Extract an Essential Skeleton of a Character as a Graph from a Character Image

Kazuhisa Fujita

University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo, 182-8585 Japan

Abstract

This paper aims to make a graph representing an essential skeleton of a character from an image that includes a machine printed or a handwritten character using the growing neural gas (GNG) method and the relative neighborhood graph (RNG) algorithm. The visual system in our brain can recognize printed characters and handwritten characters easily, robustly, and precisely. How can our brains robustly recognize characters? In the visual processing in our brain, essential features of an object will be used for recognition. The essential features are crosses, corners, junctions and so on. These features may be useful for character recognition by a computer. However, extraction of the features is difficult. If the skeleton of a character is represented as a graph, the features can be more easily extracted. To extract the skeleton of a character as a graph from a character image, we used the GNG method and the RNG algorithm. We achieved to extract skeleton graphs from images including distorted, noisy, and handwritten characters.

Keywords: *Skeletonization, Character Recognition, Self Organizing Map.*

1. Introduction

Why can we robustly recognize characters from a rotated, a distorted, and a noisy image including characters? This ability is provided by robust visual recognition mechanism in the brain. In the visual processing in the brain, common essential features of an object are used for recognition. Essential features of an object are crosses, corners, junctions, circles, and so on [5]. In pattern recognition by a computer, we may achieve to provide more robust image recognition if we effectively use the common essential features. The purpose of this study is to extract essential structures of a character from an image as a graph, that here we called a skeleton graph, in order to easily use essential structures of a character for character recognition by a computer. A skeleton graph represents the skeleton of a character. Each skeleton graph extracted from the images including same characters will be similar. Thus, using similarity of skeleton graphs allows us to achieve to develop more robust character recognition system. In this study, we propose the method of extraction of a skeleton graph of a character from an image including a character in order to develop more robust character recognition system directly using similarity of structures of skeletons.

Extraction of a skeleton from a character image is called skeletonization. Skeletonization is generally executed before recognition process by a learning machine [8]. Skeletonization is a general morphological method that is used to thin a broad stroke of a character image and to extract only a bone of a character from a character image. The major functions of skeletonization in image processing are to reduce data size and to make more easily extract morphological features. This conventional method allows us to extract a skeleton "image" of a character from an image.

To achieve to extract a skeleton graph, we employed the growing neural gas (GNG) method that is topology learning algorithm and one of self organizing map (SOM) methods. SOM [9] can be developed based on topology conserving classifiers. However, the network structure of SOM is static (generally, n-dimensional lattice) and the network structure cannot represent topology of input space. The GNG method improves this problem because the GNG method allows to flexibly increase or to flexibly decrease nodes and edges of the network. The GNG method has been proposed by Fritzke [4]. The GNG method has been widely applied to clustering or topology learning, such as reconstruction of 3D models [7], landmark extraction [2], and object tracking [3]. We applied the GNG method to skeletonization.

In the present study, we demonstrated making a skeleton graph of a character using our proposed method. Under noisy circumstance, our approach could also produce satisfactory result. This achievement may allow us to robustly extract an essential skeleton from a character images.

2. Methods

2.1 Scheme

To make the skeleton graph, we used three steps. The first step, execute image processing that consists of binarizing and trimming. The second step, roughly extract a skeleton graph from a character image using the GNG method. The third step, remove redundant edges and rewire nodes using relative neighborhood graph (RNG) algorithm [12]. The GNG method is high ability to extract topological features of characters and easily method to assemble. However, the graph generated by the GNG method has a few redundant edges because the GNG method tends to make triangle clusters [1,6]. To resolve this problem, we used the RNG method that has the ability to extract a perceptually meaningful structure. Using this method, redundant edges reduce and only essential structures are extracted from a character image. Through the three steps, the skeleton graph can represent essential structures of a character.

