Weighted Distance Mapping (WDM)

Egon L. van den Broek[1,2], Theo E. Schouten[3], Peter M. F. Kisters[2], and Harco Kuppens[3]

[1] Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
T: +31 20 598 7743, F: +31 20 598 7653
egon@few.vu.nl http://www.few.vu.nl/~egon/
[2] Nijmegen Institute for Cognition and Information
P.O. Box 9104, 6500 HE Nijmegen, The Netherlands
T: +31 24 361 5476, F: +31 24 361 6066
p.kisters@gmail.com
[3] Nijmegen Institute for Computing and Information Science
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
{T.Schouten,H.Kuppens}@cs.ru.nl http://www.cs.ru.nl/~{ths,harcok}/

Abstract. A new method is introduced for describing, visualizing, and inspecting data spaces. It is based on an adapted version of the Fast Exact Euclidean Distance (FEED) transform. It computes a description of the complete data space based on partial data. Combined with a metric, a true Weighted Distance Map (WDM) can be computed, which can define a probability space. Subsequently, distances between data points can be determined. Using edge detection, borders (or boundaries) between categories (or clusters) of data can be found. Hence, Voronoi diagrams can be created. Moreover, the visualization of such WDMs provides excellent means for data inspection. Several examples illustrate the use of WDMs as well as their efficiency. So, a new, fast, and exact data analysis method has been developed that yields the means for a rich and intuitive method of data inspection.

1 Introduction

With the increasing amounts of information in the current society, the need for data mining becomes more and more important. Both automated and manual procedures were developed for these purposes. These procedures not seldomly rely on clustering techniques (introduced by [1]), used in a wide range of disciplines.

The clusters obtained through, for example, dilation and erosion operations, provide a way to describe the structure present in data, based on a certain feature representation. However, they rely on the availability of data for approximating an appropriate coverage of the corresponding data space.

In practice, often only partial data is available. Nevertheless, a description of the complete data space is required. In such a case, approximations have to be made concerning those parts of the data space that lack data. This paper presents a promising new approach: Weighted Distance Mapping (WDM), which provides the means to describe and inspect the complete data space, based on any arbitrary number of data points.

As will be shown in this paper, WDMs can be utilized for four purposes: (1) to describe the complete data space based on a limited number of data points that fill only a small part of the complete data space, (2) for rapid visualization of data spaces, (3) to determine distances between data, using any metric, and (4) for the extraction of edges (or boundaries) between categories. These four features can be useful in a wide range of applications; e.g., robot navigation [2,3] and segmentation of a data space based on valuable but limited experimental data (e.g., color assigned to color categories), as described in Section 7 and used by [4,5].

Before WDM is introduced, morphological processing and (Euclidean) distance transforms are briefly described. In addition, the Voronoi diagram is briefly discussed, as a primitive distance map. Next, an adapted version of the algorithm that provides Fast Exact Euclidean Distance (FEED) transformations [6] is introduced. It is applied to obtain the WDMs.

In Section 5, FEED is compared with the city-block distance, as a baseline, and with Shih and Wu's [7] 2-scan method, as a state-of-the-art fast Euclidean distance (ED) transform. Next, FEED is applied on the probabilistic categorization of a color space, based on experimental data. The paper ends with conclusions and a brief exposition of advantages and disadvantages of WDMs generated by FEED.

2 From morphological processing to distance transforms

The operations dilation (also named dilatation) and erosion are fundamental to morphological processing of images. Many of the existing morphological algorithms are based on these two primitive operations [8]. The processes of dilation and erosion are illustrated in Figures 1 and 2.

Given two sets A and B in $\mathbb{Z}^2$, the dilation of A by B, is defined as

$$A \oplus B = \{x \mid (B)_x \cap A \neq \emptyset\}, \tag{1}$$

where $(B)_x$ denotes the translation of B by $x = (x_1, x_2)$ defined as:

$$(B)_x = \{c \mid c = b + x, \text{for some } b \in B\} \tag{2}$$

Thus, $A \oplus B$ expands A if the origin is contained in B, as is usually the case.

The erosion of A by B, denoted $A \ominus B$, is the set of all x such that B translated by x, is completely contained in A, defined as

$$A \ominus B = \{x \mid (B)_x \subseteq A\} \tag{3}$$

Thus, $A \ominus B$ decreases A.

