

Detecting Centres of Maximal Geodesic Discs on the Distance Transform of Surfaces in 3D Images

Gabriella Sanniti di Baja¹ and Stina Svensson²

¹ Istituto di Cibernetica, Italian National Research Council
Via Toiano 6, IT-80072 Arco Felice (Naples), Italy
gsdb@imagn.cib.na.cnr.it

² Centre for Image Analysis, Swedish University of Agricultural Sciences
Lägerhyddvägen 17, SE-75237 Uppsala, Sweden
stina@cb.uu.se

Abstract. We introduce the distance transform for surfaces in 3D images, i.e., the distance transform where every voxel in the surface is labelled with its geodesic distance to the closest voxel on the border of the surface. Then, the distance transform is used to identify the set of centres of maximal geodesic discs in the surface. The centres of maximal geodesic discs can be used to give a compact representation of any surface. In particular, they can provide a useful representation of the surface skeleton of solid volume objects.

1 Introduction

In [6], the *constrained distance transform* was introduced for computing the minimal path from any point in the image to a goal point, avoiding *obstacles*. This can be used to find optimal paths between obstacles (e.g., for robot navigation), measure perimeters, and determine convex hulls. A special case of the constrained distance transform is the distance transform of line patterns (DTLP), introduced in [12] and further investigated in [9]. In DTLP, distance information is propagated along the line pattern, starting from the end points of the line pattern itself (the background is used as an obstacle to the propagation). In presence of meeting lines that convey different distance information into their common meeting point, the minimum distance or the maximum distance can be used to continue distance propagation on the outgoing line. By selecting the maximum distance, the DTLP can be used, for instance, to find the longest path through a line pattern. The minimum distance can be used to guide hierarchical decomposition.

Here we extend the DTLP to surfaces in 3D images. In case of concurrent distinct distance values, we use the minimum distance to continue propagation. Every voxel in the surface is labelled with the distance to the closest voxel on the border of the surface. This implies that only *open* surfaces, i.e., surfaces that do not enclose any background component, can be considered. The first step in the algorithm is to detect the border of the surface. Then, once the distance transform of the surface is available, we identify therein the centres

of the maximal geodesic discs (CMGDs). These, similarly to the centres of the maximal discs in the distance transform of a 3D object, can be used to provide a new representation of the surface. In fact, the CMGDs constitute a subset of the surface which, in turn, can be seen as the union of the maximal geodesic discs. The CMGDs can be suitably linked to each other to obtain a curve representation with as many components as there are components in the surface. If this is done, one can easily treat images containing more than one object component, without need of resorting to a preliminary connected component labelling. Moreover, connected curves can be traced, which is not possible for sparse voxels, and can be vectorized so saving memory occupation. Thus, the representation of the surface in terms of connected sets including the CMGDs is convenient. We remark that the new representation does not coincide with the skeleton of the surface. In fact, loops can be created which do not correspond to tunnels of the surface, so that the topology is not preserved. Nevertheless, it can be seen as a new linear shape descriptor of surfaces in 3D images, where the curves can be interpreted as the main symmetry axes of the surfaces.

To extract the new representation, it is convenient that the border of the surface is defined analogously to the contour of a 2D plane pattern. The approach to extract the new representation can then be seen as analogous to the one to extract the skeleton of the pattern from its distance transform. The contour of a pattern is the set of pattern pixels having a neighbour in the background. These pixels belong to protruding contour arcs, i.e., have no neighbours internal in the pattern, or belong to arcs delimiting the pattern itself, i.e., have neighbours internal in the pattern. To have a similar definition of border in case of a surface, we should identify there curves and edges, respectively corresponding to the protruding contour arcs and the arcs delimiting the pattern. Curves should be prevented from propagating distance information, as their 2D counterpart, the protruding arcs, consist exclusively of centres of maximal discs through which no distance information can be further more propagated. To this purpose, all curves have to be preliminarily identified and temporarily removed from the surface. It is then possible to identify only the border voxels (edges) from which distance information can be propagated to the surface. The curves are added again after the distance transform is computed. The distance label of their voxels is set to the value of the voxels in the edges of the surface, as also curves belong to the border.

