

Effective Piecewise Linear Skeletonization of Sparse Shapes

Wenyu Qu¹, Zhiyang Li^{2,*}, Junfeng Wu², Yanan Wu³ and Zhaobin Liu²

¹School of Computer Software, Tianjin University, China

²School of Information Science and Technology, Dalian Maritime University, China

³87 Unit, 91550 of PLA, Dalian, China

Conventional image skeletonization techniques implicitly assume the pixel level connectivity. However, noise inside the object regions destroys the connectivity and exhibits sparseness in the image. We present a skeletonization algorithm designed for these kinds of sparse shapes. The skeletons are produced quickly by using three operations. First, initial skeleton nodes are selected by farthest point sampling with circles containing the maximum effective information. A skeleton graph of these nodes is imposed via inheriting the neighborhood of their associated pixels, followed by an edge collapse operation. Then a skeleton fitting process based on feature-preserving Laplacian smoothing is applied. Finally, a refinement step is proposed to further improve the quality of the skeleton and deal with noise or different local shape scales. Numerous experiments demonstrate that our algorithm can effectively handle several disconnected shapes in an image simultaneously, and generate more faithful skeletons for shapes with intersections or different local scales than classic methods.

Keywords: Skeletonization, Sparse shapes, Farthest point sampling, Multi-scale

1. INTRODUCTION

Shape analysis and recognition constitutes a fundamental problem of the areas such as pattern recognition, image processing and computer vision [1-6]. A central issue in shape analysis and recognition is how to represent the object. Skeleton capturing both the topology and prominent geometry of an object in a concise manner is widely used, because it can ease the characteristics detection in recognition and classification [7-9].

Many proposed skeletonization techniques utilize the connectivity of image pixels inside the object regions [7, 8]. However, they may fail to extract a correct skeleton or yield many artifacts if the regions lack pixel level connectivity, known as sparse shapes [10]. The sparse shapes may be generated by noise occurring inside object regions, poor lighting conditions, incorrect thresholding, texts faded due to aging or poor ink quality, etc. Figure 1 illustrates a non-sparse image and its sparse counterpart. The right one is an image with a sparse shape “A” caused by an artificially introduced internal noise. Median image filtering or morphological set operations can be used to reconstruct

the lost connectivity of the pixels. However, if the sparseness is significant, these operations might introduce new problems, such as distortions to the topology feature of the object or error links between different objects [10].

To obtain skeletons from images which may lack pixel level connectivity, principal curves [11], first introduced in statistics, present a solution. Most skeletonization algorithms in this area

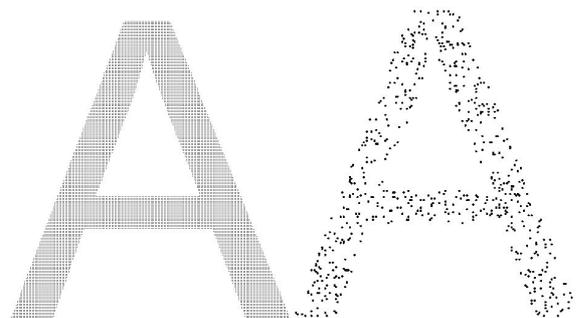


Figure 1 Comparison of a non-sparse image and its sparse counterpart.

*Correspondence Author. E-mail: lizy0205@gmail.com

[10, 12-14] exploit a two-layer iteration process including locating a set of vertices based on the principle of principal curves and connecting the vertices by the minimum spanning tree (MST) of them to form an approximate skeleton. Because the minimum spanning tree is unable to represent closed topological property, it is dynamically modified by detecting closed regions. Meanwhile, the number of the vertices needs to be adaptively changed to obtain a good approximation of the skeletal representation for a given shape distribution. Thus, although these algorithms can extract skeletons from shapes with self-intersections or loops, they suffer the following three main drawbacks. First, this skeletonization strategy is not easy to control. Most of the skeletonization algorithms have parameters that interact with some complexity during the process of inserting or deleting vertices and the modification of the MST. Second, the topology captured by a modified MST will generate unnecessary links between separate objects (Figure 10-b), since MST connects all the vertices together according to its definition. Thus, for identifying a handwritten word, the word has to be segmented into letters in advance. Finally, most of the algorithms lack a clear and intuitive formulation to fit and smooth a skeleton. Besides, these algorithms are not good at feature-preserving.

In order to overcome the above shortcomings as many as possible, an effective algorithm is presented in this paper. Three techniques, farthest point sampling, edge collapse and feature-preserving Laplacian smoothing, are combined with a refinement mechanism to obtain the skeletons of sparse shapes. The algorithm captures the topology of original models by farthest sampling with adaptive radii (the radii of sampling circles) and the followed edge contraction processes, and later, increases the approximation of the skeletal representation by smoothing the locations of the nodes via Laplacian. Testing on a large amount of typical data including multi-scale models (composed of shape parts with different scales), self-intersecting models and images of faded texts, our algorithm can produce skeletons with high quality.

The principal contributions are summarized as follows:

- The proposed skeletonization strategy consisted of node sampling and edge contraction performs better compared with the MST-based methods. It easily captures closed topological property and avoids error links between different objects.
- Since the refinement process adaptively samples nodes reflecting the local scale based on prior radii, our algorithm can handle multi-scale models.
- Based on the Laplacian method which can be especially designed for the feature points, the algorithm can preserve the features well.

This paper begins with a brief overview of related work on skeletonization for sparse shapes. Section 3 presents our skeletonization algorithm. Section 4 goes through the details of the three steps of the proposed algorithm. Section 5 contains implementation and some experimental results. Finally, Section 6 summarizes the paper.

