

# *Robust Skeletonization through Exact Euclidean Distance Transform and its Application to Neuromorphometry*

This paper presents how robust 1-pixel-wide and 8-connected skeletons can be obtained simultaneously with exact distance transform calculation. The proposed approach is based on the new concept of *exact dilation*. Two alternative algorithms for exact Euclidean distance transform calculation allowing exact dilation are described: a simpler approach based on the *SEDR* (sorted exact distance representation) data structure, which allows exact distance transform calculation; and a more effective strategy based on border propagation. In both techniques the distances are assigned strictly according to sequences of increasing exact distances in the orthogonal lattice. Because of the high accuracy allowed by such procedures, progressive dilations, high quality and accurate 1-pixel-wide and 8-connected skeletons can be obtained corresponding to the frontiers between previously labelled distinct connected objects. Although this method can be useful for determining generalized Dirichlet tessellations, which is also illustrated in this article, its full potential is harnessed by previously segmenting the contours of connected objects by removing their points corresponding to curvature peaks, which are obtained by using an effective multi-scale curvature estimation technique. In such a way, not only high-quality 1-pixel-wide and 8-connected skeletons are obtained for any shape, but also the whole approach becomes considerably robust to small distortions in the object contours, thus avoiding one of the great shortcomings in traditional skeletonization methods. The application of such methods to an important problem in computational neuroscience and neuromorphometry, namely the automated extraction of tapered dendrograms, is described and illustrated. Considerations regarding distances in orthogonal lattices, typical problems in skeletonization, and the practical implementation of the proposed techniques are also included.

© 2000 Academic Press

**Luciano da Fontoura Costa**

*Cybernetic Vision Research Group,  
IFSC, USP, Caixa Postal 369,  
São Carlos, SP, 13560-970, Brazil  
E-mail: [luciano@ifsc.usp.br](mailto:luciano@ifsc.usp.br)*

## **Introduction**

Distance transforms and skeletons have received special attention over the last decades as promising alternatives for shape representation and analysis. By assigning to

each pixel in the image background the respective shortest distance to the contours of the objects, the *distance transform* [1–3] provides a useful expanded image representation which takes into account not only the proximity and symmetries intrinsic to each object,

but also between distinct objects. The importance of the distance transform as a technique for image processing and analysis has been reflected in many applications, including skeletonization and reconstruction [1,3–10]. Dirichlet tessellations [3,11], mathematical morphology [1,3], geodesic distance calculation and robot trajectory planning [3,12–14] and matching [3,15]. Although many types of distance have been considered, the Euclidean distance is by far the most natural to humans and visual information. The restrictions to the representation of Euclidean distances in the orthogonal lattice, as well as the anisotropies implied by its discrete geometry, have motivated much effort toward defining alternative and more suitable distances [2,3,16,17]. At the same time, fewer works and applications have considered the exact Euclidean distance transform [10].

The interest in skeletons has to a great extent been fostered by the pioneering developments by Blum [18–20], who proposed this concept in 1967 as a means of expressing shape symmetry, while also establishing the now broadly known grass-fire analogy. This interesting analogy provides a powerful conceptualization of many types of skeletons as the position where the propagating fire (actually any wave) fronts, initiated at the object contours, meet. An additional reason behind the relevance of skeletons in visual processing is provided by the important role symmetry is known to play in human visual perception [18–22]. In a study by Wright [9] skeletons have been divided into four main classes: (i) symmetry axis transform — SAT, where the contour elements correspond to the center of all the circles inscribed in the shape; (ii) smoothed local symmetries — SLS; (iii) process-inferring symmetry analysis — PISA [23]; and (iv) symmetry set [9,24]. SAT skeletons, the type considered henceforth, correspond to the centers of maximal inscribed disks within the object (or between objects), including at least two contact points. Despite the considerable potential application of skeletons in image processing and computer vision, the issue of obtaining high quality contours has remained a complicated and challenging one [9,25,26], even though multi-resolution and gray-level versions of skeletonization have also been considered in the literature [27–30]. The following qualities are expected from a suitable skeleton: (a) [3] centered within the shape and retains its topology; (b) be a thin subset of the shape; and (c) allows reconstruction of the shape. While relatively trivial in continuous spaces, such properties are often rather difficult (in some cases impossible) to be guaranteed in orthogonal lattices. Property (c), for instances, can not be achieved unless the object internal countour elements

(i.e. distance zero) are included as part of the respective skeleton, but this becomes pointless because the object contour alone is already shape preserving.

The main skeletonization algorithms have been classified [9] as: thinning [26,27], wave propagation [8], distance transform-based, and analytical techniques. A new class of approaches based on potential fields has been described more recently in [30]. The present work addresses the third category, i.e. distance transform-based skeletonization approaches. Previous developments in this area include [1,4,5,7–10,32–34]. The pioneering work by Rosenfeld and Pfaltz [1] used city block distance and determined skeletons by ridge finding. Danielsson [6] devised a distance based skeletonization technique which looks for the center of maximal disks by using tables supplying the largest disk contained in each specific disk. The developments of Arcelli and di Baja [4,5] have focused on non-Euclidean distances. In Klein and Kübler [7], disks of increasing sizes are superposed onto every object element in order to simulate grass-fire propagation. A nice approach to skeletonization, called SKIZ (skeletonization by influence zone) simultaneous with Euclidean distance calculation has been proposed [32] and improved by using a better Euclidean distance technique [33,34]. That technique propagates not only the distances, but also the labels respective to distinct connected objects in the image, in such a way that the skeletons can be obtained by edge detection in the final labeled image. More recent approaches include those advanced by Leymarie and Levine [8], where skeletons are obtained through the evolution of snakes taking into account distance maps; the use of level set methods [35] the search for distance ridges by using matched filters [9]; and the accurate but relatively complex method of finding maximal disks proposed [10]. Most of these approaches exhibit very high sensitivity to small distortions and detail in the object boundaries (see later in this paper). Attempts at coping with such a problem have included the simplification of contours [25].

The underlying motivation of the present paper derives from the concept of *exact dilations* of spatially sampled frontiers. Given a digital ball (i.e. a ball of specific radius in an orthogonal lattice), only a finite set of distinct distances to its center are permitted. By sorting these distances in ascending order, it is possible to perform region dilation with the maximum accuracy allowed by the spatially sampled representation in images, which is called *exact dilation*. As described in this article, this concept provides the key not only to

effective exact Euclidean distance transform calculation but also paves the way to robust and high-quality skeletonization. Two algorithms are considered for distance transform: one operating by repeatedly scanning the object internal contour, and another which follows the sorted distances of the elements of the dilated region and requires vector addition. While the first algorithm is verified to be simpler and allow exact results, the second algorithm is faster and provides the interesting property of allowing compositions of distance transform. The execution speed allowed by the sequential versions are reasonably high (about 15s for  $256 \times 256$  images using interpreted MATLAB — better performance should be achieved by using compiled languages) allowing real-time applications in many image processing and vision problems.

Since both the described algorithms for Euclidean distance transform calculation are capable of implementing exact dilations of binary objects, they can be easily modified in order to allow the skeletons (SAT) corresponding to the axis between distinct objects to be generated simultaneously with the distance transform calculation. As in the SKIZ method [32–34], labels are assigned to the surrounding region as the distance transform is being computed, the respective skeletons being obtained by a special edge detection algorithm. However, unlike those approaches, more accurate skeletons are here obtained because of the exact dilation procedure and a hierarchical labeling of the propagating areas. While this methodology immediately allows the determination of high-quality, 1-pixel-wide and 8-connected generalized Dirichlet tessellation, its potential is only fully harnessed when the contours of connected shapes are previously segmented (opened) at curvature extrema. As described in Leyton [36], an interesting relationship has been known to exist between symmetry axes and curvatures. More specifically, Leyton has advanced the symmetry-curvature duality theorem, which implies that the curvature extrema correspond to the terminations of local symmetry axes. The idea underlying the skeletonization method proposed in the present article reflects the symmetry-curvature theorem in the sense that only the more robust and meaningful skeletons (avoiding the noise implied by the image spatial quantization), namely those induced by opening the smoothed object contour at its curvature extrema, are obtained. In addition, by varying the Gaussian standard deviation, a multiscale family of skeletons can be obtained. It is shown in this article that this approach not only guarantees high-quality 1-pixel-wide, 8-connected skeletons, but is also substantially robust

to distortions in the object boundaries, a problem which has severely constrained the application of traditional skeletonization algorithms. This important result is a direct consequence of the fact that the collision of the frontiers (in the grass-fire sense), caused by the same segmented portion of the object, is completely avoided.

The present paper also includes the application of the developed algorithms to a relevant problem in neuro-morphometry and computational neuroscience, namely the derivation of tapered dendrograms. While standard dendrograms are extremely important in modeling the electrochemical behavior of neural cells [37,38] and in synthesizing morphologically realistic artificial neural cells [39], the enhanced version presented in this paper allows not only the lengths of the dendritic segments to be taken into account, but also their respective *local width* (i.e. the width at each point along the dendritic segment) instead of the average width traditionally used.

