

Approximating the Skeleton for Fine-to-Coarse Shape Representation

Luca Serino, Carlo Arcelli, and Gabriella Sanniti di Baja

Institute of Cybernetics “E. Caianiello”, CNR, Naples, Italy
{l.serino,c.arcelli,g.sannitidibaja}@cib.na.cnr.it

Abstract. A method to generate a hierarchical skeleton structure is presented. The curve skeleton of a 3D object is used, where each voxel is labeled with the radius of the associated ball, i.e., with its distance from the complement of the object. Polygonal approximation is accomplished on all skeleton branches represented in a 4D space, where the coordinates are the (x,y,z) coordinates plus the radius r associated with each skeleton voxel. In this way, skeleton branches are divided into geometrically straight line segments, whose voxels are characterized by either constant or linearly increasing/decreasing radius. By increasing the threshold used for polygonal approximation the hierarchical skeleton structure is generated, which allows the user to get a fine-to-coarse shape representation.

Keywords: Skeleton, shape representation, polygonal approximation.

1 Introduction

The skeleton is a well known representation system and has been used in the framework of shape analysis for 2D and 3D objects. A wide literature is available as concerns different skeletonization methods, devised in the continuous and in the digital space, and the use of the skeleton in several application fields [1].

The skeleton is a homotopic subset of the object. It consists of the union of curves, termed skeleton branches, each of which can be seen as in correspondence with a part (main body or limb) of the object. Skeleton branches can be classified in two categories, respectively including peripheral and internal branches. The former branches are in correspondence with limbs of the object and are characterized by the presence of end points on their tips, i.e., points of the skeleton having only one neighbor in the skeleton. The latter branches are in correspondence with main bodies of the object from which the limbs protrude and are delimited by the so called branch points, i.e., points of the skeleton with more than two neighbors in the skeleton. Branch points are the skeleton points where different branches meet.

Skeleton points are symmetrically placed within the object. For 2D objects, skeleton points are all aligned along symmetry axes of the object. In turn, for 3D object this is the case only if the object consists exclusively of parts with tubular shape. In the general case, symmetry points of 3D objects are placed along planes and

axes, so that only a linear subset of the set of the symmetry points can be included in the skeleton.

An important skeleton feature for shape analysis is related to the reconstruction ability of the skeleton. This means that the represented object can be fully reconstructed starting from its skeleton. To have such an important feature, each skeleton point should be associated with its distance from the complement of the object, i.e., with the radius of the ball that, centered on the skeleton point, touches the boundary of the object in at least two distinct parts. Moreover, the skeleton should include all the symmetry points of the object. Actually, the skeleton of a 3D object generally is not characterized by full reconstruction ability, since only a subset of the symmetry points of the object can be included in the skeleton. For example, refer to Fig. 1, where an input object, its skeleton and the object reconstructed from the skeleton are shown. Though full recovery is not possible, the reconstructed object is still a meaningful approximation of the input object. Thus, differences between input objects and reconstructed objects can be disregarded and in this paper we will consider the objects reconstructed by the skeletons as if they were our input objects.



Fig. 1. An input object, left, its skeleton, middle, and the reconstructed object, right

Another interesting skeleton feature, particularly useful in the framework of the structural approach to object recognition, is the possibility to represent shape in a hierarchical manner. In fact, different skeletons of the same object can be computed, each of which characterized by a different level of detail. Thus, the user can select the skeleton having enough detail to solve the current task.

Two main approaches exist to build a hierarchical skeleton structure. The most common approach is based on skeleton pruning. Given the initial skeleton, its branches are associated a parameter measuring the perceptual relevance of the corresponding object part. Then, peripheral branches can be pruned in increasing perceptual relevance order so generating a sequence of skeletons, each of which is a proper subset of the previous skeleton and whose structure becomes simpler and simpler. The second approach divides each branch of the skeleton into a number of parts by means of a polygonal approximation or by using an approximation with curves of higher order. By increasing the value of the threshold used during the approximation, rougher and rougher representations of the skeleton are obtained.

