Discrete Curve Evolution on Arbitrary Triangulated 3D Mesh

Sergii Poltaretskyi[1,2,3], Jean Chaoui[1,2,3], and Chafiaa Hamitouche-Djabou[1,3]

[1] Télécom-Bretagne, 655 Avenue du Technopole, 29200 Plouzané, France
sergii.poltaretskyi@telecom-bretagne.eu
[2] IMASCAP, 65 Place Copernic, 29280 Plouzané, France
[3] Laboratoire de Traitement de l'Information Médicale, INSERM UMR 1101, 29609 Brest, France

Abstract. Discrete Curve Evolution (DCE) algorithm is used to eliminate objects' contour distortions while at the same time preserve the perceptual appearance at a level sufficient for object recognition. This method is widely used for shape similarity measure, skeleton pruning and salient point detection of binary objects on a regular 2D grid. Our paper aims at describing a new DCE algorithm for an arbitrary triangulated 3D mesh. The difficulty lies in the calculation of a vertex cost function for an object contour, as on a 3D surface the notion of Euclidean distance cannot be used. It is also very difficult to compute a geodesic angle between lines connecting vertices. We introduce a new cost function for border vertex which is only based on geodesic distances. We apply the proposed algorithm on vertex sets to compute an approximation of original contours, extract salient points and prune skeletons. The experimental results justify the robustness of our method with respect to noise distortions.

Keywords: discrete curve evolution, landmark points, vertices sets, triangulated mesh, skeleton pruning.

1 Introduction

Image Processing is one of the most rapidly evolving areas of information technology today, with growing applications in all areas of business, defense, health, space, etc. It also forms a core area of research within the computer science and engineering disciplines with more than 30 percent of the scientific publication volume in the world. It forms the basis for all kinds of future visual automation.

Very often image processing deals with object recognition, analysis and matching. A wide range of techniques that are described in the current literature operate with objects represented on regular grids, such as 2D pixel or 3D voxel images. In contrast, very few work is dedicated to object analysis on a non-regular grids; for example a 3D mesh represented with triangles, where each vertex could have an arbitrary number of connections.

The purpose of this study is to propose an efficient method to determine characteristic landmarks of binary sets of vertices lying on an arbitrary triangulated surface (blue points on Fig.1(a)). A very interesting method is described

E. Barcucci et al. (Eds.): DGCI 2014, LNCS 8668, pp. 410–421, 2014.

by Latecki et al. [5,6,7] where authors proposed iteratively evolve a boundary of a 2D object by deleting one point at each iteration. A point that is deleted at a current step is the one with the lowest cost function. The cost function is designed so that it reflects a contribution of a boundary vertex to the global object shape and it is computed for all boundary vertices. As a result we obtain a subset of vertices that best represents the shape of a given contour and can be called landmarks. Authors named their algorithm as Discrete Curve Evolution (DCE) as at every step it evolves a discrete curve to represent only the most significant parts.

The cost function for DCE is very simple and involves computing only Euclidean distances and angles. The proposed method is fast, easy to implement and demonstrates impressive results. All these facts inspire us to apply this technique to a shape defined on a triangulated surface. Unfortunately, on an arbitrary triangulated mesh the notion of Euclidean distances and angles do not exist which makes impossible to compute the cost function, and hence use the DCE. Tothe best of our knowledge, there is no publication in the literature concerning the DCE for vertices sets on polygonal mesh. In this paper we introduce a new cost function that can be easily computed on a triangulated surface and yields better results on regular 2D grids compared with the classical DCE.

