2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Using local convexities as anchor points for 3D curve skeletonization

Luca Serino and Gabriella Sanniti di Baja
Institute for High Performance Computing and Networking, CNR
Naples, Italy
luca.serino@cnr.it, gabriella.sannitidibaja@cnr.it

***Abstract* — A new algorithm is introduced to compute the curve skeleton of 3D objects by using the notion of local convexity. The centers of maximal balls detected on the distance transform of the object are filtered to select as anchor points only those located on sharp local convexities of the object's boundary. Then, the skeleton is obtained by means of topology preserving removal operations. Pruning is finally accomplished to remove from the skeleton scarcely significant peripheral branches.**

Keywords — 3D object; local convexity; curve skeleton; pruning

I. INTRODUCTION

The curve skeleton of a 3D object is a homotopic subset of the object consisting of curves placed along the main symmetry axes of the object. Skeletonization is based on the notion of symmetry point and on the growth process that were introduced by Blum, initially for 2D objects and successively for objects in higher dimensions, to define the Medial Axis Transform, MAT [1]. A symmetry point p is an object's point at the same distance r from at least two distinct sections of the boundary of the object. The growth process associates to p all object's points having a distance from p smaller than or equal to r. In this way, a maximal ball centered on p and with radius r is built. The ball touches the boundary in at least two distinct parts and is entirely contained in the object. The MAT is the locus of the symmetry points, and the envelope of all the maximal balls coincides in shape and size with the object.

Papers dealing with continuous and discrete approaches to the computation of the 3D curve skeleton as well as with the use of the skeleton for applications can be found in the literature [2-15]. In this paper, we follow the discrete approach and deal with the computation of the curve skeleton in voxel images. In particular, we suggest a method that is a mixture of skeletonization based on iterated voxel removal and skeletonization based on the use of the distance transform of the object.

Each iteration of skeletonization by means of iterated topology preserving voxel removal consists of two sub-iterations: during the first sub-iteration, border voxels are identified; during the second sub-iteration, the found border voxels are sequentially checked and are removed provided that they are not necessary for topology preservation or to account for relevant object's shape information. Skeletonization terminates when all object voxels are border voxels and they all result un-removable. An evident drawback of this skeletonization approach is its high computational cost when applied to rather thick objects. In fact, in this case a large number of iterations are necessary to obtain the skeleton. Another drawback is that finding a reliable criterion to preserve from removal the object's points accounting for shape information is not an easy task. A generally followed solution to this problem is to prevent from removal the so called end points, by setting a threshold on the maximum number of object's neighbors of the border voxels that should be preserved from removal. However, such a criterion does not guarantee an isotropic behavior of skeletonization. In fact, local configurations of border voxels with the same structure but with different orientation, may or may not be mapped into end points depending on the order in which voxels are visited.

To solve the above drawbacks, we resort to the use of the Distance Transform, DT, and to the selection of the Centers of Maximal Balls, CMB, in the DT. In fact, the DT is a multi-valued replica of the object, where each voxel is assigned the value of its distance from the complement of the object. Each object voxel in the DT can be interpreted as the center of a ball with radius equal to the distance value of the voxel itself and touching the object boundary. Clearly, a CMB is a symmetry point. Thus, a discrete approximation of the MAT is possible by selecting the CMB in the DT.

We use the DT to identify as anchor points a suitable set of CMB, so as to guarantee that the most relevant information on object's shape be reflected by the skeleton. Moreover, we exploit the DT to avoid repeated inspections of the image by directly accessing in increasing distance value order the voxels that undergo removal, so as to significantly reduce the computational cost of iterated thinning.

Actually, not all the CMB should be taken as anchor points for 3D curve skeletonization for at least two reasons: 1) unless we consider exclusively objects composed by snake-shaped parts, the CMB concentrate along symmetry planes and symmetry axes. Thus, CMB filtering is necessary at least as concerns the CMB placed along symmetry planes; 2) some CMB are due only to the discrete nature of the digital space and do not carry shape information. For example, sections of a cylinder in the continuous space are circles and only the centers of these circles are CMB, but sections of a cylinder in the digital space are polygons and, besides the centers of these polygons, also some spurious CMB are detected due to the weak convexities at the vertices of the polygons. Thus, also some convexity based filtering of the CMB is necessary.