The basic terms used in this article are illustrated in Figure 1. Let  $B$  be an  $N \times N$  binary image and let its foreground and background elements have values 0 and 1, respectively. The *object* (*shape*, or *foreground*) elements, which will henceforth have value 1, are represented in gray, and the *background* elements (value 0) are identified in white. The process for *internal border* detection considered here consists in marking as a contour element every foreground element with coordinates  $(x, y)$  for which  $B(x-1, y) + B(x+1, y) +$

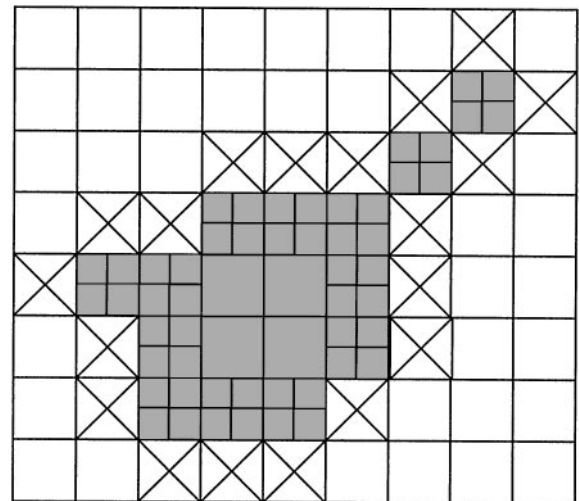


Figure 1. Illustration of the terms adopted in this article.

$B(x, y-1) + B(x, y+1) < 4$ . *External border* detection is performed by applying the above described method to the complement of the image. In Figure 1, the object external contour is identified by “×” and the internal contour by “+”. Both the internal and external contours are 8-connected. An *internal point* is defined as any foreground element not belonging to the object internal contour. Any foreground element which has one or two neighbours will be called a *1-pixel-wide element*. In Figure 1, the internal points correspond to the gray cells without a “+”. The two right-most and left-most cells are 1-pixel-wide elements.

### Some Considerations about Exact Distances in Orthogonal Grids

Let  $U$  be the orthogonal lattice having its elements indexed by vectors  $(x, y)$ , where  $x$  and  $y$  are integers. It is possible to assign to every lattice element a real number corresponding to its distance to the lattice origin  $(0, 0)$ . Although the results in this article can be immediately extended to any type of distance, we shall henceforth be constrained to the *Euclidean distance*, represented as  $\text{dist}_E\{(x_1, y_1); (x_2, y_2)\} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ . The *digital ball* of finite radius  $R$  ( $R$  is an integer value) is defined as the set of elements of  $U$  such as  $\text{dist}\{(x, y); (0, 0)\} \leq R$ . The *distance digital ball of radius  $R$*  corresponds to the representation of the distances within the digital ball of radius  $R$ . Figure 2 presents all the Euclidean distances within the  $7 \times 7$  square centered at the lattice as well as the distances in the digital ball of radius 3, represented by the shaded region. The number of distinct distances in a distance digital ball of radius  $R$ , henceforth represented as  $N_R$ , is plotted in terms of  $R$  (up to  $R = 30$ ) in Figure 3. The plot also includes the total number of distinct distances which is relatively small when compared with the total number of cells (about only 10%), which indicates a high degree of distance symmetries and allows the distinct distances to be treated individually as done in the present paper.

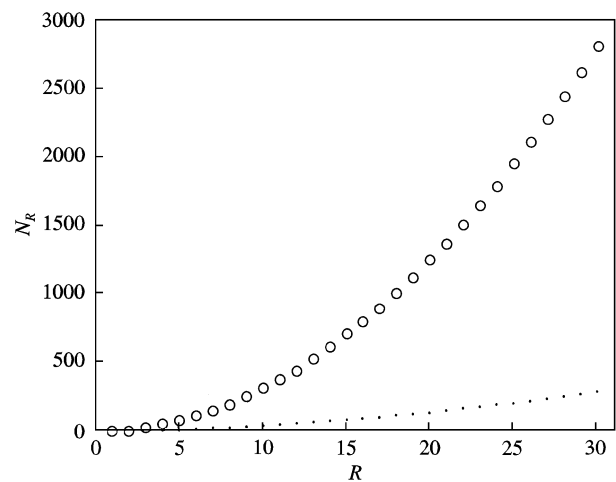
It is clear from Figure 2 that the number  $N(d)$  of lattice points exhibiting the same exact distance  $d$  to the origin varies in terms of  $d$ . However, because of symmetries in the orthogonal grid, this number is always an integer power of 4. For instance, the number of lattice points having distance  $\sqrt{5}$  is 8 (see Figure 2). Figure 4 shows the number of equidistant points in terms of their respective distinct distances for  $R = 30$ . It

$3\sqrt{2}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$3\sqrt{2}$
$\sqrt{13}$	$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$	$\sqrt{13}$
$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$
3	2	1	0	1	2	3
$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$
$\sqrt{13}$	$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$	$\sqrt{13}$
$3\sqrt{2}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$3\sqrt{2}$

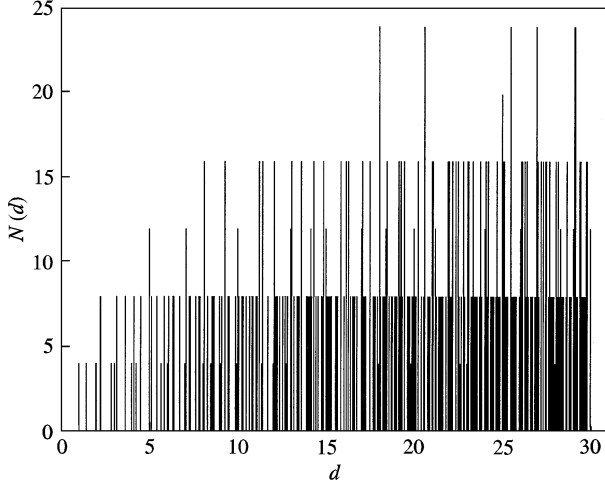
**Figure 2.** All the Euclidean distances in a  $7 \times 7$  square centered at the lattice origin and the distance ball of radius 3 (shaded).

can be readily observed that the number of equidistant points (symmetries) tends to increase with the distance. At the same time, the difference between subsequent distinct distances tends to get smaller.

The *distance transform* of a binary image in a continuous space is accurately defined as the process that assigns to each of the infinite points in the Cartesian space the minimal distance from this point to the objects



**Figure 3.** The number  $N_R$  of distinct distances in a digital ball (dots) and the total number of cells in the respective digital ball (circles) in terms of the radius  $R$ .



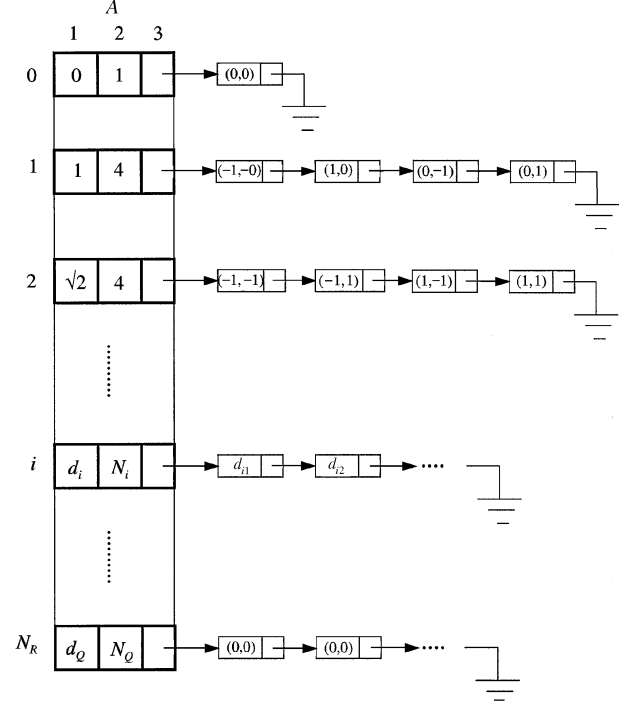
**Figure 4.** Number  $N(d)$  of points in the digital ball of radius  $R$  presenting the same distance  $d$  to the center.

in the image. For a single convex object and assuming Euclidean distance (the type of distance considered henceforth), it can be readily verified that the (infinite) set of points with a specific distance  $d$  reproduces the outline of the original shape at a larger scale. This inherent and important feature of the continuous distance transform will be referred to as *shape preserving property*. The distance transform of binary shapes in spatially quantized spaces is defined in exactly the same way, except that many (actually an infinite number) of the infinite distances in continuous spaces will not be represented — for instance, distances in the whole interval  $(1, \sqrt{2})$  will not be found on the lattice. For the same reason, the shape preserving property does not perfectly hold for Euclidean distance transforms in sampled spaces, where only approximations of the outer boundaries of the shape can be obtained in terms of the contours of the region defined by those points of the lattice with distances smaller or equal to a specific distance value.