With the first approach (e.g., [2]), the objects recovered by the successively pruned skeletons differ from the input object for the absence of an increasing number of peripheral parts. Thus, the hierarchical skeleton structure represents an object characterized by simpler and simpler shape until a very rough shape is obtained, coinciding with what could be understood as the main body of the object. In turn, by

following the second approach (e.g., [3]), no object's limbs are lost when approximating the skeleton with a higher value for the threshold, but the geometry of each object part is represented in a rougher and rougher manner.

The literature on multi-scale skeletonization driven by pruning includes contributions mainly for the 2D case. In the 3D case, pruning is mainly used as a necessary post-processing step to clean the skeleton. In fact, the skeleton is strongly conditioned by the presence of boundary noise, especially with continuous skeletonization methods such as those based on the use of the Voronoi Diagram. An extension of pruning to build a hierarchical skeleton structure also in the 3D space seems to be straightforward. In turn, as regards the second approach, to our knowledge only methods effective in the 2D space are available. In this paper we follow the second approach to provide a hierarchical skeleton structure for 3D objects.

2 Basic Notions

We use binary voxel images in cubic grids. The $3 \times 3 \times 3$ neighborhood of a voxel p includes the six face- the twelve edge- and the eight vertex-neighbors of p .

The distance between two voxels p and q is given by the length of a minimal path from p to q . If the same weight is used for all unit moves, the D^{26} distance is obtained, which is the 3D version of the 2D chessboard distance. In turn, if the weights 3, 4 and 5 suggested in [4] are respectively used to measure the unit moves from p towards its face-, edge- and vertex-neighbors along the path, the $\langle 3,4,5 \rangle$ -distance is obtained. The $\langle 3,4,5 \rangle$ -distance provides a good approximation to the Euclidean distance.

The skeleton S of an object P is a homotopic subset of P , consisting of curves centered in P , whose voxels are labeled with their distance from the complement of P . We use the algorithm [5] to compute the $\langle 3,4,5 \rangle$ -distance labeled skeleton.

A voxel p of S is an end point if has only one neighbor in S , a normal point if has two neighbors in S , and a branch point if has more than two neighbors in S .

A skeleton branch is a connected subset of S entirely consisting of normal points, except for the two extremes of the branch that are end points or branch points. If both extremes are branch points, the skeleton branch is an internal branch. All other skeleton branches are peripheral branches.

The reverse distance transformation [6] is a process that associates to a set of voxels, labeled with integer distance values, the envelope of the balls centered on the voxels and having radii equal to the corresponding distance values. We use the $\langle 3,4,5 \rangle$ -distance when applying the reverse distance transformation to the skeleton.

3 Polygonal Approximation

Different approaches to compute the polygonal approximation of a digital curve are available in the literature. The split type approach [7] is particularly convenient when working with open curves, as it is the case for skeleton branches. In fact, in this case the two starting points from which the process recursively splits a curve are the extremes of the curve itself. This guarantees that the vertices identified along the

curve are not influenced by the selection of the two starting points, as it would be the case for a closed curve.

Roughly speaking, a split type polygonal approximation algorithm can be described as follows. The two extremes of the input open curve are taken by all means as vertices of the polygonal approximation. The Euclidean distance of all points of the curve from the straight line joining the two vertices is computed. The point with the largest distance is taken as a new vertex, provided that such a distance overcomes an a priori fixed threshold θ (to be set depending on the desired approximation quality). Otherwise the process terminates. If a new vertex is detected, such a vertex divides the curve into two sub-curves, to each of which the above split type algorithm is applied. The splitting process is repeated as far as points are detected having distance larger than the threshold from the straight lines joining the extremes of the sub-curves to which the points belong. When recursion is completed, the curve results to be approximated by a number of segments that, in the limit of the adopted tolerance, are rectilinear. The curve is represented by the ordered sequence of all and only the points that have been detected as vertices.