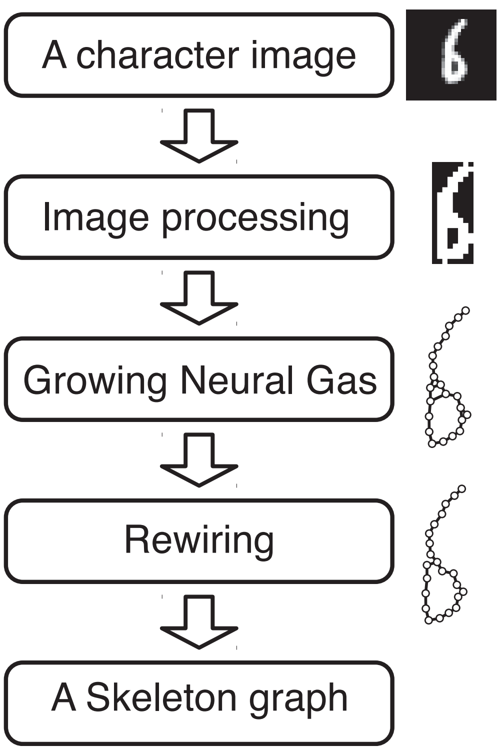

Fig. 1 The scheme to make a skeleton graph.

2.2 Growing Neural Gas

The growing neural gas (GNG) method has been proposed by Fritzke [4]. The GNG method is one kind of SOM methods and extracts topology or classifies data. The network of the GNG flexibly varies and its structure represents data structure. Using these features, we extracted the skeleton of a character from a character image as a graph.

The input space of the GNG network is an image that is a two-dimensional pixel space sized $W \times H$. The network consists of a set A of nodes. Each node $c \in A$ has an associated reference vector w_c. Node's reference vector w_c is denoted by $\boldsymbol{w}_c = (w_{cx}, w_{cy})$. The two parameters represent the node position over the image. The reference vectors must fulfill:

$$0 \leq \boldsymbol{w}_{cx} < W, 0 \leq \boldsymbol{w}_{cy} < H. \quad (1)$$

There are edges between pairs of nodes. These connections are not weighted and not directed. The edges defined topological structure of the network.

Every node is examined to calculate which one's reference vectors are most like the input vector through the following process.

1. Starting with only two nodes that are connected each other. Positions of the nodes are random in $\mathcal{R}$.
2. The input vector $\boldsymbol{x}_i = (x_i, y_i)$ is chosen at random from pixels on a character.
3. The criterion for neighborhood is Euclidean distance between an input vector and a reference vector of a node. The number k of the winning (nearest) node is defined by,

$$k = \arg\min_i \ \|\boldsymbol{w}_i - \boldsymbol{x}\|. \quad (2)$$

   Simultaneously, find the second nearest node s.
4. Increase the age of all the edges connecting with the wining node.
5. Add the squared distance between the input vector and the reference vector of the winning node to a local counter variable:

$$\Delta \mathrm{error}_k = \|\boldsymbol{w}_k - \boldsymbol{x}\,\|^2 . \quad (3)$$

6. The winning node k is rewarded with becoming more like the input vector.

$$\boldsymbol{w}_k(t+1) = \boldsymbol{w}_k(t) + \lambda(t)(\boldsymbol{x} - \boldsymbol{w}_k). \quad (4)$$

   All direct neighbors n of k are also rewarded.

$$\boldsymbol{w}_n(t+1) = \ \boldsymbol{w}_n(t) + \lambda(t)(\boldsymbol{x} - \boldsymbol{w}_n), \quad (5)$$

   where t is the learning frequency and λ is the learning coefficient. λ decays with the learning frequency.

$$\lambda(t) = \lambda_0 \times \left(1 - \frac{t}{T}\right), \quad (6)$$

where λ_0 is the initial learning coefficient and T is the preset maximum training step.