### The Sorted Exact Distance Representation — SEDR

When the  $N_R$  distinct distances within a digital ball of radius  $R$  are sorted in ascending order, the respective position (starting from 0) of each distance in the list will be called *distance index*. For example, the sorted distances and respective distance indexes for the digital ball in Figure 2 is:

Sorted distinct distances : 0 1  $\sqrt{2}$  2  $\sqrt{5}$   $2\sqrt{2}$  3  
 Distance indexes : 0 1 2 3 4 5 6



**Figure 5.** The general structure of an SEDR.

Given a digital ball of radius  $R$ , its respective *sorted exact distance representation* — *SEDR* — is defined as being the data structure containing: (a) the sorted distinct distances; (b) the number of equidistant (with respect to the lattice origin) elements for each respective distinct distance; and (c) the coordinates of those equidistant elements relative to the lattice origin. The word “exact” in SEDR stands for the fact that every distance is stored with real accuracy and not merged into intervals in order to achieve connected circles or disks, as done in other studies [7,10]. Although many alternative data structures could be used to represent an SEDR (see Implementation and Performance), we shall consider here an  $N_R \times 3$  array  $A$  such as its element  $A(i, 1)$  contains the  $i$ th distinct distance in ascending order (observe that  $i$  is the distance index);  $A(i, 2)$  contains the number of elements relative to the ball center. Figure 5 illustrates the general structure of an SEDR.

### Two Algorithms for Exact Euclidean Distance Transform Determination

Both algorithms described in this section accept as input the internal contour of the binary objects in the image; represented in terms of a list (in any order) of their

coordinates, i.e.  $C = [(x_1, y_2); (x_2, y_2); \dots; (x_L, y_L)]$ ; producing as output the respective distance maps.

The first algorithm resembles the approach described by Klein and Kübler [7], where 8-connected disks of increasing sizes are superposed into the object contours. However, that approach considers neither the sorted distinct distance nor the exact dilation principles adopted in the present article, which proved to be essential. The Euclidean distance transform calculation proceeds by updating an  $N \times N$  binary image  $D$ , starting with all its pixels set to 0, in the following way. For each of the subsequent distance index  $i$  in the SEDR (its radius  $R$  should be defined by the maximum distance of interest), a total of  $A(i, 2)$  vectors are generated by adding each of the respective coordinates in the linked list pointed by  $A(i, 3)$  to each element  $(x, y)$  in the list  $C$ . Every time these vectors point to a cell in  $D$  which is empty (i.e. has value 0), the value  $A(i, 1)$  is assigned to that cell. As a matter of fact, the distance index instead of the exact distance value can be stored in order to allow an integer, rather than real, array to be used. The process stops after all the distinct distances in the SEDR have been considered. At the end of such processing, the image  $D$  contains the exact Euclidean distance transform up to the maximum considered distance  $R$ . The ratio between the total distance assignments (i.e. when an empty cell is found) and the total of tested cells will be henceforth called *hit rate*.

The above described incremental expansion of the distance-assigned area while considering each subsequent distinct distance in the SEDR will be denominated *exact dilation*. This concept is particularly important when interpreted from the perspective of wavefronts propagating from the contour, such as in the grass-fire analogy. The basic idea here is that the exact dilation allows the representation in the orthogonal lattice of *all* the possible distinct wavefronts that would be obtained by strictly monotonic increasing distances (along the normal field induced by the objects internal contours) even at infinitesimal displacements. In other words, considering the grid-intersecting quantization of the continuous border of the dilating area, the frontiers generated by the exact dilation corresponds to the finest evolution in the orthogonal lattice of the wavefronts emanating from the contours.

This feature, which is closely related to the shape preserving property described earlier, provides the key not only for effective and exact Euclidean distance transform calculation, but also paves the way to the

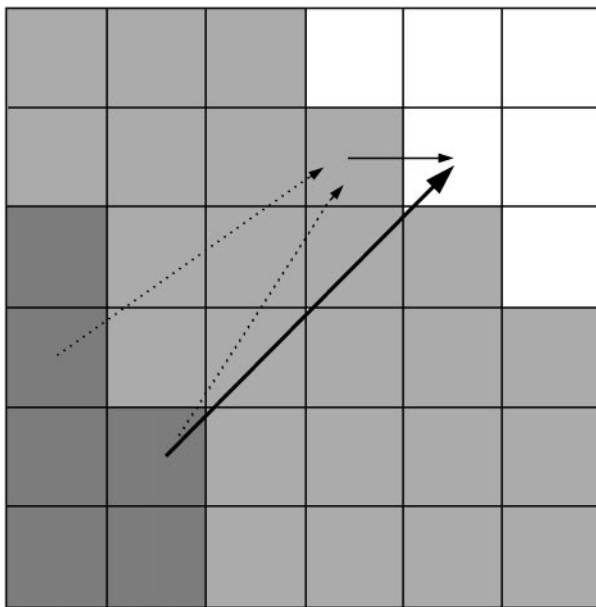
robust and high-quality skeletonization method proposed in this article. As a matter of fact, it also has important implications for achieving fully precise operations in mathematical morphology and even the calculation of geodesic distances.

We shall also consider an alternative algorithm for exact Euclidean distance calculation. This method differs from the above described one in the sense that the distance assignments proceed directly by scanning the outer contour of the dilated region instead of repeatedly following the list of the object inner contour components, allowing optimal hit rate at the expense of some overhead corresponding to finding the smallest distances (as explained below). Related approaches have been reported [33–35] where values are assigned as the object external frontier is dilated. Let us refer to the list of current distances in the dilating external contour as  $S$ . As with the first algorithm, the alternative approach starts with the inner contours of the objects in the image, and the distance map is constructed in  $D$ . The external contour elements have their distances calculated and stored into  $S$ , always in ascending order. At each subsequent step, each of all the lattice points exhibiting the smallest distance value in  $S$  are updated with the respective value and removed from the stored list. The distance between the shape contour and each of the eight neighboring cells of each updated cell which is still empty are computed, its value and relative coordinated of the equidistant points matching the shape contour is assigned to the respective cell in  $D$ , and the cell representation is stored in the stored list. The determination of the relative coordinates matching the original countour can be obtained by using the SEDR and the method for distance index determination that is described in the Appendix. The process continues until no distance smaller than  $R$  remains in  $S$ .

Of course, the important step here regards the computation of the distances between each empty cell in the external contours and the image objects. This can be achieved by storing into each element of the image  $D$  not only the distances, but also the relative coordinates of the object points related to that distance. This additional information can be incorporated at little computational expense during the exact distance transform calculation at the moment of updating the empty cells. Once such information is available, the distance to a new empty cell can be straightforwardly determined as that corresponding to the smallest distance obtained by vector addition of all the relative coordinates in all the adjacent (8-neighborhood) non-empty cells and the

distance from the sea cells to the empty cell. This process is illustrated in Figure 6 with respect to one of the neighboring non-zero cells, where the dark gray squares correspond to the object, the light gray squares indicate cells to which their respective distance has already been assigned the white squares represent empty cells, the two dashed arrows indicate the relative coordinates of the two equidistant internal contour points, and the solid arrow indicates the distance between the chosen non-empty cell and one of the empty cells. The empty cells in question receives as distance the smallest vector sum of the dashed and solid arrow (this process includes every neighboring non-empty neighboring cell), indicated by a solid thin arrow in the figure. The thus obtained distance is exact to the level of round-off noise implied by vector addition in real precision.

Figure 7 illustrates the eight first steps (b–i) in the exact dilation of the simple object in (a) as well as the for first stages (j–m) in progressive distance transforms considering a digital ball or radius 1 (i.e. a cross). It is clear from this example that growing the frontier in terms of exact dilations is much more precise than successive distance transforms with a fixed-size digital ball. As a matter of fact, the dilations with the cross is so coarse that the interior of the shape is filled up soon at



**Figure 6.** Vector addition of distances as a means of determining the distance to an empty cell.  $\blacksquare$  = object;  $\square$  = cells with distance assigned;  $\square$  = empty cells;  $\therefore \blacktriangleright$  relative coordinates of internal contour points;  $\rightarrow$  distance between non-empty cell and empty cell.