For the sake of simplicity, let us consider the object shown in Fig. 2 left, whose skeleton consists of a unique branch. By comparing Fig. 2 left and Fig. 2 middle, we may observe that the geometry of the skeleton reflects the geometry of the object. In fact, curvature changes along the skeleton correspond to curvature changes along the boundary of the object. Thus, a polygonal approximation of the skeleton would divide it into straight line segments. Each segment would be in correspondence with a region of the object characterized by the absence of curvature changes along its boundary, and could be seen as the spine of that region.

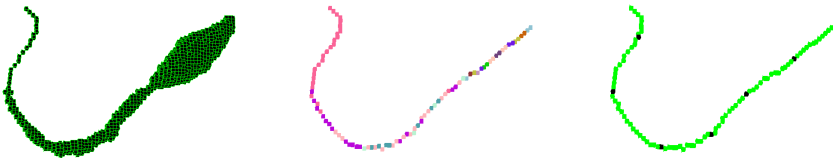


Fig. 2. From left to right, an object, its skeleton (different colors denote different distance values), and the vertices (black) resulting in the skeleton after polygonal approximation in 4D

Still with reference to Fig. 2, we also note that the different distance values r_i of the voxels p_i successively met when tracing the skeleton from the first extreme to the second extreme (i.e., the different radii of the balls associated with the skeleton voxels) take into account the changes in width of the object. By plotting the ordered pairs (p_i, r_i) on a Cartesian plane, a 2D curve is obtained and a split type polygonal approximation can be performed to set vertices wherever the radii fail to be aligned in the limits of the adopted tolerance. The skeleton could then be divided into subsets such that the radii of the corresponding voxels are either constant, or change in a linear manner. Each subset could be interpreted as the spine of a region of the object that is characterized either by constant thickness or by thickness that linearly increases/decreases.

If changes in geometry and distance values along a skeleton branch are simultaneously considered, polygonal approximation will divide the skeleton branch into subsets characterized by linearity both in geometry and in the distribution of distance values. In this way, each segment of the skeleton branch will correspond to a simple region characterized by the following two properties: 1) absence of significant curvature changes along the boundary and 2) thickness that either is constant or linearly increases/decreases. To this purpose, we here perform polygonal approximation in a 4D space, where the four coordinates are the three Cartesian coordinates and the radii of the voxels of the 3D skeleton branches.

For the sake of completeness, we note that while a skeleton branch is obviously a connected set in the 3D space, this is no longer guaranteed when the skeleton branch is represented in the above 4D space. There, a skeleton branch may result in a sparse set of points. However, this does not influence the performance of polygonal approximation.

The computation of the Euclidean distance d of a point C from the straight line joining two points A and B in the 4D space is done by using the following expression:

$$d^2 = \|AC\|^2 - P_{ABC} * P_{ABC} / \|AB\|^2$$

where $\|AB\|$ is the norm of the vector AB, and P_{ABC} is the scalar product between vectors AB and AC. If the point C is the one at maximal distance from the straight line joining A and B, C is taken as a vertex of the polygonal approximation provided that its distance d satisfies the condition $d > \theta$.

Once all vertices have been detected in the 4D space, the skeleton voxels in the 3D space corresponding to them are marked as vertices. In the limits of the adopted tolerance, these vertices divide the skeleton branch into straight line segments, whose voxels are characterized by radii that either are constant or linearly increase/decrease. See Fig. 2 right, where the vertices are shown in black. Thus, a skeleton branch can be efficiently represented by orderly giving only the spatial coordinates and radii of the found vertices.

Each segment of a skeleton branch is interpreted as the spine of a simple region of the object part associated to the whole skeleton branch. A simple region is shaped as a cylinder (Fig. 3 middle left), when radii are all equal along the spine (Fig. 3 left, where all skeleton voxels, depicted with the same color, have the same radius), or as a truncated cone (Fig. 3 right), when radii linearly increase/decrease along the spine (Fig. 3 middle right, where different colors denote different radii). Of course, the simple region is delimited by two half balls centered on the two vertices of the spine.