2. RELATED WORK

The conventional skeletonization algorithms [8, 15-18] mainly focus on images without sparseness in objects regions, which require formal criteria for the boundary of the objects or for separating the foreground from the background. However, towards images with sparse shapes, a pixel with zero valued pixels in its neighborhood may not lie on the boundary of a region. Thus, it is difficult to distinguish the foreground from the background in such shapes, and neither 4-connectivity nor 8-connectivity neighbors exist.

For the sake of brevity, we focus on the skeletonization algorithms that do not depend on the connectivity of pixels. Inspired by the hint of human visual symmetry, a modified thinning algorithm for noisy digital patterns was proposed by Chen and Yu [19]. For each pixel, they found the effective circular range, computed the symmetry score of the pixel distribution and converted the symmetry information into a gray-scale image which was thinned to obtain the skeleton. Datta *et al.* [20] proposed to use a dynamic self-organizing map (SOM) for the extraction of skeletal shapes from a 2D dot pattern, which means lacking pixel level connectivity compared to common image. The SOM was initialized with a linear topology and later changed according to the local topology of the input pattern.

Computing the skeletal description of shapes is similar to approximating the principal curve of the shape distribution [11, 21-23]. Principal curves are defined as self-consistent smooth curves which pass through the middle of the data points. Singh *et al.* [10] did an outstanding work in this area. A batch formulation of the self-organizing mapping was used to obtain the skeleton or called the principal curve of sparse shapes. They assigned a given number of SOM units randomly, and connected them via a modified MST to form an initial skeleton graph. Then the locations and connections of these units were optimized by evolving the SOM algorithm iteratively. The method is applied to various kinds of images with sparse shapes, such as binary images of planar industrial parts [24], handwritten characters, and faded text [25]. The SOM algorithm can extract skeletons from shapes with self-intersections. However it would produce poor results if the object contains different local shape scales and generate unnecessary links between separate objects.

To extract a skeleton from shapes with different local scales, Palenichka and Zaremba [12] proposed a structured SOM algorithm, which iteratively updated the SOM weights while progressively decreasing the span of a SOM kernel function according to the local shape scales. Yet their algorithm was complicated and described vaguely. To give a simple fitting objective function, Kégl and Krzyzak [26] proposed an explicit one. The skeleton graph was optimized by minimizing an intuitive objective function that captures the two competing criteria of smoothing the skeleton and fitting it to the shape, while the objective function was non-linear. Motivated by solving the problem of highly curved and self-intersecting curves, Liu and Jia [13] turned to principal curve and presented a bottom-up strategy to construct a principal graph. The principal graph was spanned by MST and modified by detecting closed regions and intersectional regions to generate loops and crossings. The method was applied in image skeletonization and tested on images of handwritten characters and objects.

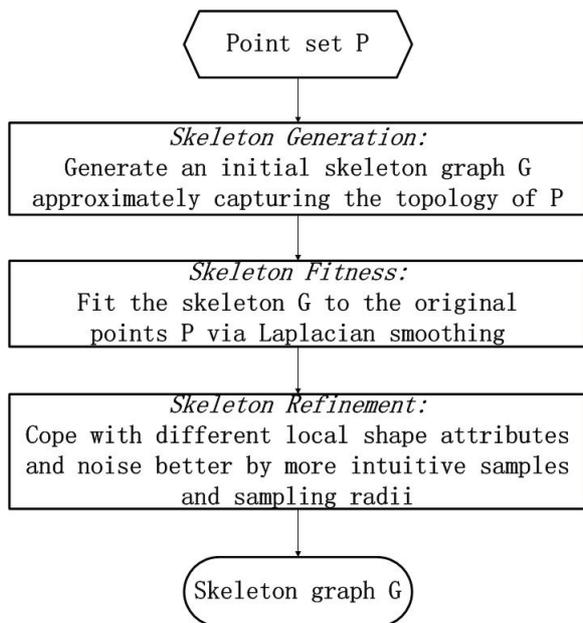


Figure 2 The flowchart of the proposed algorithm.

3. THE PROPOSED METHOD

The objective of our skeletonization algorithm is to fit a set of smooth, piecewise-linear curves to the skeleton of the image with sparse shapes. The curves here are represented by a Euclidean graph in the plane spanned by the vertices found during the skeletonization process. A Euclidean graph $G_{V,E}$ in the d -dimensional Euclidean space is a pair (V, E) . $V = \{\mathbf{v}_1, \dots, \mathbf{v}_m\}$ is a set of vertices (sometimes called nodes for avoiding confusing), and $E = \{(\mathbf{v}_{i_1}, \mathbf{v}_{j_1}), \dots, (\mathbf{v}_{i_k}, \mathbf{v}_{j_k})\} = \{\mathbf{e}_{i_1 j_1}, \dots, \mathbf{e}_{i_k j_k}\}$, $1 \leq i_1, j_1, \dots, i_k, j_k \leq m$ is a set of edges, where each edge \mathbf{e}_{ij} joins two vertices \mathbf{v}_i and \mathbf{v}_j . For the sake of simplicity, the term *graph* is used as an abbreviation for Euclidean graph in the rest of the paper.

Since the 2D binary image with sparse shapes lacks pixel level connectivity, the pixel coordinates are interpreted as points in R^2 , i.e., a point set P . Our skeletonization process begins with P , and produces a 1D graph G as the piecewise-linear approximation for the shape skeleton of P . The whole algorithm consists of three steps: skeleton generation, skeleton fitness and skeleton refinement. Figure 2 shows the flowchart of the proposed algorithm.

The skeleton generation step consists of node sampling and edge contraction. Specifically speaking, the step firstly selects a subset nodes of P based on farthest point sampling with circles of adaptive radii, and builds the graph G spanned by these nodes via inheriting the neighborhood of their associated points, then iteratively collapses unnecessary edges of G by their midpoints until no triangles exist to obtain a thin version of the shape. The sampling radius r of a node is determined by finding the effective circular range which contains the maximum effective information. The detailed description of the generation step is given in Section 4.1.

