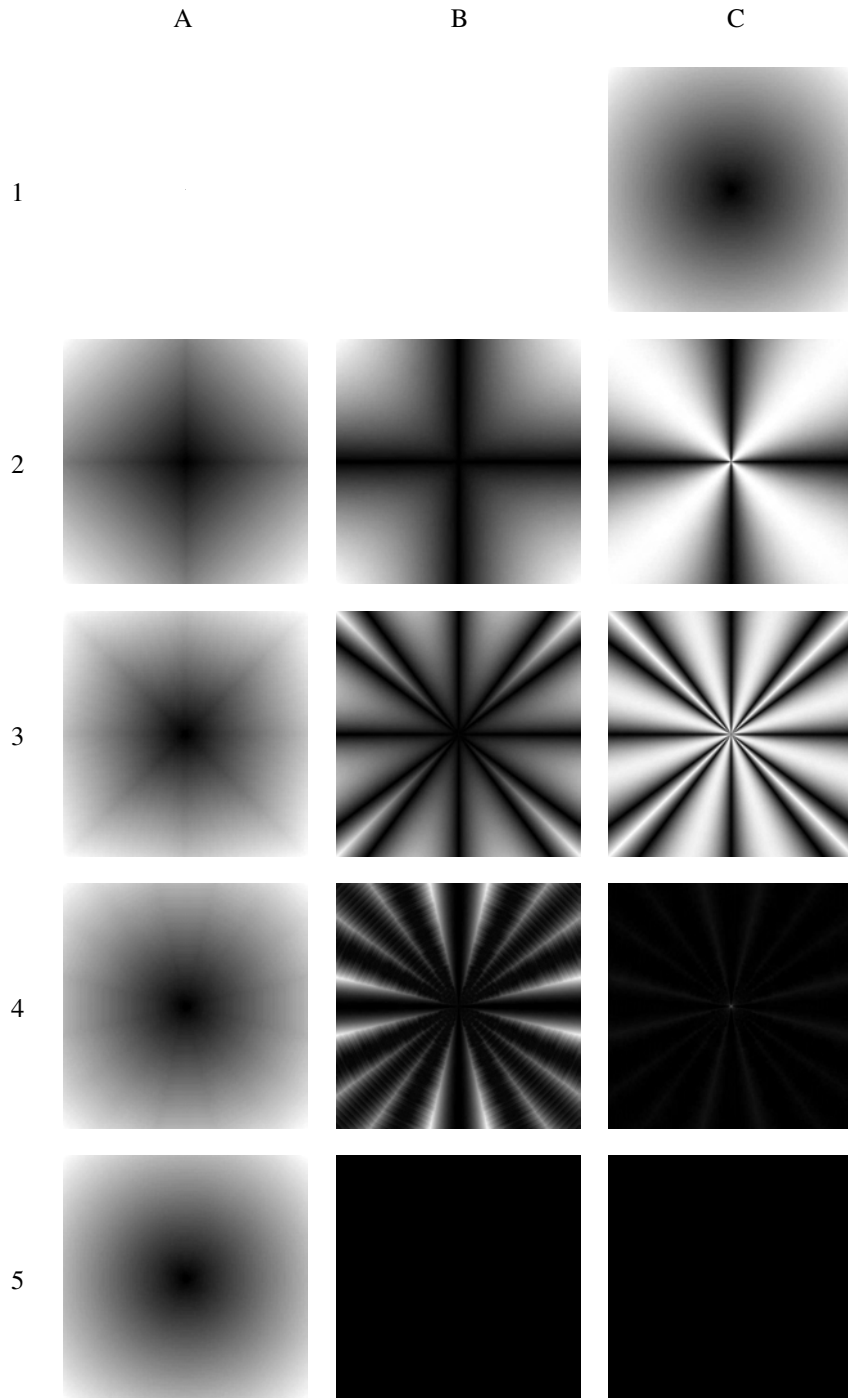


Fig. (2). An input image consisting of 1 pixel (# A1) and its true Euclidean distance image (EDI; # C1). The four rows below this, visualize the distance transforms (DT) # 2: city block [1, 2], # 3: chamfer 3-4 [21, 22], # 4: hexadecagonal region growing [23], and # 5: Shih and Wu's Euclidean DT [24]. From left to right, the columns indicate: # A: resulting distance image (DI_{R_i}), # B: absolute difference with EDI (i.e., $|EDI - DI_{R_i}|$), and # C: relative difference with the EDI (i.e., $|EDI - DI_{R_i}|/EDI$). See Sections 1–4 for more information. Table 1 provides the error statistics for the images visualized here. Note that the (EDI (# C1) is also the output of Danielsson [17], Maurer *et al.* [33], FEED [12, 20], and Lucent's LLT [68]; see also Table 1. Further, note that to optimize the percept of the visualizations, the original intensity values i of all the images shown here $\{I_x\}$ were transformed as follows: $255(i - \min_{I_x}) / \max_{I_x} - \min_{I_x}$.

Table 1

Errors of distance transforms (DT) in a 524×524 image (i.e., 274,576 pixels), consisting of 1 pixel in the center (see also Fig. (2)). The DT resulting from the following eight algorithms are presented: city-block, chamfer 3-4, Danielsson, hexadecagonal region growing (HexaD), Maurer *et al.*, Shih and Wu's 2-scan (EDT-2), FEED, and Lucent's LLT.

DT algorithms		error compared to (exact) ED				
		-ED (in % pixels)	absolute error (in pixels)		relative error (in %)	
			average	max	average	max
city block	[1,2]	99.62	61.52	153.48	29.56	41.42
Danielsson	[17]	-	-	-	-	-
chamfer 3-4	[21,22]	99.43	6.63	21.19	3.35	5.72
HexaD	[23]	99.41	1.41	7.47	0.73	41.42
Maurer <i>et al.</i>	[33]	-	-	-	-	-
EDT-2	[24]	-	-	-	-	-
FEED	[12,20]	-	-	-	-	-
LLT	[68]	-	-	-	-	-

Note. With - is denoted that no (or 0) errors have been generated.

true Euclidean DT or Voronoi surfaces. Such a DT is calculated as follows:

$$DI(b) = \min_{o \in O} D(b, o), \quad (1)$$

where D can be any metric and b is a background pixel. Fig. (2) presents some DIs of a single pixel (in the center of the image; look closely) as well as their deviation from the ED [1, 2, 12, 20–24]. So, as is also illustrated in both Fig. (2) and Fig. (3), the DI itself relies on the metric chosen (e.g., the Euclidean distance, ED).

As it is explained above, DT are a rather fundamental concept in computational geometry and, consequently, in image processing and computer science in general. On the one hand, this makes research on DT interesting for a broad range of applications and, consequently, industry's interest can be expected as well as patent applications from their side. On the other hand, research in this field often concerns improvements of algorithms in terms of speed / computational complexity [25], as we will also see later in this article. In applications, these are factors influenced by a variety of factors, which make the infringement of a patent on DT hard to detect. From this perspective, the economic value of patent applications on DT is questionable. So, the feasibility of patent applications with DT as their core is like a coin with two sides: infringement versus impact. This makes it interesting to put the worlds of scientific articles and industrial patent applications next to each other, as will be done in the current article.

This article is organized as follows. First, in Section 2, we will continue with the introduction of DT and DI, with the aim to create a common background of the topic. Next, Section 3 briefly touches upon some of the most important work done on DT/DI in the previous century. This section is followed by a report on the developments in the last decade in Section 4. Then, a review of the key patent applications that emerged on DT/DI will be discussed and their relation with research conducted in academic community will be depicted in Section 5. We close this article with a discussion in Section 6.

2. DISTANCE TRANSFORM

The DT is a basic operation in computer vision, pattern recognition, and robotics. For instance, if the object pixels represent obstacles, then the DT tells us how far a point is from these obstacles. This information is for example useful when one needs to segment a medical image [9–11, 13–15] or tries to move a robot in the free space and to keep it away from the obstacles [26–28].