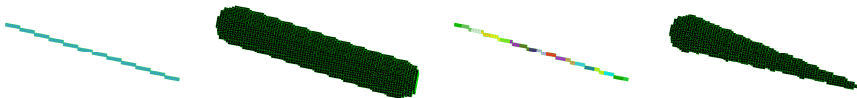


Fig. 3. Straight line skeleton segments and their corresponding simple regions

4 The Hierarchical Skeleton Structure

The procedure described in the previous section is applied to all skeleton branches, so originating an approximated version of the entire skeleton. By using different values for the threshold θ , differently approximated skeletons are obtained. In particular, the more the threshold increases, the more the approximated skeleton actually represents a rough version of the object.

The smallest value that can be taken into account for the threshold is $\theta=1$. In fact, due to the chord property [8], if the distance of each point of a digital curve from the straight line joining the extremes of the curve is at most equal to 1, then the digital curve is a digital straight line segment. To fix the maximum possible value for the threshold, we compute for each skeleton branch the maximal distance of its voxels from the straight line joining the extremes of the branch. Then, the maximum μ among all computed maximal distances is taken as the maximum value for the threshold θ . In fact, with this value for the threshold each skeleton branch is obviously approximated by the straight line segment joining the two extremes of the branch, since no skeleton voxel can be found characterized by distance larger than μ . It would be useless to consider a larger threshold.

Starting from the minimal value, the threshold is increased by a fixed incremental amount (set to 0.5 in this work), until μ is achieved. In principle, each obtained polygonal approximation of the skeleton should be recorded to build the hierarchical skeleton structure. However, the current polygonal approximation is skipped if it has the same number of vertices as the previously recorded polygonal approximation. In fact, it would be identical to the previous approximation.

Once all threshold values have been taken into account, the hierarchical skeleton structure is available. At a given level of the hierarchy, which is in correspondence with a given threshold value, each skeleton branch is represented by the sequence of vertices detected along that branch during the polygonal approximation by using that threshold.

To visualize the representations and to understand how the different polygonal approximations more or less faithfully represent the initial skeleton, we build the approximated skeletons starting from the available information, i.e., the ordered sequences of spatial coordinates and radii of the vertices.

Let (x_i, y_i, z_i) and $(x_{i+1}, y_{i+1}, z_{i+1})$ be the spatial coordinates of two successive vertices v_i and v_{i+1} and let r_i and r_{i+1} be the associated radii. Let Δx , Δy , Δz and Δr be the absolute values of the differences between homologous coordinates of the two vertices.

The number L of voxels in a minimal length path linking v_i and v_{i+1} is given by the maximum among Δx , Δy and Δz , i.e., by the D^{26} distance between v_i and v_{i+1} . We need to establish how many moves should be done via each of the three kinds of neighbors. In this respect, we note that, since a move via vertex-neighbors requires that the three spatial coordinates all change by 1, the total number of vertex-moves is conditioned by the minimum among Δx , Δy and Δz . For the sake of simplicity, let us suppose that the minimum is Δx . Then, Δx is the number of moves via vertex-neighbors. With similar arguments, the number of moves via edge-neighbors will be the minimum out

of the remaining two differences of coordinates Δy and Δz , each of which diminished by Δx . Finally, the remaining moves, if any, are via face-neighbors. Thus, we can easily compute the number of moves necessary in total to link v_i and v_{i+1} and how many of these moves occur via each of the three types of neighbors.

However, several minimal length paths with the same number of moves for each kind of neighbors can be built. Thus, to build the minimal length path aligned along the digital straight line joining v_i and v_{i+1} , the different moves have to be suitably alternated along the path.

To this aim, let us denote by m_f , m_e and m_v the number of moves in face, edge and vertex direction respectively. Let us consider the minimum among m_f , m_e and m_v . For simplicity, we suppose that the minimum is m_v . Let us also consider the sum of m_f and m_e . The ratio $(m_f + m_e)/m_v$ counts the number of moves in face and edge directions, after which one move via a vertex-neighbor has to be done. To determine the way in which the moves via face- and edge-neighbors alternate, we compute the minimum between m_f and m_e . For simplicity, let us suppose that the minimum is m_e . Then the ratio m_f/m_e indicates after how many moves via face-neighbors a move via edge-neighbor should be done. Special cases where the ratios are not integer numbers or the denominator is zero are properly treated.