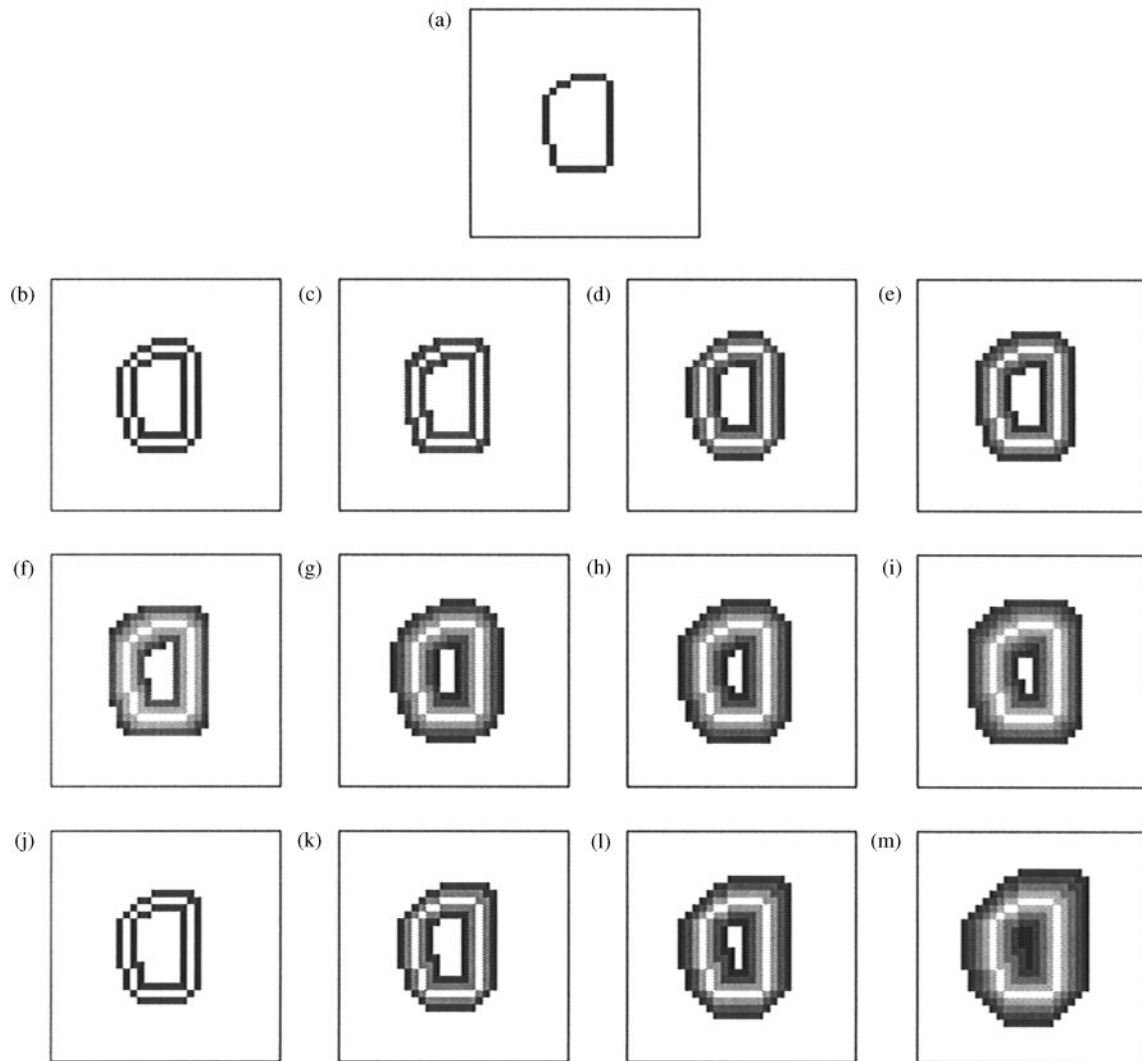
the fourth dilation, represented in Figure 7(m). On the other hand, the successive frontiers obtained during the exact distance transform are very special in the sense that they represent *all* the frontiers which can be obtained in the orthogonal grid by dilating the original shape with respect to the whole succession of increasing exact distinct distances. This allows the external and internal borders of the dilated shape to correspond much more closely to its original shape, i.e. the exact distance transform provides better quality regarding the shape preserving property. Compare, for instance, the external boundaries of the dilated areas in Figures 7(i) and (m).

### Some Considerations about Skeletonization

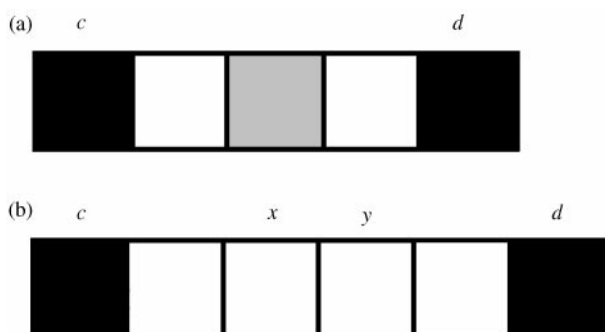
One of the main difficulties (if not the main one) in determining 1-pixel-wide skeletons is illustrated in Figure 8. While it is easy to define the point that is equidistant to the points  $c$  and  $d$  in Figure 8(a), which is represented by shading, a rather complex situation arises in (b).

The two following solutions can be considered for dealing with this problem: (i) to assign as middle point both cells  $x$  and  $y$ ; and (ii) give priority to one of the pixels  $c$  or  $d$  and assign as the skeleton element the point which is closer to the pixel with higher priority. For example, if  $d$  is prioritized, only  $y$  would be assigned as skeleton. While alternative (i) represents a more balanced representation, alternative (ii) allows the important advantage that a single point is obtained. These results are immediately extensible to 2D binary shapes, the second strategy guaranteeing 1-pixel-wide skeletons. The current article adopts the above alternative (ii). In 2D images, the hierarchy of separate object internal contours can be easily accomplished by using any standard labeling algorithm [40,41].

Another important observation regards the processing of binary objects containing 1-pixel-wide elements. Although such portions could be understood as skeletons of null width (which would imply the whole internal contour of the skeleton to be considered as part of the skeleton), we shall henceforth assume that the binary objects in the original image have no 1-pixel-wide element. Elements not satisfying this condition can easily be made suitable by dilating these objects using the  $3 \times 3$  cross as the structuring element.



**Figure 7.** Original binary image (a) and the eight first steps (b–i) in its exact dilation corresponding to the distances  $1$ ,  $\sqrt{2}$ ,  $2$ ,  $\sqrt{5}$ ,  $2\sqrt{2}$ ,  $3$ ,  $\sqrt{10}$  and  $\sqrt{13}$ . The four first progressive distance transforms (j–m) with  $R = 1$ , starting with the image in (a). In all images darker gray levels correspond to higher distance values (white = zero distance).



**Figure 8.** Two situations encountered in skeletonization.

### Obtaining High-Quality, 1-Pixel-Wide Dirichlet Tessellations

Once the connected objects in the image have been labeled, the respective generalized Dirichlet tessellation can be immediately obtained at minimal additional computational cost from any of the above two methods for calculating the exact Euclidean distance map. The underlying idea is simple and effective, consisting of assigning as Dirichlet limiting contours all the points where propagating distance-assigned regions relative to two distinct objects collide. As a matter of fact, Dirichlet frontiers correspond to the skeletons of



the background region between two distinct image objects. The method for determining the Dirichlet tessellation takes advantage of the laveling procedure discussed in the previous section and is described in detail in the following.

Firstly, the distinct regions in the binary image are labeled by using any suitable labeling/region growing algorithm [40,41]. Let us represent the thus obtained  $M$  distinct labels by the set  $\Lambda = \{l_1, l_2, \dots, l_M\}$ . In addition to the image  $D$  where the distance transform is calculated, an auxiliary image  $W$  with the same size is also kept at all times. This image  $W$  should initially contain the labeled internal borders of the objects in the image. Now, the exact Euclidean distance transform is calculated by using any of the two algorithms described for exact distance transform determination. In order to reflect the hierarchical assignments strategy described in the previous section, the contour elements of the region with level  $l_1$  are processed before those belonging to region  $l_2$ , and so on (or *vice versa*). This processing schedule can be easily achieved by storing the contour elements respective to the labeled regions subsequently into list  $C$ . Every time a cell  $(x, y)$  is updated into  $D$ , the label of the respective object being processed is also stored at the same coordinates  $(x, y)$  in  $W$ . At the end of the distance transform calculation, image  $W$  will contain the influence areas of each distinct object in the image, identified in terms of the original labels. Because of the adopted processing order, as well as the exact dilation and the single-point assumptions (i.e. the second strategy outlined earlier), the borders between the respective influence regions result clear and with excellent continuation. As a matter of fact, these properties allow the Dirichlet tessellation to be immediately obtained which is characterized by 1-pixel-wide, 8-connected, contours. This final processing step, here called *special edge detection*, is presented in terms of the following pseudo-code:

For each labeled region identified by  $l_i$ , starting with  $l_1$ :

Consider a Dirichlet border element every foreground cell with coordinates  $(x, y)$  for which  $(W(x-1, y) \neq W(x, y))$  or  $(W(x, y-1) \neq W(x, y))$  or  $(W(x-1, y) \neq W(x, y))$  or  $(W(x, y-1) \neq W(x, y))$ ;

Assign label  $l_i$  to every pixel  $W(x, y)$  whose value is larger or equal to  $l_i$ .