The graph G obtained from the generation step captures the approximate topology of the shape, and roughly follows the real skeleton. However, it is not smooth and might contain some

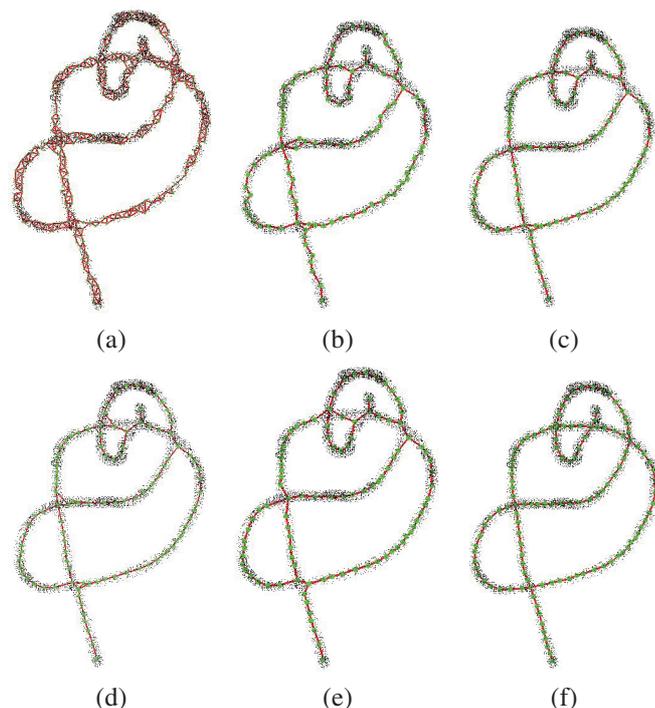


Figure 3 The overview of our skeletonization algorithm. (a) The skeleton graph after farthest sampling in generation step. (b) The skeleton graph after edge contraction in generation step. (c) The skeleton graph after fitness step. (d) The skeleton graph after farthest sampling in refinement step. (e) The skeleton graph after edge contraction in refinement step. (f) The final skeleton graph.

imperfections, such as spurious branches, short loops and two junction vertices in intersectional regions connected by a short path. We propose to fit the skeleton via minimizing Laplacian-based quadratic energy, for its simplicity and easy to implement. We discuss the fitness process in Section 4.2.

Kégl and Krzyzak [26] proposed a collection of restructuring operations to deal with the imperfections mentioned above and improve the structural quality of the skeleton. However, the restructured result is sensitive to parameters when the point set P contains noise or different local shape scales. In order to alleviate these influence, we exploit a skeleton refinement mechanism (See Section 4.3) and redo generation step with the more intuitive samples and sampling radii. Then, we remove all mentioned imperfections and fit the skeleton to further improve the quality of the final skeleton.

Figure 3 shows an overview of our skeletonization algorithm.

4. METHODOLOGY

4.1 Skeleton Generation

The task of this step is to generate an initial skeleton graph $G = (V, E)$, approximately capturing the topology of P . In order to locate the sampling node set V more uniformly in P , Cao *et al.* [27] utilized the farthest point sampling technique by a fixed-radius circle. Inspired by [19], we sample P using the farthest point sampling with optimal circles containing the maximal effective information. In the rest of this subsection, we first

discuss how the sampling radius r is defined at a sampling node, and then give the detailed descriptions of the node sampling and edge contraction processes.

The sampling radius. The sampling radius r is an important parameter of the node sampling process. But given a point $p(x_i, y_i)$, it is difficult to determine its radius r_i for the characteristic of sparse shapes mentioned above. We apply the principle of the maximum entropy to explore the "point information", which is contained in a circular neighborhood of the point $p(x_i, y_i)$. That is, the sampling radius r_i is selected when the points in the current sampling circle contains the maximal entropy.

In information theory, entropy is a measure of the uncertainty associated with a random variable. For a discrete random variable X with possible values x_1, \dots, x_n and the probabilities $P(x_i) = P_i$, its entropy is defined as follows:

$$H(X) = E_X[I(x)] = \sum_{i=1}^n P_i \log \frac{1}{P_i}. \quad (1)$$

According to the definition of entropy, given a point $p(x, y)$ and a radius r , the point information $I_r(x, y)$ within the circular range with the radius r can be computed by the following Equation 2.

$$I_r(x, y) = \sum_i P_i (\log \frac{1}{P_i}) \delta[i], \quad (2)$$

$$\delta[i] = \begin{cases} 1 & \text{a point here} \\ 0 & \text{otherwise} \end{cases}.$$

In order to compute $I_r(x, y)$, we follow the two assumptions proposed by Chen and Yu [19]. First, the probabilities are equiprobable for points appearing in the same ring of the circular range, and are inversely proportional to the square of distance for points appearing in different ring areas of the circular range. That is

$$P_{r,i} = \frac{1}{i^2} P_{r,1}, 1 \leq i \leq r, \quad (3)$$

and

$$\sum_{i=1}^r P_{r,i} n_i = 1, \quad (4)$$

where $P_{r,i}$ is the probability for the points appearing in the i th ring, n_i is the number of points in the i th ring. Second, the Gaussian function can be used to simulate the contribution of the point to the information decreasing from the center to the boundary of the circular range. Thus, $I_r(x, y)$ defined in Equation 2 may be finally modified as:

$$I_r(x, y) = \sum_{i=1}^r \frac{3}{2i} n_i P_{r,i} (\log \frac{1}{P_{r,i}}), \quad (5)$$

where $\frac{3}{2i}$ approximates to the Gaussian function. The optimal sampling radius $r_{x,y}$ for a point $p(x, y)$ is then determined by finding the maximal information:

$$r_{x,y} = \arg \max_r I_r(x, y). \quad (6)$$

For implementation, given a point $p(x, y)$, $r_{x,y}$ is initialized as 1 and iteratively incremented by 1, until reaches a maximum