The new representation can be applied to any open surface. In particular, it can be applied to the surface skeleton of a 3D solid object. This latter case is treated in this paper.

2 Definitions and Notions

In this paper, 3D bilevel images, i.e., volume images consisting of object and background, will be considered. Each voxel has three types of neighbours in its $3 \times 3 \times 3$ neighbourhood: six voxels sharing a face, twelve voxels sharing

an edge, and eight voxels sharing a point with the central voxel. We will use 26-connectedness for the object and 6-connectedness for the background.

Each voxel in the object can be labelled with the distance to its closest background voxel. This labelling can be performed by computing the distance transform (DT) of the object, [2]. The distance depends on the chosen metric. Two simple metrics, commonly used, are the D^6 and the D^{26} metrics, where the distance between two voxels, v and w , is equal to the number of steps in a minimal 6- and 26-connected path, respectively, between v and w . The corresponding DTs, here called DT^6 and DT^{26} , can be computed by propagating distance information from the background in one forward and one backward scan over the image. Each voxel is assigned the minimum of the label of the voxel itself and the labels of its already visited neighbours increased by one. For DT^6 only face neighbours are considered.

DT^6 and DT^{26} give rough approximations to the Euclidean distance. To have a better approximation a *weighted distance transform* can be used. For the weighted DT, we use different weights, w_1 , w_2 , and w_3 , for the distance to a face, an edge, and a point neighbour, respectively. Computation of the DT using weights $w_1 = 3$, $w_2 = 4$, and $w_3 = 5$, has been shown to give a good approximation to the Euclidean distance, [2]. The DT using $w_1 = 3$, $w_2 = 4$, and $w_3 = 5$ will here be denoted 3 – 4 – 5 DT. When computing the DT, the contribution given by each already visited neighbour of a voxel is its corresponding label increased by the relative weight. To be consistent, we use weights also for DT^6 and DT^{26} . For DT^6 , $w_1 = 1$, $w_2 = 2$, and $w_3 = 3$, while for DT^{26} , $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$.

The distance label of a voxel v in the DT can be interpreted as the length of the radius of a ball, centred on the voxel and fully enclosed in the object. In the following, we will denote by v both the voxel and its associated distance label. A voxel is *centre of a maximal ball* (CMB) if the corresponding ball is not completely covered by any other single ball in the object, [1]. This can be checked by a suitable label comparison. A voxel v is a CMB if all its neighbours n_i , $i = 1, \dots, 26$, with proper weights w_j , $j = 1, 2, 3$, satisfy

$$n_i < v + w_j.$$

For any voxel v , N^{26} is the number of 26-connected object components in the $3 \times 3 \times 3$ -neighbourhood of v . \overline{N}_f^{18} is the number of 6-connected background components, having v as a face-neighbour, in the $3 \times 3 \times 3$ -neighbourhood of v where the point neighbours are disregarded (in other words only 18 neighbours of v are taken into account). For a deeper discussion, see [7,10,4]. Here we use the algorithm introduced in [3] for computing \overline{N}_f^{18} and N^{26} .

3 Computing the Distance Transform of Surfaces

Before computing the DT of a surface, we should detect its border. In fact, the minimum distance will be propagated starting from it. The labels in the DT

will indicate the distances to the background along paths in the surface, passing through the border. Therefore, we set the border voxels to w_1 instead of 0. The same convention is used for the end points when computing the DTLP, [12,9].

Surface skeletons can consist of both surfaces and curves, depending on the shape of the original object and the used metric. When computing the DT of the surface skeleton, only the voxels in the edges should be allowed to propagate distance information to the surface. Curves do not border any surface portion and we should prevent that any propagation occurs from them. Thus, we detect and temporarily remove the curves, before detecting the edge voxels from which propagation can be done. After the DT has been computed, the curves are added and the distance label of their voxels is set to w_1 .