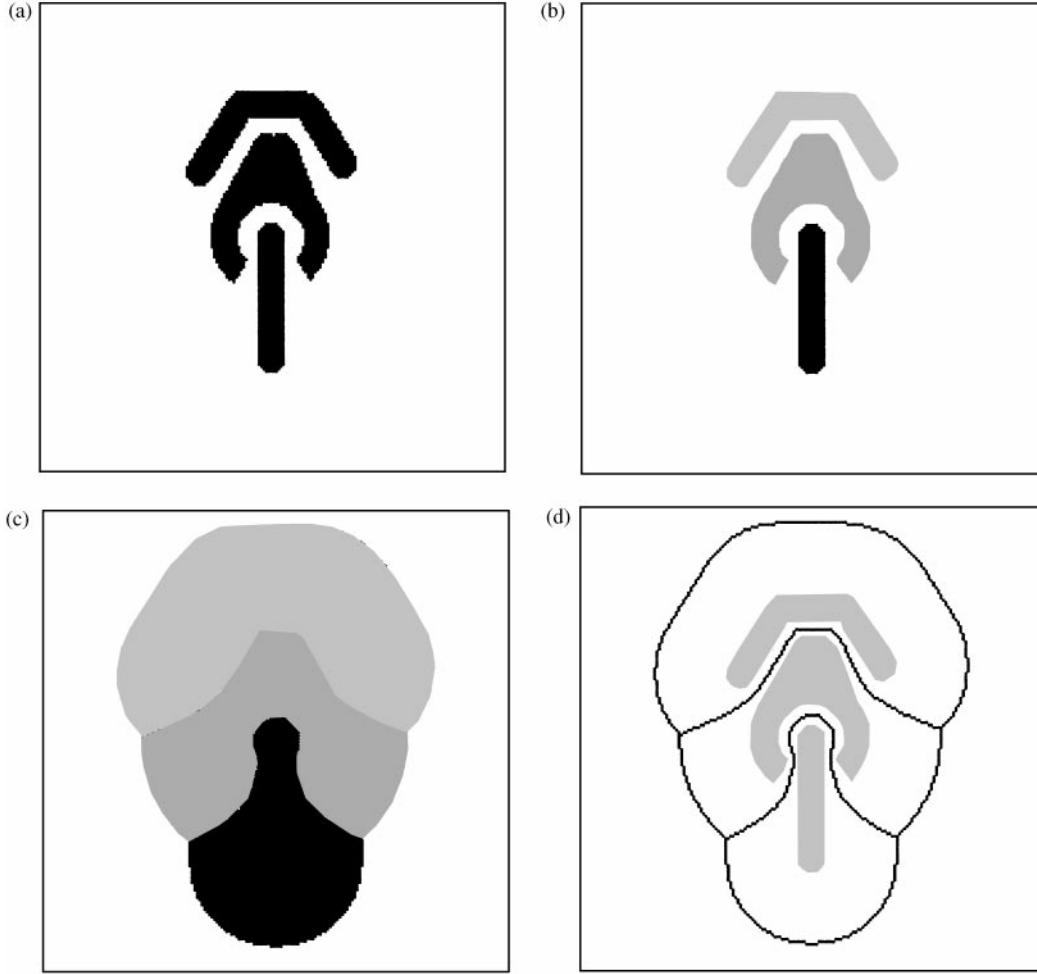
The so obtained Dirichlet frontiers correspond to the center of the maximal circles inscribed between any two

distinct objects in the image, i.e. the obtained skeletons are of the SAT type. This can be easily verified by considering that the dilating regions meet each other at the middle point along the shortest distance to each pair of objects. Of course, the adopted hierarchical labeling assumption implies that one of the centers of maximal disks corresponding to otherwise 2-pixel-wide skeletons are left out.

Let us illustrate the above described concepts and technique through a real example. Consider the image in Figure 9(a), whose initial labeled regions are shown in Figure 9(b). The final configuration of  $W$  is shown in (c) and the respective Dirichlet tessellation considering  $R = 30$  superposed onto the original image (in gray) is presented in (D). High-quality 1-pixel-wide and 8-connected borders are thus obtained.

### Towards High-Quality, Robust, and 1-Pixel-Wide Skeletons

Though the skeletons obtained by the method for Dirichlet tessellation discussed in the previous section correspond to the tessellation among the distinct connected objects in the image, the excellent features of the obtained skeletonization can indeed be extended and generalized to single binary objects in the image. Indeed, such a segmentation allows not only high-quality contours (1-pixel-wide, 8-connected, good continuation), but also a high level of robustness to distortions in the object contour. Considering that one of the most problematic aspects of skeletonization is precisely a high level of sensitivity to such distortions, the technique proposed herein represents an especially relevant alternative to effective skeletonization. The key for such an important property relies on the fact that the previous segmentation of the object contours effectively prevent skeletons to arise within each labeled region. In order to take advantage of such important properties allowed by the above strategy, it is essential to use an effective technique for contour segmentation. This article adopts the multiscale curvature-based technique described in [42–44], which calculates the first and second derivatives needed for curvature calculation in terms of the Fourier transform of the shape internal contours represented parametrically, as indicated in Eqn (1). Here  $k(t)$  is the parametrized point curvature;  $\mathfrak{F}$  and  $\mathfrak{F}^{-1}$  stand for the Fourier transform and its inverse;  $s(t)$  is the complex Peano representation of the object internal contour;  $g_\sigma(t)$  is the Gaussian with standard deviation  $\sigma$ ,  $*$  stands for complex conjugate; and  $Im\{\}$



**Figure 9.** Illustration of Dirichlet tessellation by exact dilations: (a) original binary image; (b) labeled image; (c) labeled images after label spreading by the exact Euclidean distance transform; and (d) Dirichlet tessellation with  $R = 30$  superposed onto the original image.

means imaginary part. This technique is particularly effective, demanding  $O(L \log L)$  basic operations ( $L$  is the number of contour elements) and allowing multiscale behavior controlled by  $\sigma$ . For  $L = 800$ , the curvature estimation takes about 0.3s in MATLAB.

than about 10% of maximum curvature in contour. Actually, this parameter can vary — see Cesar and Costa [42] for a discussion on the application of this methodology. The standard deviation of the Gaussian has been set to 2. Once the positive curvature

$$k_{\sigma}(t) = \frac{-\text{Im}\{\mathfrak{S}^{-1}\{\mathfrak{S}\{g_{\sigma}(t)\} \cdot \mathfrak{S}\{s(t)\} \cdot (j2\pi f)\} \cdot \mathfrak{S}^{-1}\{[\mathfrak{S}\{g_{\sigma}(t)\} \cdot \mathfrak{S}\{s(t)\} \cdot (-4\pi^2 f^2)]^*\}\}}{\|(\mathfrak{S}^{-1}\{\mathfrak{S}\{g_{\sigma}(t)\} \cdot \mathfrak{S}\{s(t)\} \cdot (j2\pi f)\})\|^3} \quad (1)$$

Once the contour curvature is calculated, higher positive values (actually, this sign depends on the way in which the contour is pursued — it is henceforth adopted counter-clockwise sense) indicate vertices (convex) of the original contour. Such curvature extremes are identified by taking the first derivative of the curvature and selecting the points with value higher

points have been identified, they are removed from the contour to be labeled and processed by the distance transform.

The SAT concept assumes that the maximal inscribed disks are tangent to the object on at least two of its points. However, it is argued here that, by originating

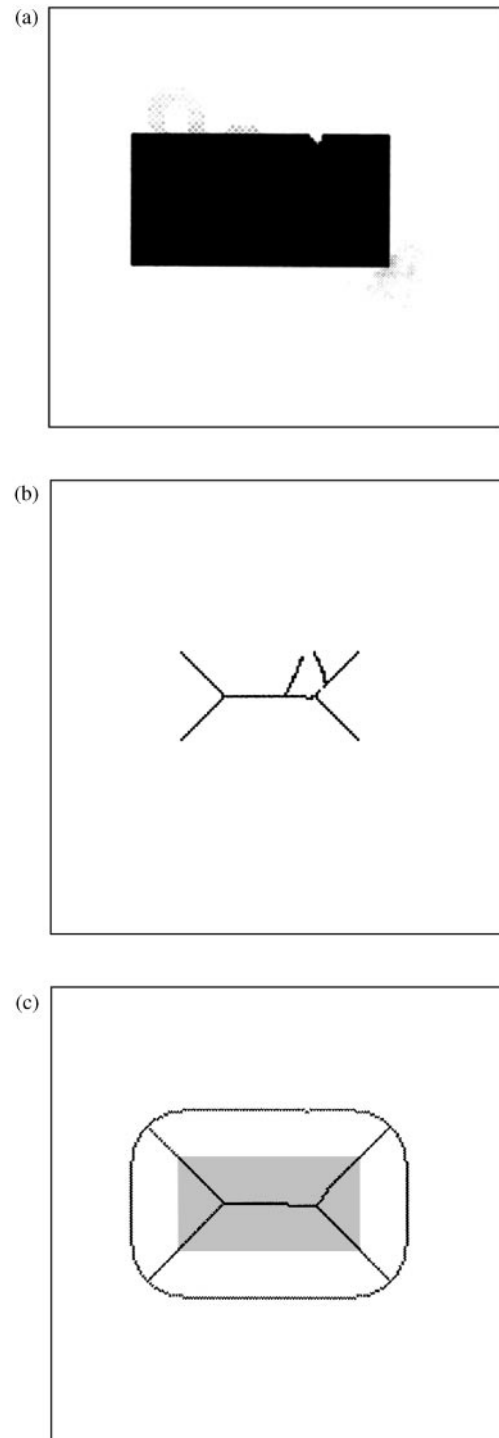
the in the conceptualization of the SAT in continuous spaces, this criterion has been the main source of complication in skeletonization. The problem arises because it is virtually impossible to define tangent points when shapes are represented in orthogonal lattices, for parametrized versions of the object contours are not even continuous, and thus not differentiable. Moreover, digital straight lines which are tangent to a digital circle often present two common points at the interface. In practice, it becomes virtually impossible to identify distinct tangent points in digital images. Strictly speaking, every pair of subsequent points in a digital contour is a potential candidate to define a maximal circle. The skeletonization method advanced in the present article avoids such complications simply by previously segmenting the object contour at its curvature extrema and identifying as skeleton elements all the points where the dilating frontiers meet, which can be immediately done while computing the exact Euclidean distance transform. Because these points still correspond to the center of maximal disks (except for the fact that the hierarchical labeling described earlier implies that some of these centers will be skipped in order to achieve 1-pixel-wide skeletons), there is not need to check for tangents or any related concept.