Based on these two morphological operations the *4-n* and the *8-n* dilation algorithms were developed by Rosenfeld and Pfaltz [9] for region growing purposes. These region growing algorithms are based on two distance measures: the

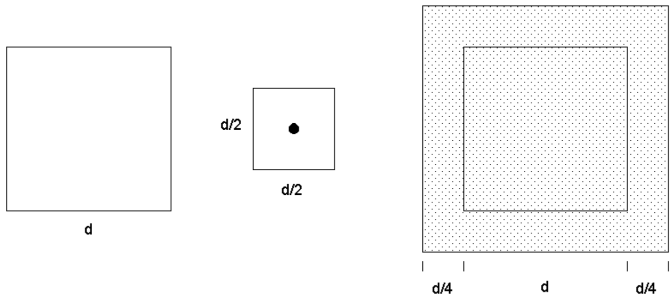


Fig. 1: The process of dilation illustrated. The left figure is the original shape A. The square in the middle is the dilation marker B (dot is the center). The middle of the marker runs over the boundary of A. The result of dilation of A by B ($A \oplus B$) is given by the solid shape on the right, in which the inner square projects the original object A.

city-block distance and the chessboard distance. The set of pixels contained in the dilated shape, for respectively *4-n* and *8-n* growth for an isolated pixel at the origin, are defined as:

$$C_4(n) = \{ (x, y) \in \mathbb{Z}^2 : |x| + |y| \leq n \}, \tag{4}$$

$$C_8(n) = \{ (x, y) \in \mathbb{Z}^2 : |x| \leq n, |y| \leq n \}, \tag{5}$$

where n is the number of iterations.

To obtain a better approximation for the ED, Rosenfeld and Pfaltz [9] recommended the alternate use of the city-block and chessboard motions, which defines the octagonal distance. The octagonal distance provides a better approximation of the ED than the other two distances.

Thirty years later Coiras et al. [10] introduced hexadecagonal region growing, again a combination of *4-n* and *8-n* growth (see Figure 3). The latter uses the

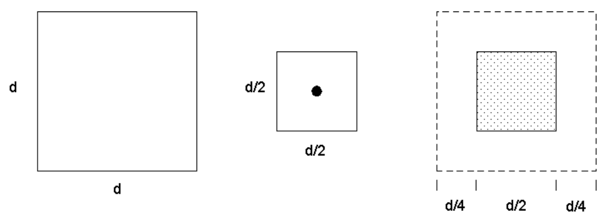


Fig. 2: The process of erosion illustrated. The left figure is the original shape A. The square in the middle is the erosion marker B (dot is the center). The middle of the marker runs over the boundary of A. The result of erosion of A by B ($A \ominus B$) is given by the solid shape on the right, in which the outer (dotted) square projects the original object A.

```
for k:=1 to R
    for every pixel p in the boundary
        if NOT [(p is a vertex) AND (k modulo 5=0)
            AND (k modulo 45!=0)]
            if [(k modulo 2=0) AND (k modulo 12!=0)
                AND (k modulo 410!=0)]
                grow p as 8-n
            otherwise
                grow p as 4-n
```

Fig. 3: Algorithm for hexadecagonal growth (source: [10]).

identification of vertex pixels for vertex growth inhibition. This resulted in an approximation of circular region growing up to 97.4%. In other words, Coiras et al. were able to determine the exact ED, in a vast majority of the cases.

3 Euclidean Distance transformation (EDT)

Region growing algorithms can be applied to obtain distance transformations. A distance transformation [9] creates an image in which the value of each pixel is its distance to the set of object pixels O in the original image:

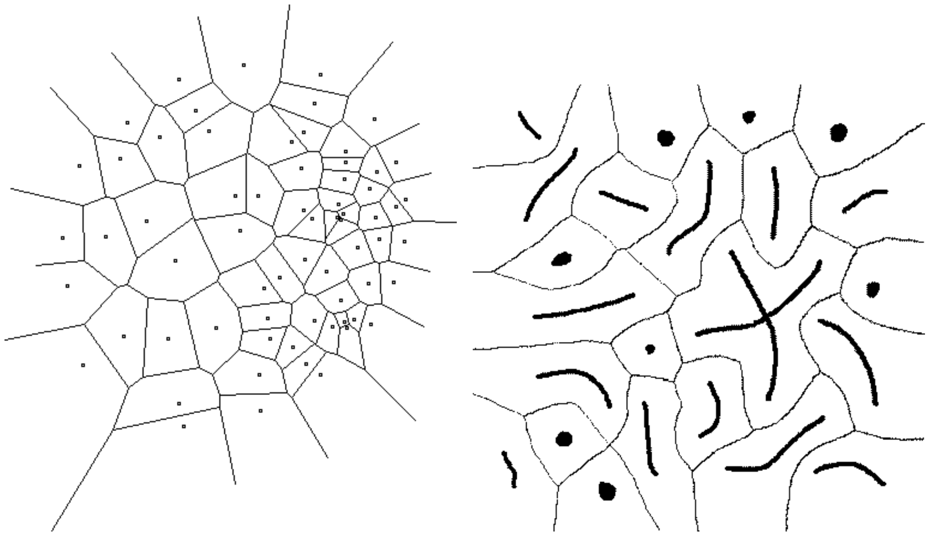
$$D(p) = \min\{\mathrm{dist}(p, q), q \in O\} \tag{6}$$

The Euclidean distance transform (EDT) has been extensively used in computer vision and pattern recognition, either by itself or as an important intermediate or ancillary method in applications ranging from trajectory planning [11] to neuromorphometry [12]. Examples of methods possibly involving the EDT are: (i) skeletonization [13]; (ii) Voronoi tessellations [14]; (iii) Bouligand-Minkowsky fractal dimension [15]; and (iv) Watershed algorithms [16], and robot navigation [2,3].