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

Our 3D curve skeletonization method is accomplished in three phases respectively dealing with 1) computation of a subset of the object consisting of surfaces and curves, SURF, 2) computation of a subset of SURF consisting exclusively of curves, CURV, and 3) pruning scarcely significant peripheral branches of CURV to originate the curve skeleton, SKEL.

To obtain SURF, the following processes are accomplished. The DT is computed and a convexity based filtering of the CMB is used to select the anchor points. Topology preserving removal operations based on the notion of simple point are applied to the voxels that are not anchor points. Final thinning is applied to the obtained nearly thin set to gain unit thickness.

To compute CURV, the border voxels delimiting SURF are iteratively identified. Border voxels that are not placed along curves or on sharp convexities are possibly removed by means of topology preserving removal operations. Final thinning is then accomplished to gain unit thickness.

Finally, pruning based on the evaluation of the number of unit wide slices of the object that would not be recovered from the pruned skeleton is accomplished to originate SKEL.

The algorithm in this paper follows a similar strategy as the algorithm in [10], but the specific criteria are different. In particular, for the computation of SURF the notion of local convexity is extended from the 3×3×3 to the 5×5×5 neighborhood; the computation of CURV is accomplished by resorting to iterated border voxel removal; the pruning criteria is completely new.

II. Basics Notions and Definitions

We refer to binary voxel images in cubic grid, where the object consists of voxels with value 1 and the background consists of voxels with value 0.

The 3×3×3 neighborhood $N_{26}(p)$ of a voxel p includes the six face- the twelve edge- and the eight vertex-neighbors of p. The set $N_{18}(p)$ includes only the six face- and the twelve edge-neighbors of p. The set $N_{124}(p)$ includes all voxels except p in the 5×5×5 block centered on p.

The <3,4,5>-path-based-distance between two voxels p and q, is the length of a minimal 26-connected path from p to q, where the unit moves from p towards face-, edge- and vertex-neighbors along the path are weighted 3, 4 and 5, respectively. As it has been shown in [16,17], the small maximal error of this discrete path-based distance justifies its use in place of the Euclidean distance and provides the advantages offered by the simplicity of path-based distances.

The distance transform DT is obtained by assigning to each voxel of the object its <3,4,5>-path-based-distance from the background.

The ball of radius r centered on a voxel p with distance value $d(p)=r$ in the DT, is built by applying to p the reverse distance transformation [18].

A voxel p in the DT is center of a maximal ball, CMB, if for each of its neighbors q it results: $d(q) - d(p) < w$, where w is 3, 4, or 5 depending on whether q is a face-, an edge-, or a vertex-neighbor of p [19].

The layers of the DT are subsets of the DT including voxels with any of the three possible values between two successive multiples of the smallest weight w=3. The k-th layer is the set of voxels q such that $(k\text{-}1)\times 3 < d(q) \le k\times 3$ [19].

Voxels in the k-th layer of the DT can be interpreted as belonging to the *border* that would characterize the object at the k-th iteration of skeletonization accomplished by iterated voxel removal; voxels belonging to any successive layer h, $h>k$, can be interpreted as belonging to the *inside* at the k-th iteration.

Inspired by previous work on concavity filling [20], in [10] we gave the following definition: a border voxel p is placed on a locally planar surface if $N_{26}(p)$ includes nine background neighbors, eight border neighbors and nine inside neighbors. Thus, as soon as the number of inside neighbors is less than nine, the border voxel p is on a *local convexity* of $N_{26}(p)$. The smaller is the number of inside neighbors, the sharper is the convexity. Accordingly, a threshold θ can be fixed on the maximum number of inside voxels that a border voxel p can have to be considered as placed on a sharp convexity in $N_{26}(p)$. In this work, we extend the notion of local convexity in $N_{124}(p)$ and say that a border voxel p is on a locally planar surface if $N_{124}(p)$ includes 50 background neighbors, 24 border neighbors and 50 inside neighbors. In turn, a border voxel p is on a *local convexity* to $N_{124}(p)$ as soon as the number of inside neighbors is less than 50. Obviously, the smaller is the number of inside neighbors in $N_{124}(p)$, the sharper is the local convexity where p is located.