To select the first move to be accomplished starting from v_i , we compute the maximum among m_f , m_e and m_v . For simplicity, suppose that such a maximum is m_f . Then, the first move in the path is in face direction. Since six face-neighbors of v_i exist, the proper face-neighbor to be accepted in the path is the one at minimal distance from v_{i+1} . To select the remaining voxels constituting the path, it is enough to analyze only a subset of the neighborhood of the current voxel, which is determined based on the position of the previous voxel in the path. Among the neighbors of the current voxel, only those that can be reached with the proper move are considered and the one at minimal distance from the vertex v_{i+1} is selected as the next voxel in the path.

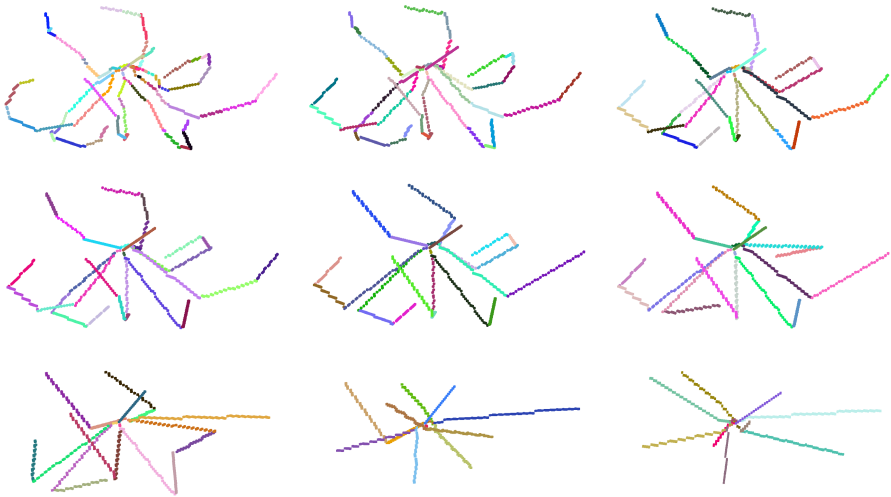


Fig. 4. Polygonal approximations of the skeleton with different threshold values

Since the hierarchical skeleton structure is recorded by the ordered sequences of spatial coordinates and radii of the vertices, the only available radii are those pertaining any two successive two vertices v_i and v_{i+1} of a given approximation. To distribute linearly distance values to all voxels in the path that we have just shown how to build, we make use of the number L of voxels in the path, i.e., the D^{26} distance between v_i and v_{i+1} , and of Δr , i.e., the absolute value of the difference between r_i and r_{i+1} . The ratio $\Delta r/L$ indicates the increment in radius that, starting from the smallest radius (say the radius r_i that pertains to vertex v_i), will allow us to ascribe the proper radius to each voxel in the segment until we reach the second extreme of the segment (v_{i+1} to which the radius r_{i+1} pertains). Then, the so computed radii are approximated by the closest integers, in order each representation in the hierarchical structure can be regarded as a digital skeleton.

As an example, see Fig. 4, where nine different polygonal approximations are shown. Colors are used to distinguish skeleton segments. Actually, 17 different approximations were available for the input skeleton, but for space restriction only a few approximations are shown, namely those corresponding to $\theta=2.5$, $\theta=4.5$, $\theta=6.5$, $\theta=8.5$, $\theta=10.5$, $\theta=13.5$, $\theta=24.5$, $\theta=32.5$ and $\theta=34.5$. Since for $\theta=34.5$ the nine branches of the skeletons are approximated by nine straight line segments, it would have no sense to consider larger values for the threshold.

To show how faithfully the object is represented when the skeleton is approximated by using an increasing threshold value, we apply the reverse distance transformation to the approximated skeletons. This is possible since we have assigned integer distance values to the voxels constituting the spines approximating the skeleton branches. In Fig. 5 the objects recovered starting from the skeletons of Fig. 4 are orderly shown.