DT can be applied in any number of dimensions [21, 29–34] as well as on image sequences (i.e., video) [12]. Distance is a fundamental notion with such functions. The L_p distance metric is defined as follows:

$$d_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}, \quad (2)$$

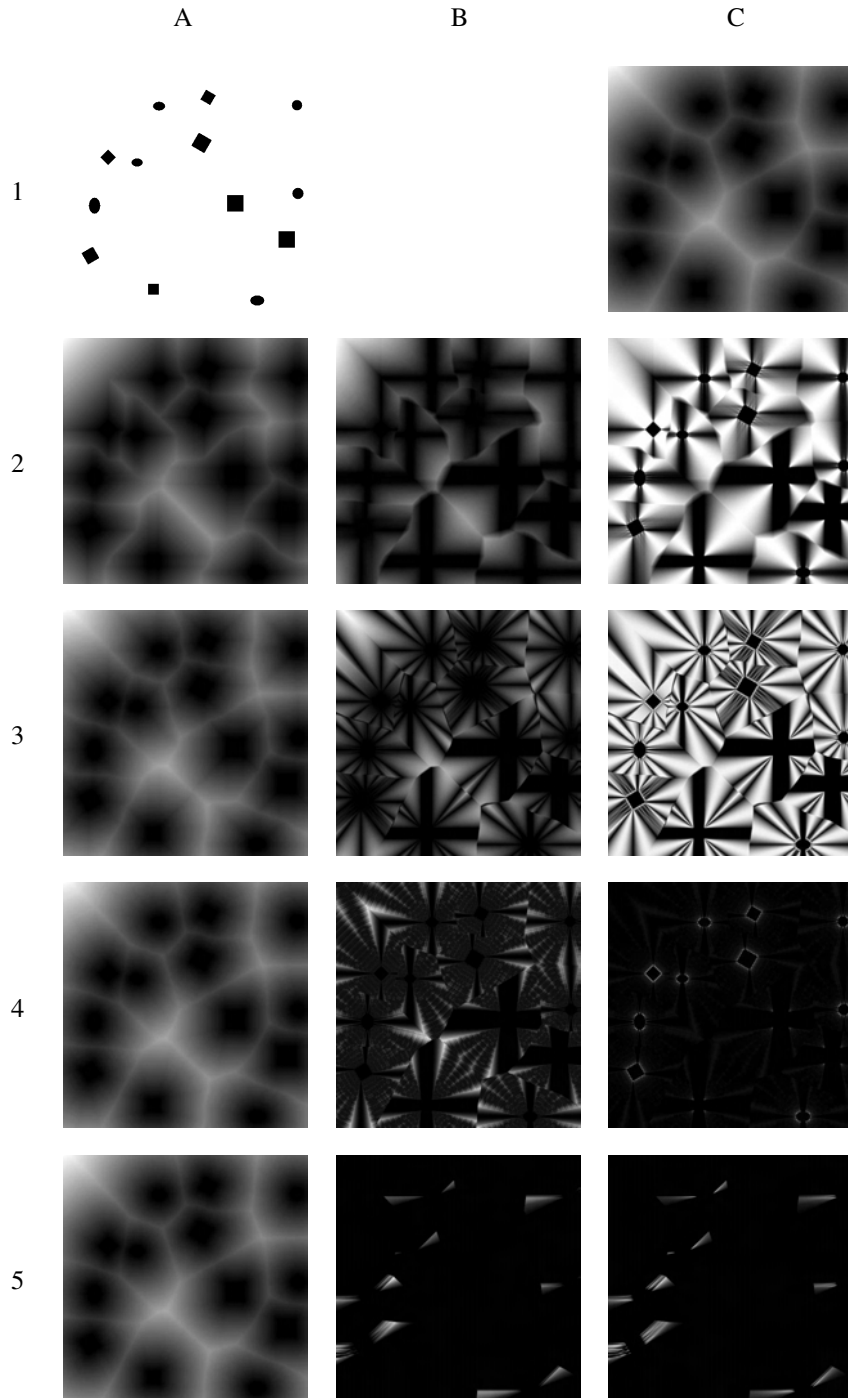


Fig. (3). An input image consisting of some simple test objects (# A1) and their true Euclidean distance image (EDI; # C1). The four rows below this, visualize the distance transforms (DT) # 2: city block [1, 2], # 3: chamfer 3-4 [21, 22], # 4: hexadecagonal region growing [23], and # 5: Shih and Wu's Euclidean DT [24]. From left to right, the columns indicate: # A: resulting distance image (DI_R), # B: absolute difference with EDI (i.e., $|EDI - DI_R|$), and # C: relative difference with the EDI (i.e., $|EDI - DI_R|/EDI$). See Sections 1–4 for more information. Table 2 provides the error statistics for the images visualized here. Note that the (EDI (# C1) is also the output of Maurer *et al.* [33], FEED [12, 20], and Lucent's LLT [68]; see also Table 2. Danielsson [17] had some small errors; however, they are hard to visualize in this manner, with this resolution. Further, note that to optimize the percept of the visualizations, the original intensity values i of all the images shown here $\{I_x\}$ were transformed as follows: $255(i - \min_{I_x}) / \max_{I_x} - \min_{I_x}$.

Table 2

Errors of distance transforms (DT) in a 524×524 image (i.e., 274,576 pixels), consisting of some simple test objects (see also Fig. (3)). The DT resulting from the following eight algorithms are presented: city-block, chamfer 3-4, Danielsson, hexadecagonal region growing (HexaD), Maurer *et al.*, Shih and Wu's 2-scan (EDT-2), FEED, and Lucent's LLT.

DT algorithms		error compared to (exact) ED				
		-ED (in % pixels)	absolute error (in pixels)		relative error (in %)	
			average	max	average	max
city block	[1, 2]	85.01	11.95	71.76	21.68	41.42
Danielsson	[17]	0.09	δ_a	0.17	δ_r	6.07
chamfer 3-4	[21, 22]	84.60	1.65	9.44	2.96	5.72
HexaD	[23]	83.86	0.41	3.70	1.02	41.42
Maurer <i>et al.</i>	[33]	-	-	-	-	-
EDT-2	[24]	5.03	0.04	3.09	0.09	7.28
FEED	[12, 20]	-	-	-	-	-
LLT	[68]	-	-	-	-	-

Notes. The exact average errors produced by Danielsson's algorithm [17] are $\delta_a = 0.000031$ (absolute) and $\delta_r = 0.000510$ (relative).

With - is denoted that no (or 0) errors have been generated.

where \mathbf{x} and \mathbf{y} are n -tuples, i is used to denote their n coordinates (or dimensions), and $1 \leq p \leq \infty$.

Although the L_p distance metric can be defined in an n -dimensional space (see Eq. 2), in practice often a two-dimensional (2D) or three-dimensional (3D) space is required [32, 35–37], as most digital images are 2D or 3D. For higher dimensional spaces (i.e., $n > 3$), applications are less apparent and, hence, little patent applications will be filed in this area. Therefore, this article focussed mainly on 2D DT and to a lesser extend on 3D DT, which is an established field on its own [32, 35].

The golden standard for 2D DT is the Euclidean DT (EDT). Often one wants to determine the exact ED. The Euclidean metric (d_E) is directly derived from Eq. 2 with $p = 2$, which results in:

$$d_E((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (3)$$

However, even in 2D, finding the DT with respect to the Euclidean metric is rather time consuming. In order to tackle the computational burden of EDT, two strategies have been adopted: i) approximation of exact EDs and ii) parallel implementations [38–43]. This overview article focuses on the first strategy.

To determine the quality of a DT, its deviation (or error) from its golden standard the EDT has to be determined; see also Figs. (2) and (3). This deviation can be defined in several ways, such as the:

- average error (absolute and/or relative),

- maximum error (absolute and/or relative),
- number of pixels in the DI with an incorrect distance assigned to it, and
- the variance in errors

of the difference between the DT and EDT. A range of factors determine which measure one should take. The area of application is the most important factor. Of course, a range of other measures can be defined. Regrettably, in most papers, the error measure is not defined; an exception to this is [23]. In this paper, the error measures are made explicit; see also Tables 1–3. All three tables denote the first three measures of the four just mentioned.

Although Eq. 1 is straightforward, it is hard to develop an algorithm that calculates the DT quickly [7, 25, 44–47]. In practice, the calculation of DT starts with the initialization of the algorithm. Assign an initial integer distance $DI(x, y)$ to each pixel (x, y) of picture I , and initialize these as follows:

$$\begin{aligned} DI(x, y) &= 0 & \text{if } I(x, y) \in O \\ DI(x, y) &= \Delta & \text{if } I(x, y) \notin O, \end{aligned} \quad (4)$$

with $\Delta > \sqrt{X^2 + Y^2}$, where X and Y are respectively the number of columns and the number of rows of the picture grid [5, 48], which together define the image size. This initialization is generic, suitable for most DT. After the initialization, DI can be generated.

Although often unmentioned, it should be noted that in most cases a (standard) rectangular grid is assumed. However, alternatives have also been explored; for ex-

Table 3

Errors and timing results of distance transforms (DT) on a set of 160 (size: 524×524 , with 3%–77% object pixels) artificially generated images, such as also shown in Fig. (3)). The DT resulting from the following eight algorithms are presented: city-block, chamfer 3-4, Danielsson, hexadecagonal region growing (HexaD), Maurer *et al.*, Shih and Wu's 2-scan (EDT-2), FEED, and Lucent's LLT.

DT algorithms		error compared to (exact) ED					timing			
		–ED (in % pixels)	absolute error (in pixels)		relative error (in %)		in ms/image		in ns/pixel	
			average	max	average	max	total	rms*	total	rms*
city block	[1, 2]	59.10	4.55	90.38	14.46	41.42	1.86	0.22	6.76	0.81
Danielsson	[17]	0.23	δ_a	0.33	δ_r	11.80	6.45	0.81	23.50	2.95
chamfer 3-4	[21, 22]	58.17	0.64	12.70	1.93	5.72	5.30	0.54	19.30	1.98
HexaD	[23]	57.41	0.22	5.93	1.35	41.42	14.11	2.77	51.39	10.08
Maurer <i>et al.</i>	[33]	–	–	–	–	–	10.10	0.75	36.79	2.72
EDT-2	[24]	5.24	0.04	12.47	0.13	7.70	14.24	3.52	51.84	12.82
FEED	[12, 20]	–	–	–	–	–	2.80	0.28	10.20	1.01
LLT	[68]	–	–	–	–	–	9.00	0.95	32.77	3.46

Notes. rms* (i.e., root mean square) indicates the variation in time due to the content of the images.