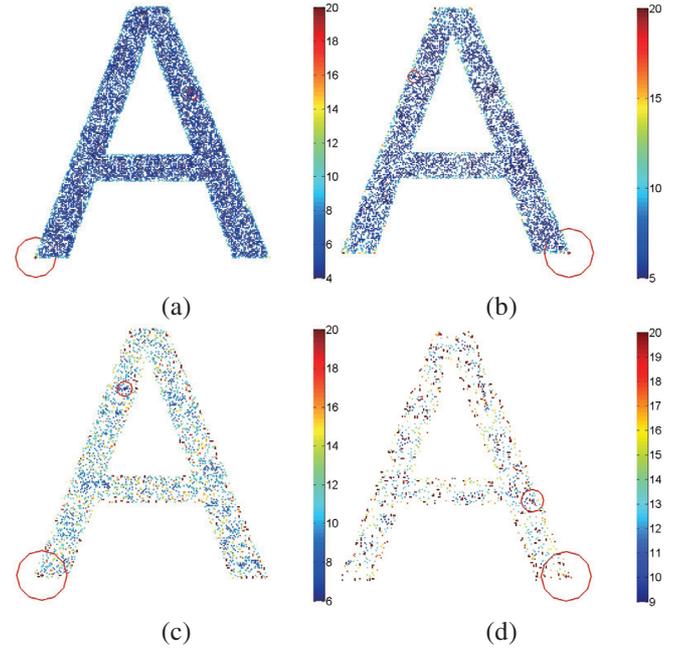


Figure 4 The optimal circle with radius r containing maximum effective information captures the intrinsic property of the underlying 2D shapes with different level of sparseness. (a) 40% noise. (b) 60% noise. (c) 80% noise. (d) 90% noise. The index of the colorbar is the corresponding radius represented by the color. Circles with maximum radius and minimum radius are drawn in each images.

size T . T is fixed as 20 for all the experiments in this paper. Then, $r_{x,y}$ is selected as the one which has the maximal value of $I_r(x, y)$.

In Figure 4, we show the optimal radii of the points of "A" under different internal noise levels. The radius roughly increases its size from the center to the boundary of the shape, and captures the intrinsic property of the underlying 2D shapes. Meanwhile, the sizes of the radii are suitable for the following node sampling process.

Node sampling. The sampling process initializes randomly, proceeds with the covered point which has the farthest distance between it and the current sample nodes, and makes a decision on whether to select the point as a member of sample nodes with a sampling circle of the mentioned radius r or not. The sampling process ends when all the points in P have been covered by the sample circles. The flowchart is illustrated in Figure 5, where V represents sample node set, S represents the points included in the current sampling circle, C represents the points having been covered by the sampling circles, D consists of the distances between the points in P to these nearest nodes in V , and is initialized with zero.

Edge contraction. After the sampling process, each sample node $v_i \in V$ represents a set of associated points $P_i \in P$, such that $P = \bigcup_i P_i$, $P_i \cap P_j = \phi$. $\bigcup_i P_i$ is a partition of P , similar to a Voronoi diagram. The edge set E is constructed by connecting v_i and v_j if their associated points share common K -nearest neighbors (KNN), which imitates the construction of a Delaunay triangulation from the Voronoi diagram (Figure 3-a). A small value K is chosen to avoid error links between two separated objects.

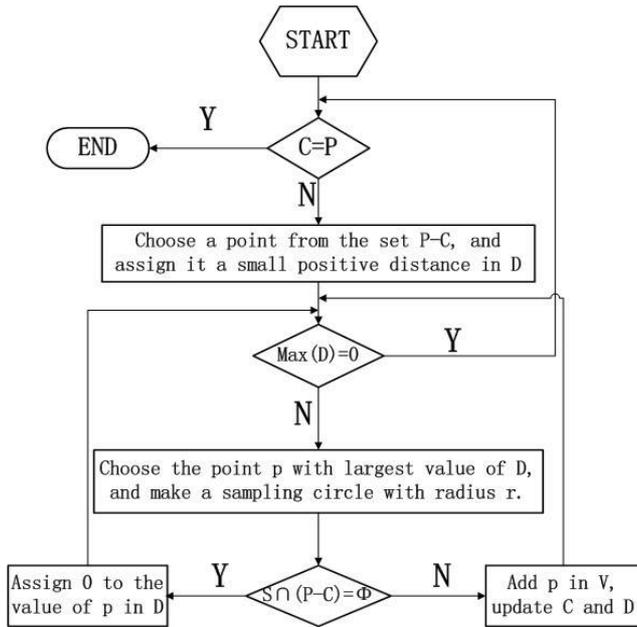


Figure 5 The flowchart of the farthest point sampling process.

Finally, we iteratively collapse unnecessary edges by their midpoints until no triangles exist to simplify the skeleton graph G (Figure 3-b). The points associated with the two endpoints of the edge are assigned to the newly created point during the process. In order to make the final curve skeleton nodes uniformly distributed and preserve the shape better, the edge with minimum length is firstly collapsed.

4.2 Skeleton Fitness

To better fit the skeleton to the original points, we optimize the skeleton graph G by minimizing the following quadratic energy. See Figure 3-c as an example.

$$E(V) = LV^2 + \sum_i w_i^2 v_i - v'_i \|^2. \quad (7)$$

where V is the vertices of the graph G , L is the Laplace matrix of the graph G , providing the smoothing constrains. Equation 7 defines a Least-squares mesh [28], and minimizes the thin-plate energy [29], which is similar to the thin plate spline.