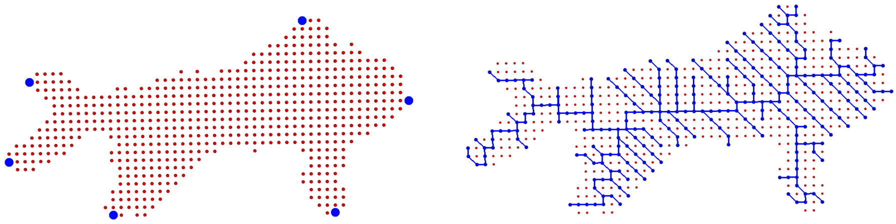
One very important application of the DCE is skeleton pruning. It is well described in the literature for a regular 2D grid [3], but there is no publication regarding sets of vertices on triangulated surfaces. This paper describes how the DCE can be used on polygonal meshes for skeleton pruning. First we propose to address a problem of skeleton extraction for sets of vertices as it is not a trivial task. Only few methods in the literature have been dedicated to this problem on triangulated surfaces. Rossl et al. [2] have presented method in which some mathematical morphology operators have been developed and applied to a set of vertices on triangulated meshes. The proposed method is very efficient and simple to implement, however it has several drawbacks leading to disconnected skeleton. A recent work described in [1] improves the previously proposed method of skeleton extraction. Authors claim that in [2], vertex classification is not sufficient as there are still unmarked vertices that are not considered in the skeletonization. To overcome the problem, a new class of vertices is proposed and new rules for topological thinning process are introduced. The proposed method has been tested on relatively homogeneous and on irregular meshes. Obtained skeletons reflect correctly the topology and geometry of input vertices' sets that proofs robustness of the approach.

An important factor of skeleton computing algorithm is its sensitivity to object's boundary deformation, as a minor noise or variation of boundary often generates redundant branches. To demonstrate it, we apply a method described in [1] to a vertices set. We show the obtained results on Fig.1(b) where lot of unnecessary branches were produced, that in turns could hamper further processing. Second major contribution of this paper is an application of our DCE algorithm for skeleton pruning on arbitrary triangulated surfaces.

Skeleton pruning is a well-known problem in 2D binary image processing. These are many different techniques proposed in the current literature; we will not list all of them as it is not the scope of current paper, an interested reader can find a good overview in [3]. We are looking for a method that yields excellent results and can be implemented for a set of vertices on a triangulated surface. We found two algorithms [3,10] that helped us to develop a solution for our problem. The first method is called Skeleton Pruning by contour partition [3]. The main idea consists in partitioning an object contour into segments and remove skeleton vertices whose generating points all lie on the same contour segment. According to Blum's definition of the skeleton [4], every skeleton point is linked to boundary points that are tangential to its maximal circle, so called generating points. The most important question is how to partition a boundary contour into segments as it plays a key role on resulting skeleton. Authors proposed to use the DCE and demonstrate excellent results. The second method [10] also uses the DCE algorithm, but for a difference purpose. Here, authors detect landmark points that actually represent skeleton ending points. Then these points are propagated inside an object with a condition that always maintains equal distance to objects' boundaries.

In this work we propose a solution for skeleton pruning that is also based on the DCE; but in our case we use a specific DCE algorithm that can also work on a triangulated surface.

Therefore, for the sake of understanding in Section 2 we briefly describe the DCE algorithm for contours on a regular 2D grid. Section 3 describes a new DCE algorithm that can be used on arbitrary triangulated surfaces. In Section 4 we show an application of the newly created DCE method to skeleton pruning on a polygonal mesh and we demonstrate obtained results.



(a) Landmarks example

(b) An example of a skeleton extracted from a set of vertices on a 3D mesh

Fig. 1. Examples of landmarks and skeleton for a set of vertices on a triangulated mesh

2 Discrete Curve Evolution

Discrete Curve Evolution (DCE) algorithm introduced in [5,6,7], was developed to neglect the distortions of objects' contours in digital images while at the same time preserve the perceptual appearance at a sufficient level for object recognition. An obvious solution is to eliminate these distortions by contour approximation (or curve evolution). The basic idea of the method is very simple. At every evolution step, a contour point with the smallest relevance measure is removed, and two of its neighbor points become connected and form a contour's line segment. The relevance measure K is given by:

$$K(s1, s2) = \frac{\beta(s1, s2)\ \ l(s1)\ \ l(s2)}{l(s1) + l(s2)} \tag{1}$$

where β is the turn angle at the common vertex of two contour's segments $s1$ and $s2$ and $l(.)$ is the length function, normalized with respect to the total length of a polygonal curve. The main property of this measure is that higher the value of $K(s1, s2)$, the larger the contribution of the curve of arc $s1 \cup s2$ to the shape [5,6].