The number of 26-connected components of object voxels in $N_{26}(p)$, and the number of 6-connected components of background voxels having p as face-neighbor and computed in $N_{18}(p)$ are respectively denoted by $cc_{26}(p)$ and by $cc_{18}(p)$.

An object voxel p is a simple point if the topology of the object does not change if p is removed. From an operative point of view, p is simple if $cc_{26}(p)=1$ and $cc_{18}(p)=1$, [21,22].

The voxels of the curve skeleton are called end points, normal points and branch points, depending on whether they have one, two or more than two neighbors in the skeleton.

III. Phase 1: Computation of SURF

The following processes are accomplished to compute SURF: DT computation, CMB extraction, CMB filtering via local convexity, topology preserving voxel removal, final thinning.

The first task, computation of the distance transform DT, is accomplished by using the standard 2-raster-scan algorithm [16]. Once the DT is available, CMB detection is performed by taking into account the distance values of each voxel and of its neighbors in $N_{26}(p)$.

Among the CMB, we select as anchor points those placed on sharp local convexities in $N_{124}(p)$, i.e., only the CMB for which the number of inside neighbors in $N_{124}(p)$ is at most equal to an a priori fixed threshold θ_1. In this paper we regard a local convexity of $N_{124}(p)$ as sharp if the dihedral angle is at most 90 degrees. Since for a 90 degree dihedral angle the number of inside voxels in $N_{124}(p)$, is eight, we set θ_1=8 and consider p as

placed on a sharp convexity of $N_{124}(p)$ only if the number n of inside neighbors of p is such that $n \leq 8$.

Actually, besides any CMB p placed on a sharp local convexity, we select as anchor point also any neighbor of p in $N_{26}(p)$ that is a CMB and has the same distance value as p. This additional selection of anchor points is done to obtain a skeleton with a more perceptually satisfactory structure.

In Fig. 1, a 3D object used as running example and the selected anchor points are shown.

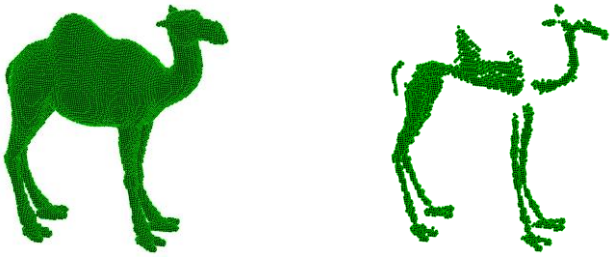

Fig. 1. A 3D object, left, and the selected anchor points, right.

To obtain SURF, we remove from the object in increasing distance value order all simple voxels that are not anchor points. Actually, to obtain a skeleton where components of CMB grouped into superficial structures are connected to each other without altering this structure, we proceed as follows. Let r be the currently examined distance value and suppose that all removable voxels with distance value r have been removed. Before checking for removal voxels with distance value $r+1$, any voxel with distance value r, already accepted as skeletal voxel, marks as skeletal voxels its neighbors in $N_{26}(p)$ that are the most suitable for linking it to the inside. These neighbors are detected as those maximizing the directional derivative of the skeletal voxel with distance value r with respect to each of its neighbors with distance value larger than r. The directional derivative from p to a neighbor q is computed as $(d(q)-d(p))/w$, where w is 3, 4, or 5 depending on whether q is a face-, an edge- or a vertex-neighbor of p.

Once all distance values have been processed, a nearly thin set is obtained, where thickening may occur in face- or edge-direction. See Fig. 2 left.

Fig. 2. The nearly thin set, left, and the unit wide set SURF consisting of surfaces and curves, right.

To identify thickening in face- or in edge-direction, we use directional templates, each consisting of four voxels aligned along one of the six face-directions or along one of the twelve edge-directions. The two voxels in the middle of the template are object voxels, while the remaining two voxels are background voxels. See Fig. 3. Final thinning is done, as in [10], by resorting to six (twelve) directional removal processes that act on thickening in face-direction (edge-direction). The resulting set SURF is shown in Fig. 2 right.

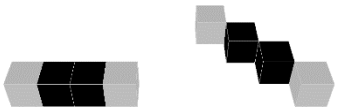

Fig. 3. Templates to detect thickening in face- and edge-direction.

