Image Skeletonization Based on Curve Skeleton Extraction

Xiaoyan Hu, Bo Sun, Huiqin Zhao, Bin Xie, and Hao Wu

College of Information Science and Technology, Beijing Normal University
{huxy,tosunbo}@bnu.edu.cn

Abstract. Skeletonization is a transformation of an object in a digital image into a simplified representation of the original object. The skeleton of an image object is an abstraction of the original object which largely preserves the extent and connectivity of the original region while throwing away most of the boundary and interior pixels. In this paper, we propose a new method to calculate skeleton from 3D space instead of image space which has only two dimensions. Our method start with a contour of an object in an image, then inflate this two dimensional shape to a three dimensional mesh, and then apply a 3D mesh curve skeleton extraction algorithm to this intermediate three dimension mesh model. Finally, we project the resulting 3D curve skeleton back to image space and get the skeleton of the original shape or object in the image. Our method is noise insensitive. A little perturbation on shape would not change the structure of the resulting skeleton. Our method is relatively fast because it only generates a geometry mesh in contrast to compute a Voronoi graph. Our method preserves the topology as well as the shape.

1 Introduction

Skeletonization is a transformation of an object in a digital image into a simplified representation of the original object. The skeleton of an image object is an abstraction of the original object which largely preserves the extent and connectivity of the original region while throwing away most of the boundary and interior pixels. The notion skeleton was introduced by H. Blum as a result of the Medial Axis Transform (MAT) or Symmetry Axis Transform (SAT) [13]. The MAT determines the closest boundary point(s) for each point in an object. An inner point belongs to the skeleton if it has at least two closest boundary points. Skeletonization makes the object image easier to be read and use in many image processing and analysis applications such as feature analysis and pattern recognition and classification.

There are three major categories of skeletonization techniques to produce a skeleton representation of a certain object or region in an image. 1. Skeleton can be extracted based on distance transforms by detecting ridges in distance map of the boundary points. 2. Calculating the Voronoi diagram generated by the boundary points. 3. Using morphological thinning techniques based on erosion operations. Although these methods can compute skeleton for a certain object in an image, they are very sensitive to noise and small disturbance on boundary.

J.A. Jacko (Ed.): Human-Computer Interaction, Part I, HCII 2011, LNCS 6761, pp. 580–587, 2011.

In this paper, we propose a new method to calculate skeleton from 3D space instead of image space which has only two dimensions. Our method started with a contour of an object in an image, then we inflate this two dimensional shape to a three dimensional mesh, and then apply a 3D mesh curve skeleton extraction algorithm to this intermediate three dimension mesh model. Finally, we project the resulting 3D curve skeleton back to image space and we get the skeleton of the original shape or object in the image. The 3D curve skeleton extraction algorithm we used in this paper is mesh contraction method. This method is proposed by Oscar et al. [5], and can extract curve skeleton efficiently from a 3D mesh. One advantage of "mesh contraction" method is that it is noise insensitive. Another advantage of this curve skeleton extraction algorithm is that it is very fast. When applying this algorithm on a mesh, it would take only several iterations (about ten iterations) to convergence to a 3D curve skeleton which make the skeletonization interactive process. By using mesh contraction algorithm, our skeletonization method inherits these advantages. 1. Our method is noise insensitive. A little perturbation on shape would not change the structure of the resulting skeleton. 2. Our method is relatively fast because it only generates a geometry mesh in contrast to compute a Voronoi graph. 3. Our method preserves the topology as well as the shape.

2 Related Work

In this section, we discuss relate work in two fields which are relative to our method. One is traditional image skeletonizing methods and the other is curve skeleton extraction of objects in 3D space.

In image processing and computer vision, the skeleton of a region is proposed by Blum [13] and is defined via medial axis transformation (MAT) which is also known as grass fire transformation. There are three major categories of skeletonization techniques to produce a skeleton representation of a certain object or region in an image. 1. Skeleton can be extracted based on distance transforms by detecting ridges in distance map of the boundary points. [9, 22, 10, 12, 13] compute skeleton using wave front propagation or distance transform。 2. Another way is calculating the Voronoi diagram generated by the boundary points [16, 23]. 3. Using morphological thinning techniques based on erosion operations. [3, 24, 4].