To demonstrate the DCE and to be able to compare it with an algorithm proposed in this paper we created a 2D shape on a triangulated grid shown on Fig.2(a). To be able to use the relevance measure defined as (1), our triangulated grid is located on a plane so that Euclidean distances and angles can be easily computed. Figure 2 shows 3 stages of boundary evolution where an input image clearly has noisy borders. At first, this method allows us to smooth object's borders by removing noisy vertices (compare Fig.2(a) with Fig.2(b) and Fig.2(c)).

If we continue to evolve the curve, we will linearize digital arcs that are relevant to the curve shape, which will result in a successive simplification of the curve shape. Since in every evolution step, the number of digital line segments in the curve decomposition decreases by one, the evolution converges to a convex polygon, which defines the highest level in the shape hierarchy.

Figure2(d) shows an iteration stage where we left with 6 vertices that represent boundary contour. We can notice that the algorithm removed a top vertex that is an important point for the shape and normally should be presented for further image processing tasks such as object recognition.

3 DCE on an Arbitrary Triangulated Surface

To apply DCE algorithm to a contour formed by vertices on a triangulated surface we propose to use the same idea as for contours on a regular 2D grid. The main difference between these two cases is that on a 3D surface the notions of Euclidean distances as well as Euclidean angles do not exist. We need to find a new relevance measure that can be easily computed on a 3D surface.

If we simply try to translate a relevance measure from a regular 2D grid onto a triangulated domain we need to find equivalent measurements. A Euclidean distance could be easily replaced with a geodesic distance and can be efficiently computed with the help of Fast Marching Method (FMM) [8]. Another parameter that needs to be translated is the Euclidean angle. Compare to distances, a computation of geodesic angles is not a trivial problem. To overcome this issue

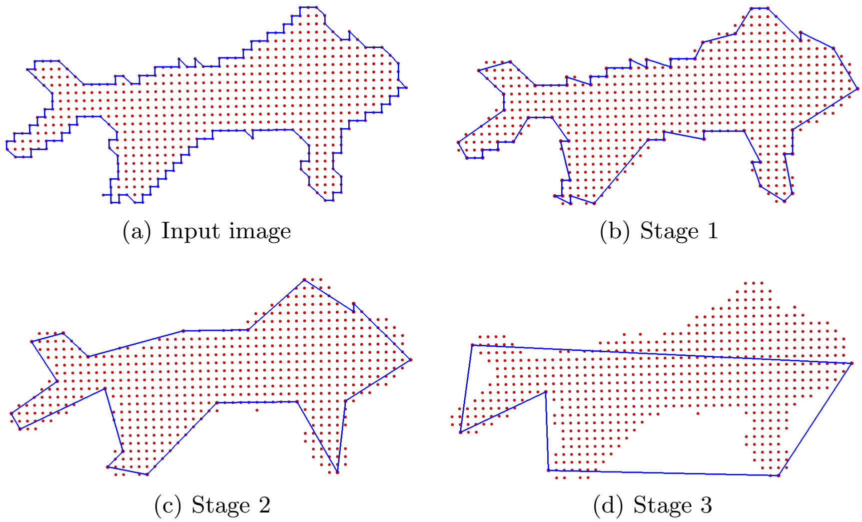


Fig. 2. Three stages of Discrete Curve Evolution with the relevance measure proposed in [6]

we propose a new measure that uses only geodesic distances to compute a cost function for each vertex.