IV. Phase 2: Computation of CURV

The following processes are accomplished to compute CURV: iterated detection of the anchor points, topology preserving voxel removal, and final thinning.

The set SURF, originated by the first phase of skeletonization, is a set consisting of surfaces and curves in the 3D space. Voxels already belonging to curves in SURF should definitely be kept in CURV. Hence, they should be taken as anchor points. Voxels belonging to the surfaces of SURF should undergo the iterated removal process so as to obtain the one-dimensional set CURV. Of course, removal should regard at each iteration only voxels delimiting the current surfaces, provided that these voxels are not necessary to account for object's shape. Thus, voxels placed on sharp convexities should be taken as anchor points.

To identify all the anchor points at the current iteration, we use the notion of simple point. A voxel p is not simple if $cc_{26}(p) \neq 1$ or $cc_{18}(p) \neq 1$. The former case certainly occurs if p is on a curve of SURF; the only exception for curve voxels is when p is on the tip of a curve, where it is $cc_{26}(p)=1$. Thus, voxels for which it is $cc_{26}(p) \neq 1$ or the number of object neighbors is equal to one are identified as belonging to curves and are taken as anchor points. In turn, p may be not simple if $cc_{18}(p) \neq 1$. This is certainly the case if p is a voxel that is internal in a surface and, as such, it should not be removed to avoid the creation of a spurious tunnel. To avoid its removal, we can consider as anchor point at the current iteration also any internal voxel. As for the remaining voxels that are all simple voxels, we select as anchor point any voxel p placed in a sharp convexity in the $5\times5\times5$ neighborhood $N_{124}(p)$. As already done when working on the boundary of the 3D object, we regard a convexity on the border of a surface as sharp in $N_{124}(p)$ when the angle is at most 90 degree. When this is the case, four voxels of $N_{124}(p)$ are internal in the surface. Thus, we set to 4 the value of the threshold θ_2 used to identify border voxels on sharp convexities. This corresponds to take p as anchor point if the number n of internal neighbors in $N_{124}(p)$ is such that $n \leq 4$.

Once all the anchor points at the current iteration have been identified, the voxels that are not anchor points are sequentially inspected and are removed if they are simple. The iterated process of anchor point detection and voxel removal terminates when all voxels are anchor points. The resulting nearly thin set is shown in Fig. 4 left.

Final thinning is then accomplished by means of directional processes able to reduce the nearly thin set to the desired unit wide set CURV. The same templates used during final thinning in the first phase of skeletonization are employed to identify parts characterized by 2-voxel thickness in face- and edge-

direction. The resulting set CURV consisting exclusively of curves is shown in Fig.4 right.

Fig. 4. The nearly thin set, left, and the unit wide set CURV consisting of curves, right.

V. PHASE 3: COMPUTATION OF SKEL

Peripheral curves should exist in CURV only in correspondence with object parts perceived as individually meaningful, e.g., limbs and relevant boundary convexities. However, notwithstanding the selection of anchor points placed on sharp local convexities, CURV may still include peripheral curves that do not correspond to perceptually significant object's parts. Thus, the third phase of the skeletonization process is devoted to pruning with the aim of removing scarcely significant branches from the skeleton, based on a significance measure of skeleton branches.

Most of the significance measures suggested in the literature have been proposed for pruning the skeleton of 2D objects. For the skeleton of 3D objects, a measure of significance involving the ratio between the number of centers of maximal balls along a skeleton branch and the total number of voxels in the branch has been suggested [10]; a noise pruning method is also available, [23], that first unglues skeleton branches at junctions and then retrieves the meaningful branches lost during ungluing using a connectivity analysis from seeds carrying significant shape information.

In this paper we introduce a new pruning method based on the evaluation of the loss in object recovery caused by pruning. To avoid altering the topology, only peripheral curves undergo pruning. Since removal of peripheral curves may cause other curves initially internal in CURV to become peripheral curves, pruning is iterated as far as prunable skeleton branches are identified. Each iteration of pruning is accomplished as follows.

First, every peripheral curve of CURV is identified as the set of skeletal voxels that can be traced starting from an end point until a branch point is reached. Each peripheral curve is assigned an identity label.