Figure 10 illustrates the robustness of the proposed approach to skeletonization achieved by segmenting and labeling the shape contours. The original binary object is show in (a), and a result which is typical of traditional skeletonization is presented in (b). It is clear that the small distortion in the contour of the shape propagates towards the central axis, which is caused by the collision of the dilating waves generated at the vicinity of the distortion. The binary skeleton obtained by the method described in this article, which is much more robust, is presented in (c) superposed to a filtered version of the original object.

### Applications to Neuromorphometry

The techniques for exact Euclidean distance transform and skeletonization proposed in this article are useful in a wide range of situations in image processing and vision. This section illustrates the application of those techniques in neuromorphometry, namely the quantization of shape properties of neural cells and structures.

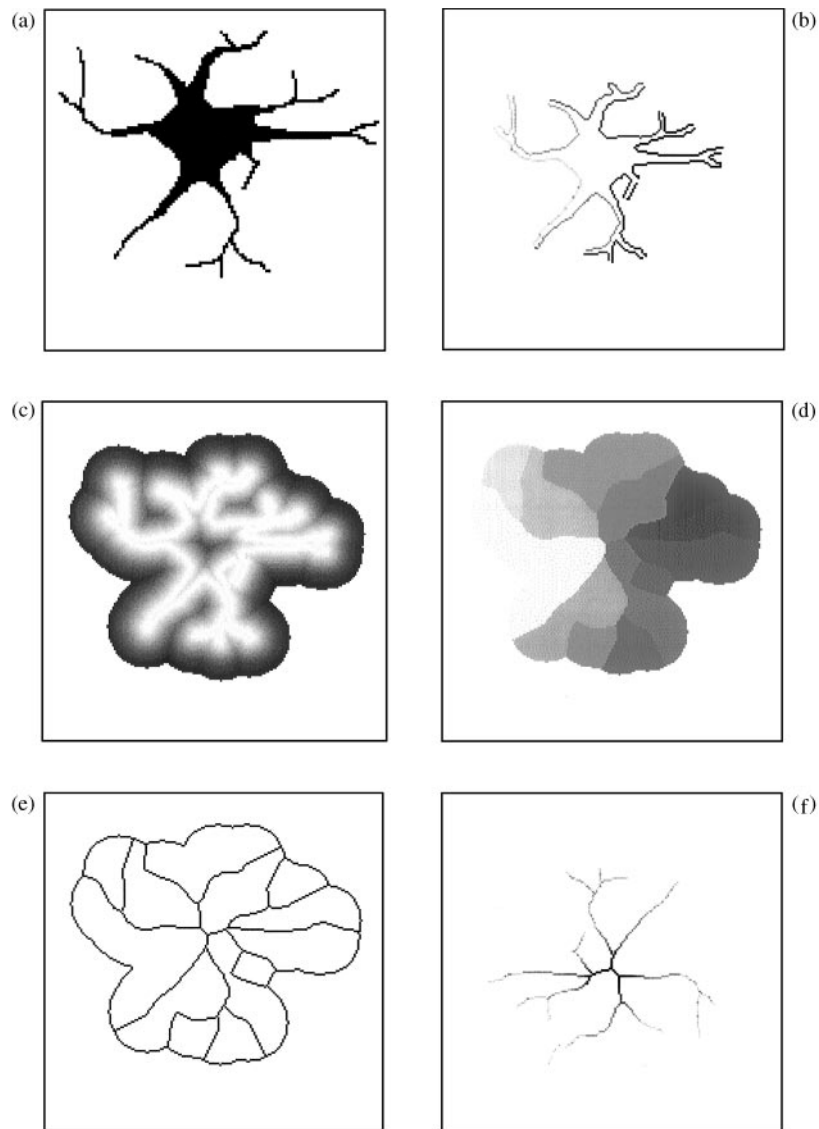
Neural cells are characterized by a number of prolongations (called *processes*) corresponding to the dendrites and the axon. From the perspective of the



**Figure 10.** Illustration of the robustness of the proposed skeletonization: (a) original binary shape; (b) result typically obtained by using traditional skeletonization methods; and (c) binary skeletons obtained by the technique presented in this article.

transmission cable model, the function of each neuron is specified by the extensions and widths of the neural processes [37,38]. Actually, each process is understood as a passive or active transmission line whose parameters are determined by the above mentioned shape properties of the respective cells. The evolution of the electrical signal in such cables is modeled in terms of partial differential equations having the its parameters determined by the extension and width of the neural processes. As traditionally done (the compartmental

approach), each process is approximate as a series of cylinders with fixed width. The data structure representing such connected cylinders is called a *dendrogram* [45–47]. While in neuroscience dendrograms have traditionally been derived by hand, the first and only automated approach (as far as we know) for dendrogram generation was described in Costa *et al.* [48], where the dendrogram is extracted by using curvature-based identification of branch points and extremities in the dendritic arborization (represented by its internal

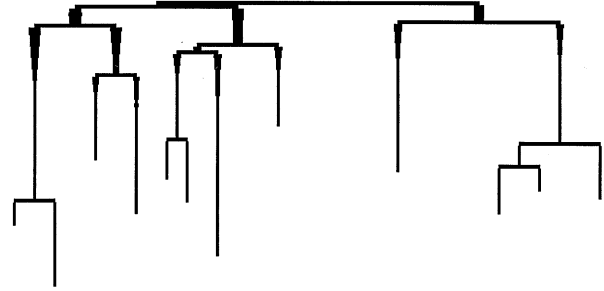


**Figure 11.** Illustration of the process of determining tapered dendrograms for electrical neural simulation: (a) the original binary representation of the neuron; (b) segmented and labeled internal contours; (c) the distance transform of the contours of the shape in (a); (d) dilated regions obtained by using exact dilations; (e) the internal and external skeletons; and (f) the isolated internal skeleton, separated from the external skeleton by performing a logical “and” between the original binary shape and the skeletons in (e) and weighted by the distance transform of the contour of the shape in (a).

contour) and subsequent merge by using a formal grammar approach. That approach, however, uses formal grammar methods which are not completely objective and demands additional processing for determining the width of the dendritic segments and skeletons. This section shows how the skeletonization methodology developed in the present paper can be effectively applied to determine a better representation of the neural processes in terms of tapered dendrograms (including also the width of each of the points along the dendritic segments), from which more complete models of neural activity can be obtained. Such improved dendrograms are also extremely important for the derivation of statistical description of shapes to be used in the synthesis of artificial neural cells [39,48]. Although we shall be restricted to 2-dimensional neural cells, it should be observed that many biological neurons, such as ganglion and Purkinje cells, are characterized by planar organization of the dendritic arborization.

The methodology will be illustrated for the neuron in Figure 11(a). The respective labeled segmented (partitioned by curvature) contours are shown in (b). It should be observed that the two shorter extremities (upper and right portion of the cell) have been deliberately disconsidered in order to illustrate the possibility of selecting the skeleton branches (this has been done by re-joining the contour segments at the respective high curvature points). The distance transform is shown in (c) and the exact dilation of the contour portions are shown in (d). The obtained internal and external skeletons are shown in (e), and the isolated internal skeleton, obtained by a logical “and” between the original shape and the skeletons in (e) and graded by the distance transform of the original shape contour is presented in (f). The respectively obtained tapered dendrogram, obtained through a simple recursive tracking of the skeleton (which is facilitated by the fact that the skeleton segments are one-pixel-wide) is shown in Figure 12.

It is also observed that the obtained skeleton can be used to recover a reasonable reconstruction of the original shape. This can be straightforwardly achieved by updating into an initially empty image all its points covered by the relative coordinates of the SEDR indexed by the distance value of each skeleton element. Figure 13 presents the reconstruction of the object in Figure 11(a) obtained by this procedure. the small holes inside the image can be easily filled up by standard morphological techniques, as illustrated in Figure 13(b). It should also be observed that the small differences



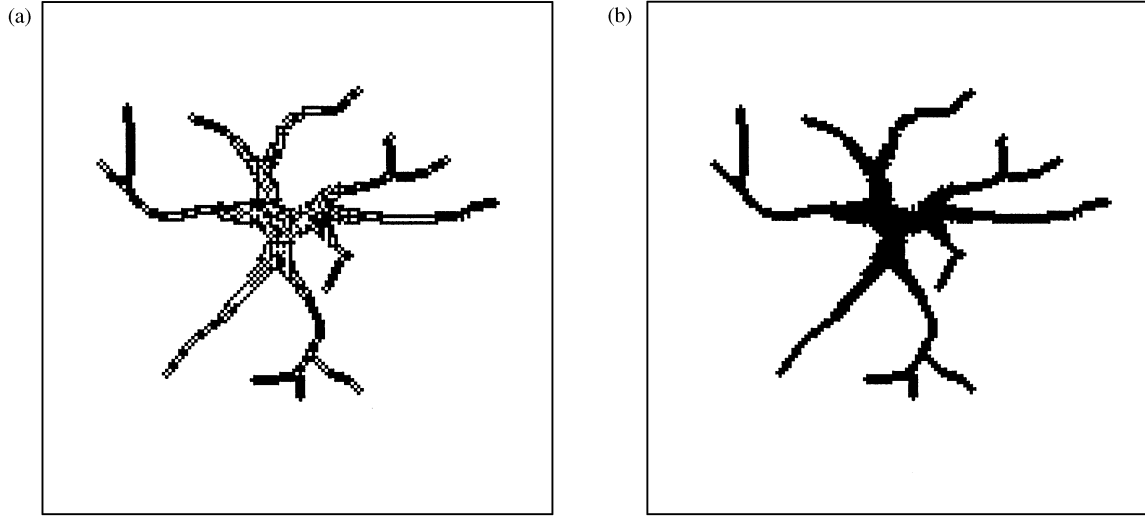
**Figure 12.** The tapered dendrogram respective to the neural shape in Figure 11(a). While the hierarchical structure of the dendritic arborization is represented by the tree structure, the respective width of the segments is indicated by the width of each vertical segment.

between the original and reconstructed object are caused by the two following reasons: (i) the skeleton has not included the contour elements (i.e. zero distance); and (ii) the hierarchical assignment strategy described in Section 7 may imply a displacement of one pixel for the skeletons which would otherwise have non-unitary width.

