

Linear representation of discrete surfaces in 3D

Carlo Arcelli, Gabriella Sanniti di Baja, Luca Serino
Istituto di Cibernetica "E.Caianello", CNR
Via Campi Flegrei 34, 80078 Pozzuoli, Naples, Italy
(c.arcelli, g.sannitidibaja, l.serino)@cib.na.cnr.it

Abstract

A method to compute a linear medial representation of a complex surface in the 3D discrete space is presented. The method involves voxel classification, surface labeling, anchor point detection, and voxel removal.

1. Introduction

In the image processing literature, much attention has been devoted to the medial representation of objects, mirroring an analogous interest in the physical and biological fields. Most of the research activity in medial representation has been inspired by the work of Blum [1], dealing with the notion of symmetry point and a growth process, and aimed at identifying the locus of the centers of the spheres, bitangent two sections of the boundary of the object and entirely contained in the interior of the object. In other words, the medial linear representation consists of the points placed along the symmetry axes that lie on the object.

Similarly to other representation schemes, e.g., the boundary representation, the medial representation associates to an object with a given dimensionality a subset with lower dimensionality. Differently from other representation schemes, the medial representation directly represents the object interior. Other relevant features of a medial representation are: i) the possibility to hierarchically rank its constituting components, so as to provide a multiscale representation of the object, ii) the topological equivalence between object and representative structure, and iii) the ability to capture the most salient aspects of the object's shape, producing a stylized version of the object that makes the representation appealing and intuitive also to non computer scientists. The last two properties are particularly useful when the medial representation is used for object recognition and classification tasks.

Our interest is towards a medial linear representation of complex surfaces in the 3D discrete space. A complex surface consists of intersecting 2D manifolds, *sheets*, and 1D manifolds, *lines*. Surface voxels can be termed *curve voxels* if they belong to lines; *branching voxels* if they are placed at intersections among lines, or among lines and sheets; *junction voxels*, if they are placed at the intersection among at least three sheets; *edge voxels*, if

they border sheets; and *internal voxels*, otherwise. We call *internal sheets* the sheets delimited exclusively by junction voxels, and *peripheral sheets* the sheets at least partially delimited by edge voxels, see Fig. 1.

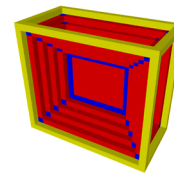


Fig. 1. A surface with peripheral and internal sheets. Yellow, blue and red denote edge, junction and internal voxels, respectively.

In this paper, we introduce a method to compute the medial linear representation of a complex discrete surface S . A number of elements, detected in S as anchor points, should be ascribed to its representation, to reflect the geometrical structure and shape of S . Symmetry points in the sheets of S should be detected as anchor points, since we follow the approach of Blum. Curve, branching, and junction voxels in S should also be taken as anchor points, since they carry relevant structural information. Moreover, voxels necessary to link the anchor points should be identified, to ensure topological equivalence between S and its representation. The method involves voxel classification; computation of a sort of geodesic distance transform, so as to structure S by a suitable labeling; anchor point detection; and sequential voxel removal by topology preserving operations.

This paper widens the applicability of a recently suggested method for the computation of the curve skeleton of 3D objects [2]. The main novelties regard an improved voxel classification scheme, the introduction of a simple criterion to segment the surface into its constituting manifolds, and a straightforward way to obtain surface labeling.

2. Notions

We use binary voxel images in cubic grids, where the object, actually a complex surface S , is the set of 1's and the background is the set of 0's. The 26-connectedness is chosen for S and the 6-connectedness for the background.

The discrete counterpart of a continuous surface includes voxels that, with the exception of the edge voxels

and of the voxels on tips of lines, cannot be removed without altering topology. We note that, differently from the continuous case, intersections of sheets in a discrete surface may include configurations consisting of $2 \times 2 \times 2$ object voxels (see Fig. 2).

We call $N(p)$ the $3 \times 3 \times 3$ neighborhood of any voxel p . $N(p)$ includes the 6 face-, the 12 edge- and the 8 vertex-neighbors of p . In turn, we call $N^*(p)$ the set including the 6 face- and the 12 edge-neighbors of p .

We denote by n_p the number of 26-connected object components computed in $N(p)$, and by n_p^* the number of 6-connected components of background voxels having p as face-neighbor and computed in $N^*(p)$.

A voxel p of S is *simple* if the surface including p is homotopic to the surface deprived of p . Voxel simplicity means that the numbers of components and tunnels of S are the same, independently of whether p is in the surface or in the background. In [3,4] it has been shown that a voxel is simple if $n_p=1$ and $n_p^*=1$.