Several methods for calculation of the EDT have been described in the literature [17,9,3,18], both for sequential and parallel machines. However, most of these methods do not produce exact distances, but only approximations [19]. Borgefors [20] proposed a chamfer distance transformation using two raster scans on the image, which produces a coarse approximation of the exact EDT. To get a result that is exact on most points but can produce small errors on some points, Danielsson [21] used four raster scans.

In order to obtain an exact EDT, two step methods were proposed. Two of the most important ones are:

- Cuisenaire and Macq [19] first calculated an approximate EDT, using ordered propagation by bucket sorting. It produces a result similar to Danielsson's.



(a) (b)

Fig. 4: Voronoi diagrams of (a) a set of points and (b) a set of arbitrary shapes as determined by way of region growing.

Second, this approximation is improved by using neighborhoods of increasing size.

- Shih and Liu [22] started with four scans on the image, producing a result similar to Danielsson's. A look-up table is then constructed containing all possible locations where no exact result was produced. Because during the scans the location of the closest object pixel is stored for each image pixel, the look-up table can be used to correct the errors. It is claimed that the number of error locations is small.

3.1 Voronoi diagrams

Such exact EDT can be applied to obtain distance maps such as the Voronoi diagram (see Figure 4)[4]. The Voronoi diagram $V(P)$ is a network representing a plane subdivided by the influences regions of the set of points $P = \{p_1, p_2, ..., p_n\}$. It is constructed by a set of Voronoi regions $V(p_i)$ which is, for any i, defined by

$$V(p_i) = \{x \in \mathbb{Z}^2 : \; |x - p_i| \leq |x - p_j|, \text{ for all } j\} \tag{7}$$

Voronoi diagram generation of a space with arbitrary shapes (see Figure 4) is hard from the analytical point of view [23,24], but is easily solved by applying a growth algorithm.

[4] The Voronoi web page: `http://www.voronoi.com`

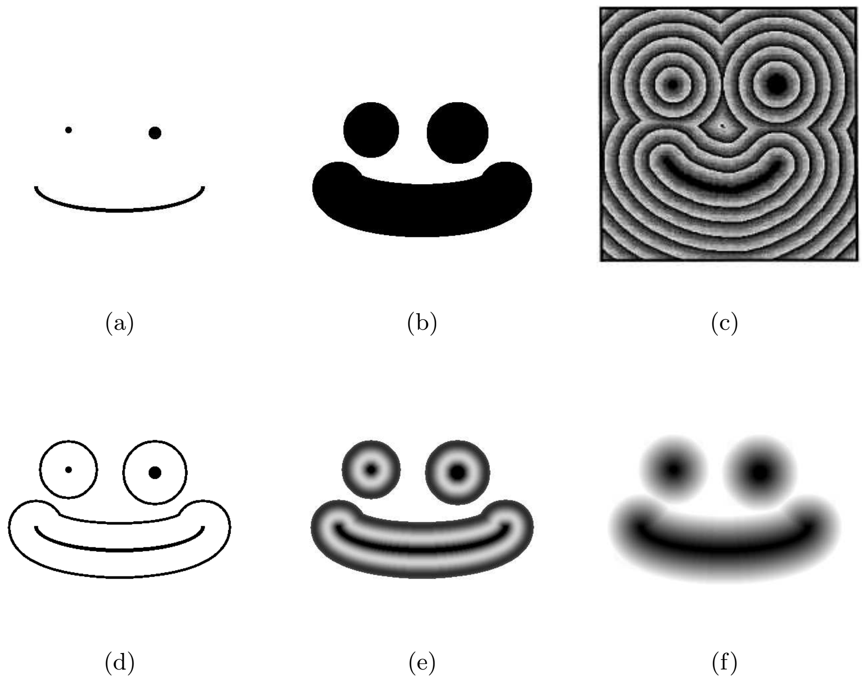


Fig. 5: (a) is the original image. (b) is the same image after dilation by hexadecagonal region growing. (c) is a distance map as presented by Coiras et al. [10]. (d), (e), and (f) are weighted distance maps (WDM). (d) provides the the extremes (i.e., the original pixels and the boundary of the dilated image). (e) presents a discrete distance map. (f) presents an gradual decrease in weight with the increase of distance from the original pixels.