The reverse distance transformation is applied to the branch points found as the terminal voxels of all peripheral curves. The balls built in correspondence of branch points that are neighbors of each other or have a <3,4,5>-path-based-distance smaller than the sum of the corresponding distance values partially overlap. Thus, a number of connected components, called kernels, possibly different from the number of peripheral curves is found. As an example, see Fig. 5 top, where for an easier visualization we show the skeleton, left, and the kernels, right, found at the beginning of the second pruning step. We note that only 4 kernels are obtained when applying the reverse distance transformation to the branch points found as terminal points of the current 12 peripheral branches.

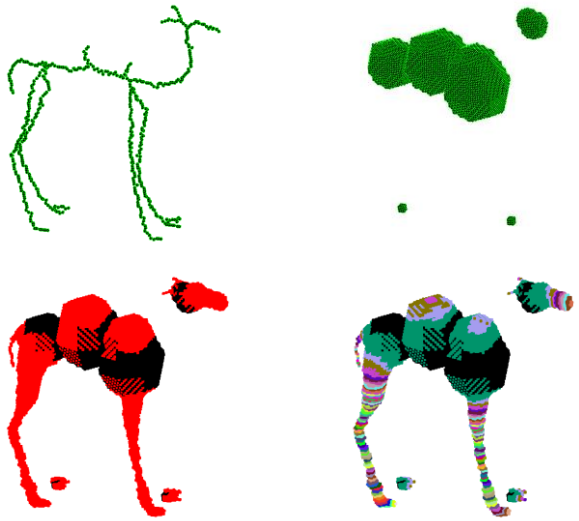

Fig. 5. The skeleton after the first pruning step, top left, and the kernels associated to the branch points delimiting the peripheral branches, top right. The limbs protruding from the kernels, bottom left, and their labeling by means of the <1,1,1>-path-based-distance from the kernels, bottom right.

The reverse distance transformation is also separately applied simultaneously to all peripheral curves. Propagation of the identity labels assigned to the curves is accomplished while performing the reverse distance transformation, so as to individually identify the regions associated to the different curves. Voxels that could be assigned more than one identity label, e.g., the voxels in the ball centered on a branch point where different curves meet, are all assigned a common identity label denoting voxels in overlapping regions. At the end of the reverse distance transformation, the voxels in the overlapping regions are removed, so originating the regions, called limbs, whose voxels are recovered exclusively by the peripheral curves. See Fig. 5 bottom left, where the limbs are colored in red.

To evaluate the significance of a peripheral curve, we count the number of "slices" of the corresponding limb that jut from the kernel including the branch point delimiting that peripheral curve. The larger is the number of slices, the larger is the perceptual relevance of the limb. To count the number of slices, we compute the distance transform of the limbs with respect to the kernels. Since the image includes besides limbs and kernels also a third set, the background, that cannot be traversed by the flow of distance information, an ordered propagation algorithm, e.g., [24], is used to compute the distance transform. We use the <1,1,1>-path-based-distance with equal weights for face-, edge- and vertex-neighbors to count the number of slices jutting from the kernel in any of the three principal directions. See Fig. 5 bottom right, where different colors denote the slices detected by means of the <1,1,1>-path-based-distance. If a distance value d exists in the distance transform of a limb, this means that the limb juts from the kernel for at least d slices.

Setting a threshold δ on the minimum number of slices that should protrude from the kernels to consider the limbs as perceptually significant obviously depends on the needs of the user for the specific application. Moreover, the value of δ

certainly depends also on image resolution. If the same image is available at different resolutions, different values for δ should be adopted. Finally, fixing for all limbs a unique value for δ is not advisable. We should take into account that kernels may have different sizes and the relevance of any limb should take into account, besides the number of protruding slices of that limb, also the importance in terms of volume of the relative kernel. In any case, peripheral curves mapped into limbs jutting at most two slices are generally regarded as noisy skeleton branches, whichever application and image resolution are considered.