The exact average errors produced by Danielsson's algorithm [17] are $\delta_a = 0.003$ (absolute) and $\delta_r = 0.00075$ (relative).

With – is denoted that no (or 0) errors have been generated.

ample, triangular and hexagonal grids [48–50] and sparse grids [51]. For an overview of the possible grids, we refer to [5].

In principle, DT are binary; that is, only two types of pixels are distinguished: those that belong to an object (i.e., $I(x, y) \in O$) and those that do not belong to an object (i.e., $I(x, y) \notin O$), often denoted as background pixels. However, in practice, often multiple objects or classes are present and need to be distinguished [19, 52, 53]. For this purpose, a multi object or multi class DT is required. Fortunately, a straightforward solution can be implemented that solves this issue. The class or object label of the input pixel $I(x, y)$ that provides the minimum distance can be placed in a second DI^+ (i.e., a matrix) [19, 52]. So, with assigning a new $DI(x, y)$ to $I(x, y)$, also DI^+ needs to be updated.

With the introduction (Section 1; see also Fig. (1)) and the current section, the authors hope to have given a brief introduction on the basic elements of computational geometry related to DT. With depicting a processing pipeline of image segmentation (see Fig. (1)) we hope to have stressed the importance of DT for this fundamental operation in image processing and, consequently, for image processing in general. In the next two sections, we will discuss the academic research conducted in the last 30+ years of the previous century (Section 3) and, subsequently, in the last decade (Section 4). After these sections, an overview will be provided of the research as conducted in industry and described in its key patent applications (Section 5).

3. ON MORE THAN 30 YEARS OF RESEARCH

With more than 30 years of research on DT, it falls far beyond the scope of this article to provide an extensive review. Therefore, we will highlight some of the most important works done on DT in the previous century. For an excellent recent review, we refer to [4]. We hope that this selective review provides some additional understanding on DT, in particular EDT, as well. Moreover, an explanation of this work is required to understand the key patent applications that emerged in this field, in particular, in the last decade, as will be discussed in Section 5.

Rosenfeld and Pfaltz [1, 2] introduced the first distance functions and their accompanying proofs and algorithms, which could be utilized for the generation of DI for digital (2D) images. The two distance functions that have become most famous are the city-block distance (d_4):

$$d_4((x_1, y_1), (x_2, y_2)) = |x_1 - x_2| + |y_1 - y_2| \quad (5)$$

and the chessboard distance (d_8):

$$d_8((x_1, y_1), (x_2, y_2)) = \max\{|x_1 - x_2|, |y_1 - y_2|\}, \quad (6)$$

where (x_1, y_1) and $(x_2, y_2) \in \mathbb{Z}^2$.

The city-block distance allows measuring only in horizontal and vertical directions (see also Figs. (2)

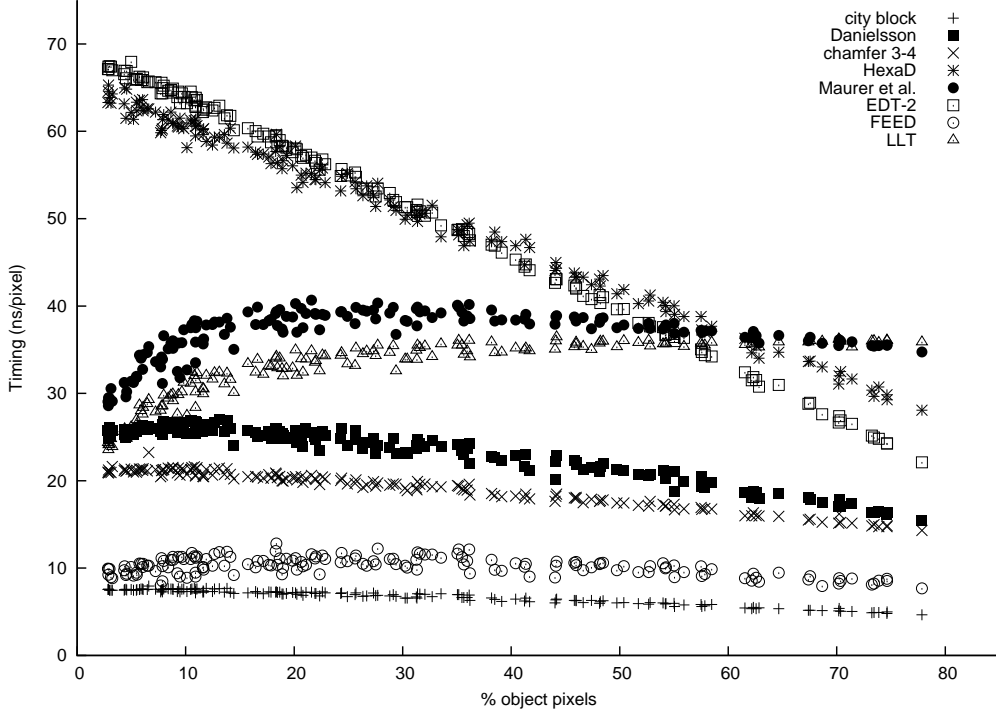


Fig. (4). The execution times as function of the percentage (%) of object pixels in the 160 images, for the following eight DT algorithms: city-block [1, 2], Danielsson [17], chamfer 3-4 [21, 22], hexadecagonal region growing (HexaD) of Coiras *et al.* [23], Maurer *et al.*'s algorithm [33], Shih and Wu's Euclidean DT (EDT-2) [24], FEED [12, 20], and Lucent's LLT [68]. This shows that the execution time is dependent on the content of the images; for example, the percentage of object pixels, as shown here, but also the border pixels.

and (3)), while the chessboard distance also takes diagonal directions into consideration. So, the d_4 or d_8 distance of two points is the number of steps required to reach either point from the other, where only city-block or chessboard movements can be used, respectively.

To obtain a better approximation for the ED, Rosenfeld and Pfaltz [1, 2] defined the octagonal distance (d_{oct}): the alternate use of the city-block and chessboard motions. Geometrically, the corresponding "disks" of d_4 , d_8 , and d_{oct} are diamonds, squares (see also Fig. (2)), and octagons. Hence, d_{oct} provides the best approximation of the ED out of these three distances.

Twenty years after Rosenfeld and Pfaltz [1, 2], Borgefors [21, 22] introduced her chamfer DT:

$$d_C((x_1, y_1), (x_2, y_2)) = \begin{cases} \Delta_y d_2 + (\Delta_x - \Delta_y) d_1 & \text{for } \Delta_y \leq \Delta_x \\ \Delta_x d_2 + (\Delta_y - \Delta_x) d_1 & \text{for } \Delta_y > \Delta_x \end{cases} \quad (7)$$

where $\Delta_x = |x_1 - x_2|$ and $\Delta_y = |y_1 - y_2|$. Optimal values should be chosen for d_1 and d_2 (under the assumption: $d_2 < 2d_1$) to approach the ED as well as possible. What the optimal values are depends on the application at hand and the trade-off between computational complexity and accuracy. On how to optimize the chamfer DT, we refer to the original article of Borgefors [21, 22]. An alternative heuristic to obtain optimal values for d_1 and d_2 can be found in [54], which is a patent application.

Note that Borgefors' [21, 22] elegant chamfer DT can be applied with several metrics, such as the city block (with $d_1 = 1$ and $d_2 = \infty$) and chessboard (with $d_1 = d_2 = 1$) metrics; see also Eqs. 5 and 6 (cf. Eq. 7). Figs. (2) and (3) present visualizations of chamfer DT with $d_1 = 3$ and $d_2 = 4$, as it was introduced in [21, 22]. Since its introduction until this very day, the algorithm provided in Appendix 1 of [21] frequently

has been applied both in science and industry. This illustrated by numerous scientific articles as well as by various patent applications; for example, [54–57]. A reason for this its time complexity of $O(nm)$, with nm being the number of pixels in the image.

Borgefors' algorithm requires an initialization, as defined by Eq. 4. Subsequently, the algorithm uses a forward and a backward pass that replace $DI(x, y)$ in DI , as follows:

```

% Forward pass:
for y = 1 to Y - 1
  for x = 0 to X - 1
    DI(x, y) = min{DI(x - 1, y - 1) + d2,
                  DI(x - 1, y) + d1,
                  DI(x + 1, y - 1) + d2,
                  DI(x, y - 1) + d1,
                  DI(x, y)}
% Backward pass:
for y = Y - 1 to 1
  for x = X - 1 to 0
    DI(x, y) = min{DI(x, y + 1) + d1,
                  DI(x - 1, y + 1) + d2,
                  DI(x + 1, y) + d1,
                  DI(x + 1, y + 1) + d2,
                  DI(x, y)},

```

(8)

where d_1 and d_2 depend on the metric of choice.