4 Fast Exact Euclidean Distance (FEED)

In contrast with the existing approaches such as those of Shih and Liu [22] and Cuisenaire and Macq [19], we have implemented the EDT starting directly from the definition in Equation 6. Or rather its inverse: each object pixel q, in the set of object pixels (O), *feeds* its ED to all non-object pixels p. The naive algorithm then becomes:

initialize $D(p) \ = \ if \ (p \in O)$ then $0,$ else ∞
  foreach $q \in O$
    foreach $p \notin O$
      update : $D(p) = \min(D(p), \mathrm{ED}(q,p))$

However, this algorithm is extremely time consuming, but can be speeded up by:

- restricting the number of object pixels q that have to be considered
- pre-computation of $\mathrm{ED}(q, p)$
- restricting the number of background pixels p that have to be updated for each considered object pixel q

This resulted in an exact but computationally less expensive algorithm for EDT: the Fast Exact Euclidean Distance (FEED) transformation. It was recently introduced by Schouten and Van den Broek [6]. For both algorithmic and implementation details we refer to this paper. For an improved version and an extension to fast handling of video sequences, we refer to [3].

In its naive implementation, FEED proved already to be up to $3\times$ faster than the algorithm of Shih and Liu [22]. Providing that a maximum distance in the image is known a priori, it is even up to $4.5\times$ faster.

To be able to utilize FEED for the creation of WDMs, we have applied a few small modifications to the implementation, compared to the algorithm as introduced in Schouten and Van den Broek [6]. FEED was adapted in such a way that it became possible to define a metric on which the WDM was based. The result of the application of various metrics is illustrated in Figures 5, 7, and 8.

5 Benchmarking FEED

Shih and Wu describe in their paper "Fast Euclidean distance transformation in two scans using a 3×3 neighborhood" [7] that they introduce an exact EDT. They propose their algorithm as the, to be preferred, alternative for the fast EDT as proposed by Cuisenaire and Macq [19]. Shih and Wu's algorithm is the most recent attempt to obtain fast EDTs. Therefore, this algorithm would be the ultimate test for our FEED algorithm.

We have implemented Shih and Wu's algorithm (EDT-2) exactly as they described and tested it on a set of images on both processing time and errors in the EDs obtained, see Figure 6 for two example text-images. As a baseline, the city-block (or Chamfer 1,1) distance was also taken into account.

In Table 1 the timing results can be found for the city-block measure, for Shih and Wu's two scans (EDT-2), and for FEED. As was expected, with a rough estimation of the ED, the city block distance outperformed the other two algorithms by far (see Table 1). More surprising was that FEED was more than twice as fast as EDT-2.

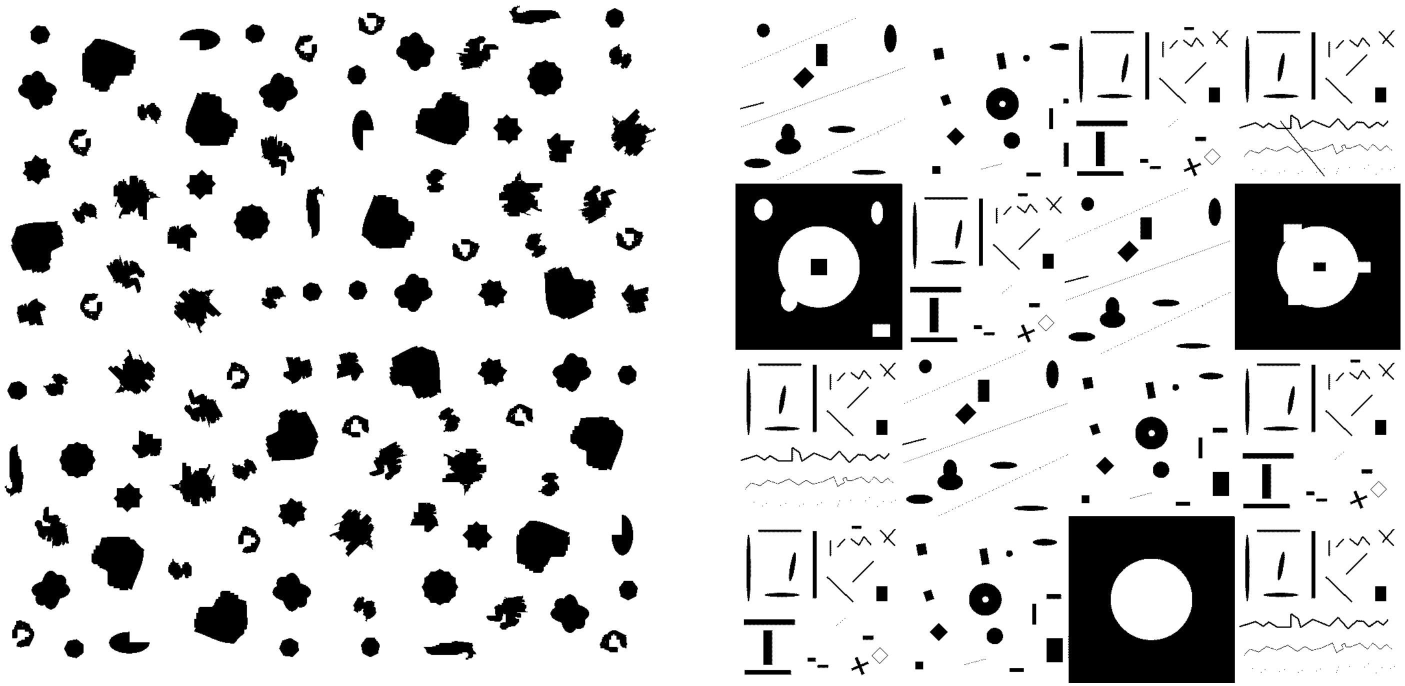


Fig. 6: Two examples of the set of test images used for the comparison of the city-block (or Chamfer 1,1) transform, Shih and Wu's 2-scan method (EDT-2), and the Fast Exact Euclidean Distance (FEED) transform.