Although these methods can compute skeleton for a certain object in an image, they are very sensitive to noise and small disturbance on boundary and the resulting skeleton can drastically different in its structure. This becomes a serious problem in digital image process. Several methods have been proposed to pruning “false” branches that are believed to be caused by noise in the outline. [16, 22, 10] Liu and Geiger et al. [26, 11] proposed a method based on self-similarity of a smooth outline curve to stabilize the skeleton extraction. Tang et al. [25] extracts skeleton from ribbon like shapes based on a new wavelet function.

In digital geometry processing, the extraction of curve skeletons from 3D models is a fundamental problem and is still a research challenging to find a simple and robust method that compute curve skeletons for 3D geometry mesh [7, 5]. Methods to extract curve skeleton for 3D objects can be roughly classified into two main categories, based on the underlying representation of 3D objects: volumetric methods and geometric methods.

Many exist methods work on volumetric discrete representation such as voxelized models [17, 21, 27] or distance field function [14, 28, 6, 18].

Geometric method work directly on polygon meshes or point sets. There two main categories methods, one obtain an approximate medial surface Voronoi diagram and prune it to get a curve-skeleton.[1, 8, 19], The other make use of Reeb graph based techniques via various real value functions defined on model surface [20, 15, 2]. Recently, Oscar et al. developed a simple method based on Laplacian smoothing to robustly extract curve skeleton [5]. Our method is mainly based on their work. We refer the reader to the comprehensive survey of [7] for more detailed discussion on curve skeleton extraction for 3D models.

3 Overview

The first step of our method is to transform 2D shape into a 3D mesh. In this stage, we apply snake algorithm to generate contour of object in source image. After we got contour, a two dimensional mesh will be generated within this contour. One simple way to generate this 2D mesh is to generate a quad grid per pixel within the contour and tessellate each quad grid into two triangles. This step results in a triangle mesh in image space. The next step is inflating it in positive and negative direction along the normal of image plane. Here we use an extruding function to control the process of extruding the 2D mesh and to control the displacement of each grid point. 2D Gaussian is a good choice. We tweak the width, height of a standard 2D Gaussian to match the scale of the mesh which corresponded to the original shape to be skeletonized in the image.

In skeleton extraction stage we adopt mesh contraction algorithm developed by Oscar et al. This method works on three dimensional meshes and can extract one dimensional curve skeleton within several contraction iterations. The process of mesh contraction is handled by a discrete Laplace equation. By solving this discrete Laplace equation we get a contracted mesh which is visually thinner than the original mesh. By repeating this process on to the contracted mesh several iterations we would get an extremely contracted 3D mesh witch look like a one dimensional curve skeleton. In Oscar's method a connectivity surgery is applied to the extremely contracted 3D mesh to get a real 1D curve skeleton. The extremely contracted 3D mesh contain a large number of degenerate vertices, so we omit the connectivity surgery step and parallel project the resulting 3D mesh onto image plane directly. Overlapped vertices would be projected to same pixel, and the projected pixels in image space constitute the skeleton representation of the original object or shape.

4 3D Mesh Generation

We use snake algorithm to calculate the contour of an image object. This contour represent the shape of this image object, we considered the problem of skeletonizing a 2D shape as a projection of skeleton extraction of an object in 3D space. We need to reconstruct the 3D geometry of the 2D image object. The reconstruction would be an under-constrained problem because there are many (in fact infinite) 3D shapes satisfy the only constrain of exactly projecting onto the image object. Our 3D mesh generation method does not try to find a best 3D shape by introducing some prior condition or additional constrains. Instead, we just inflate the 2D shape using some weight function. The method is very simple: we first tessellate the 2D shape into a 2D mesh, and then move these mesh vertices off along the normal direction of the image plane.

Our tessellation of an image object is applied by scanning the image object to find out all pixels that are lying on the contour or inside this object. These pixels form uniform rectangle grids if we consider the pixels as vertex and draw edges between one pixel and its neighbor pixels in vertical and horizon direction. This grid can be seemed as quadrilateral mesh and we tessellate it further by adding edges between one pixel and its 45o diagonal neighbor pixel. Then we get a triangle mesh of the image object.