To keep the skeleton adherent to the original points, we exploit L_1 approximation that is robust to outliers, i.e., v'_i is the L_1 median of the associated points of v_i , and is defined to be any point which minimizes the sum of Euclidean distances to all the associated points.

$$v'_i = \arg \min_x \sum_{j \in N(i)} p_{\{j\}-x}, \quad (8)$$

$N(i)$ represents the set of points associated with v_i . The penalty coefficient w_i in Equation 7 plays the balancing role between the smoothness and the approximation of the skeleton. w_i is set equal to a constant value or designed according to the local geometry, to preserve the features. We demonstrate it in Figure 10.

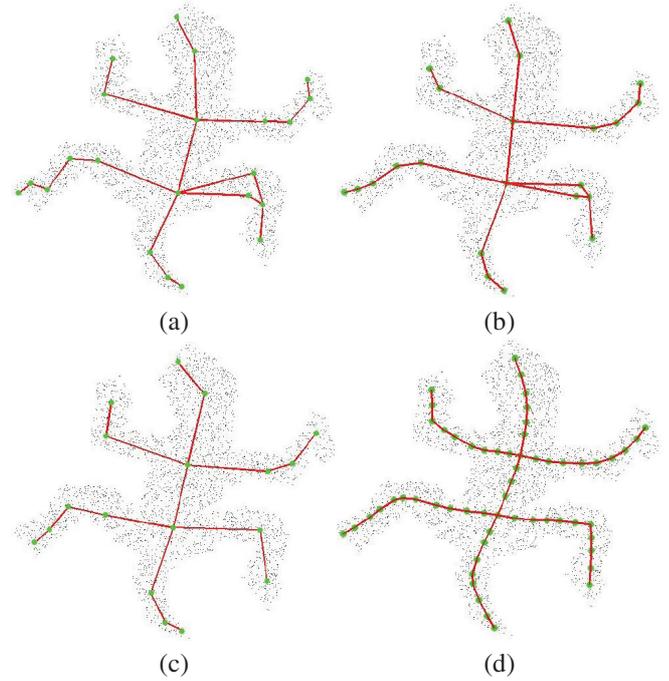


Figure 6 The skeleton refinement step re-samples nodes with prior vertices and radii reflecting local shape scales. (a) The skeleton graph after generation step. (b) The skeleton graph after fitness step. (c) The skeleton graph after refinement step. (d) The final skeleton graph after long edge processing.

The system Equation 7 is equivalent to solving a linear system in least squares sense,

$$AV = b, \quad (9)$$

where

$$A = \begin{pmatrix} L \\ F \end{pmatrix}, \quad F_{ij} = \begin{cases} w_i & j = i \\ 0 & otherwise \end{cases},$$

$$b_k = \begin{cases} 0 & k \leq n \\ w_i v'_i & k = n + i \end{cases}.$$

The unique analytical solution of Equation 9 is $V = (A^T A)^{-1} A^T b$. Since Laplace matrix L in this paper is computed by Fujiwara weights [30] to compensate for irregular edge lengths, which is related to V , Equation 9 is iteratively solved as Au *et al.* [31] suggested.

4.3 Skeleton Refinement

This step is utilized to cope with different local shape attributes and noise better by more intuitive samples and sampling radii. The above skeleton graph provides us with a candidate V_{old} for V and a proper sampling radius r_i for the point v_i in V_{old} , where r_i is set equal to the minimum length of edges linked to v_i . These prior knowledge reflect the scales of local shapes. Take Figure 6 for an example. The lizard is a multi-scale model with large scales in the trunk regions and low scales in the foot regions. Figure 6-b shows that the prior radii adaptively changes these lengths with the scales.

We redo the generation step and fitness step with these prior knowledge. The regeneration step first samples all the nodes

v_i in V_{old} with the sampling radius r_i , and labels the points in P which have been covered by the sampling circles, and then samples the unlabeled points with the radii equaling to the radii of these nearest nodes in V_{old} . The sampling process finally terminates when all points in P are labeled. A graph G is built and then collapsed with the method mentioned before. Since the prior sampling radii reflect the local scales of the shape, the refinement step with these radii avoids poor edges in G to a certain extent. See Figure 3-d, e and Figure 6-c.

Although the refinement step is effective, our algorithm may still suffer from some imperfections, such as short branches, short loops or two junction vertices in intersectional regions connected by a short path. In this paper, we utilize the restructuring operations proposed by Kégl and Krzyzak [26] to handle such cases. See Figure 3-f. More robust method to cope with intersectional regions can be found in [13].

After the above operations, the resulting skeleton graph G captures the topology of the object correctly. However, the shape approximation of the skeletal representation may be not good in the long edges areas. We follow the approach suggested in [10], and iteratively add a new vertex at the middle of an edge if the length of the edge is larger than a threshold δ_{max} . After that, we associate the points in P with their nearest vertex v_i and fit the skeleton again by the Laplacian smoothing method mentioned above. Take Figure 6-d as an example.

5. RESULT

The experiments are carried out on a computer with a Core2 Quad 2.33GHz CPU and 2 GB memory. We have implemented our algorithm in Matlab 2009, and used the following parameters in default cases: the maximum length of edges $\delta_{max} = \min(r)$, r is the sampling radii, the number of neighbors $K = 5$, penalty coefficient $w_i = 1$.

Analysis of the result. In Figure 7, we test our skeletonization algorithm on images with different level of sparseness to confirm the feasibility of the algorithm. In order to simulate the sparseness of the object, the noise was randomly added in the object and it changed the black pixels to white pixels. The amount of the noise is varied between 0 to 95% (SNR = 0.22 dB) of the original number of pixels in the object. SNR represents signal to noise ratio,

$$\text{SNR} = 10 * \lg \frac{\text{Signal}}{\text{Noise}}. \quad (10)$$

The skeletonization result degrades gracefully as the sparseness of ‘‘A’’ in the image increases. It shows that our algorithm deals with sparseness well.