In 1980, Per-Erik Danielsson [17] proved that near-to Euclidean DI can be generated by effective sequential algorithms. Although neither generic nor as elegant as other algorithms (e.g., Borgefors' algorithm, see also Eq. 8, and the algorithm of Maurer *et al.* [33]), on approaching the ED it outperformed all the other algorithms produced so far, including Borgefors' later on developed algorithm; see Fig. (4). In the worst case Danielsson's algorithm has an error that is only a fraction of the grid constant, as Danielsson explains nicely himself [17]. Being that close to the ED, throughout the years this algorithm has become the algorithm to beat. This makes it, even more than 30 years after its introduction, until this very day, an algorithm that has been applied often both in science and industry (cf. [43, 54–56]).

The precision of Danielsson's algorithm has its downside. It uses a descriptor consisting of two com-

ponents: $|x_1 - x_2|$ and $|y_1 - y_2|$, which increases the algorithm's computational complexity; see also Table 3. As shown in Eq. 9, Danielsson had to modify its raster scanning. This increases the computational complexity of the algorithm even further. So, although approximating the ED closely, its computational complexity is a problem for various application areas.

Danielsson's algorithm used during both the initialization and the two scans over the image, a vector value ($DIv(x, y)$) per pixel. Here, the norm of the vector is its distance. These vectors are initialized as $(0, 0)$ for object pixels and (Z, Z) for background pixels, where Z is an large enough integer. The two scans, each requires three passes over each row:

```

% First picture scan:
for y = 1 to Y - 1
  for x = 0 to X - 1
    DIv(x, y) = min{DIv(x, y),
                  DIv(x, y - 1) + (0, 1)}
  for x = 1 to X - 1
    DIv(x, y) = min{DIv(x, y),
                  DIv(x - 1, y) + (1, 0)}
  for x = X - 2 to 0
    DIv(x, y) = min{DIv(x, y),
                  DIv(x + 1, y) + (1, 0)}
% Second picture scan:
for y = Y - 2 to 0
  for x = 0 to X - 1
    DIv(x, y) = min{DIv(x, y),
                  DIv(x, y + 1) + (0, 1)}
  for x = 1 to X - 1
    DIv(x, y) = min{DIv(x, y),
                  DIv(x - 1, y) + (1, 0)}
  for x = X - 2 to 0
    DIv(x, y) = min{DIv(x, y),
                  DIv(x + 1, y) + (1, 0)}.

```

(9)

Finally, the DI is calculated as $DI(x, y) = |DIv(x, y)|$.

After almost twenty years [17], Coiras *et al.* [23] introduced hexadecagonal region growing, which was an interesting alternative for the existing approaches. Figs. (2) and (3) visualize the DT based on Coiras' hexadecagonal region growing (see also Eq. 10), including its errors. Their work continues part of the work that was done by Kulpa and Kruse, 15 years

before [58]. In their article “Algorithms for circular propagation in discrete images” Kulpa and Kruse [58] present several algorithms. In the last section of their article, they mention that a “. . . schema can be called ‘hexadecagonal’ . . .” but do not provide an algorithm for it. Coiras *et al.* [23] analyzed this hexagonal DT and, subsequently, provided an algorithm for the empirical hexagonal growth presented in [58].

Similar to the original work of Rosenfeld and Platz [1, 2], Coiras *et al.* [23] also proposed a combination of d_4 and d_8 growth. Coiras *et al.* [23] used the identification of vertex pixels for vertex growth inhibition. This resulted in an approximation of the EDT up to 97.4%, at least so they claim (cf. Figs. (2) and (3) and Tables 1 and 2). As such, it approximates the EDT better than the chamfer 5-7-11 model, introduced in [21, 22] that served as the ‘standard’ for hexadecagonal distance for more than a decade. Coiras *et al.*’s algorithm for hexadecagonal growth [23] is defined as follows:

```

for i = 1 to R
  for o ∈ β(O)
    if ¬(o! = V ∧ i mod 5 = 0 ∧ i mod 45! = 0)
      if (i mod 2 = 0 ∧ i mod 12! = 0 ∧
          i mod 410! = 0)
        then
          grow o with d8
        else
          grow o with d4,

```

(10)

where R denotes the number of iterations (or the radius of the region growing process), $\beta(O)$ denotes the boundary pixels of object O , and \mathbf{V} denotes *vertex* (i.e., the point opposite to and farthest from the base in a figure). Further, note that this algorithm can be easily optimized such that modulus computations are required only once per iteration i instead of once per boundary pixel $o \in \beta(O)$.

Although the principle underlying hexadecagonal region growing is interesting, its performance is disappointing. Danielsson’s algorithm (see Eq. 9 is only 20% slower than chamfer 3-4 (see Eqs. 7 and 8), Coiras *et al.*’s algorithm (see Eq. 10) requires $2\times$ the time chamfer 3-4 needs. For more information on tim-

ing results, we refer to Table 3 and to Fig. (4), which both provides a comparison with various other algorithms.

All the previously described algorithms are based on raster scanning or region growing using only information from a limited area around each considered pixel. In that way they achieved time complexity $O(nm)$, with nm being the number of pixels in the image. It also means that they can be extended to 3D and higher dimensional images and also images with anisotropic pixels and voxels. But in that way they could not overcome the problem of disconnected (Euclidean) Voronoi regions. So, they all produce approximations of the EDT, in some cases, like Danielsson’s, this can be described as semi-exact EDT in the sense that for most pixels an exact EDT is achieved but for a small fraction of the pixels a (slightly) wrong value is delivered; see also Tables 1–3. Many authors have developed extensions to the above type of algorithms to correct this situation.

In 1998, Shih and Liu [59] presented their method to obtain EDT. They started with four scans of the image. This produced a similar result as Danielsson’s algorithm [17]; see also Eq. 9. Next, a look-up table method was used to correct the wrong pixels. For a large majority of cases, they were able to determine exact EDT. One year later, Cuisenaire and Macq [60] also introduced an exact EDT. First, they calculated an approximate EDT, using ordered propagation by bucket sorting. This procedure produces a result similar to Danielsson’s [17]. Second, they applied neighborhoods of increasing size to improve. However, these and similar approaches lead to complicated algorithms with a high time complexity.

In parallel to the above developments, so called independent scanning algorithms were developed. This principle was devised by Rosenfeld and Pfaltz [1, 2]. They started with processing each row independently from each other calculating for each background pixel its squared ED to the nearest object pixel in the row. Then they processed each resulting column independently from the others in a complicated way to produce the final 2D EDT. The idea behind this approach was that in principle an exact EDT with time complexity of $O(nm)$ could be reached provided that for each column the number of feature pixels taken into account could be reduced to order $O(n)$. Progress was made into that direction but not achieved in the previous century. The resulting algorithms did achieve exact EDT but not the desired $O(nm)$ and were rather complex.

All this academic research resulted in a wide range of applications, where DT were applied. Generally speaking, chamfer distance was applied particularly often. Probably mainly because many variants of them were developed; so, they became well known. Also the accuracy could be improved by increasing the size of the considered neighborhood around each pixel until that was sufficient for a particular application.

In the last decade of the previous century, DT found their way to applications such as route planning [61] and (robot) navigation [26], collision prevention [27], handwriting recognition [62], image segmentation [9] (see also Fig. (1)), skeletonization [63], Voronoi tessellations [30, 64], Watershed algorithms [65], and MRI data analysis [66]. Next, we will discuss some prominent academic research as reported in the last decade and refer to advances made on existing applications and the introduction of new applications, compared to those just mentioned.

4. THE LAST DECADE

Since their introduction by Rosenfeld and Platz [1], DT have received a heavily fluctuating amount of attention throughout the years. With the start of this century, however, DT and EDT in particular, have again gained in interest. In addition to established names in the field, a number of new names have reported their work in the field. Some of this work will be depicted in this section. Again (cf. Section 3), we will refrain from providing an exhaustive review, as this is far beyond the scope of this article. For an excellent recent review, we refer to [4]. Alternatively, we will briefly denote some of the most noteworthy work done on DT in the last decade.

Shortly after [59] and [60], Costa *et al.* [67] presented a method to determine EDT, using the concept of exact dilations. Their work was closely followed by Borgefors and colleagues, who presented several DT in two special issues of journals: [44, 45].

In the same year as FEED was launched [20], Shih and Wu [24] introduced their two scan method, with which they claimed to be able to obtain true exact EDTs. However, their algorithm does not do so in all cases (cf. Figs. (2) and (3)). Van den Broek *et al.* [19] determined that their claim was only justified in roughly 99% of the cases. This is also confirmed by the timing and error results reported earlier in this article; see Table 3.