To generate a 3D triangle mesh which is ready to extraction of curve skeleton from it we need a closed surface mesh instead of the mesh we get so far. Our strategy is to clone the mesh and let these two meshes share vertices that are lying on the contour. By doing so, we finally get a flattened closed 3D triangle mesh. The only thing left to do is to inflate it to a real 3D mesh. We move vertices of the flattened 3D mesh in positive and negative direction along the image normal. A weighting function is used to control how far each vertex to move. In this paper, we select a simple strategy that using Gaussian as our weighting function.

5 Curve Skeleton Extraction

Extraction of curve-skeleton is a fundamental problem in computer graphics especially in computer animation. Mesh contraction based curve skeleton extraction contract a 3D mesh to a 1D curve skeleton by applying a implicit Laplacian smoothing with global positional constrains. In general, Laplacian smoothing without constrain would smooth out all detail (as well as features which is important to skeleton extraction) and the resulting 3D mesh would converge into a single point. To preserve skeleton information of the original object, global positional constrains are need to keep features where they are.

Laplacian smoothing is a operation that move every vertices on a mesh by solving discrete Laplacian equation: $\mathbf{LV}'=0$, where $\mathbf{L}$ is a n×n curvature-flow Laplace operator and can be written as follow:

$$\mathbf{L}_{ij}=\begin{cases}\omega_{ij}=cot\alpha_{ij}+cot\beta_{ij} & \text{if } (i,j)\in \mathrm{E}\\ \sum_{(i,k)\in \mathrm{E}}^{k}-\omega_{ik} & \text{if } i=j\\ 0 & \text{otherwise}\end{cases} \tag{1}$$

Where α_{ij} and β_{ij} are the opposite angles with respect to the edge (i,j) [5]. According to [5], $\delta=\mathbf{LV}=[\delta_1^T,\delta_2^T,...,\delta_n^T]^T$ approximate the (inward) curvature-flow normals, and solving equation (1) means removing the normal components and so that contracting the mesh geometry. As descript as above, applying Laplacian smoothing without constrain would make the 3D shape to converge to a single point. So extra global positional constrains are add in [5]. The resulting linear system to be solved is written as below:

$$\begin{bmatrix} \mathbf{W}_L \mathbf{L} \\ \mathbf{W}_H \end{bmatrix} \mathbf{V}' = \begin{bmatrix} 0 \\ \mathbf{W}_H \mathbf{V} \end{bmatrix} \tag{2}$$

Where W_L and W_H are diagonal weighting matrices with respect to Laplacian smoothing term and global positional constrains. The upper half of equation (2) means contract the 3D mesh using Laplacian smoothing and the lower half mean retain the original shape. Equation (2) is over-determined and can be solve in least-squares sense to minimizing the quadratic energy:

$$\left\| \mathbf{W}_L \mathbf{L} \mathbf{V}' \right\|^2 + \sum_i \mathbf{W}_{H,i}^2 \left\| v_i' - v_i \right\|^2 \tag{3}$$

Solving equation (2) or minimizing (3) we get a shrinked mesh while retain the main shape features compared to the original mesh. By solving equation (2) for the result mesh we get a more shrinked mesh and after several iterations the original mesh can be eventually contracted into a 1D skeleton. We refer readers to the paper [5] for more detail on both math and implementation information.

The resulting 1D skeleton contains a large number of collapsed faces and degenerated faces, so Oscar et al. proposed two post processing step named connectivity surgery and embedding refinement to get a true skeleton structure of the original 3D shape. In this paper, however, we only need a 1D skeleton of the generated 3D shape nevertheless whether it is a really 1D structure or a degenerated geometry shape. Thus we omit the followed steps and stop our skeleton extraction task as we find a visually 1D curve skeleton.

6 Projection

Once we get a visually 1D curve skeleton (although this one is a degenerated 3D shape in fact), we perform a parallel projection from 3D space back to image space along normal of image plane. Because we didn't perform connectivity surgery and embedding refinement to the contracted mesh, so the projection operations are performed to all vertices of the contracted mesh (in which case the number of vertices is the same as original mesh).

Location and orientation of image plane is determined by O_i, $\mathbf{n}$, $\mathbf{x}$ and $\mathbf{y}$. As illustrated in Fig 1. , O_i is the origin of image plane, i.e. the left bottom corner of the input image. $\mathbf{n}$ is unit normal vector of image plane. $\mathbf{x}$ and $\mathbf{y}$ are unit vectors along image horizontal and vertical directions. Let P be a vertex on the 1D skeleton, and $p = (P - O_i)$ be the vector from O_i to P, and p' be the projection point of P on image.