In Figure 8, we test our skeletonization algorithm with different values of the parameter K . As we can see in Figure 8-a and 8-c, the number of edges in the skeleton graph after farthest sampling step increases with the parameter K increasing, and the final skeleton graphs of the shape are almost the same. Thus, our algorithm is insensitive to the number of neighbors K .

Examples of images with multi-scale objects and faded texts are shown in Figure 9 and Figure 10, which contain 2170 and 3345 points respectively. We compare our algorithm with the classic SOM method [10]. We try our best to adjust the main

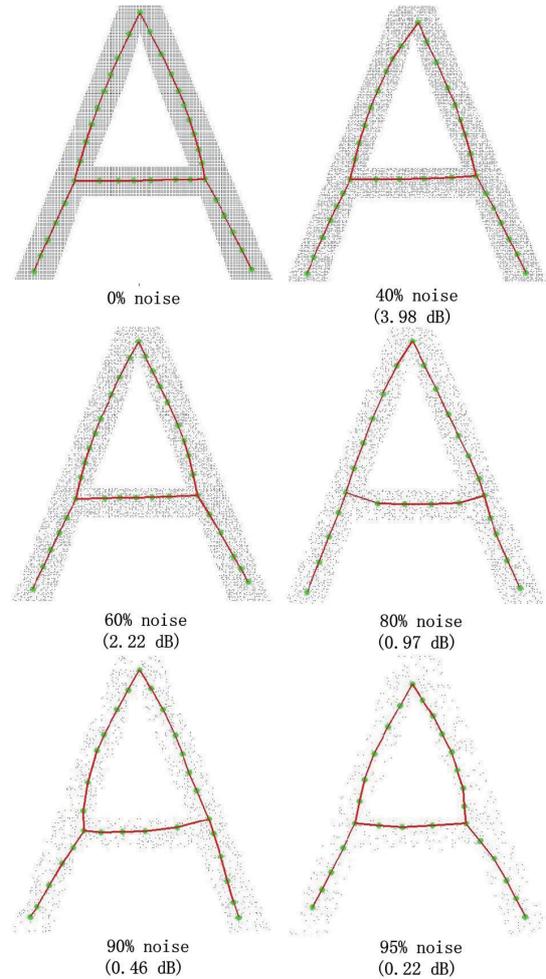


Figure 7 Our skeletonization algorithm handles images with different levels of sparseness introduced by internal noise.

parameters of the SOM method: the initialization number of map units N , minimum length of edge δ_{min} , maximum length of edge δ_{max} , iteration times T . However, the SOM method gives poor results for objects with different local scales and links separate characters due to the drawbacks mentioned above. While our algorithm handles these cases well, since the generation step captures the underlying topology correctly, and the refinement step refines the skeleton respecting the local scales.

The processing time spent for the examples demonstrated in Figure 9, 10 are 0.54s, 1.08s, respectively. While the runtime for the same models in Figure 9, 10 are 1.97s and 4.11s when using the SOM method.

Table 1 shows the computing time of the three steps for all models presented in this paper, which indicates that our algorithm can produce good results with satisfied runtime.

Furthermore, in Figure 10-c, following the method proposed in [26], we classify vertices into end-vertex, corner-vertex, line-vertex, T-vertex, Y-vertex according to the local geometry, and design penalty coefficient w_i for each cases. Specially speaking, at an end-vertex or a corner-vertex, the penalty coefficient w_i is simply set to 100, to preserve the local geometry and avoid over-smoothing. At a line-vertex or a Y-vertex, the smoothness of the polygonal curve is ensured to set $w_i = 1$ as the default cases. At a T-vertex, to penalize one of the three angles for its deviation

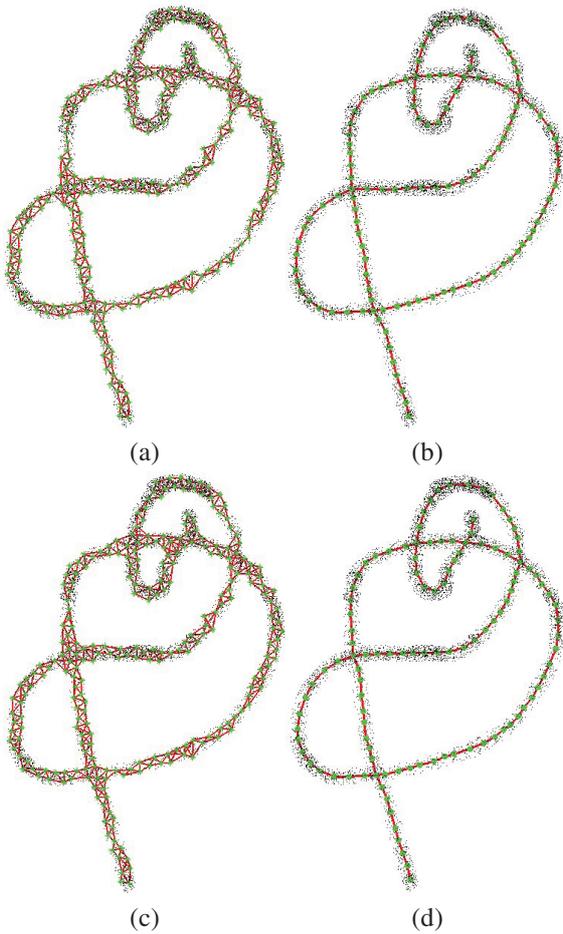


Figure 8 Our method is insensitive to parameter K changing. (a) The skeleton graph after farthest sampling in generation step ($K=10$). (b) The final skeletonization result of (a). (c) The skeleton graph after farthest sampling in generation step ($K=20$). (d) The final skeletonization result of (c).