Early in this century, much progress was made with the independent scanning approach to obtain exact EDT in a fast way; see also Table 3 and Fig. (4). In 2003, Maurer, Qi, and Raghavan [33] introduced an EDT for arbitrary dimensions (cf. [21, 32, 34]). Besides using independent scanning also called dimensionality reduction, this algorithm is based on Breu *et al.*'s partial Voronoi diagram calculation [18]. Hence, Maurer *et al.*'s algorithm can be regarded as a generalization of the algorithm by Breu *et al.* to arbitrary dimensions. Their general recursive EDT algorithm produces the squared EDT for isotropic voxels of arbitrary dimensions. For a fixed number of dimensions they indicate that it is easier using consecutive code loops for which they also provide optimizations. Note that all computations can be implemented in integer arithmetic. Further they also provide an adaption to anisotropic voxels, possibly requiring floating point operations. Besides using the Euclidean or L_2 metric, the algorithm can also be adapted to any other L_p metric, like the city-block and chessboard metrics. The time complexity is proven to be $O(N)$, with N being the number of voxels.

In 2009 Lucet [68] presented several sequential exact EDT algorithms based on fundamental transforms of convex analysis: the Legendre Conjugate or Legendre-Fenchel transform and the Moreau envelope or Moreau-Yosida approximate. The two new algorithms also use dimensionality reduction and also achieve $O(nm)$ time complexity (cf. Table 3 and Fig. (4)). The LLT algorithm is applicable to any image, the less general NEP algorithm requires convex data to function correctly but is then faster than the LLT algorithm. Both methods can also be extended to arbitrary dimensions and can be implemented in integer arithmetic.

One year after [33], Schouten and Van den Broek [20] presented their Fast Exact Euclidean Distance (FEED) transformation. With FEED, they introduced an algorithm, which obtained a true exact EDT in a computationally cheap way, see also Figs. (2)–(4) and Tables 1–3.

The naive implementation of FEED is rather straightforward. First, FEED is initialized as follows:

$$DI(b) = \text{if } (b \in O) \text{ then } 0 \text{ else } \infty, \quad (11)$$

where b are background pixels and O is the set object pixels o ; see also Eq. 1. Subsequently, it calculates the EDT starting directly from its definition (see Eq. 1), or

rather its inverse:

```

foreach  $o \in O$ 
  determine:  $A_o$ 
  update: foreach  $a \in A_o$  do
     $DI(a) = \min\{DI(a), ED^2(o, a)\}$ ,

```

where A_o is the area where o should feed distances to.

To avoid square roots, ED^2 is used instead of ED . Further, note that with $o \in O$ in Eq. 12, only the border pixels of O need to be considered because

$$\min\{ED(b, o)\} == ED(b, o_b), \quad (13)$$

where o_b is a border pixel of O (i.e., having at least one of its four 4-connected pixels in the background). However, to make FEED truly computationally efficient, several additional speed ups are required. These fall beyond the scope of the current article. For specifications of FEED's speed-ups we refer to [20, 28].

FEED has been compared with a broad range of other algorithms, among which those that are mentioned in the current paper. See also Figs. (2) and (3) for a visualization of five algorithms on the same images and Table 1–3 as well as Fig. (4) for timing results. This collection of algorithms comprised both exact EDT, excellent approximations of EDT, and rough estimations of EDT, as illustrated in Figs. (2) and (3) and calculated as shown in Tables 1–3. Time after time FEED proved to be not only the fastest exact EDT but also faster than all approximations of EDT it was benchmarked against. The current timing results conform this again; see Table 3 and Fig. (4). Moreover, even when compared with rough estimation such as those defined by Eqs. 5–6, FEED performed excellently (cf. Table 3 and Fig. (4)).

However, FEED has its downside as well. Seen the requirements on reducing the size of A_o to obtain minimal execution time (see Eq. 12), the process for it can not be optimized for a given image. It can only be optimized for a sample of the type of images one wants to process for certain applications. Thus, the time complexity cannot be proven in a theoretical way. FEED's speed has only been proved through experimental results (cf. Table 3). Although this has been done repeatedly [12, 19, 20, 28, 37, 52, 69] (cf. Table 3 and Fig. (4)),

in contrast with the other methods, it cannot be proved that the arithmetic complexity is $O(nm)$.

Currently, after more than 45 years of research on DT, a large number of DT algorithms is available. For eight of them, we have tested our implementations on a set of 160 artificial generated images of size 524×524 pixels; see Table 3. The number, size, position and type of objects were varied to cover a range from about 3% to 78% object pixels. In Table 3 accuracy and timing results are given, as determined on a PC with an Intel Core 2 Duo E6550P[®] 2.33GHz processor (2×32 KB data and 2×32 KB instruction L1 cache, 4096 KB L2 cache) and with 1024 MB memory, using the *gcc* compiler. Although all the algorithms have a theoretical proven or experimentally indicated time complexity of $O(n^2)$, there is a large variation in execution between them, as is shown in Table 3 and illustrated in Fig. (4). In Fig. (4) the execution times are given as function of the percentage of object pixels. This shows that the execution time is dependent on the content of the images. For certain algorithm it decreases, for others it increases with increasing percentage of object pixels.

Judging from the accuracy and timing information alone (see Table 3), one would expect that the use of the non-exact algorithms will decrease rapidly. However, there are other factors playing a role, such as the ease of implementation or the integration of the DT algorithm with other algorithm in the processing chain of an application. For example, in our experience Danielsson [17] is much easier to implement starting from its publication than Maurer *et al.* [33], FEED [20], and Lucent's LLT [68] are. Regarding an application, if that would require only distances up to a certain maximum M to be determined and larger distances classified as "large", than HexaD [23] and FEED [20] can be easily adapted to provide that with an increased speed. This by limiting R in Eq. 10 to M and by restricting A_o in Eq. 12 directly to a circle with radius M . The other methods would still require that the whole image is processed; consequently, for these methods there are hardly any means to speed up processing. Further, when speed is of utmost crucial, exploitation of modern hardware developments like multi-core CPUs or GPUs might be different for the different types of DT algorithms. The above arguments made a elaborate discussion of the development of the non-exact ED algorithms useful, despite the existence of very fast exact ED algorithm.

The vast amount of academic research conducted on DT since the start of this century resulted in multiple algorithms, which computational complexity is

$O(nm)$ (even in nD), as is discussed in the current section. These developments also resulted in a further extension of the range of applications (cf. Section 3), including: route planning [70], (robot) navigation [28, 56], video surveillance [69, 71], handwriting recognition [72], internal radiation therapy [13], image segmentation [10, 11] (see also Fig. (1)), skeletonization [73], Bouligand-Minkowsky fractal dimension [74], neuromorphometry [73, 75], MRI data analysis [76], and volume rendering [77]. Some application areas of DT were simply explored in alternative ways, some were brand new (cf. Section 3). Next, we will discuss how research conducted in industry contributed to the developments in DT by way of reviewing the patent applications granted in the last 20 years.

5. PATENT APPLICATIONS

DT are a rather fundamental concept in computational geometry and, consequently, in computer science in general. This makes research on DT interesting for a broad range of applications (see also Sections 3 and 4), which makes DT interesting for industry. However, work in this field concerns improvements of algorithms either in precision of the approximation of EDT or in speed / computational complexity [25] (cf. Tables 1–3 and Fig. (4)), where many other factors often determine both the application’s speed and precision. Consequently, the infringement of a patent on DT may be hard to detect. This raises questions to whether patent applications in this field can be of sufficient economical value. In sum, the endeavor of filing patent applications on DT has its pros and cons, which makes an analysis on them valuable.

This section will report on an exhaustive search for key patent applications on DT specific. We will refrain from reporting on an exhaustive search for patent applications that apply DT to assure the appropriate narrow scope for this article. Dozens of patent applications were found concerning either DT themselves, their application, or related techniques. A selection of 10 key patent applications that emerged will be discussed in order of publication. Note that various others are mentioned throughout this article as well.

In 1990, Fujioka and Watanabe [78] (Kabushiki Kaisha Toshiba, Kawasaki, Japan) had their patent application granted, which they had filed two years before. Their invention concerned a “method and apparatus for obtaining an object image and distance data of a moving object”. This is by far the oldest patent ap-

plication on DT the authors are acquainted with. The description of their invention touches upon the essence of DT algorithms. Fujioka and Watanabe [78] state that their invention comprises: “an image memory for storing a reference monitor image of a designated monitor region”, which we have denoted as I in Sections 2 and 3 and “a distance map memory for storing a distance map”, which we have denoted as DI in Sections 2 and 3. They continue by elaborating on both I and DI : “the monitor image of the designated monitor region and the distance map comprising a plurality of blocks having distance data from a predetermined reference point to points in the monitor region corresponding to each of the blocks”. Further, they specify “an object image detector for detecting an object image of the moving object . . .”, as it is nowadays considered a standard procedure (cf. [28, 69, 71]). Lastly, they state that it is required to have “a distance detector for detecting the distance from the reference point to the moving object, from the detected object image and the distance map . . .”. Taken together, this patent application touches upon the essence of DT and illustrates one of its core applications: navigation and object detection. Nevertheless, the authors are not acquainted with even a single reference in scientific literature to this patent application. So, it seems that this is the first time this work was unveiled to the scientific community.