The geodesic distance between two voxels p and q of a surface is measured as the length of a minimal discrete path linking p to q and consisting exclusively of voxels of the surface. The length of the path can be measured by weighting with suitable integer weights the moves towards the different kinds of neighbors encountered along the path, to take into account that moves towards different neighbors have different Euclidean lengths. We use the weights $w_f=3$, $w_e=4$ and $w_v=5$, as suggested in [5], to respectively measure moves from a voxel towards its face-, edge- and vertex-neighbors.

The geodesic distance transform GDT of a surface provides a labeling of the surface. To compute GDT, three sets are identified in the image: an object of interest (OI), i.e., the set of surface voxels that receive and propagate distance information; a reference set (RS), i.e., the set of voxels from which distance information is initially propagated; and a set of barrier voxels (BV), i.e., the set of voxels through which distance information does not flow.

In a standard distance transform, computed when the two sets OI and RS exhaust the image, the symmetry points are identified as the centers of the maximal balls of the object. Specifically, an object voxel p with distance value p is a center of a maximal ball if for each of its object neighbors n_i , with distance value n_i , (for $i=f, e, v$) there results $n_i < p + w_i$, [5]. We use this criterion to identify the symmetry points also in GDT.

3. Voxel classification and surface labeling

To classify the voxels of S , the following checks are orderly done:

Any object voxel p having at most two disjoint object neighbors is classified as *curve voxel*.

Any not yet classified object voxel having a neighboring curve voxel is classified as *branching voxel*.

Any not yet classified voxel p such that $n_p^* \neq 1$ is classified as *internal voxel*.

Any internal voxel p whose object neighbors are all branching voxel is re-classified as *branching voxel*.

Any internal voxel p having at least three 6-connected components of background voxels in $N(p)$, or being any of the eight object voxels in a $2 \times 2 \times 2$ configuration is re-classified as *junction voxel*.

Any internal voxel having a face- or an edge-neighbor classified as junction voxel is re-classified as *extended junction voxel*.

Any not yet classified voxel is classified as *edge voxel*.

An example of the obtained classification is shown in Fig. 2, where green, orange, blue, grey, red and yellow are used to identify curve, branching, junction, extended junction, internal and edge voxels.

The classification allows us to identify all the manifolds constituting the complex surface. To this purpose, all junction, extended junction, branching and edge voxels are removed from the surface, so that a number of components equal to the number of manifolds is obtained. In particular, extended junction voxels have a key role to separate intersecting sheets, where voxels of a sheet have edge-neighbors in another sheet. Each manifold is obtained by adding to the corresponding component the extended junction, branching and edge voxels having a neighbor in the component, as well as junction and edge voxels adjacent to any already added extended junction voxel. Moreover, it is possible to distinguish internal and peripheral sheets, since for an internal sheet none of the added voxels is an edge voxel.

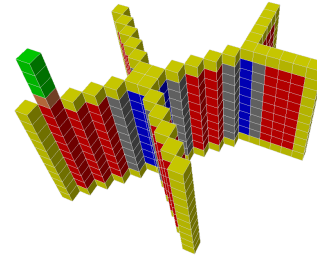


Fig. 2. Voxel classification (see text).

To mark as anchor points the symmetry points of S , we resort to a surface labeling by means of the geodesic distance transformation.

For the peripheral sheets of S , the symmetry points have to be identified with respect to the sets of edge voxels associated to the sheets. To this aim, for peripheral sheets we compute the geodesic distance transform so as to assign to each internal voxel of a sheet a value equal to its distance from the closest voxel in the associated set of edge voxels.

To find symmetry points in internal sheets, GDT has to be computed so as to assign to each internal voxel of a sheet a value equal to its distance from the closest voxel in the set of junction voxels delimiting that sheet. To this

aim, we consider, among the voxels added to build the internal sheets, those arranged to form closed curves as actually delimiting internal sheets. We re-classify them as *delimiting junction voxels*. In turn, extended junction voxels and junction voxels arranged to form open curves are seen as internal voxels. We re-classify them as *new internal voxels*.