Our method can be applied to any surface skeleton of a solid 3D object. We prefer a skeleton that has a number of desirable properties, as it is the case for the surface skeleton introduced in [11], which will be used through the rest of the paper. The surface skeleton is centred within the original object with respect to the D^6 metric, topologically equivalent to the original object, almost symmetric, and allows complete object recovery. This last property, unfortunately, implies that the skeleton can be two-voxel thick. Thus, before detecting the border of the surface skeleton, this has to be reduced to a one-voxel thick surface. To this purpose, we use the local operations introduced in [4] to obtain a one-voxel thick surface skeleton starting from the two-voxel thick set. Thinning is achieved in six scans of the image, one for every face direction. During each scan, only configurations that are two-voxel thick in the scanning direction are taken into account, see Fig. 1. Voxels in these configurations are sequentially removed provided that their removal does not change the topology.

The above thinning is, however, not enough to obtain a one-voxel thick surface in presence of L-shaped regions, see Fig. 2. To remove voxels in such regions another step is necessary, based on the use of an L-shaped configuration detector. During this step, only voxels having $\overline{N}_f^{18} = 1$ (grey voxels in Fig. 2) and at least three face neighbours in the skeleton are considered. Among them, the voxels with two face neighbours in the skeleton, that are edge neighbours of each other and have both $\overline{N}_f^{18} = 1$ (\bullet) or have both $\overline{N}_f^{18} \neq 1$ (\circ), are marked as candidate to removal. Removal of marked voxels is then done sequentially, provided that topology is not altered. The resulting set is shown to the right in Fig. 2.

Once the one-voxel thick surface is obtained, the border is detected in two steps. In the first step, *curve voxels* are defined as voxels having $\overline{N}_f^{18} = 1$ and $N^{26} \geq 2$ (the case $N^{26} > 2$ occurs at curve junctions), [7,10]. Since propagation of distance information should not occur from curves, all voxels with $\overline{N}_f^{18} = 1$ and $N^{26} \geq 2$ are temporarily removed. Note that the tips of the curves, end points, are not detected by the previous criteria and hence remain in the surface skeleton. This does not create any problem, since they are voxels isolated from the rest of the surface. In the second step, voxels having $\overline{N}_f^{18} = 1$ and $N^{26} = 1$ are detected as *edge voxel*. With reference to Fig. 2, grey voxels are voxels having $\overline{N}_f^{18} = 1$ and $N^{26} = 1$, hence they would be identified as edge

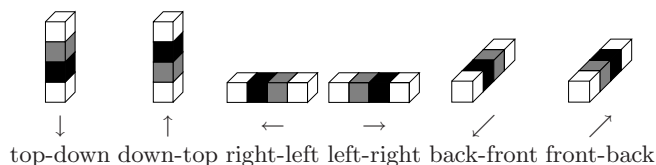


Fig. 1. Two-voxel thick configurations in the six scanning directions. The current voxel of the surface skeleton is shown in grey, the other voxel in the skeleton in black and background voxels in white

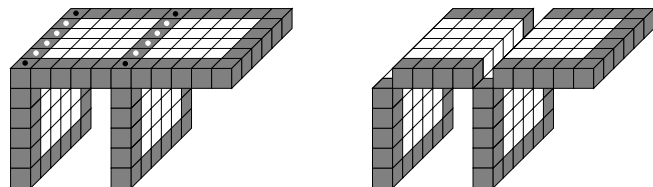


Fig. 2. Surface skeleton before, left, and after, right, removal of voxels in L-shaped configurations. Grey voxels satisfy $\overline{N}_f^{18} = 1$. See text

voxels. Removal of voxels in L-shaped configurations is indispensable to avoid that these voxels are wrongly classified.

To compute the DT of a surface, we use, analogously to the “regular” DT case, the weights w_1 for the distance to a face neighbour, w_2 for the distance to an edge neighbour, and w_3 for the distance to a point neighbour. Edge voxels are labelled w_1 . Distance information is propagated from the edge over the surface. Each voxel in the surface is assigned the minimum of the label of the voxel itself and the labels of its already visited neighbours, also belonging to the surface, increased by the corresponding weight. The propagation is repeated, alternately in forward and backward fashion, until no more changes occur. The number of scans needed depends on the complexity of the surface. For discussion about the 2D case, [8]. The number of iterations of forward and backward scans needed for the examples shown in Figs. 5 and 6 are 6 and 2, respectively.