Table 1: Timing results for three images on the city-block transform, Shih and Wu's 2-scan method (EDT-2) and for FEED.

Images	Algorithms		
	City-block	EDT-2	FEED
standard	8.75 s	38.91 s	17.14 s
rotated	8.77 s	38.86 s	18.02 s
larger obj.	8.64 s	37.94 s	19.94 s

However, the aim of this research was to utilize exact EDT. Hence, next to the timing results, the percentage of errors made in obtaining the ED is of interest to us. The city-block transform resulted for all three images in an error-level of less than 5%; see Table 2. Shih and Wu's claimed that their two scan algorithm (EDT-2) provided exact EDs. In 99% of the cases their claim appeared justified. However, errors occur in their algorithm, which are reported in Table 2. So, FEED appeared to be the only algorithm that provided the truly exact ED for all instances.

Table 2: Errors of the city-block (or Chamfer 1,1) transforms and of Shih and Wu's two scan algorithm (EDT-2). Note that no errors of FEED were mentioned since FEED provides truly exact EDs.

Images	Algorithms	
	City-block	EDT-2
standard	2.39%	0.16%
rotated	4.66%	0.21%
larger obj.	4.14%	0.51%

6 Weighted distance mapping (WDM)

This section describes the WDM method. Distance maps in general are, for example, used for skeletonization purposes [25] or for the determination of pixel clusters. Given a metric, WDM provides a distance map representing a distance function, which assigns a weight to all of the points in space. Such a weight can be a probability, for example, determined by $\frac{1}{\mathrm{ED}}$, $\sqrt{\mathrm{ED}}$, or $\log(\mathrm{ED})$.

So, using distance functions, distance maps can be created [10]. This is done by growth models based on these distance functions. These distance maps give an excellent overview of the background-pixels that are close to a certain object pixel: A distance map divides the space in a set of regions, where every region is the set of points closer to a certain element than to the others.

6.1 Preprocessing

In order to reduce processing time or to enhance the final WDM, preprocessing algorithms of choice can be applied. For instance, noise reduction and preclustering algorithms.

In the first case, a range can be provided in which is scanned for other points. If no points with the same label are found within this range, this point is rejected as input for WDM.

In the second case, when data points having the same label, are within a range (as was provided), and no other points with another label lay between, then the data points can be connected. When this is done for all data points with the same label, a convex hull is generated for this cluster of points. Next, the same label can be assigned to all points within this convex hull. Hence, instead of labeled data points, labeled objects serve as input for the WDM.

6.2 Binary data

Let us first consider a set of labeled binary data points (i.e., each point in space either is or is not an object point). An isolated pixel with value *0* at the origin is grown using FEED, up to a chosen radius. Each grown pixel then receives a value according to its distance to the origin. As default the ED is used, but any metric could be used. The resulting image defines a mask B.

The output image is initialized with the input image, assuming *0* for an object pixel and a maximum value for a background pixel. Then a single scan over the input image A is made. On each pixel of A with value *0* (an object pixel) the mask is placed. For each so covered pixel, the output value is updated as the minimum of the current value and the value given by the mask.

The resulting output image contains then for each background pixel its minimum distance to the set of object pixels according to the metric of choice. In the case of binary data, the WDM can be stored in one matrix. In Figure 5, some results of the WDM, using different metrics, are shown using the same input image as Coiras et al. [10].

6.3 Multi class data

Now, let us consider the case that multiple labeled classes of data points are present and, subsequently, WDM is applied for data space segmentation. In such a case, the class of the input pixel that provides the minimum distance can be placed in a second output matrix.

The minimum distance value then indicates the amount of certainty (or weight) that the pixel belongs to the class. This can be visualized by different color ranges, for each class. In addition, a hill climbing algorithm can be applied, to extract edges from the distance image and so generate a Voronoi diagram (see Section 3.1).

To determine the class to which the ED is assigned, the update step of FEED was changed to:

$$\begin{aligned} \text{update}: \text{ if } (\mathrm{ED}(q,p) \ < \ D(p)) \\ \text{then } (D(p) \ = \ \mathrm{ED}(q,p); \\ C(p) \ = \ C(q)\ ) \end{aligned}$$

where C is a class matrix.