To obtain our formula we analyze four cases shown on Fig.3. Figures 3(a), 3(b), 3(c) have spikes formed by two segments with exactly the same length but different distances between non-connected vertices that form these spikes. We observe that a spike on Fig.3(a) is not relevant compared to overall boundary while spikes on Fig.3(b) and Fig.3(c) are important. Figures 3(c) and 3(d) contain spikes with the same distances between non-connected vertices but different segments' lengths that form these spikes; where we can observe that a pick on Fig.3(d) more important than on a Fig.3(c). We conclude that a vertex cost is directly proportional to a sum of length of two segments that are connected to the studying vertex and is inversely proportional to a distance between non-connected vertices of these segments. We define a formula for a cost function as the next:

$$K = \frac{a+b}{\sqrt{c}} \tag{2}$$

where a, b, and c are shown on Fig.4(a).

It is important to note that to find a distance c between non-connected vertices of a spike we search for the shortest path between these points only inside an object that is defined by the studied contour, Fig.4(a). When a vertex is found in a concave region a distance c will be equal to the sum of segments that form a cavity, Fig.4(b):

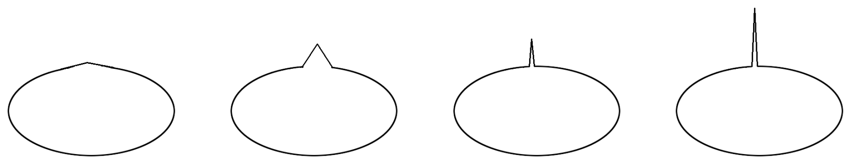
$$c = a + b \tag{3}$$

A very important point is why we use a square root of c in (2) instead of just c. As we can see, the last condition implies that we are no longer able to distinguish between "concave" vertices. Let us imagine two different concave regions:

a) $a = 10$ cm, $b = 20$ cm and $c = 30$ cm
b) $a = 1$ cm, $b = 0.5$ cm and $c = 1.5$ cm

A cost function without the square root will give a cost value 1 for both cases, while the case "a" is more important as it represents bigger segment. A cost function with the square root will give a cost value 4.48 for the case "a" and 1.22 for the case "b". Here we can easily decide to remove a vertex from the case "b".

We now have a relative measure that can be used on an arbitrary triangulated surface as it is very easy to compute geodesic distances between points.



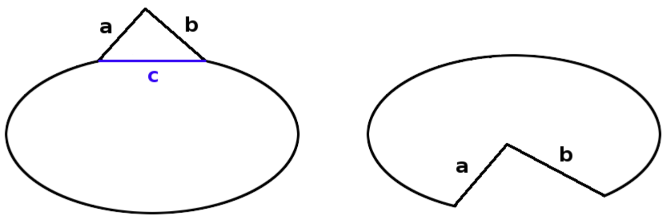
(a) A spike with edge sizes equal to 1 distance unit and a big angle

(b) A spike with edge sizes equal to 1 distance unit and a medium angle

(c) A spike with edge sizes equal to 1 distance unit and a sharp angle

(d) A spike with edge sizes equal to 2 distance units and a sharp angle

Fig. 3. The influence of a vertex connecting two segments on the shape of the curve depends on segment lengths that form a spike and the distance between the non-connected vertices of these segments