4 Representing the Surface

Any voxel in the DT of a surface can be seen as the centre of a geodesic disc with radius equal to the distance label of the voxel, and fully contained in the surface. A geodesic disc centred on a voxel v can be computed by applying the reverse distance transformation, [4], using v as a seed and the background of the surface as an obstacle. In Fig. 3, top, a surface is shown that consists of planes that meet each other along segments in face, edge, and point directions. The geodesic discs shown in Fig. 3, bottom, are centred at the intersection of the

above segments and correspond to different DTs, DT^6 , DT^{26} , and 3-4-5 DT. The radius of the disc is the same for all DTs.

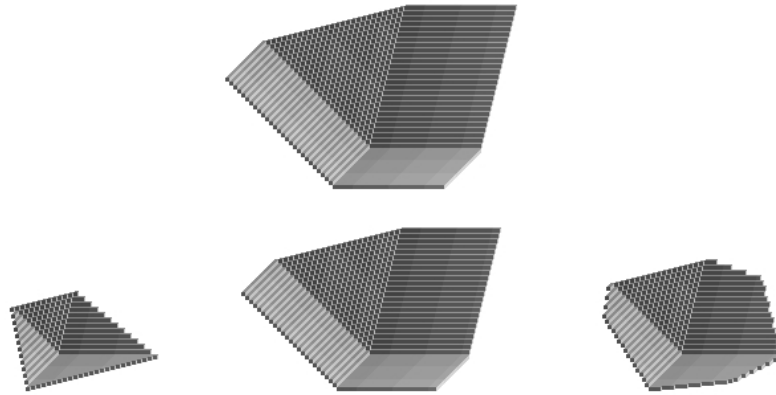


Fig. 3. Above, a surface consisting of four planes meeting each other along segments in face, edge, and point directions. Below (from left to right), geodesic discs centred at the intersection of the above segments and with radius 25, using DT^6 , DT^{26} and 3-4-5 DT, respectively

4.1 Centres of Maximal Geodesic Discs

A centre of a maximal geodesic disc can be defined in the same way as the centre of a maximal disc for 2D objects and the centre of a maximal ball for 3D objects, [1]. Thus, a *centre of a maximal geodesic disc* (CMGD) is a voxel whose corresponding geodesic disc is not completely covered by any other single geodesic disc. A voxel v in the DT is a CMGD if all its neighbouring voxels n_i , $i = 1, \dots, 26$, in the surface, with the proper weights w_j , $j = 1, 2, 3$, satisfy

$$n_i < v + w_j.$$

In Fig. 4, the sets of CMGDs for the surface shown in Fig. 3, top, are shown for DT^6 , DT^{26} , and 3-4-5 DT, respectively.

The union of the geodesic discs corresponding to the set of CMGDs is equivalent to the original surface.

4.2 Connecting the Centres of Maximal Geodesic Discs

To have a representation of the surface with the same number of components as those of the surface, the CMGDs should be linked to each other. This could be



Fig. 4. From left to right, the set of CMGDs for the DT of the surface shown in Fig. 3, top, using DT^6 , DT^{26} , and 3 – 4 – 5 DT, respectively

done, e.g., by growing paths in the direction of the steepest gradient in the DT starting from each CMGD.

Indeed, besides the CMGDs, also other voxels, here called *saddle voxels* and *branch point voxels*, are important to gain the connectedness of the final representation. Saddle voxels are defined as voxels having in the surface skeleton more than one component of neighbours with higher label, or more than one component of neighbours smaller label. A branch point voxel is defined as a voxel having more than two components of neighbours with the same label as the voxel itself. In all cases, we use 26-connectedness.