To compute GDT of the whole surface S we use two sets RS1 and RS2, two sets OI1 and OI2, and two sets BV1 and BV2. Specifically, RS1 includes the edge voxels (we assign the label w_f to these voxels), OI1 includes the internal voxels and the extended junction voxels of the peripheral sheets (extended junction voxels receive, but do not propagate, distance information so as to prevent flowing of information from voxels in a sheet to their edge neighbors in other intersecting sheets); BV1 includes all remaining voxels of the image. In turn, RS2 includes the delimiting junction voxels (we assign to these voxels a large label, e.g., $M \times w_f$, where M is the largest among the three dimensions of the image); OI2 includes the internal voxels, the new internal voxels and the extended junction voxels of the internal sheets (now the extended junction voxels both receive and propagate distance information); BV2 includes all remaining voxels of the image.

We resort to a recursive algorithm that, starting from two buffers B1 and B2 (initially including the voxels of RS1 and RS2, respectively) identifies for each voxel in the buffers the neighbors in OI1 and OI2 to which distance information should be propagated, and labels these neighbors with the proper distance value. Any voxel of OI1 that has received distance information is added to B1, unless is an extended junction voxel. Voxels of OI2 are added to B2 anyway, as soon as they are labeled. When a voxel in B1 (B2) has been considered and has possibly provided distance information to its neighbors, the voxel is removed from the buffer, which will result to be empty at the end of the process.

The rationale for considering junction voxels as barrier voxels is to avoid that distance information, transferred to the voxels of a sheet, could propagate through the junction voxels also to other intersecting sheets. For instance, with reference to Fig. 3, we should guarantee that the violet voxels in sheet A, are labeled with the length of the shortest path to the set of (yellow) voxels bordering A, notwithstanding the fact that they are actually closer to the set of (yellow) voxels bordering B. In this way, we can correctly identify in every sheet the voxels that are symmetry points with respect to the set of voxels bordering that sheet.

Once the voxels of S are classified and GDT is computed, we mark as anchor points in GDT the symmetry points, as well as the voxels of S classified as curve, branching and junction voxels. Note that delimiting junction voxels and new internal voxels are not marked as anchor points, otherwise the iterative topology preserving voxel removal process would not be effective for voxels

of internal sheets. Also note that we avoid to identify the symmetry points in proximity of junction voxels. In fact, marking these symmetry points as anchor points would only create unwanted thickening of the representation in correspondence with intersections. As an example, see Fig. 3, where the violet voxels, all with the same distance from the set of edge voxels bordering A, would be detected as symmetry points by the criterion in Section 2. Finally, in proximity of intersections with internal sheets, even if some voxels, originally classified as junction voxels, have been re-classified as delimiting junction voxels, detection of unwanted symmetry points does not occur. In fact, the label $M \times w_f$ assigned to the delimiting junction voxels ensures that in a peripheral sheet a voxel p with a neighbor labeled $M \times w_f$ cannot fulfill the symmetry point criterion $n_i < p + w_i$.

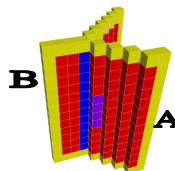


Fig. 3. Violet voxels should receive distance information from the edge voxels bordering A.

4. Extracting the linear representation

The medial linear representation is obtained by an iterative removal process applied to the voxels of S that are not anchor points. The remaining voxels of S are visited in increasing label value order. At each iteration, voxels with the same label are sequentially checked against removal and are removed if are simple points.

We note that in the presence of internal sheets, starting from the iteration in which the checked voxels are those labeled $M \times w_f$, the sequential removal process may create a bias in the position of the linear representation. The representation, instead of being medially placed, would be shifted depending on the adopted raster fashion.

To solve this problem, some parallelism is taken into account during removal of voxels with label $M \times w_f$ or larger. Specifically, a preliminary test is done in parallel on all checked voxels to inhibit from removal those simultaneously characterized by $n_{p^*} \neq 1$. Then, only the remaining voxels undergo sequential removal. The inhibition and removal process is repeatedly performed before passing to the next iteration, as far as current voxels can be removed. In fact, due to removal of some voxels, previously inhibited voxels may be no longer characterized by $n_{p^*} \neq 1$.

The removal process terminates when all labels have been considered. The obtained representation can be 2-voxel thick at parts and may include parts, which do not account for significant shape information. Two processes, concerning final thinning and pruning, are used to obtain

a linear representation, rid of unnecessary peripheral curves. These processes are not illustrated in detail for space limitation. Final thinning is achieved by means of directional processes, during each of which thickening in a given direction is detected, and topology preserving removal operations are used to gain unit thickness in that direction. Pruning removes peripheral curves and is based on the ratio R between the number of significant voxels (i.e., voxels that were marked as anchor points) and the total number of voxels in the curves.