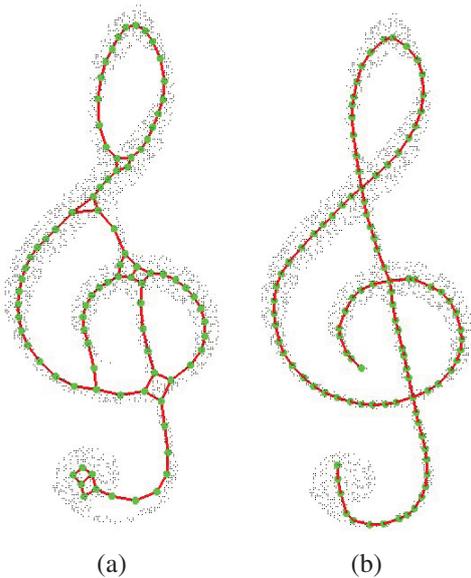


Figure 9 Comparison with SOM method for images with multi-scale objects. (a) SOM method ($N=30, \delta_{\min}=5, \delta_{\max}=10, T=40$). (b) Our method.

Table 1 Time statistics of the proposed skeletonization scheme on different models (seconds).

Models	# Points	Generation	Fitness	Refinement	Total
Figure 3	6603	0.91	0.05	0.41	1.37
Figure 6	3271	0.57	0.05	0.30	0.92
Figure 7-a	12044	3.44	0.08	0.65	4.17
Figure 7-c	4818	0.87	0.02	0.15	1.04
Figure 7-f	603	0.04	0.01	0.02	0.07
Figure 9	2170	0.29	0.04	0.21	0.54
Figure 10	3345	0.61	0.07	0.40	1.08
Figure 11-a	16510	5.17	0.10	0.82	6.09
Figure 11-b	6604	0.89	0.06	0.40	1.35

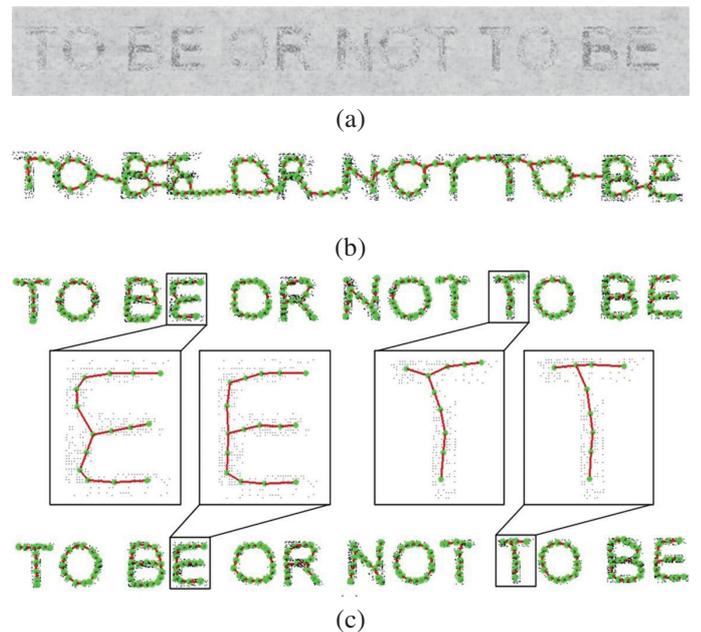


Figure 10 Comparison with SOM method for image with multi-unconnected objects. (a) Original image of faded text. (b) SOM method ($N=50, \delta_{\min}=5, \delta_{\max}=15, T=40$). (c) Our method with constant penalty coefficient (up) and feature-preserving penalty coefficient (down).

from straight angle, we assume no perpendicular edge existing when computing the Laplace matrix, and then treat it as a line-vertex. As we can see in this figure, the skeletonization results keep the features near the crossing regions of "E" and "T" better.

Limitations. Complex shapes present difficulties for most skeletonization algorithms, especially when the shapes are with large sparseness. Since we capture the approximate topology of the shapes by building and collapsing edges, our skeletonization algorithm cannot preserve the details of shape well sometimes. An example is shown in Figure 11. Although our method is able to extract a correct curve skeleton of the major parts of the gear, many gear teeth are lost during the skeletonization process.

Another limitation of our algorithm is that single letter's skeleton may be disconnected, if the sparse shape is extremely non-uniformly distributed. Adjusting the parameter K carefully by hand may overcome it. We will also try to solve this problem by using direction fields.

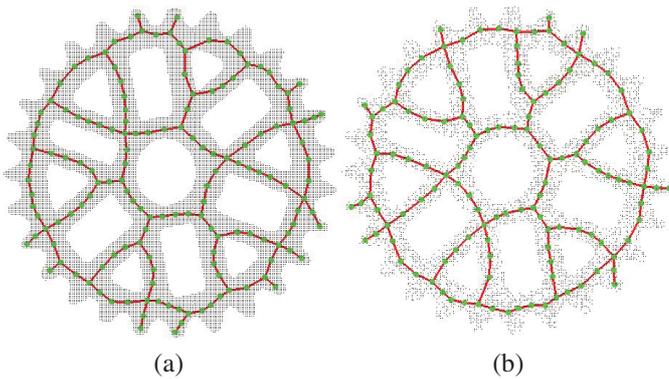


Figure 11 The skeletonization result of the gear image with different levels of sparseness. (a) Without noise. (b) 60% noise.

6. CONCLUSION

We propose an effective and efficient feature-preserving skeletonization algorithm for sparse shapes, which also works well for non-sparse shapes. The skeletonization strategy consisted of node sampling and edge contraction handles complicated cases, such as objects with multi-unconnected parts and self-intersections. And the refinement process improves the quality of acquired skeletons when the object regions contains noise or different local shape scales. Furthermore, it provides an intuitive and linear method to tune the smoothness and fitness of the final skeleton. Accordingly, our algorithm can be used in a wide variety of recognition and classification tasks.