Five years after Fujioka and Watanabe [78], Bick and Giger [16] (Arch Development Corporation, Chicago, IL, USA) had their patent granted on “a method for the automated segmentation of medical images, including generating image data from radiographic images of the breast.” Medical image segmentation [9–11, 13–15] is one of image processing’s most important fields of application. A significant amount of progress has been achieved in this field, as is illustrated by this patent application. 15 years ago the segmentation processing pipeline as introduced in this patent (see also Fig. (1)) made it worth granting the patent application. Nowadays, such a processing pipeline is considered common knowledge in (medical) image processing (cf. [14]). This patent application is, as the inventor stated, applicable to breast mammograms, including the extraction of the skin line, the correction for non-uniform exposure conditions, hand radiographs and chest radiographs. However, its applications go well beyond this and stretch over all the application areas that require image segmentation, with at most minor changes to a processing pipelines needed (cf. Fig. (1)).

Sekiguchi, Sano, and Yokoyama [79] filed their patent “Method of and apparatus for region detec-

tion in three dimensional voxel data” in 1994 and got it accepted for publication in 1996. As such, it was the only patent application published on 3D DT that we found that was published in the previous century (cf. [32, 35–37]). That this were indeed the early days of bringing 3D DT to practice is illustrated by the inventors, who state that the goal of their invention is “... minimizing the human operation to be achieved for the extraction processing to guarantee reliability of extraction.” This concerns region extraction or segmentation of medical images (e.g., attained from X-ray, CT and MRI) [14, 15]. This was also the case with the patent of Bick and Giger [16]; however, Sekiguchi *et al.* [79] invented another processing pipeline, adapted to 3D images.

In 1999, a patent of Rucklidge and Jaquith [80] of Xerox Corporation (Stamford, CT, USA) was published. It concerned a fast, low-overhead implementations of a powerful, reliable image matching engine based on the Hausdorff distance [81]. This work follows earlier work of Rucklidge and colleagues (e.g., [81]). The image matching engine is fed a pattern that has to be recognized in images. Subsequently, a database of images can be supplied in which the pattern needs to be recognized if present. The image is preprocessed with the processor using various morphological dilation operations (cf. Fig. (1)). This can produce a set of preprocessed images. Subsequently, the engine conducts a hierarchical search for the pattern in the database of images. To limit the engine’s computational load, DT are applied within bounding boxes. They apply this principle throughout their complete processing pipeline. Moreover, they suggest fast recursive algorithms and parallel implementations, which both stresses the high computational complexity of such engines.

Braspenning *et al.* [82] of Koninklijke Philips Electronics N.V. (Eindhoven, the Netherlands) proposed a method to segment digital images; see also Fig. (1). As they denote themselves, this is a basic procedure in digital image processing, which is required for a lot of applications. Please see Fig. (1) for its processing pipeline. It extends current work in that it assigns to each $I(x, y)$ the objects closest to it (cf. the note on multi class DT in Section 2). Subsequently, it goes one step further and assigns not only the object each pixel is closest to but also to which side of this object.

Four years after Braspenning *et al.* [82], Liang and Bogoni [83] of Siemens Corporation (Iselin, NJ, US) also had their patent, titled: “System and method for toboggan-based object segmentation using distance

transform” granted; see also Fig. (1). Their main contribution lays in that they apply DT not on multi class, but still binary, images but on (arbitrary) multi-level images, where each level of intensity could denote a dimension, as the authors phrase. Liang and Bogoni [83] propose to define a number of thresholds, which reduces the number of levels (or dimensions) to the preferred one. So, in principle, they simply propose to quantize a multi-level intensity image to a lower-level intensity image. Although quite straight forward, this is indeed a procedure that can show its use in practice.

In both the same year as Liang and Bogoni [83] (i.e., 2009) and one year after this (i.e., 2010), Lee and Phan [84, 85] (Bellevue, WA, USA) had two related patent applications granted on respectively a “image region partitioning using pre-labeled regions” and a “method for adaptive image region partitioning and morphological processing”, such as dilation (or dilatation) and erosion (cf. [6–8]). Their work introduces a “zone of interest (ZOI)”, a bounding box that limits the neighborhood in which calculations have to be executed. This principle significantly speeds up the generation of distance maps. A similar principle is used with FEED [12, 20] to speed up that algorithm. The ZOI is created sequentially in two passes. Moreover, this method also allows multi class DT; see also Section 2. It further extends this principle as it allows the use of different metrics for the distinct objects or classes.

In the same two years as Lee and Phan [84, 85], Bitar [55] and Bitar and Marty [54, 56] got three of their patents published. Here, we will discuss one them: a “Method for determining chamfer mask coefficients for distance transform”. This patent is very interesting as it continues the work of Borgefors and colleagues [21, 22] on chamfer DT. As elegant as chamfer DT may be, the calculation of parameters d_1 and d_2 (see also Eq. 7) is costly; see also 3. Bitar and Marty [55] introduced an algorithm to reduce the computational complexity of these calculations significantly.

Also in 2010, Porikli [86] (Mitsubishi Electric Research Laboratories, Inc., Cambridge, MA, USA) got their patent application on a “method for generating distance maps using scan lines” granted. Unlike the conventional approaches their method extracts the minimum distances with no explicit distance computation, using either multi-directional dual scan line propagation or wave propagation methods. The precision of the dual scan propagation method can be set according to the available computational power. Alternatively, a wavefront from object points can be started that propa-

gates outwardly at each step, while recording the number of steps as the minimum distance (cf. [17, 87–90]). However, unlike for example FEED [20], the computational load of Porikli’s algorithm does not depend on the number of object points, which makes it constant in performance. Please note that, in addition to the patent, this work is also well described in [47].

Taken together, in the last decade various interesting DT patent applications have been granted. This section only discussed a handful of them but already illustrated how vivid this topic of research is, also for industrial purposes. The patent applications discussed are not so distinct from the developments discussed in the previous two sections. Moreover, this section illustrated that up to now the academic research conducted in the first 30 years still serve as the field’s foundation (cf. Section 3 and this section).

Also with patent applications, issues as speed-accuracy trade-off and multi class DT are a topic of interest. This former is well illustrated by the work of Bitar and Marty [55] who present an algorithm to optimize the calculation of chamfer parameters d_1 and d_2 ; see also Eq. 7–8. Among other issues, also the issue of multi-level intensity images has been challenged in the patent application of Liang and Bogoni [83]. Last and perhaps most noteworthy is the application of DT for image segmentation; see also Fig. (1). This illustrates the fundamental nature of DT, as image segmentation itself is already considered as a fundamental operation in image processing.

6. CURRENT & FUTURE DEVELOPMENTS

With this article we have tried to provide a very brief introduction into distance transforms (DT) in Section 2, including a specification of the most prominent algorithms in the field in Section 3. Next, a few interesting results from the first decade of this century were discussed (Section 4). Moreover, through two data sets (see Figs. (2) and (3)) and their accompanying statistics (see Table 1 and 2) as well as through a benchmark (see Table 3 and Fig. (4)), the intriguing complexity of the fundamental concepts DT and DI is well illustrated. Eight noteworthy algorithms (i.e., city block [1,2], Danielsson [17], chamfer 3-4 [21,22], hexdecagonal region growing [23], Maurer *et al.* [33], EDT-2 of Shih and Wu [24], FEED [12, 20], and Lucent’s LLT [68]) developed to calculate DT/DI illustrate this; see Fig. (4). Finally, we have discussed the key patent applications that emerged on DT in Sec-

tion 5. This last section illustrated that up to the current day the work conducted in the first 30 years still form the field’s foundation (cf. Section 3 and 5).

On one hand, distance transforms (DT) are a basic operation in computational geometry. On the other hand, they are applied within various applications; see for example, [53, 69, 91]. In the last case, DT are either applied by themselves or as an intermediate method such as: route planning and (robot) navigation [26, 28, 56, 61, 70], collision prevention [27], video surveillance [69, 71], internal radiation therapy [13], handwriting recognition [62, 72], (medical) image segmentation [9–11, 13–15] (see also Fig. (1)), skeletonization [63, 73], Voronoi tessellations [30, 64], Bouligand-Minkowsky fractal dimension [74], Watershed algorithms [65], neuromorphometry [73, 75], MRI data analysis [76, 92], and volume rendering [36, 77], to mention but a few.

DT can be applied in an arbitrary number of dimensions [21, 32–34] and even for image sequences (i.e., video) [12]. However, most often they are applied in 2D or 3D [32, 35–37]. The reason for this is simple; most digital images are 2D, some are 3D, as it is well illustrated by the areas of application just mentioned. For 4D and higher dimensional spaces, fewer direct applications are apparent and, hence, few patent applications will be filed in this area. Therefore, this article focused on 2D DT, which is an established field on its own [32], as we have seen throughout the article.