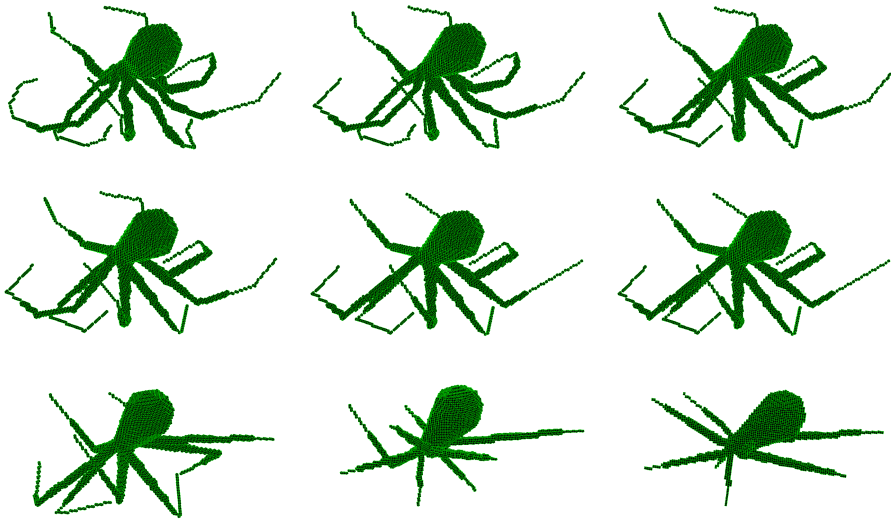


Fig. 5. Objects recovered by applying the reverse distance transformation to the input skeleton and its approximated versions

As expected, when the threshold increases the approximated skeleton becomes a more schematic representation of the input object. For example, since the roughest polygonal approximation of the skeleton (Fig. 4 bottom right) consists only of nine straight line segments, the corresponding recovered object (Fig. 5 bottom right) is the combination of nine simple parts (as many parts as many are the branches of the input skeleton) and information on the fact that, for example, the tentacles of the octopus are bent is no longer available. Actually, the straight line segments joining the extremes of the branches associated with the tentacles of the octopus are not at all centered within the actual tentacles of the input object.

To build a hierarchical skeleton structure able to represent the object's parts without excessive loss of shape information, one should use as maximum threshold a value definitely smaller than μ . Even better, one should fix a different maximum threshold value for each skeleton branch, which should obviously be smaller than the maximal distance of the skeleton voxels along that branch from the straight line joining the extremes of the branch. A possible solution is to use as maximum threshold value for each branch the largest radius associated with the skeleton voxels along the branch. In this way the risk of replacing the input skeleton, definitely centered within the object, by a skeleton whose voxels are likely to be placed outside the object is significantly reduced.

We work with a discrete skeleton whose voxels are labeled with the $\langle 3,4,5 \rangle$ -distance, while during polygonal approximation we detect vertices by computing the Euclidean distance. Thus, if we want to set the maximal threshold value in function of the maximal distance value of the skeleton voxels in a branch, we need to properly evaluate the Euclidean length of the radii. Roughly speaking, if we divide by 3 the distance value r_i of a skeleton voxel p_i , we obtain the number of moves in face direction to reach the complement of the object starting from p_i . Thus, a reasonable approximation of the Euclidean length of the radius of p_i is given by $(r_i/3)+0.5$, where the amount 0.5 is added to take into account that also the voxel p_i contributes to the length of the radius.

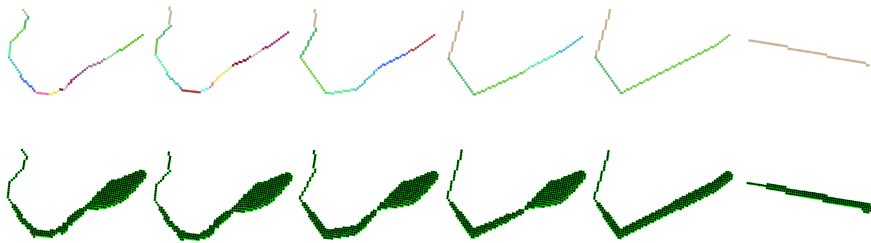


Fig. 6. Top, approximated versions of the skeleton, not all equally meaningful. Bottom, the corresponding reconstructed objects.