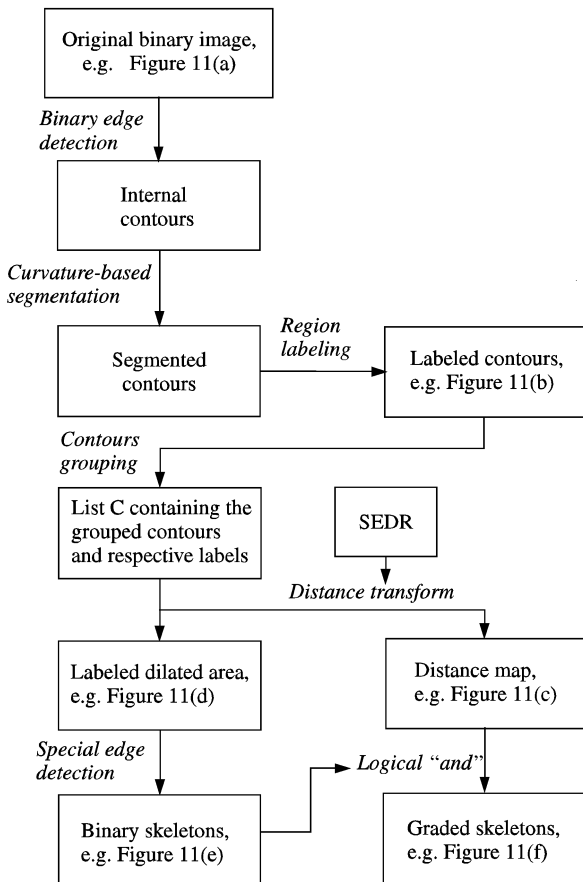
## Implementation and Performance

This section includes some considerations about the implementation and performance of the proposed techniques for exact Euclidean distance transform calculation. Special attention is paid to characterizing the described techniques in terms of their execution speed and required data structures.

The block diagram in Figure 14 illustrates the principal operations and data involved in the proposed skeletonization method. Virtually every process involved is simple and suitable for sequential or parallel execution. At each successive stage (except the distance transform), only two images are needed: the previous one and another containing the respective results (i.e. operations such as binary edge detection are not possible to be executed “in-place” by using only one image). Even the distance transform extended for skeletonization requires only two images, because it receives as input the elements of the list *C*. Consequently, the data structures needed during the whole process includes only two images, the list *C*, and the SEDR. As a matter of fact, the image used to store the distance maps does not need to have real precision, since the distance index instead of the exact distances



**Figure 13.** Reconstruction of the neuron in Figure 11(a) from its respective skeleton before (a) and after (b) filling.



**Figure 14.** The processes (italics) and data involved in the proposed skeletonization approach.

can be stored (there is a one-to-one relationship between each exact distance and the distance index). Also, in cases where only the skeletons are needed, the image  $D$  and respective computation can be completely skipped.

There is an interesting possibility regarding the storage of the SEDR. As observed earlier, the number of relative coordinates with respect to each distinct distance is always a multiple of 4. This can be immediately used for implementing the SEDR in a standard array rather than an array involving a linked list for representation of the relative coordinates. The idea is to represent the SEDR for a specific radius  $R$  in terms of an array with dimension  $N_R \times 6$ , such as its element  $A(i, 1)$  contains the  $i$ th distinct distance in ascending order (observe that  $i$  is the distance index);  $A(i, 2)$  contains the number of elements in the lattice with that distance to the origin; and  $A(i, 3)$  to  $A(i, 6)$  contains the respective relative coordinates of the elements having the distance indicated in  $A(i, 1)$ . The point here is to repeat the line  $A(i, j)$  as many times as needed to accommodate the whole set of relative coordinates. While the fact that such subsequent cells  $A(i, j)$  will consequently have the same value poses no problem to the distance transform and skeletonization, it will not be compatible with the technique for distance index determination described in the Appendix.

The effectiveness of the first proposed approach to exact distance transform calculation is a direct

consequence of its respective hit rate. The higher this number, the better the performance. It has experimentally verified that the hit rate, which is a function of the shape of the object internal contour and  $R$  (worse for higher values), can be as low as  $1/80$ . Yet, sequential implementations of this approach in MATLAB 5.1 on an IBM-PC Pentium II (300 MHz, 256 Mbytes RAM) have allowed reasonably good execution speeds. The average total execution time for binary images such as in Figure 11(a) has been verified to be about 15 s. Contrariwise, the second alternative to distance transform calculation allows optimal hit rate, and took about 1 s to execute.

### Concluding Remarks

The important concepts of *sorted distinct distances*, represented by the proposed SEDR data structure, and *exact dilations*, as well as the hierarchical labeling of the contours, have provided the key to the interesting features achieved for the proposed methods for distance transform and skeletonization. Two algorithms for exact Euclidean distance transform have been presented and illustrated which allow sequential execution speed compatible with many applications in image processing and vision. Considerations about the performance of these algorithms in terms of the required data structures and allowed execution speeds have also been included. The second algorithm allows faster execution times. By being capable of implementing exact dilation, both these algorithms can be slightly modified in order to allow simultaneous determination of high-quality SAT skeletons between distinct labeled regions of the image. This possibility was applied first for determining Dirichlet tessellations between distinct connected objects in binary images and then extended and generalized to single objects in binary images and then extended and generalized to single objects by performing previous contour segmentation (namely opening the contour at its peaks of curvature). The adopted multiscale method for curvature estimation has proven to be decisive in achieving proper identification of the critical points for opening the contours. The proposed skeletonization methodology has allowed high-quality, 1-pixel-wide and 8-connected skeletons which are also fully robust to discrepancies in the contours of the labeled regions. In addition, the proposed methodology can be straightforwardly applied to gray-scale images by using the slicing strategy [28].

The potential of the proposed techniques has been illustrated with respect to an important application in neuromorphometry and computational neuroscience, namely the determination of tapered dendrograms capable of representing not only the extension of the dendritic (an eventually axonal) segments, but also their respective local width. Such neural representations have enormous potential for allowing not only the synthesis of morphologically more realistic neural cells, but also for paving the way for more complete and detailed differential electrophysiological models.

Undergoing related developments to be reported when the opportunity arise include the extension of the techniques to 3-D shapes, use of other monotonic distances, propagation of labels assigned to each contour element, calculation of geodesic distances, as well as other applications in neuromorphometry and computational neuroscience.

### Acknowledgment

Luciano da Fontoura Costa is grateful to FAPESP (Proc. #96/05497-3) and CNPq (Proc. #30142/92-3) for financial help.

### References

1. Rosenfeld, A. & Pfaltz, J. L. (1996) Sequential operations in digital picture processing. *Journal of the Association for Computing Machinery*, **13**(4): 471–494.
2. Borgefors, G. (1986) Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, **34**: 344–371.
3. Borgefors, G. (1994) Applications using distance transforms. In: *Aspects of Visual Form Processing* (C. Arcelli, L. P. Cordella and G. S. di Baja, Eds.), 84–108, World Scientific.
4. Arcelli, C. & Baja, G. S. (1994) Ridge points in Euclidean distance maps. *Pattern Recognition Letters*, **13**: 237–243.
5. di Baja, G. S. (1994) Well-shaped, stable, and reversible skeletons from the (3, 4)-distance transform. *Journal of Visual Communication and Image Representation*, **5**: 107–115.
6. Danielsson, P. E. (1980) Euclidean distance mapping. *Computer Vision, Graphics, and Image Understanding*, **14**: 227–248.
7. Klein, F. & Kübler, O. (1987) Euclidean distance transformations and model-guided image interpretation. *Pattern Recognition Letters*, **5**: 19–29.
8. Leymarie, F. & Levine, M. D. (1992) Simulating the grassfire transform using an active contour model. *IEEE PAMI*, **14**: 56–75.