Some examples showing the obtained linear representations are given in Fig. 4. To favor visualization, the orientations of the surfaces and of their corresponding representations are not necessarily the same.

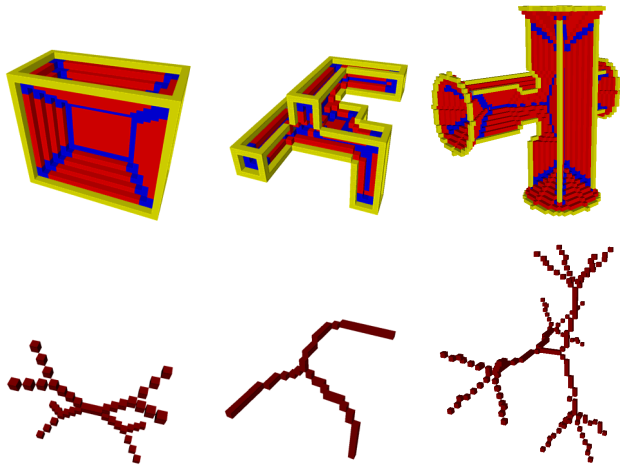


Fig. 4. Surfaces showing edge, internal and junction voxels, top, and their medial representations, bottom.

5. Concluding remarks

A method to compute a medial linear representation of a discrete surface S has been described. Our method is intended for surfaces for which the main symmetry axes lie on the surfaces themselves. In this case, the subset of the surface consisting of the symmetry points gives a perceptually meaningful representation of the surface. On the contrary, this does not happen for a simple tubular surface, e.g., a piece of a pipeline, where the main symmetry axis is centrally placed in the portion of the background surrounded by the surface. In fact, in this case our representation would consist only of the ring equidistant from the two components of edge voxels delimiting the tubular surface. Of course, the method cannot be applied to surfaces surrounding cavities, due to topological reasons.

To obtain the representation, the voxels in S are classified and a propagation of distance information is done towards internal voxels, starting from edge and delimiting junction voxels. Voxels in each peripheral sheet derive distance information exclusively from the edge voxels bordering the peripheral sheet itself. In turn,

voxels of internal sheets derive distance information from the delimiting junction voxels bordering the internal sheets. Voxel classification and labeling allow us to guide the iterative removal process, by accessing at each iteration only voxels with the same label value, as well as to select as anchor points the symmetry points, the curve and the junction voxels. The resulting representation is centered within S , is topologically homotopic to S , reflects the geometry of S , and is 1-voxel thick. The use of the weights 3, 4 and 5 to compute GDT guarantees that a reasonable stability under surface rotation characterizes the representation. Pruning makes the algorithm robust with respect to noise and, by using different threshold values for the ratio R , could be used to rank hierarchically subsets of the representation.

The performance of the algorithm with respect to stability and robustness can be seen in Fig. 5, with reference to a rectangular shape in different orientations and bendings and in the presence of noise. The linear representation has in all cases the same structure and shape.

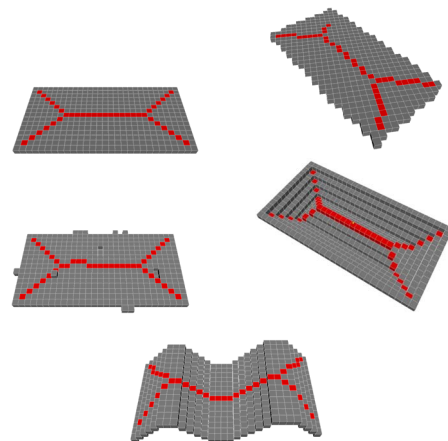


Fig. 5. Linear medial representation (red voxels) superimposed on the surface (gray voxels).

References

- [1] H.Blum, Biological shape and visual science, *J. Theor. Biol.*, 38, 205-287, 1973.
- [2] C.Arcelli, G.Sanniti di Baja, L.Serino, From 3D discrete surface skeletons to curve skeletons, in: A. Campilho and M. Kamel (Eds.), *ICIAR 2008*, LNCS 5112, Springer-Verlag, Berlin, 507-516, 2008.
- [3] G.Bertrand, G.Malandain, A new characterization of three-dimensional simple points, *Pattern Recognition Letters*, 15/2, 169-175, 1994.
- [4] P.K.Saha, B.B.Chaudhuri, Detection of 3D simple points for topology preserving transformations with application to thinning, *IEEE Trans. PAMI*, 16, 1028-1032, 1994.
- [5] G.Borgefors, On digital distance transform in three dimensions, *CVIU*, 64/3, 368-376, 1996.