Then the projection p' of vertex P can be achieved using equation (4):

$$\mathbf{p}' = \mathbf{p} - \alpha\mathbf{n} \quad \text{where} \quad \alpha = \mathbf{p} \cdot \mathbf{n}; \tag{4}$$

As we get projection point p' of P, the pixel position can be calculated as the dot product between projected point p' and axis of image plane.

$$x = \mathbf{p}' \cdot \mathbf{x}; \qquad y = \mathbf{p}' \cdot \mathbf{y} \tag{5}$$

By labeling all hit pixels we finally get the skeleton of original image object.

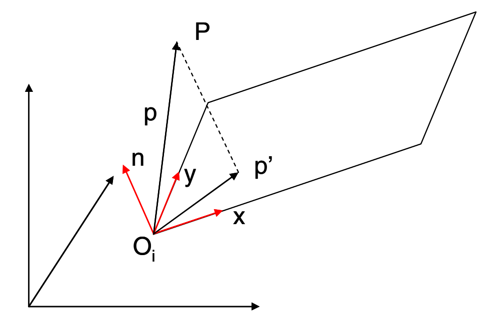

Fig. 1. Project vertex back to image plane: P is a vertex of 1D skeleton of the contracted mesh, it is projected onto image plane and then its image coordinate is calculated with respect to image left-bottom corner O_i

7 Result

Figure 2 shows the original image with a rectangle object in it, the corresponding 3D mesh generated using our method and the resulting skeleton. Figure 3 shows the result of applying our method onto the same rectangle with two little protuberances. We can see that the resulting skeletons are generated correctly even when the boundary is disturbed. Figure 4 and Figure 5 are basically the same as figure 2 and 3, but the objects are thinner.