By taking into account the above considerations, we set the value of δ as follows. Let d be the largest distance value found in the distance transform of the j-th limb l_j associated with the peripheral curve c_j. Let us denote by m the maximum number of slices that the user accepts to lose when recovering the object from the pruned curve skeleton, at the resolution of the images in the application at hand. Let the i-th kernel k_i be the one including the terminal branch point of the peripheral curve c_j. Let us denote by K_i the volume of k_i, measured by the number of voxels constituting k_i. Finally, let us denote by Σ the sum of the volumes of all the kernels. We set $\delta = m \times K_i / \Sigma$, so as to take into account the relevance of each kernel. As for the value of m, we have set m=10, since we have experimentally found that such a value provides in the average satisfactory results when working with images of size 128×128×128, as the ones used in this work. We regard as prunable the peripheral curve c_j with its terminal branch point in k_i if the following condition is satisfied:

$$d \leq 2 \text{ OR } d \leq \delta \qquad (1)$$

Fig. 6. The skeleton before pruning, left, and at the end of pruning, right.

Once the significance evaluation has been accomplished for all peripheral branches and the scarcely significant ones have been pruned, new peripheral branches may be created. Thus, pruning is newly applied. If at any iteration of pruning only one kernel is found, only the check $d \leq 2$ is done to measure the relevance of peripheral curves. Pruning terminates when no more removable peripheral branches are identified. The resulting pruned skeleton SKEL is shown in Fig. 6 right. The skeleton before pruning is shown again in Fig. 6 left to better appreciate the performance of pruning. For the running example, the process terminates after two iterations of pruning.

VI. Experimental Results

We have tested the skeletonization algorithm on a large set of objects taken from repositories of 3D shapes of public domain, such as the Princeton Shape Benchmark [25] and the McGill 3D Shape Benchmark [26]. A small dataset of eight images of size 128×128×128, the obtained curve skeletons before and after pruning are shown in Fig. 7.