The CMGDs, saddle voxels, and branch point voxels are called *intrinsic*. They are detected and marked in one scan of the DT, before starting path growing. During path growing, suitable voxels, here called *induced* are detected and marked on the DT. The DT is repeatedly inspected until no more voxels are marked. For each already marked voxel, v , either intrinsic or induced, its neighbourhood is inspected. The voxel(s) n_i , $i = 1, \dots, 26$, on the DT, that by using the proper weights w_j , $j = 1, 2, 3$, maximise the gradient

$$grad(n_i) = \frac{n_i - v}{w_j}$$

are marked as belonging to a growing path. We denote the highest gradient $grad_{max}$.

Due to the fact that we are growing paths on surfaces, where 18-connectedness is possible, we can have paths in certain directions that alternately include one single voxel and a pair of edge connected voxels. This might create diverging paths, as the steepest gradient could occur along non-connected, diverging directions from of the two voxels. We force, if possible, path growing to continue in only one direction, whenever two or more connected neighbours equally maximise the gradient. Thus, if a voxel can propagate on a number of neighbours n_i , all with $grad(n_i) = grad_{max}$, and constituting a unique 26-connected component, we originate a single path by choosing, as the next voxel in the path, only one neighbour of the component of n_i 's.

In Figs. 5 and 6, two examples of the set of CMGDs and the corresponding connected curve representations are shown. We started from 3D solid objects,

a cylinder and a digital Euclidean ball, respectively, and computed the surface skeletons by using the algorithm introduced in [11]. These were reduced to one-voxel thick surfaces, as described in Section 3. The surface skeletons were then also simplified by pruning some non-significant peripheral curves by means of the algorithm introduced in [5]. The metric used is D^6 .

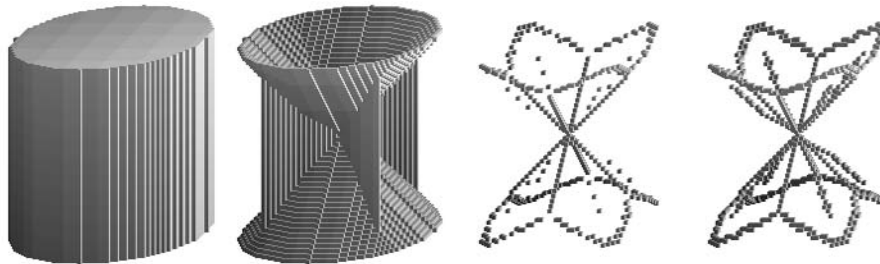


Fig. 5. From left to right: A cylinder with radius 25, and height 50, the simplified surface skeleton, the set of CMGDs, and the connected curve representation obtained after path growing

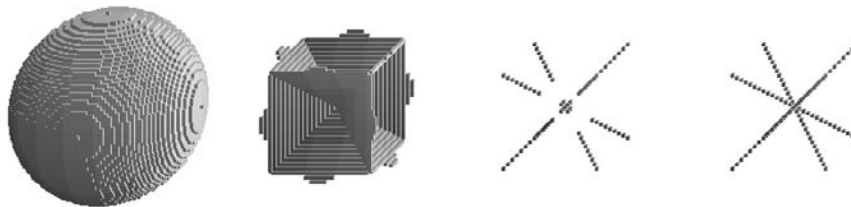


Fig. 6. From left to right: A digital Euclidean ball with radius 28, the simplified surface skeleton, the set of CMGDs, and the connected curve representation obtained after path growing

In Fig. 5, right, it is possible to see paths alternately including one voxel and a pair of edge connected voxels. We also note that the curve representation is not topologically equivalent to the surface skeleton and, hence, to the object. In fact, a number of *loops* characterise the curve representation, that do not correspond to tunnels in the surface skeleton and the object. The curve representation is not the curve skeleton of the object.

Note that it is generally preferable to remove the markers from the CMGDs with label w_1 (except for the case in which they belong to curves) before performing path growing. In fact, these CMGDs might cause the growth of a number of

noisy paths. The marker has been removed from CMGDs with label w_1 in the examples shown in this paper.

5 Conclusion

We have introduced the distance transform of surfaces in 3D images and have used it to extract the centres of maximal geodesic discs. The centres of maximal geodesic discs provide a new representation of a surface in general, and of the surface skeleton of 3D solid objects in particular. To compute the distance transform, we detect the border. This is done in two steps. First the curves are identified and temporarily removed, then the edges are identified and used for propagating distance information.