This article illustrated that the true groundbreaking work on DT has been done in the previous century (cf. Sections 3 and 4 and see Fig. (4)). However, it must be acknowledged that also in the last 10 years, the research on DT has been vivid, as has been illustrated by the interesting articles published and the key patents that have emerged; see also Sections 4 and 5. Note that only a small sample has been included of the research (i.e., both articles and patent applications) conducted in the previous and the current century.

When comparing Sections 4 and 5 there appears to be hardly any overlap between the authors of scientific articles and the inventors of granted patents. Exceptions are the work of Rucklidge and colleagues [80] and Porikli [86]. Both of them filed their algorithms well before publishing it (e.g., [47]) and saw their patent application granted after their article was published. So, in general it seems to be of interest to consult recent patents in computer science. For sure, it is worth the effort for computational geometry, for DT in particular, as has been illustrated by the current article.

DT are a core concept in computational geometry and, consequently, in computer science in general. On one hand, this sets DT at the roots of a broad range of applications. On the other hand, having its principles well defined, progress in this field concerns improvements in precision or speed. Most often both the application's speed and precision is determined by many factors; consequently, the infringement of a patent on DT is sometimes judged as hard to detect. Nevertheless, DT as research topic is vivid and the interest of both science and industry undoubtedly paves the way to a bright future for it.

ACKNOWLEDGMENTS

The authors thank Khurshid Zaman, the editor of the journal, for inviting us to write this article. Moreover, we acknowledge Ms. Munira Tuba and Ms. Noureen Azher for their cooperation, patience, and prompt replies on our questions. We gratefully acknowledge the three anonymous reviewers who each provided us detailed feedback of the highest possible quality on an earlier version of this article. Their comments and suggestions have indeed improved this article substantially; hence, although anonymous it needs to be stressed that their contributions have been extremely valuable. Further, we thank Frans van der Sluis (Human Media Interaction, University of Twente) for his comments on a previous version of this article. Last, we gratefully acknowledge Lynn Packwood for her careful proof reading (Human Media Interaction, University of Twente).

CONFLICT OF INTEREST

The authors have no conflicts of interest to declare.

REFERENCES

- [1] Rosenfeld A, Pfaltz JL. Sequential operations in digital picture processing. *J ACM* 1966; 13(4):471–94.
- [2] Rosenfeld A, Pfaltz JL. Distance functions on digital pictures. *Pattern Recognit* 1968; 1(1):33–61.
- [3] Rosenfeld A. Some notes on digital triangles. *Pattern Recognit Lett* 1983; 1(3):147–50.
- [4] Fabbri R, da F. Costa L, Torelli JC, Bruno OM. 2D Euclidean distance transform algorithms: A comparative survey *ACM Comp Surveys* 2008; 40(1):Article 2.
- [5] Klette R, Rosenfeld A. *Digital geometry: Geometric methods for digital image analysis*. Morgan Kaufmann Publishers: San Francisco 2004.
- [6] Cuisenaire O. Region growing Euclidean distance transforms. *Lect Notes Comput Sci (Image Anal Process)* 1997; 1310:263–70.
- [7] Cuisenaire O. Locally adaptable mathematical morphology using distance transformations. *Pattern Recognit* 2006; 39(3):405–16.
- [8] Soille P, Talbot H. Directional morphological filtering. *IEEE Trans Pattern Anal Mach Intell* 2001; 23(11):1313–29.
- [9] Deklerck R, Cornelis J, Bister M. Segmentation of medical images. *Image Vision Comput* 1993; 11(8):486–503.
- [10] Mohana Rao KNR, Dempster AG. Modification on distance transform to avoid over-segmentation and under-segmentation. In: *Proc VIPromCom: The 4th EURASIP-IEEE Reg 8 Int Symp Video/Image Process Multimedia Commun*. Zadar, Croatia: Croatian Soc Electron Marine – ELMAR; 2002; 295–301.
- [11] Rogowska J. 5 (Part II). In: Bankman IN, Ed. *Overview and Fundamentals of Medical Image Segmentation*. 2nd ed. Burlington, MA, USA: Academic Press / Elsevier, Inc. 2008; 73–90.
- [12] Schouten TE, van den Broek EL. Incremental Distance Transform (IDT). In: Erçil A, Çetin M, Boyer K, Lee SW, Eds. *Proc 20th IEEE Int Conf Pattern Recognit (ICPR)*. Istanbul, Turkey: Piscataway, NJ, USA: IEEE Comput Soc Press 2010; 237–40.
- [13] van den Broek EL, Schouten TE. Modeling Internal Radiation Therapy. In: *BioInformatics 2011: Proc Int Conf BioInf Models Meth Algorithms*. Rome, Italy: INSTICC; 2011; [in press].
- [14] Makram-Ebeid S, Rouet J-M, Fradkin M. Medical viewing system and image processing for integrated visualisation of medical data. *US0163357 (2005)*.
- [15] Zhang L. Distance transform based vessel detection for nodule segmentation and analysis. *US7747051 (2010)*.
- [16] Bick U., Giger ML. Automated method and system for the segmentation of medical images. *US5452367 (1995)*.
- [17] Danielsson PE. Euclidean distance mapping. *Comput Graph Image Process* 1980; 14(3):227–48.
- [18] Breu H, Gil J, Kirkpatrick D, Werman M. Linear time Euclidean distance transform algorithms. *IEEE Trans Pattern Anal Mach Intell* 1995; 17(5):529–533.
- [19] van den Broek EL, Schouten TE, Kisters PMF, Kuppens HC. Weighted Distance Mapping (WDM). In: Canagarajah N, Chalmers A, Deravi F, Gibson S, Hobson P, Mirmehdi M, Marshall S, Eds. *Proc IEE Int Conf Visual Inf Eng (VIE2005)*. Glasgow, United Kingdom: Wrightsons – Earls Barton, Northants, Great Britain 2005; 157–64.
- [20] Schouten TE, van den Broek EL. Fast Exact Euclidean Distance (FEED) Transformation. In: Kittler J, Petrou M, Nixon M, Eds. *Proc 17th IEEE Int Conf Pattern Recognit (ICPR 2004)*. Cambridge, United Kingdom 2004; 3:594–7.
- [21] Borgefors G. Distance transformations in arbitrary dimensions. *Comput Vision Graph Image Process* 1984; 27(3):321–45.
- [22] Borgefors G. Distance transformations in digital images. *Comput Vision Graph Image Process* 1986; 34(3):344–71.
- [23] Coiras E, Santamaria J, Miravet C. Hexadecagonal region growing. *Pattern Recognit Lett* 1998; 19(12):1111–7.
- [24] Shih FY, Wu YT. Fast Euclidean distance transformation in two scans using a 3×3 neighborhood. *Comput Vision Image Understanding* 2004; 93(2):195–205.
- [25] Hajdu A, Hajdu L. Approximating the Euclidean distance using non-periodic neighbourhood sequences. *Discrete Math* 2004; 283(1–3):101–11.