Figure 7a presents a set of six arbitrary shapes, where Figure 7b is the ED map generated by the algorithm and Figure 7c is the classification as provided by the algorithm. Figure 7d combines Figures 7a-c and presents the true WDM, providing: (1) the weights, (2) the classification, with that (3) the segmentation of the complete data space, and (4) the original data.
The resulting WDM can serve a four purposes:

1. It provides a probability space. The complete data space is described by providing probabilities to unknown regions in the data space; e.g., $\frac{1}{\mathrm{ED}}$, $\sqrt{\mathrm{ED}}$, log(ED). This results in fuzzy boundaries (see Figures 5e-f, 7b,d, and 8b).
2. Determination of the edges between categories (see Figure 7c,d). In addition, Voronoi diagrams can be generated (see Figure 4a-b and c.f. Figures 8a and 8c).
3. Distances between data can be determined, using any metric (see Figures 5c-f and 8d).
4. Visualization of the categorized (fuzzy) data space, as illustrated in Figures 5e-f, 7b,d, and 8b.

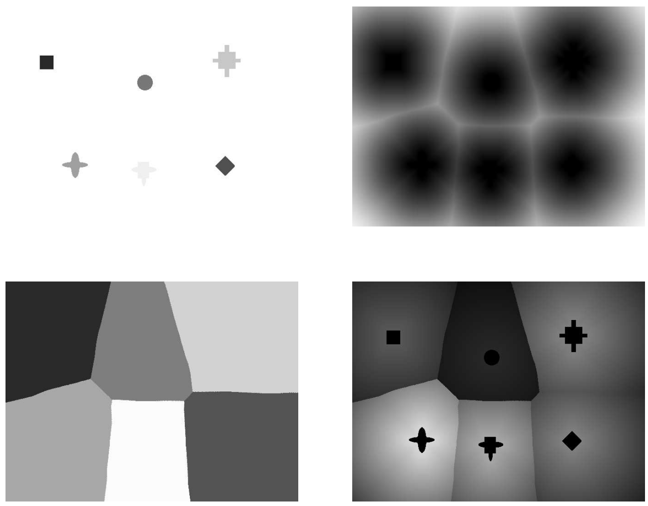
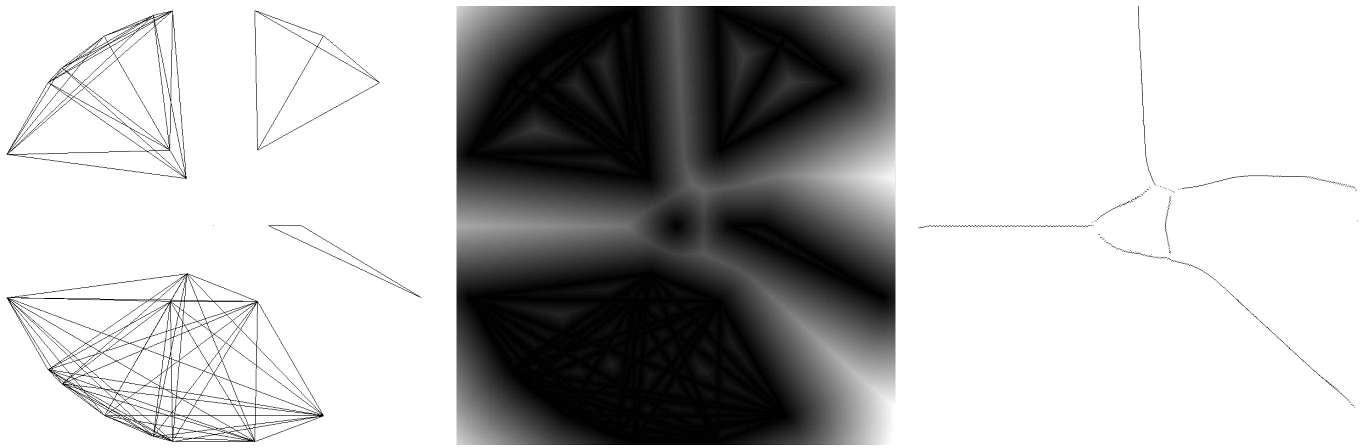


Fig. 7: From top to bottom: the original image, the basic ED map, the fully segmented and labeled space, and the labeled Weighted Distance Map (WDM), in which the original objects are projected and where pixel intensities denotes the weight.



(a) (b) (c)

Fig. 8: (a) The original image in which all data points (of four color categories) assigned to the same color category are connected with each other, using a line connector. (b) The basic ED map of (a), in which the intensity of the pixels resembles the weight. (c) The boundaries between the four classes, derived from the ED map as presented in (b). A hill climbing algorithm was used to extract these boundaries. Note that (c) is the Voronoi diagram of (a).

7 An application: segmentation of color space

WDM as described in the previous section has been validated on various data sets (see for example Figure 5 and 7). We will illustrate its use for the categorization of color in the 11 color categories (or focal colors) [4,26,5]. This was one of the data sets on which the mapping was validated. For more information on the topic 11 color categories see, for example, the World Color Survey [27].