(a) Spike (b) Cavity

Fig. 4. Two types of contour topology

3.1 Application of DCE on a Planar Triangulated Surface

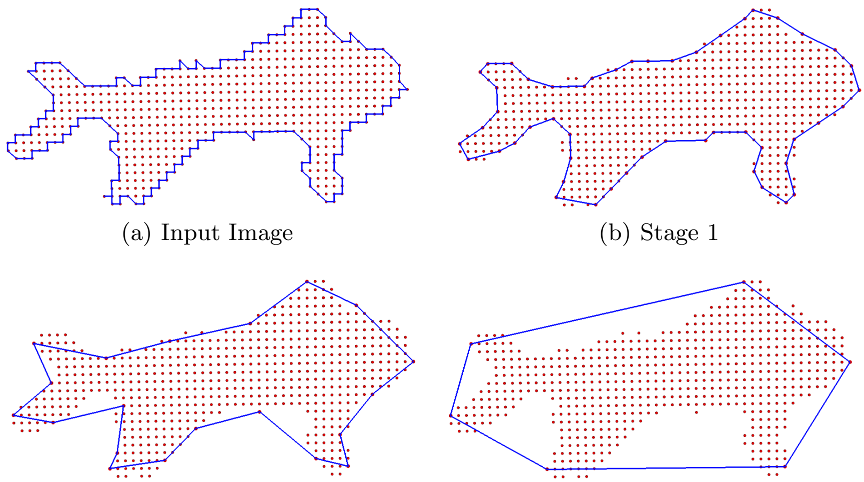
To validate our cost function we first intend to apply it to a contour shown on Fig.2(a). This contour is defined around an object that is drawn on a triangulated surface where all triangles, that define this surface, lie on the same plane. This fact allows us to apply a relevance measure defined in [5] and a measure developed in this paper to the same boundary. Figure5 shows result obtained with the current method. For a good visual comparison we demonstrate several contour evolution stages that contain exactly the same number of boundary vertices.

With the first stage both contours contain 30 vertices. Comparing our results Fig.5(b) with previous algorithm Fig.2(b) we can clearly state that our method generates a contour that is a lot smoother and approximates the input shape in a better way.

With the second stage we generate contours with 20 vertices, Fig.5(c) and Fig.2(c). Both methods create a good approximation of the input shape but we still can say that our approach generates a slightly smoother boundary.

With the third stage we produce contours with 6 vertices, Fig.5(d) and Fig.2(d). Our approach converged to a convex polygon that approximates in a good way the input contour. Previous approach lost an important part of the input boundary and did not achieve a convex shape, Fig.2(d).

We can state that method introduced in this paper outperform previously proposed algorithm on a 2D grid.