- [26] Kimmel R, Kiryati N, Bruckstein AM. Multivalued Distance Maps for Motion Planning on Surfaces with Moving Obstacles. *IEEE Trans Rob Autom* 1998; 14(3):427–36.
- [27] Rembold U, Dillmann R, Hertzberger LO, Kanade T. *Intelligent Autonomous Systems*. IOS Press: Amsterdam 1995.
- [28] Schouten TE, Kuppens HC, van den Broek EL. Timed Fast Exact Euclidean Distance (tFEED) maps. *Proc SPIE (Real Time Imaging IX)* 2005; 5671:52–63.
- [29] Coeurjolly D, Montanvert A. Optimal separable algorithms to compute the reverse Euclidean distance transformation and discrete medial axis in arbitrary dimension. *IEEE Trans on Pattern Anal and Mach Intell* 2007; 29(3):437–448.
- [30] Saito T, Toriwaki JI. New algorithms for Euclidean distance transformation of an n -dimensional digitized picture with applications. *Pattern Recognit* 1994; 27(11):1551–65.
- [31] Meijster A, Wilkinson MHF. A comparison of algorithms for connected set openings and closings. *IEEE Trans on Pattern Anal and Mach Intell* 2002; 24(4):484–494.
- [32] Hajdu A. Geometry of neighbourhood sequences. *Pattern Recognit Lett* 2003; 24(15):2597–606.
- [33] Maurer, Jr CR, Qi R, Raghavan V. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Trans Pattern Anal Mach Intell* 2003; 25(2):265–70.
- [34] Ragnemalm I. The Euclidean distance transform in arbitrary dimensions. *Pattern Recognit Lett* 1993; 14(11):883–8.
- [35] Jones MW, Barentzen JA, Sramek M. 3D distance fields: A survey of techniques and applications. *IEEE Trans on Visual and Comp Graph* 2006; 12(4):581–599.
- [36] Satherley R, Jones MW. Vector-city vector distance transform. *Comput Vision Image Understanding* 2001; 82(3):238–54.
- [37] Schouten TE, Kuppens HC, van den Broek EL. Three Dimensional Fast Exact Euclidean Distance (3D-FEED) Maps. *Proc SPIE (Vision Geom XIV)* 2006; 6066:60660F.
- [38] Bruno OM, da Fontoura Costa L. A parallel implementation of exact Euclidean distance transform based on exact dilations. *MMicroprocess Microsy* 2004; 28(3):107–113.
- [39] Chen L, Chuang HYH. An efficient algorithm for complete Euclidean distance transform on mesh-connected SIMD. *Parallel Comput* 1995; 21:841–52.
- [40] Crookes D, Brown J. I-BOL: A tool for image processing on transputers. *Transputer Appl Syst* 1993; 93:712–27.
- [41] Lee Y, Horng S, Kao T, Chen Y. Parallel computation of the Euclidean distance transform on the mesh of trees and the hypercube computer. *Comput Vision Image Understanding* 1997; 68(1):109–19.
- [42] Takala JH, Viitanen JO. Distance transform algorithm for Bit-Serial SIMD architectures. *Comput Vision Image Understanding* 1999; 74(2):150–61.
- [43] Bronstein A, Bronstein M, Devir Y, Weber O, Kimmel R. Parallel approximation of distance maps. *US0119120 (2010)*.
- [44] Borgefors G, Nyström I, Sanniti di Baja G. Discrete Geometry for Computer Imagery. *Pattern Recognit Lett* 2002; 23(6):[Special Issue].
- [45] Borgefors G, Nyström I, Sanniti di Baja G. Discrete Geometry for Computer Imagery. *Discrete App Math* 2003; 125(1):[Special Issue].
- [46] Hajdu A, Tóth T. Approximating non-metrical Minkowski distances in 2D. *Pattern Recognit Lett* 2008; 29(6):813–21.
- [47] Porikli F, Kocak T. Fast distance transform computation using dual scan line propagation. *Proc SPIE Real Time Imaging* 2007; 6496:649608-8.
- [48] Borgefors G. Distance Transformations on Hexagonal Grids. *Pattern Recognit Lett* 1989; 9(2):97–105.
- [49] Vacavant A. Fast distance transformation on two-dimensional irregular grids. *Pattern Recognit* 2010; 43(10):3348–3358.
- [50] Vacavant A, Coeurjolly D, Tougne L. Separable algorithms for distance transformations on irregular grids. *Pattern Recognit Lett* 2011; 32():[in press].
- [51] Michikawa T, Suzuki H. Sparse grid distance transforms. *Graph Models* 2010; 72(4):35–45.
- [52] Schouten TE, van den Broek EL. Fast Multi Class Distance Transforms for Video Surveillance. *Proc SPIE Real-Time Image Process* 2008; 6811:681107-11.
- [53] van den Broek EL, Schouten TE, Kisters PMF. Modeling human color categorization. *Pattern Recognit Lett* 2008; 29(8):1136–44.
- [54] Bitar E, Marty N. Method for determining optimal chamfer mask coefficients for distance transform. *US7583856 (2009)*.
- [55] Bitar E. Distance-estimation method for a travelling object subjected to dynamic path constraints. *US0031007 (2010)*.
- [56] Bitar E, Marty N. Device and method for signaling lateral maneuver margins. *US7733243 (2010)*.
- [57] Black MJ, Balan AO, Weiss AW, Sigal L, Loper MM, St. Clair TS. Methods and apparatus for estimating body shape. *US0111370 (2010)*.
- [58] Kulpa Z, Kruse B. Algorithms for circular propagation in discrete images. *Comput Vis Graph Image Process* 1983; 24(3):305–28.
- [59] Shih FY, Liu JJ. Size-invariant four-scan Euclidean distance transformation. *Pattern Recognit* 1998; 31(11):1761–6.
- [60] Cuisenaire O, Macq B. Fast Euclidean transformation by propagation using multiple neighborhoods. *Comput Vision Image Understanding* 1999; 76(2):163–72.
- [61] Zelinsky A. A mobile robot navigation exploration algorithm. *IEEE Trans Rob Autom* 1992; 8(6):707–17.
- [62] Lu Y, Shridhar M. Character segmentation in handwritten words – An overview. *Pattern Recognit* 1996; 29(1):77–96.
- [63] Kimmel R, Shaked D, Kiryati N, Bruckstein AM. Skeletonization via Distance Maps and Level Sets. *Comput Vision Image Understanding* 1995; 62(3):382–91.
- [64] Guan W, Ma S. A list-processing approach to compute Voronoi diagrams and the Euclidean distance transform. *IEEE Trans Pattern Anal Mach Intell* 1998; 20(7):757–61.
- [65] Meyer F. Topographic distance and watershed lines. *Signal Process* 1994; 38:113–25.
- [66] Hojjatoleslami SA, Kittler J. Region Growing: A New Approach. *IEEE Trans Image Process* 1998; 7(7):1079–84.
- [67] Costa LF. Multidimensional scale-space shape analysis. In: *Proc Int Workshop Synth Nat Hybrid Coding Three Dimension Imaging* 2001; 214–7.
- [68] Lucet Y. New sequential exact Euclidean distance transform algorithms based on convex analysis. *Image Vision Comput* 2009; 27(1-2):37–44.
- [69] Schouten TE, Kuppens HC, van den Broek EL. Video surveillance using distance maps. *Proc SPIE Real-Time Image Process* 2006; 6063:54-65.
- [70] Shih FY, Wu YT. Three-dimensional Euclidean distance transformation and its application to shortest path planning. *Pattern Recognit* 2004; 37(1):79–92.
- [71] Chen Z, Husz ZL, Wallace I, Wallace AM. Video object tracking based on a chamfer distance transform. In: *Proc IEEE Int*

- Conf Image Process (ICIP). San Antonio, TX, USA: IEEE Signal Process Soc 2007; III-357-60.
- [72] Zhuang Y, Zhuang Y, Li Q, Chen L. Interactive high-dimensional index for large Chinese calligraphic character databases. *TALIP* 2007; 6(2):Article 8.
- [73] Falcão, AX, da Fontoura Costa L, da Cunha BS. Multiscale skeletons by image foresting transform and its application to neuromorphometry. *Pattern Recognit* 2002; 35(7):1571-82.
- [74] Costa LF, Jr RMC. *Shape Analysis and Classification*. CRC Press, Inc.: Boca Raton 2001.
- [75] Costa LF, Manoel ETM, Faucereau F, van Pelt J, Ramakers G. A shape analysis framework for neuromorphometry. *Network-Comput Neural Syst* 2002; 13(3):283-310.
- [76] Saha PK, Wehrli FW. Measurement of trabecular bone thickness in the limited resolution regime of in vivo MRI by fuzzy distance transform. *IEEE Trans on Med Imaging* 2004; 23(1):53-62.
- [77] Pfister H, Lorensen B, Bajaj C, Kindlmann G, Schroeder W, Avila LS, *et al*. The transfer function bake-off. *IEEE Comput Graphics Appl* 2001; 21(3):16-22.
- [78] Fujioka A., Watanabe S. Method and apparatus for obtaining an object image and distance data of a moving object. US4908704 (1990).
- [79] Sekiguchi H., Sano K., Yokoyama T. Method of and apparatus for region extraction in three-dimensional voxel data. US5553207 (1996).
- [80] Rucklidge WJ, Jaquith EW. Fast techniques for searching images using the Hausdorff distance. 5999653 (1999).
- [81] Huttenlocher DP, Klanderman GA, Rucklidge WJ. Comparing images using the Hausdorff distance. *IEEE Trans Pattern Anal Mach Intell* 1993; 15(9):850-863.
- [82] Braspenning R.A.C., Ernst F.E., van Overveld C.W.A.M., Wilinski P. Segmentation of digital images. US6963664 (2005).
- [83] Liang J., Bogoni L. System and method for toboggan-based object segmentation using distance transform. US7609887 (2009).
- [84] Lee S.J.J., Phan T. Image region partitioning using pre-labeled regions. US7580556 (2009).
- [85] Lee S.J.J., Phan T. Method for adaptive image region partition and morphologic processing. US7813580 (2010).
- [86] Porikli FM. Method for generating distance maps using scan lines. US7809165 (2010).
- [87] Verbeek PW, Verwer BJH. Shading from shape, the eikonal equation solved by grey-weighted distance transform. *Pattern Recognit Lett* 1990; 11(10):681-690.
- [88] Wright MW, Cipolla R. Skeletonization using an extended Euclidean distance transform. *Image and Vision Comput* 1995; 13(5):367-375.
- [89] Tek H, Kimia BB. Symmetry maps of free-form curve segments via wave propagation. *Int J of Comput Vision* 2003; 54(1-3):35-81.
- [90] Rasche C. An approach to the parameterization of structure for fast categorization. *Int J of Comput Vision* 2010; 87(3): 337-356.
- [91] Chen M, Lu W, Chen Q, Ruchala K, Olivera G. Efficient gamma index calculation using fast Euclidean distance transform. *Phys Med Biol* 2009; 54(7):2037-47.
- [92] van Herk M, Kooy HM. Automatic three-dimensional correlation of CT-CT, CT-MRI, and CT-SPECT using chamfer matching. *Med Phys* 1994; 21(7):1163-78.