The clustered data is derived from two experiments that confirmed the existence of the 11 color categories [28]. The 216 web-safe colors, as defined by the World Wide Web (W3C) consortium [29], were used as stimuli. In both experiments, 26 subjects were asked to assign each of the 216 web-safe colors four times to one of the 11 color categories. This resulted in 11 clusters of colors, describing a small part of the color space. For more detail concerning the experiments and the data analysis we refer to [28]. The original data can be found on `http://www.few.vu.nl/~egon/publications/pdf/CLUTS.pdf`.

The RGB coordinates of the manually categorized web-safe colors were translated to HSI-coordinates [30]. Let us consider the Hue and Saturation axes of the HSI-color space, using a slice of the HSI cylinder. In this slice five color categories (i.e., brown, red, purple, blue, and green) are projected. However, only four clusters are present. This is due to the overlap between the color categories red and brown.

A bitmap image was generated, containing white background pixels and labeled pixels representing each of the data points. For each category, the data points belonging to the same cluster, were fully connected by using a line generator. Next, WDM was applied on the image (see Figure 8). This resulted in two matrices. One of them consists of the weights determined; in the other matrix the class each point is assigned to, is stored. Their combination provides the ED map.

Last, a hill climbing algorithm extracted edges from the ED map. On the one hand, this resulted in fuzzy color categories (providing probabilities). On the other hand, the extracted edges define a Voronoi diagram.

Since a few years the interest in color in the field of image processing exploded. A ED map as presented, based on experimental data, provides an excellent way for describing the color space. Next, the perceptual characteristics of the color categories could be exploited, providing a probability distribution and, subsequently, a metric for each of the color categories separate. Such a set of features can be utilized and can in combination with a ED map, provide a true WDM.

8 Discussion

This paper started with a brief overview of morphological operations and distance transforms. Next, the algorithm which generates Fast Exact Euclidean Distance (FEED) transforms, is introduced. It is compared with the city-block measure (as baseline) and with the two scan algorithm of Shih and Wu [7], which can be considered as a state-of-the-art algorithm on fast exact ED transforms. FEED proved to computationally twice as cheep as Shih and Wu's algorithm. Moreover, in contrast with the algorithm of Shih and Wu, FEED provides in all cases exact EDs. FEED is applied to generate Weighted Distance Maps (WDM), providing a metric. Its use is illustrated by the segmentation of color space, based on a limited set of experimentally gathered data points. WDMs, as proposed, provide complete descriptions of data spaces, based on a limited set of classified data. Moreover, they can used to obtain Voronoi diagrams.

The traditional drawback in the use of exact ED transforms, due to their large time complexity, is tackled by the use of FEED. Currently, we are developing a parallel version of FEED (FEED), to utilize the generation of distance maps by merging partial maps. In potential, this could significantly reduce the processing time of FEED.

Moreover, the parallel implementation of FEED would provide the means to analyze video sequences. For example, for each object in a video, a partial map can be calculated from different frames, where the partial map for fixed objects is only calculated once. First test results indicate that a parallel version of FEED can possibly be up to 10x faster than FEED, depending on whether or not a run length encoding of the fixed objects is taken into account. Hence, FEED is by far the fastest algorithm for doing exact EDT.

As a consequence, the generation of WDMs can probably be boosted in the near future. So, WDMs three main advantages can be exploited even more: (1) Complete data space can be described, based on a limited set of data points, (2) data spaces can be visualized rapidly, providing the possibility to penetrate the data space gaining more understanding and (3) Distances between data can be determined, using any metric, and (4) Edges between categories can be determined.

These features make WDM an intuitive, flexible, and powerful tool for data mining, describing data structures, and data space segmentation either fuzzy or discrete.

Acknowledgments

The Dutch organization for scientific research (*NWO*) is gratefully acknowledged for funding Eidetic (project-number: 634.000.001), a project within the ToKeN[5] research line, in which this research was partly done. Furthermore, we would like to thank Louis Vuurpijl and Leon van den Broek for reviewing earlier versions of this manuscript.