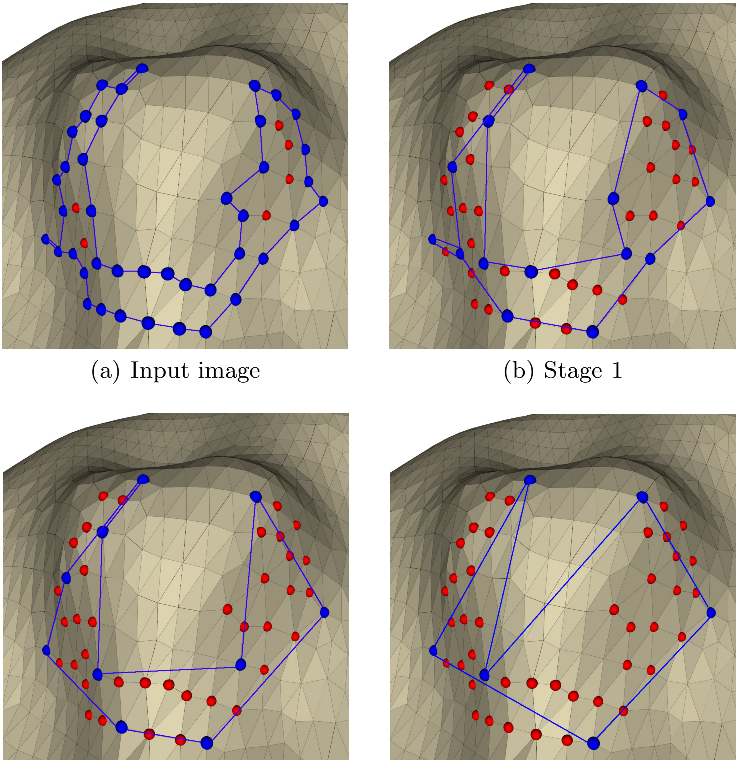
(a) Input Image (b) Stage 1

(c) Stage 2 (d) Stage 3

Fig. 5. Three stages of the proposed Discrete Curve Evolution algorithm

3.2 Application of DCE on an Arbitrary Triangulated Surface

After comparing our results with the previous method, we apply our algorithm on objects defined on arbitrary triangulated surfaces. Figure 6 shows 4 iterations of a contour evolution. As we can see all points that are removed from the contour are less significant compared to points that approximate a boundary at a current iteration.



(a) Input image (b) Stage 1

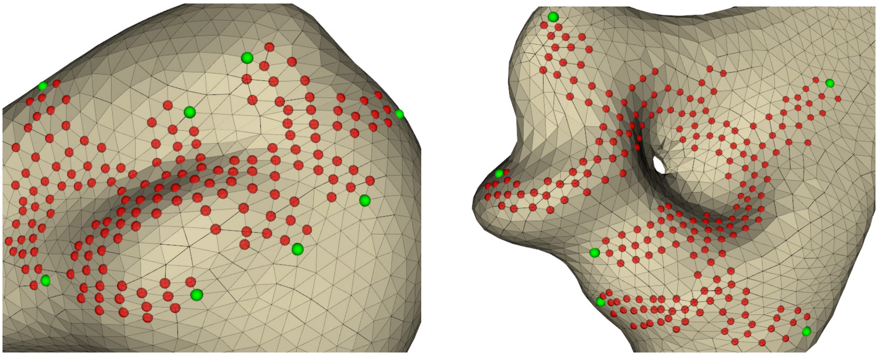
(c) Stage 2 (d) Stage 3

Fig. 6. Three stages of the proposed Discrete Curve Evolution algorithm on an arbitrary triangulated surface

Figures 7, 8 show four binary sets of vertices that lie on different meshes. With green points we highlight landmarks detected with the help of our algorithm. Obtained results demonstrate high robustness and efficiency of the proposed method.

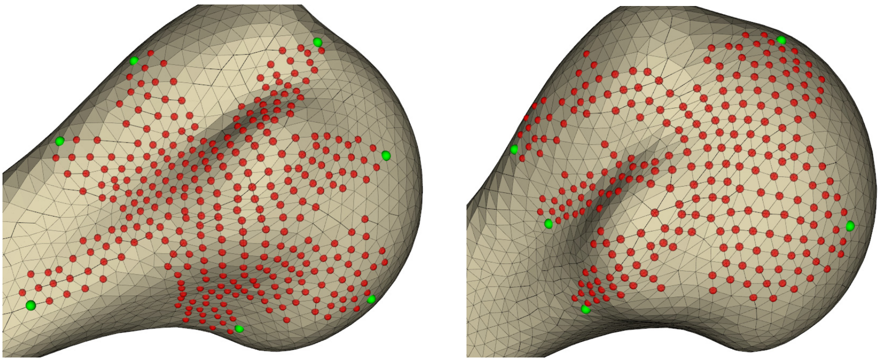
4 Application to Skeleton Pruning

The skeleton is important for object representation and recognition in various areas such as image retrieval, computer graphics, character recognition, image



(a) Example 1 (b) Example 2

Fig. 7. Landmarks points detected with the help of proposed method



(a) Example 3 (b) Example 4

Fig. 8. Landmarks points detected with the help of proposed method

processing and biomedical image analysis [4]. Skeleton-based representations are the abstraction of objects that contains shape features and topological structures. Great work have been done by lot of researchers to recognize the generic shape by matching skeleton structures, where the most significant problem is the skeleton's sensitivity to an object's boundary deformation. Little noise or variation of the boundary often generates redundant skeleton branches that may seriously disturb the topology of the skeleton's graph [3]. Lot of efforts have been made to develop methods for skeleton pruning; a good overview could be found in [3]. All these methods focused on skeleton extraction and pruning for a well defined topological grid. To the best of our knowledge, no method exists for skeleton pruning on a polygonal mesh. This section describes a new algorithm to prune a skeleton on an unstructured triangulated grid.