Acknowledgements

This work is supported by the National Nature Science Foundation of China 61370199, 61300187, 61300189 and 61370198, the Fundamental Research Funds for the Central Universities DUT15QY20, 3132016348 and 3132016352.

References

- [1] Saul, LK and Roweis, ST (2015). Think Globally, Fit Locally: Un-supervised Learning of Nonlinear Manifolds. *Journal of Machine Learning Research*, **4**(2) 119-155.
- [2] Saha, PK, Borgfors, G and Baja, GSD (2015). A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, **76**(1) 3-12.
- [3] Zhao, Y, Cao, J and Su, Z, et al. (2011). Efficient reconstruction of non-simple curves. *Journal of Zhejiang University Science C*, **12**(7) 523-532.
- [4] Delgadofriedrichs, O, Robins, V and Sheppard, A (2015). Skeletonization and Partitioning of Digital Images Using Discrete Morse Theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**(3) 654-666.
- [5] Li, Z, Qu, W and Xu, Y, et al. (2014). Efficient shape representation and retrieval in large database. *International Journal of Computer Systems Science and Engineering (CSSE)*, **29**(4).
- [6] Li, K, Nanya, T, Shen, H, Chin, FYL, et al. (2007). An Efficient Cache Replacement Algorithm for Multimedia Object Caching. *International Journal of Computer Systems Science and Engineering (CSSE)*, **22**(1-2) 47-55.

- [7] Qi, H, Li, K, Shen, Y and Qu, W (2010). An effective solution for trademark image retrieval by combining shape description and feature matching. *Pattern Recognition*, **43**(6) 2017-2027.
- [8] Cornea, ND, Silver, D and Min, P (2007). Curve-Skeleton Properties, Applications, and Algorithms. *IEEE Transactions on Visualization and Computer Graphics*, **13**(3) 530-548.
- [9] Qi, H, Lan, G and Li, K (2014). A hierarchical kernel for image classification based on layered representation. *International Journal of Computer Systems Science and Engineering (CSSE)*, **29**(4).
- [10] Singh, R, Cherkassky, V and Papanikolopoulos, N (2000). Self-Organizing Maps for the Skeletonization of Sparse Shapes. *IEEE Transactions on Neural Networks*, **11**(1) 241-248.
- [11] Hastie, T and Stuetzle, W (1989). Principal Curves. *Journal of the American Statistical Association*, **84**(406) 502-516.
- [12] Palenichka, RM and Zaremba, MB (2002). *Multi-Scale Model-Based Skeletonization of Object Shapes Using Self-Organizing Maps*. 16th International Conference on Pattern Recognition, 143-146.
- [13] Liu, X and Jia, Y (2005). A Bottom-Up Algorithm for Finding Principal Curves with Applications to Image Skeletonization. *Pattern Recogn.*, **38**(7) 1079-1085.
- [14] Feng, W, Yang, Z and Deng, J (2014). Moving Multiple Curves/Surfaces Approximation of Mixed Point Clouds. *Communications in Mathematics and Statistics*, **2**(1) 107-124.
- [15] Xia, Y (1989). Skeletonization via the Realization of the Fire Front's Propagation and Extinction in Digital Binary Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**(10) 1076-1086.
- [16] Leymarie, F and Levine, MD (1992). Simulating the Grassfire Transform Using an Active Contour Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(1) 56-75.
- [17] Lee, DT (1982). Medial Axis Transformation of a Planar Shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **4**(4) 363-369.
- [18] Ji, L and Piper, J (1992). Fast Homotopy-Preserving Skeletons Using Mathematical Morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(6) 653-664.
- [19] Chen, YS and Yu, YT (1996). Thinning Approaches for Noisy Digital Patterns. *Pattern Recognition*, **29**(11) 1847-1862.
- [20] Datta, A, Parui, SK and Chaudhuri, BB (1996). *Skeletal Shape Extraction from Dot Patterns by Self-Organization*. 13th International Conference on Pattern Recognition, 80-84.
- [21] Kegl, B, Krzyzak, A, Linder, T and Zeger, K (2000). Learning and Design of Principal Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(3) 281-297.
- [22] Verbeek, JJ, Vlassis, N and Kröse, B (2002). A K-Segments Algorithm for Finding Principal Curves. *Pattern Recogn. Lett.*, **23**(8) 1009-1017.
- [23] Sobiecki, A, Jalba, A, Telea, A (2014). Comparison of curve and surface skeletonization methods for voxel shapes. *Pattern Recognition Letters*, **47**(4) 147-1156.
- [24] Singh, R, Papanikolopoulos, N and Cherkassky, V (1998a). *Object Skeletons from Sparse Shapes in Industrial Image settings*. IEEE International Conference on Robotics and Automation, 3388-3393.
- [25] Singh, R, Wade, MC and Papanikolopoulos, N (1998b). *Letter-Level Shape Description by Skeletonization in Faded Documents*. 4th IEEE Workshop on Applications of Computer Vision, 121-126.

- [26] Kegl, B and Krzyzak, A (2002). Piecewise Linear Skeletonization Using Principal Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(1) 59-74.
- [27] Cao, J, Tagliasacchi, A, Olson, M, Zhang, H, et al. (2010). *Point Cloud Skeletons via Laplacian-Based Contraction*. IEEE Conference on Shape Modeling and Applications, 187-197.
- [28] Sorkine, O and Cohen-Or, D (2004). *Least-squares Meshes*. Proceedings of Shape Modeling International, 191-199.
- [29] Botsch, M and Sorkine, O (2008). On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 213-230.
- [30] Herholz, P, Kyprianidis, JE and Alexa, M (2015). Perfect Laplacians for Polygon Meshes. *Computer Graphics Forum*. **34**(5) 211-218.
- [31] Au, OKC, Tai, CL, Fu, H and Liu, L (2005). *Mesh Editing with Curvature Flow Laplacian*. Eurographics Symposium on Geometry Processing, 191-199.