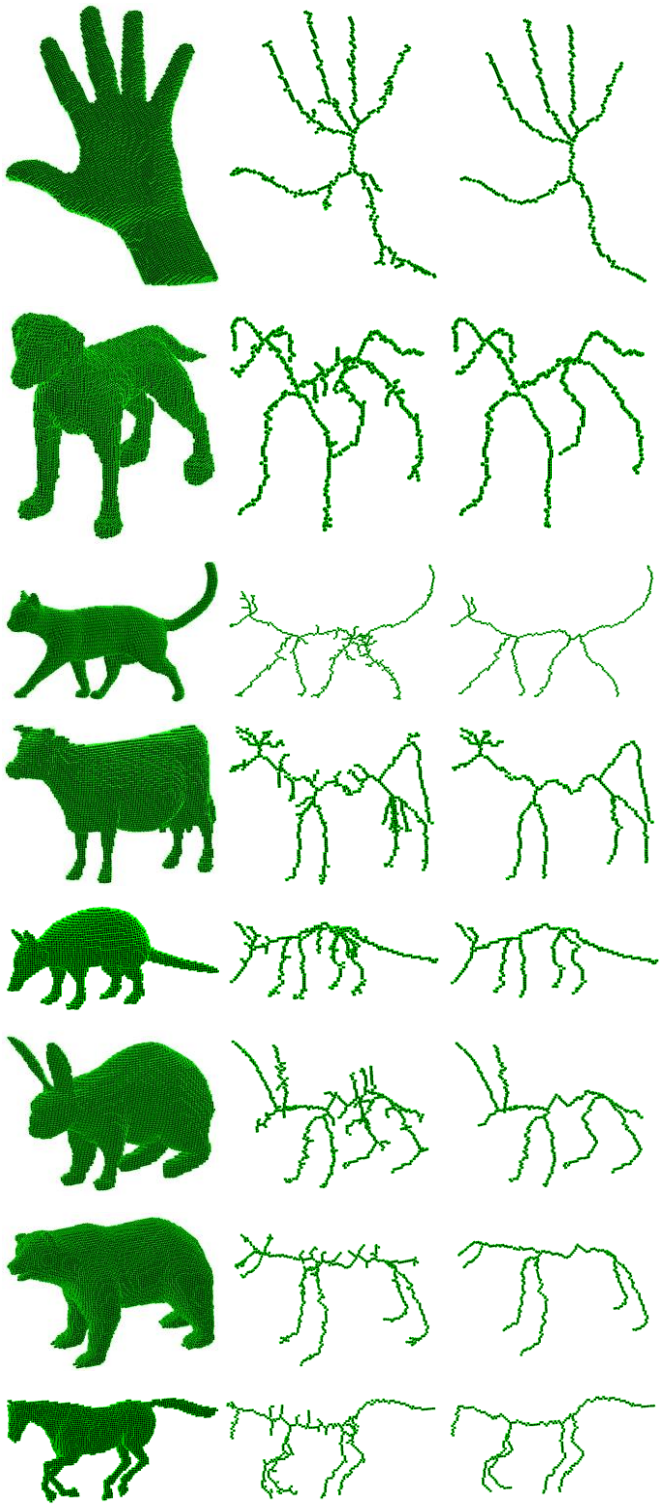

Fig. 7. From left to right, input object, its skeleton befor pruning, and after pruning.

A skeletonization method should be evaluated in terms of the properties expected for the skeleton. The 3D curve skeleton is expected: 1) to have the same topology as the object, 2) to be unit wide and centrally placed within the object, 3) to reflect object's geometry. These properties are satisfied by SKEL: 1) topology is preserved since we remove sequentially only simple

voxels, 2) skeleton centrality is guaranteed by selecting as anchor points suitably filtered CMB, which are all symmetrically placed within the 3D object, and unit thickness is satisfied by final thinning; 3) object's geometry is also reflected by SKEL, which includes branches corresponding to the main parts intuitively perceived as composing the objects.

Of course only partial recovery is possible when applying the reverse distance transformation to the skeleton, unless we deal with objects consisting only of snake-shaped parts. To check quantitatively whether a skeletonization algorithm has a reasonably good ability in object recovery, we suggest to evaluate the percentage of object's voxels that are not recovered starting from the curve skeleton. The smaller is such a percentage, the better the curve skeleton represents the object. For the sample dataset, the average percentage of non recovered object voxel is 20,10% starting from CURV and 26,08% starting from SKEL.

As for the sensitivity to noise and stability under object rotation, we point out that the images taken from the available repositories are rather clean. Otherwise, even if the suggested pruning is good enough to deal with complex cases, we recommend to use some cleaning before skeletonization to reduce the creation of spurious branches. Finally, the <3,4,5>-path-based-distance is a good approximation of the Euclidean distance and, hence, favors skeletonization stability under object rotation.

We point out that the thresholds θ_1 and θ_2 on local convexity, set to 8 and 4 in the experiments carried on in this work, can be modified to select weaker or sharper convexities. As for the threshold δ on the minimum number of slices that should protrude from the kernels to consider the limbs as perceptually significant, the value $\delta = m \times K_i / \Sigma$ used in this work has provided satisfactorily results. We are currently investigating how to adapt the value automatically to the case of images at different resolution.

VII. Conclusion

We have introduced a new skeletonization method based on the notion of local convexity. The distance transform, DT, of the object is computed and the centers of maximal balls, CMB, are detected in the DT. Filtering is then accomplished to select as anchor points only the CMB that are located on sharp local convexities. Once the anchor points have been selected, skeletonization is done by resorting to topology preserving removal operations applied to the object voxels in increasing distance value order. This process originates a set, SURF, consisting of surfaces and curves, which is then reduced to a set, CURV, exclusively consisting of curves by means of iterated anchor point detection and voxel removal. A pruning phase is also taken into account to remove skeleton branches associated to regions that are scarcely meaningful.

References

[1] H. Blum, A transformation for extracting new descriptors of shape, in Models for the Perception of Speech and Visual Form, W. Wathen-Dunn, Ed., MIT, Cambridge, MA, pp. 362–380, 1967.

[2] Y. Zhou, A. Kaufman, A. W. Toga, Three-dimensional skeleton and centerline generation based on an approximate minimum distance field, The Visual Computer, 14, pp. 303-314, 1998.

[3] J. Chuang, C. Tsai, Min-Chi Ko, Skeletonization of three-dimensional object using generalized potential field, IEEE Trans. PAMI, 22-11, pp. 1241-1251, 2000.

[4] I. Bitter, A.E. Kaufman, M. Sato, Penalized-distance volumetric skeleton algorithm, IEEE Trans. on Visualization and Computer Graphics, 7-3, pp.195-206, 2001.