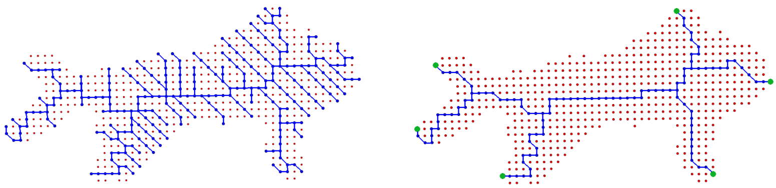
We are inspired by two different publications [3] and [10]. In [3] authors propose to use the DCE algorithm for skeleton pruning. The main idea is to remove all skeleton points whose generating points lie on the same contour segment. Another method is proposed in [10] where authors suggest to find salient points of an object contour with the help of DCE. These points represent the stable endpoints of the skeleton. Authors use these points to propagate the skeleton inside the object by selecting each time a point from the object that has equal distances to the contour parts.

We suggest to use the DCE algorithm to extract salient points of an object's boundary. We then connect these points to the closest points on the computed skeleton. The last step is the actual pruning where we remove all skeleton branches that are not connected to salient points.

Here we present few results of the proposed pruning on triangulated surface. First, we apply a method described in [1] to extract a skeleton from a set of vertices on polygonal mesh and we demonstrate results on Fig.9(a) where we observe many unnecessary branches caused by noisy boundary. Our next step is to find shape landmarks (or salient points) with the help of the proposed DCE algorithm and to connect these points to the closest skeleton point. Then, we remove all branches that do not start with the found landmarks. Figure 9(a) shows the obtained results, where we highlight detected landmarks in green. We can state that the obtained skeleton is 100 percent clean and represents complete topology and geometry of the input shape.

Now, we demonstrate few more results for different shapes on an arbitrary triangulated mesh. Figure 9(a) shows a worm shape with the calculated skeleton and landmarks. Extracted skeleton contains redundant branches that need to be removed. After applying our algorithm we obtain results shown on Fig.9(b).

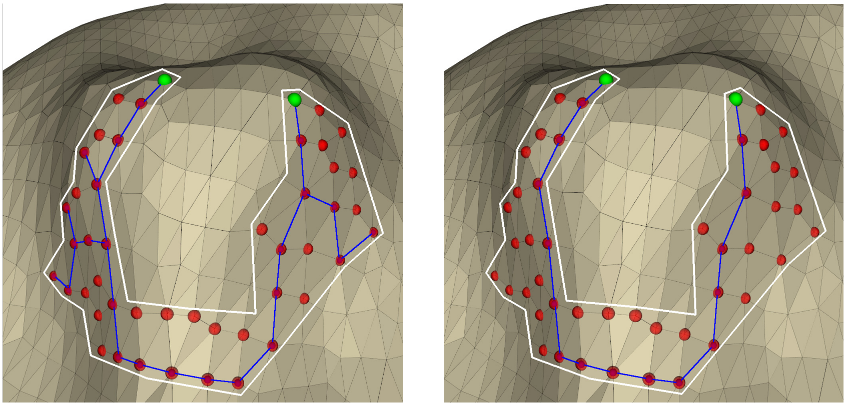
Figure 9 contains a starfish form which is more complicated than a worm shape. We can see that our method correctly identified shape landmarks and the pruned skeleton reflects the topology of an input shape.