[5] URL: `http://www.ToKeN2000.nl`

References

1. Tryon, R.C.: Cluster Analysis. Ann Arbor, Michigan: Edwards Brothers Inc. (1939)
2. Kimmel, R., Kiryati, N., Bruckstein, A.M.: Multivalued distance maps for motion planning on surfaces with moving obstacles. IEEE Transactions on Robotics and Automation **14** (1998) 427–436
3. Schouten, Th.E., Kuppens, H.C., Broek, E.L. van den: Timed Fast Exact Euclidean Distance (tFEED) maps. In Kehtarnavaz, N., Laplante, P.A., eds.: Proceedings of Real Time Electronic Imaging X. Volume 5671., San Jose, CA, USA (2005) 52–63
4. Broek, E.L. van den, Kisters, P.M.F., Vuurpijl, L.G.: The utilization of human color categorization for content-based image retrieval. In Rogowitz, B.E., Pappas, T.N., eds.: Proceedings of Human Vision and Electronic Imaging IX. Volume 5292., San Jose, CA, USA (2004) 351–362
5. Rikxoort, E.M. van, Broek, E.L. van den, Schouten, Th.E.: Mimicking human texture classification. In Rogowitz, B.E., Pappas, T.N., Daly, S.J., eds.: Proceedings of Human Vision and Electronic Imaging X. Volume 5666., San Jose, CA, USA (2005) 215–226
6. Schouten, Th.E., Broek, E.L. van den: Fast Exact Euclidean Distance (FEED) Transformation. In Kittler, J., Petrou, M., Nixon, M., eds.: Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICPR 2004). Volume 3., Cambridge, United Kingdom (2004) 594–597
7. Shih, F.Y., Wu, Y.T.: Fast euclidean distance transformation in two scans using a 3×3 neighborhood. Computer Vision and Image Understanding **93** (2004) 195–205
8. Gonzales, R.C., Woods, R.E.: Digital image processing. 2nd edn. Prentice-Hall, Inc., New Jersey (2002)
9. Rosenfeld, A., Pfaltz, J.L.: Distance functions on digital pictures. Pattern Recognition **1** (1968) 33–61
10. Coiras, E., Santamaria, J., Miravet, C.: Hexadecagonal region growing. Pattern Recognition Letters **19** (1998) 1111–1117
11. Zelinsky, A.: A mobile robot navigation exploration algorithm. IEEE Transactions of Robotics and Automation **8** (1992) 707–717
12. Costa, L.F., Manoel, E.T.M., Faucereau, F., van Pelt, J., Ramakers, G.: A shape analysis framework for neuromorphometry. Network: Computation in Neural Systems **13** (2002) 283–310
13. Kimmel, R., Shaked, D., Kiryati, N., Bruckstein, A.M.: Skeletonization via distance maps and level sets. Computer Vision and Image Understanding **62** (1995) 382–391
14. Guan, W., Ma, S.: A list-processing approach to compute Voronoi diagrams and the Euclidean distance transform. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 757–761
15. Costa, L.F., Jr., R.M.C.: Shape Analysis and Classification. CRC Press (2001)
16. Meyer, F.: Topographic distance and watershed lines. Signal Processing **38** (1994) 113–125
17. Lee, Y., Horng, S., Kao, T., Chen, Y.: Parallel computation of the Euclidean distance transform on the mesh of trees and the hypercube computer. Computer Vision and Image Understanding **68** (1997) 109–119
18. Takala, J.H., Viitanen, J.O.: Distance transform algorithm for Bit-Serial SIMD architectures. Computer Vision and Image Understanding **74** (1999) 150–161
19. Cuisenaire, O., Macq, B.: Fast euclidean transformation by propagation using multiple neighborhoods. Computer Vision and Image Understanding **76** (1999) 163–172

20. Borgefors, G.: Distance transformations in digital images. Computer Vision, Graphics, and Image Processing: An International Journal **34** (1986) 344–371
21. Danielsson, P.: Euclidean distance mapping. Computer Graphics and Image Processing **14** (1980) 227–248
22. Shih, F.Y., Liu, J.J.: Size-invariant four-scan euclidean distance transformation. Pattern Recognition **31** (1998) 1761–1766
23. Aurenhammer, F., Klein, R.: 5. In: Voronoi Diagrams. Amsterdam, The Netherlands: North-Holland (2000) 201–290
24. Berg, M., Kreveld, M., Overmans, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. 2 edn. Berlin: Springer-Verlag (2000)
25. Li, H., Vossepoel, A.M.: Generation of the euclidean skeleton from the vector distance map by a bisector decision rule. In Goldgof, D., Jain, A., Terzopoulos, D., Wang, Y.F., eds.: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (1998) 66–71
26. Derefeldt, G., Swartling, T., Berggrund, U., Bodrogi, P.: Cognitive color. Color Research & Application **29** (2004) 7–19
27. Kay, P.: The World Color Survey. URL: `http://www.icsi.berkeley.edu/wcs/` ([Last accessed on May 24, 2004])
28. Broek, E.L. van den, Hendriks, M.A., Puts, M.J.H., Vuurpijl, L.G.: Modeling human color categorization: Color discrimination and color memory. In Heskes, T., Lucas, P., Vuurpijl, L., Wiegerinck, W., eds.: Proceedings of the 15th Belgium-Netherlands Artificial Intelligence Conference, SNN, Radboud University Nijmegen (2003) 59–68
29. Stokes, M., Anderson, M., Chandrasekar, S., Motta, R.: A standard default color space for the internet - srgb. Technical report, W3C; `http://www.w3.org/Graphics/Color/sRGB.html` [Last accessed on October 02, 2004] (1996)
30. Gevers, Th., Smeulders, A.W.M.: Color based object recognition. Pattern Recognition **32** (1999) 453–464