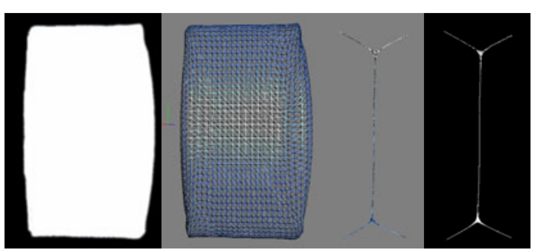

Fig. 2. Skeletonization of a rectangle

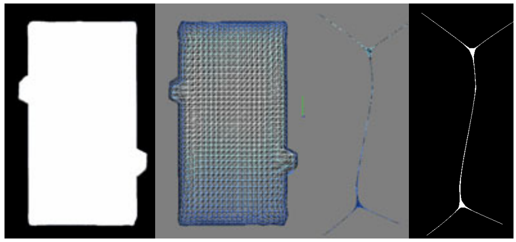

Fig. 3. Skeletonization of a disturbed rectangle

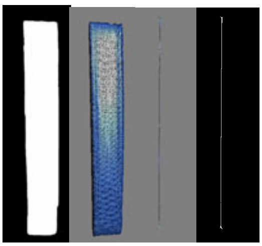

Fig. 4. Skeletonization of thin rectangle

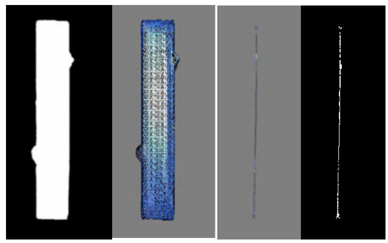

Fig. 5. Skeletonization of disturbed thin rectangle

References

1. Amenta, N., Choi, S., Kolluri, R.K.: The power crust. In: Symposium on Solid Modeling and Applications, pp. 249–266 (2001)
2. Aujay, G., H'Etroy, F., Lazarus, F., Depraz, C.: Harmonic skeleton for realistic character animation. In: Symposium on Computer Animation, pp. 151–160 (2007)
3. Arcelli, C., Sanniti di Baja, G.: A Width Independent Fast Thinning Algorithm. IEEE Trans. PAMI 7, 463–474 (1985)
4. Chen, Y.-S., Yu, Y.-T.: Thinning Approach for Noisy Digital Patterns. Pattern Recogintion 33(5), 809–819 (1996)
5. Au, O.K.-C., Tai, C.-L., Chu, H.-K., Cohen-Or, D., Lee, T.-Y.: Skeleton Extraction by Mesh Contraction. ACM Trans. Graph. 27(3) (2008)
6. Cornea, N.D., Silver, D., Yuan, X., Balasubramanian, R.: Computing hierarchical curve-skeletons of 3D objects. The Visual Computer 21(11), 945–955 (2005)
7. Cornea, N.D., Min, P., Silver, D.: Curve-skeleton properties, applications, and algorithms. IEEE Transactions on Visualization and Computer Graphics 13(3), 530–548 (2007)
8. Dey, T.K., Sun, J.: Defining and computing curve-skeletons with medial geodesic function. In: Symposium on Geometry Processing, pp. 143–152 (2006)
9. Bookstein, F.L.: The Line Skeleton. CGIP: Comp. Graphics and Image Proc. 11, 123–137 (1979)
10. Leymarie, F., Levine, M.D.: Simulating the Grass fire Transform Using an Active Contour Model. IEEE Trans. PAMI 14(1), 56–75 (1992)
11. Geiger, D., Liu, T.-L., Kohn, R.V.: Reprensentation and self-similarity of shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(1), 86–99 (2003)

12. Giblin, P.J., Kimia, B.B.: On the intrinsic reconstruction of shape from its symmetries. IEEE Trans. PAMI 25(7), 895–911 (2003)
13. Blum, H.: A Transformation for Extracting New Descriptors of Shape. In: Models of the Perception of Speech and Isual Form. MIT Press, Cambridge (1967)
14. Hassouna, M.S., Farag, A.A.: Robust centerline extraction framework using level sets. In: Proceedings on Computer Vision and Pattern Recognition, vol. 1, pp. 458–465 (2005)
15. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L.: Topology matching for fully automatic similarity estimation of 3D shapes. ACM Trans. Graph., 203–212 (2001)
16. Brandt, J.W., Algazi, V.R.: Continuous Skeleton Computation by Voronoi Diagram. CVGIP: Image Understanding 55, 329–338 (1994)
17. Ma, C.-M., Wan, S.-Y., Lee, J.-D.: Three dimensional topology preserving reduction on the 4-subfields. IEEE Transactions on Pattern Analysis and Machine Intelligence 4(12), 1594–1605 (2002)
18. Ma, W.-C., Wu, F.-C., Ouhyoung, M.: Skeleton extraction of 3D objects with radial basis functions. In: Proceedings of the Shape Modeling International, pp. 207–215 (2003)
19. Ogniewicz, R., Ilg, M.: Voronoi skeletons: Theory and applications. In: Proceedings on Computer Vision and Pattern Recognition, pp. 63–69 (1992)
20. Pascucci, V., Scorzelli, G., Bremer, P.-T., Mascarenhas, A.: Robust on-line computation of Reeb graphs: simplicity and speed. ACM Trans. Graph. 26(3), article 58 (2007)
21. Palágyi, K., Kuba, A.: A parallel 3D 12-subiteration thinning algorithm. Graph. Models Image Process. 61(4), 199–221 (1999)
22. Kimmel, R., et al.: Skeletonization via Distance Maps and Level Sets. CVIU: Comp. Vision and Image Understanding 62(3), 382–391 (1995)
23. Ogniewicz, R.L.: 1993. Discrete Voronoi Skeletons. Hartung-Gore (1993)
24. Shi, F.Y., Wong, W.-T.: Fully Parallel Thinning with Tolerance to Boundary Noise. Pattern Recogintion 27(12), 1677–1695 (1994)
25. Tang, Y.Y., You, X.: Skeletonization of ribbon like shapes based on a new wavelet function. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(9), 1118–1133 (2003)
26. Liu, T.L., Geiger, D., Kohn, R.V.: 1999. Representation and Self Similarity of Shapes. In: Proc. ICCV 1999, pp. 1129–1135 (1999)
27. Wang, Y.-S., Lee, T.-Y.: Curve skeleton extraction using iterative least squares optimization. IEEE Transactions on Visualization and Computer Graphics (2008)
28. Zhou, Y., Toga, A.W.: Efficient skeletonization of volumetric objects. IEEE Transactions on Visualization and Computer Graphics 5(3), 196–209 (1999)