(a) Extracted skeleton

(b) Pruned skeletom with proposed method

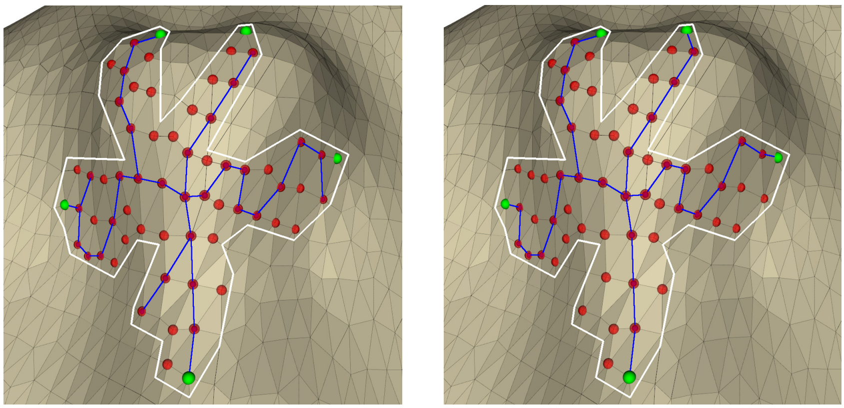
Fig. 9. Skeleton pruning technique for a set of vertices on a triangulated surface proposed in this paper



(a) Extracted skeleton with salient points

(b) Pruned skeletom

Fig. 10. Skeleton pruning technique for a shape of a worm on a 3D surface



(a) Extracted skeleton with salient points

(b) Pruned skeletom

Fig. 11. Skeleton pruning technique for a shape of a starfish on a 3D surface

5 Conclusion

In this paper we developed a new algorithm to detect characteristic landmarks of any shape defined as a set of vertices on an arbitrary triangulated surface or a set of pixels on a regular grid. The proposed method iteratively removes boundary vertices with smallest contribution to the global object shape so that at the end we keep only points that maximally represent a shape contour. We apply the proposed algorithm for several objects represented on triangulated surfaces. The obtained results justify the good reliability of our approach.

We also showed a useful application of the proposed algorithm for skeleton pruning. The demonstrated results clearly show efficiency of this approach. Extracted skeletons represent correctly objects' geometry and topology.

References

1. Kudelski, D., Viseur, S., Mari, J.-L.: Skeleton extraction of vertex sets lying on arbitrary triangulated 3D meshes. In: Gonzalez-Diaz, R., Jimenez, M.-J., Medrano, B. (eds.) DGCI 2013. LNCS, vol. 7749, pp. 203–214. Springer, Heidelberg (2013)
2. Rossl, C., Kobbelt, L., Seidel, H.P.: Extraction of feature lines on triangulated surfaces using morphological operators. In: AAAI Spring Symposium on Smart Graphics, vol. 00-04, pp. 71-75 (March 2000)
3. Bai, X., Latecki, L.J., Liu, W.-Y.: Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 449–462 (2007)
4. Blum, H.: Biological Shape and Visual Science (Part I). J. Theoretical Biology 38, 205–287 (1973)
5. Latecki, L.J., Lakämper, R.: Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution. Computer Vision and Image Understanding (CVIU) 73, 441–454 (1999)
6. Latecki, L.J., Lakämper, R.: Polygon evolution by vertex deletion. In: Nielsen, M., Johansen, P., Fogh Olsen, O., Weickert, J. (eds.) Scale-Space 1999. LNCS, vol. 1682, pp. 398–409. Springer, Heidelberg (1999)
7. Latecki, L.J., Lakamper, R.: Shape similarity measure based on correspondence of visual parts. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) 22(10), 1185–1190 (2000)
8. Kimmel, R., Sethian, J.A.: Computing Geodesic Paths on Manifolds. Proceedings of the National Academy of Sciences of the United States of America 95(15), 8431–8435 (1998)
9. Sethian, J.A.: Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences. Cambridge University Press (1996)
10. Yang, X., Bai, X., Yang, X., Zeng, L.: An Efficient Quick Algorithm for Computing Stable Skeletons. In: Congress on Image and Signal Processing (2009)