[5] S. Svensson, I. Nyström, G. Sanniti di Baja, Curve skeletonization of surface-like objects in 3D images guided by voxel classification, Pattern Recognition Letters, 23-12, pp. 1419-1426, 2002.

[6] W. Xie, R.P. Thompson, R. Perucchio, A topology-preserving parallel 3D thinning algorithm for extracting the curve skeleton, Pattern Recognition, 36-7, pp. 1529-1544, 2003.

[7] S. Bouix, K. Siddiqi, A. Tannenbaum, Flux driven automatic centerline extraction, Medical Image Analysis, 9, pp. 209-221, 2005.

[8] T. Wang, A. Basu, A note on a fully parallel thinning algorithms and its applications, Pattern Recognition Letters, 28-4, pp. 501-506, 2007.

[9] O. K-C. Au, C-L. Tai, H-K. Chu, D. Cohen-Or, T-Y. Lee, Skeleton Extraction by Mesh Contraction, ACM Trans. Graphics, 27/3, Article No. 44, 2008.

[10] C. Arcelli, G. Sanniti di Baja, L. Serino, Distance-Driven Skeletonization in Voxel Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33- 4, pp. 709-720, 2011.

[11] T. He, L. Hong, D. Chen, Z. Liang, Reliable path for virtual endoscopy: ensuring complete examination of human organs, IEEE Trans. Visualization and Comp. Graphics, 7/4 pp. 333-342, 2001.

[12] E. Sorantin, C. Halmai, B. Erdohelyi, K. Palagyi, L.G. Nyul, K. Olle, et.al., Spiral-CT-based assessment of tracheal stenoses using 3-D skeletonization, IEEE Trans. Medical Imaging, 21/3, pp. 263-273, 2002.

[13] Y. Fridman, S.M. Pizer, S. Aylward, E. Bullitt, Extracting branching tubular object geometry via cores, Medical Image Analysis, 8/3, pp. 169-176, 2004.

[14] N.D. Cornea, D. Silver, Curve-skeleton properties, applications, and algorithms, IEEE Trans. Visualization and Computer Graphics, 13/3, pp. 530-548, 2007.

[15] P.K. Saha, G. Borgefors, G. Sanniti di Baja, A survey on skeletonization algorithms and their applications, Pattern Recognition Letters, 76, pp. 3-12, 2016.

[16] G. Borgefors, On digital distance transforms in three dimensions, Computer Vision and Image Understanding 64/3, pp. 368-376, 1996.

[17] B. Verwer, Local distances for distance transformations in two and three dimensions, Pattern Recognition Letters 12/11, pp. 671-682, 1991.

[18] I. Nystrom, and G. Borgefors, Synthesising objects and scenes using the reverse distance transformation in 2D and 3D, in Image Analysis and Processing, C. Braccini et al., Eds., LNCS 974, Springer, Berlin, pp. 441-446, 1995.

[19] S. Svensson, G. Sanniti di Baja, Using distance transforms to decompose 3D discrete objects, Image and Vision Computing, 20, 529-540, 2002.

[20] G. Borgefors, G. Sanniti di Baja, Analyzing nonconvex 2D and 3D patterns, Computer Vision and Image Understanding 63/1, pp. 145-157, 1996.

[21] P.K. Saha, B.B. Chaudhuri, Detection of 3D simple points for topology preserving transformations with application to thinning, IEEE Trans. PAMI, 16/10, pp. 1028-1032, 1994.

[22] G.Bertrand, G.Malandain, A new characterization of three-dimensional simple points, Pattern Recognition Letters, 15/2, pp. 169-175, 1994.

[23] P.K. Saha, Y. Xu, H. Duan, A. Heiner, G. Liang, Volumetric topological analysis: a novel approach for trabecular bone classification on the continuum between plates and rods, IEEE Trans Med Imaging, vol. 29, pp. 1821-38, 2010.

[24] J. Piper, E. Granum, Computing distance transformations in convex and non-convex domains, Pattern Recognition, 20/6, pp. 599-615, 1987.

[25] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The Princeton Shape Benchmark, Shape Modeling International, Genova, Italy, June 2004.

[26] http://www.cim.mcgill.ca/~shape/benchMark/, McGill 3D Shape Benchmark.