The set of centres of maximal geodesic discs is connected by a path growing process, to obtain a richer representation of the surface consisting of a set of connected curves. The representation has the same number of object components as the surface, but might have loops. Hence the resulting representation is not topologically equivalent to the surface, (which is topologically equivalent to the original object when the surface is indeed the surface skeleton of the object).

Future work will be to further investigate the surface representation here introduced and, in particular, to identify the border starting directly from two-voxel thick surfaces. This is important to make our algorithm more general. We will also investigate the distance transform for surfaces where we use the maximum distance in case of concurrent distinct distance values. This will need a more complicated algorithm but could give interesting results.

Acknowledgements

Prof. Gunilla Borgefors and Dr. Ingela Nyström, both Centre for Image Analysis, Uppsala, Sweden, are gratefully acknowledged for their scientific support.

References

1. C. Arcelli and G. Sanniti di Baja. Finding local maxima in a pseudo-Euclidean distance transform. *Computer Vision, Graphics and Image Processing*, 43(3):361–367, Sept. 1988. 445, 448
2. G. Borgefors. On digital distance transforms in three dimensions. *Computer Vision and Image Understanding*, 64(3):368–376, 1996. 445
3. G. Borgefors, I. Nyström, and G. Sanniti di Baja. Connected components in 3D neighbourhoods. In M. Frydrych, J. Parkkinen, and A. Visa, editors, *Proceedings of 10th Scandinavian Conference on Image Analysis (SCIA '97)*, pages 567–572, Lappeenranta, Finland, 1997. Pattern Recognition Society of Finland. 445
4. G. Borgefors, I. Nyström, and G. Sanniti di Baja. Computing skeletons in three dimensions. *Pattern Recognition*, 32(7):1225–1236, July 1999. 445, 446, 447

5. G. Borgefors, I. Nyström, G. Sanniti di Baja, and S. Svensson. Simplification of 3D skeletons using distance information. Accepted for publication in Proceedings of SPIE International Symposium on Optical Science and Technology: Vision Geometry IX (vol. 4117), 2000. 450
6. L. Dorst and P. W. Verbeek. The constrained distance transformation: a pseudo-Euclidean recursive implementation of the Lee algorithm. In I. T. Young, J. Biemond, R. P. W. Duin, and J. J. Gerbrands, editors, *Signal Processing III: Theories and Applications*, pages 917–920. Elsevier Science Publishers B. V. (North-Holland), 1986. 443
7. G. Malandain, G. Bertrand, and N. Ayache. Topological segmentation of discrete surfaces. *International Journal of Computer Vision*, 10(2):183–197, 1993. 445, 446
8. J. Piper and E. Granum. Computing distance transformations in convex and non-convex domains. *Pattern Recognition*, 20(6):599–615, 1987. 447
9. I. Ragnemalm and S. Ablameyko. On the distance transform of line patterns. In K. A. Høgda, B. Braathen, and K. Heia, editors, *Scandinavian Conference on Image Analysis (SCIA '93)*, pages 1357–1363. Norwegian Society for Image Processing and Pattern recognition, 1993. 443, 446
10. P. K. Saha and B. B. Chaudhuri. 3D digital topology under binary transformation with applications. *Computer Vision and Image Understanding*, 63(3):418–429, May 1996. 445, 446
11. G. Sanniti di Baja and S. Svensson. Surface skeletons detected on the D^6 distance transform. In F. J. Ferri, J. M. Iñetsa, A. Amin, and P. Pudil, editors, *Proceedings of SSSPR 2000 - Alicante: Advances in Pattern Recognition*, pages 387–396, Alicante, Spain, 2000. Springer-Verlag, Berlin Heidelberg. Lecture Notes in Computer Science 1121. 446, 450
12. J.-I. Toriwaki, N. Kato, and T. Fukumura. Parallel local operations for a new distance transformation of a line pattern and their applications. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(10):628–643, 1979. 443, 446