In Fig. 6, an example is given with reference to an object whose skeleton consists of a single branch, for which the largest distance of the skeleton voxels from the straight line joining the extremes of the branch μ equals 41.5. Eight different polygonal approximations are possible, respectively obtained for $\theta=1.5$, $\theta=2.5$, $\theta=3.5$

(not shown in Fig. 6 for space restrictions), $\theta=4.5$, $\theta=5.5$, $\theta=11.5$, $\theta=12.5$ (not shown in Fig. 6 for space restrictions) and $\theta=41.5$. In Fig. 6 bottom, the corresponding reconstructed objects are shown. For this object, the largest radius along the branch equals 18, if measured with the $\langle 3,4,5 \rangle$ -distance. Its equivalent Euclidean length is 6.5. If such a value is taken as the maximal threshold value, the hierarchical skeleton structure would include only the first four approximated skeletons shown in Fig. 6 top (corresponding to the first four reconstructed objects in Fig. 6 bottom). In fact, $\theta=11.5$ is the first threshold value for which a polygonal approximation is obtained with a smaller number of vertices with respect to the approximated version obtained for $\theta=5.5$. Thus, $\theta=6.5$ originates the same approximation as $\theta=5.5$.

5 Conclusion

We have introduced a method to build a hierarchical skeleton structure, starting from the curve skeleton of a 3D object. Skeleton branches are represented in a 4D space, where the coordinates are the (x,y,z) coordinates plus the radius r associated with each skeleton voxel. Then, for each skeleton branch polygonal approximation is accomplished so as to divide skeleton branches into straight line segments, whose voxels have either constant or linearly increasing/decreasing radius. Different threshold values are used so as to obtain differently approximated versions of the skeleton. The hierarchical structure includes all the differently approximated skeletons, which allows the user to get a fine-to-coarse shape representation. The algorithm has been implemented on an Intel Core i7 (3.5 GHz, 8 GB RAM) personal computer and tested on images taken from publicly available shape repositories, e.g. [9].

References

1. Siddiqi, K., Pizer, S.M. (eds.): *Medial Representations: Mathematics, Algorithms and Applications*. Springer, Berlin (2008)
2. Shen, W., Bai, X., Hu, R., Wang, H., Latecki, L.J.: Skeleton growing and pruning with bending potential ratio. *Pattern Recognition* 44, 196–209 (2011)
3. Sanniti di Baja, G., Thiel, E.: A multiresolution shape description algorithm. In: Chetverikov, D., Kropatsch, W.G. (eds.) *CAIP 1993*. LNCS, vol. 719, pp. 208–215. Springer, Heidelberg (1993)
4. Borgfors, G.: On digital distance transform in three dimensions. *CVIU* 64, 368–376 (1996)
5. Arcelli, C., Sanniti di Baja, G., Serino, L.: Distance driven skeletonization in voxel images. *IEEE Trans. PAMI* 33, 709–720 (2011)
6. Nystrom, I., Borgfors, G.: Synthesising objects and scenes using the reverse distance transformation in 2D and 3D. In: Braccini, C., Vernazza, G., DeFloriani, L. (eds.) *ICIAP 1995*. LNCS, vol. 974, pp. 441–446. Springer, Heidelberg (1995)
7. Ramer, U.: An iterative procedure for the polygonal approximation of plane curves. *CGIP* 1, 244–256 (1972)
8. Rosenfeld, A.: Convex digital arcs. *IEEE Trans. Computers* C-23, 1264–1269 (1974)
9. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: *The Princeton Shape Benchmark*. In: *Shape Modeling International*, Genova, Italy (2004)