9. Wright, M. W. (1995) *The extended Euclidean distance transform*. PhD Thesis, University of Cambridge. (Available at <http://svrwww.eng.cam.ac.uk/reports/people/mwww.html>).
10. Ge, Y. & Fitzpatrick, J. M. (1996) On the generation of skeletons from discrete Euclidean distance maps. *IEEE PAMI*, **18**(11): 1055–1066.
11. Arcelli, C. & di Baja, G. S. (1986) Computing Voronoi diagrams in digital pictures. *Pattern Recognition Letters*, **4**: 383–389.
12. Piper, J. & Granum, E. (1987) Computing distance transformations in convex and non-convex domains. *Pattern Recognition*, **20**: 599–615.
13. Chong, K. S. & Kleeman, L. (1996) “Indoor Exploration Using a Sonar Sensor Array: A Dual Representation Approach,” *Technical Report MECSE-1996-12*, Department of Electrical and Computer Systems Engineering, Monash University. (Available at <http://calvin.eng.monash.edu.au/~cks/publication.html>)
14. Chong, K. S. & Kleeman, L. (1997) “Indoor Exploration Using a Sonar Sensor Array: A Dual Representation Approach”, 1997 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Grenoble (France).
15. Bengtsson, M. (1990) Stochastic optimization algorithms – an application to pattern matching. *Pattern Recognition Letters*, **11**: 717–724.
16. Yamashita & Ibaraki (1986) Distances defined by neighborhood sequences. *Pattern Recognition*, **19**(3): 237–246.
17. Arcelli, C. & di Baja, G. S. (1988) Finding local maxima in a pseudo-Euclidean distance transform. *Computer Vision, Graphics, and Image Processing*, **43**: 361–367.
18. Blum, H. (1967) A transformation for extracting new descriptors of shape. In: *Models for the Perception of Speech and Visual Form*. (Wathen-Dunn, W. (Ed.)), The MIT Press, pp. 362–380.
19. Blum, H. (1973) Biological shape and visual science. *Journal of Theoretical Biology*, **38**: 205–287.
20. Blum, H. & Nagel, R. N. (1978) Shape description using weighted symmetric axis features. *Pattern Recognition*, **10**: 167–180.
21. di Gesu, V. & Valenti, C. (1995) The discrete symmetry transform in computer vision. *Technical Report DMA 011 95*, University of Palermo. Available at <http://hpdma2-math.unipa.it/Valenti/html/dstrep.html>.
22. di Gesu, V. Valenti, C. & Strinati, L. (1997) Local operators to detect regions of interest. *Pattern Recognition Letters*, **18**: 1077–1081.
23. Leyton, M. (1988) A process-grammar for shape. *Artificial Intelligence*, **34**: 213–247.
24. Giblin, P. J. & Bruce, J. W. (1992) Bruce. Curves and Singularities. *Cambridge University Press*.
25. Attali, D. & Montanvert, A. (1997) Computing and simplifying 2D and 3D continuous skeletons. *Computer Vision and Image Understanding*, **67**: 261–273.
26. Lam, L. Lee, S.-W. & Suen, C. Y. (1992) Thinning methodologies - A comprehensive survey. *IEEE PAMI*, **14**: 869–885.
27. Abe, K., Mizutani, F., & Wang, C. (1994) Thinning of gray-scale images with combined sequential and parallel conditions for pixel removal. *IEEE Trans. Systems, Man, and Cybernetics* **24** (2): 294–299.
28. Gauch, J. M. (1993) *Multiresolution image shape description*. Springer-Verlag.
29. Pizer, S. M. Oliver, W. R. & Bloomberg, S. H. (1987) Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE PAMI*, **9** (4): 505–511.
30. Dill, A. R. Levine, M. D. & Noble, P. B. (1987) Multiple resolution skeletons, *IEEE PAMI*, **9**(4): 495–504.
31. Ahuja, N. & Chuang, J.-H. (1997) Shape representation using a generalized potential field model. *IEEE PAMI* **19**: 169–175.
32. Lantuéjoul, C. (1980) Skeletonization in quantitative metallography, In *Issues of Digital Image Processing*. R. M. Haralick and J.-C. Simon Eds., *Sijthoff and Noordhoff*.
33. Cuisenaire, O. (1999) *Distance Transformations: Fast Algorithms and Applications to Medical Image Processing*. PhD Thesis, Laboratoire de Telecommunications at Teledetection, Universite Catholique de Louvain, Belgium.
34. Vincent, L. (1991b) Distances defined by neighborhood sequences. *Proc. IEEE CVPR 91*, Hawaii, USA, pp. 520–525.
35. Sethian, J. A. (1996) *Sethian. Level Set Methods*. Cambridge University Press.
36. Leyton, M. (1987) Symmetry-curvature duality. *Computer Vision, Graphics, and Image Processing*, **38**: 327–341.
37. Kandel, E. R. Schwartz, J. H. & Jessel, T. M. (1991) *Principles of Neural Science*. Appleton & Lange.
38. Koch, C. & Segev, I. (1998) *Methods in Neuronal Modeling*. The MIT Press.
39. Coelho, L. R. C. & Costa, da F. (1997) Morphologically Realistic Neural Networks. *IEEE International Conference on Complex Computer Systems*, Como, Italy, 223–228.
40. Schalkoff, R. J. (1989) *Digital Image Processing and Computer Vision*. John Wiley and Sons.
41. Gonzalez, R. C. & Woods, R. E. (1992) *Digital Image Processing*. Addison-Wesley.
42. Cesar Jr., R. M. & Costa, L. da F. (1995) Piecewise linear segmentation of digital contours in  $O(N \cdot \log N)$  through a technique based on effective digital curvature estimation. *Journal of Real-Time Imaging*, **1**: 409–417.
43. Cesar Jr., R. M. & Costa, L. da F. (1996) Towards effective planar shape representation with multiscale digital curvature analysis based on signal processing techniques. *Pattern Recognition*, **29**(9): 1559–1569.
44. Cesar Jr., R. M. & Costa, L. da F. (1997) The application and assessment of multiscale bending energy for morphometric characterization of neural cells. *Review of Scientific Instruments*, **68**(5): 2177–2186.
45. Sholl, D. A. (1953) Scholl. Dendritic Organization in the Neurons of the Visual and Motor Cortices of the Cat. *Journal of Anatomy*, **87**: 387–406.
46. Poznanski, R. (1992) Modeling the electrotonic structure of starburst amacrine cells in the rabbit retina: functional interpretation of dendritic morphology. *Bull. Math. Biol.*, **54**: 905–928.
47. Velte, T. J. & Miller, R. F. (1995) Dendritic integration in ganglion cells of the mudpuppy retina. *Visual Neurosci.*, **12**: 165–175.
48. Costa, L. da F. Cesar Jr., R. M. Coelho, R. C. & Sawaki, J. T. (1998) Perspective on the analysis and synthesis of



- morphologically realistic neural networks. In: *Modeling in the Neurosciences: From Ionic Channels to Neural Networks* (R. R. Poznanski, Ed.), Harwood Academic Publishers.
49. Arcelli, C. & Serino, L. (1997) From discs to parts of visual form. *Image and Vision Computing*, **15**: 1–10.
  50. Borgefors, G. & Nyström, I (1997) Nyström. Efficient shape representation by minimizing the set of centres of macimal discs/spheres, *Pattern Recognition Letters*. **18**: 465–472.
  51. Costa, L. da F. & Velte, T. (1999) Automatic characterization and classification of ganglion cells from the salamander retina. *Journal of Comparative Neurology*, **404** (1): 33–51.
  52. Xia, Y. (1989) Skeletonization via the realization of the fire front's propagation and extinction in digital binary shapes. *IEEE PAMI*, **11** (10): 1076–1086.

## Appendix

### *Determining distance indexes from the respective distance*

An interesting and practical problem involving exact distance concerns how to obtain the distance index for a specific distance value represented in real precision. Basically, there are at least the following two alternatives for addressing such a problem: (i) to interpolate a function over the matrix (distance)  $\times$  (distance index); and (ii) to use a look-up table. the second alternative is described in the appendix.

Firstly, the smallest distance  $d_{\min}$  between any adjacent distances in the sequence of exact Euclidean distinct distances is determined (e.g. by checking all subsequent distances in the SEDR). This smallest distance determines the size of the look-up table, which will be vector  $T$  with dimension  $R/d_{\min}$ . It should be observed that although only a fraction of this vector is actually used, this does not generally pose serious computational problems since the size of this vector is small compared with the other data structures involved in the distance transform and skeletonization. Anyway, more effective allocation schemes such as hashing can always be used in critical applications. Having allocated the vector  $T$ , the SEDR is completely scanned and for each distance index  $i_d$  the integer indexing value  $i = \text{round}\{\text{SEDR}(i_d, 1)/d_{\min}\}$  is calculated, and to the cell of  $T$  indexed by  $i$  is assigned the value  $i_d$ , i.e.  $T(i) = i_d$ . Once the look-up table  $T$  is so constructed, given a distance  $d$ , belonging to the set of all possible distinct distances represented with real precision, the respective distance index is supplied by  $T\{\text{round}(d/d_{\min})\}$ . This strategy has been verified to work even for the vector additions implied by the second technique for exact Euclidean distance transform calculation. The noise margin allowed by this look-up table is  $d_{\min}/2$ , which has been verified to be fully sufficient in all the considered practical applications.