7. If k and s are connected, set the age of this edge to zero. If k and s are not connected, add the edge between these nodes.
8. Remove the edges with the age that is larger than $a_{\max}$. If the node isolated by this remove process, remove the node.
9. Every certain number of input signals generated, insert a new node. The number of nodes has limit $N_{\max}$.
   (a) Determine the neuron q with the maximum summed error.
   (b) If the maximum summed error is more than Error_0, insert new node r between q and its farthest neighbor f:

$$\boldsymbol{w}_r = (\boldsymbol{w}_q + \boldsymbol{w}_f)/2. \quad (7)$$

   (c) The values of Error_0 and $N_{\max}$ are required to hardly make redundant nodes and edges, and triangle cycles.
10. Every certain number of input signals generated, set all error variables to zero.
11. If a stopping criterion is not fulfilled, go to step 2.

Figure 2 shows the growing process of the skeleton graph generated by the GNG method. The character image includes "A". It can be seen that the GNG network learned the skeleton topology of the character.

The parameters for this simulation were: $\lambda = 0.2, N_{\max} = 40, a_{\max} = 28$.

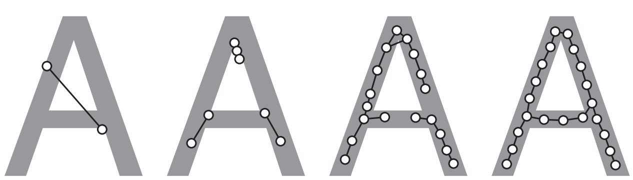

Fig.2 Different steps of convergence of the network for the character "A". These figures show the networks after 0, 4000, 12000, and 80000 steps (from left to right). At the end of the adaptation process the connection between the nodes represents the structure of "A".

2.3 Rewiring

The graph generated by the GNG method tends to have triangle cycles [1,6]. The skeleton graph generated by only the GNG method, shown in fig. 3 B, has the triangle cycles and the redundant edges. The triangle cycles and the redundant edges especially appeared on which stroke was crossed and on a broad curve line. To represent essential bone as a graph, deleting the triangle cycles and redundant edges of the skeleton graph generated by the GNG method are required. To reduce the redundant edges, we implemented the rewiring process. We kept nonredundant edges and deleted redundant edges using the Relative Neighborhood Graph (RNG) algorithm [11,12]. In the view of the RNG algorithm, each node of the graph is relative neighbor if they are near. If nodes i and j are relative neighbors, there dose not exist another node z of the set such that,

$$d(z,i) < d(i,j) \text{ and } d(z,j) < d(i,j), \quad (8)$$

where $d(i,j)$ is the Euclidean distance between i and j. When nodes i, j fulfill the equation except $d(i,j) > (w^2 + H^2)^{1/2} \times 0.15$, nodes are connected. Figure 3 C shows the skeleton graph processed by the RNG algorithm. Redundant edges reduced and an essential skeleton graph was extracted.

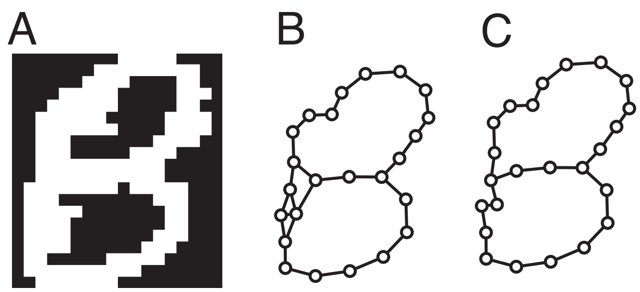

Fig. 3 A shows the handwritten digit "8". B and C illustrate the graph generated by the GNG method without and with the rewired process, respectively.

3. Results

To verify that the extracted skeleton graph represents essential structures of a character, the proposed method has been tested with the four sets of characters that are represented by binary images. The results for the four sets of images are shown in fig. 4. The first set consisted of regular images that include undistorted printed-characters. The skeleton graphs represented the essential structures of the characters, shown in fig. 4A. The second set consisted of the images of distorted and rotated printed-characters. In this case, the skeleton graphs also represented essential structures. The topology of the skeleton graphs is almost same structures of the skeleton graph generated from the regular images. The third set consisted of the images of isolated handwritten digits from the MNIST Database [10]. In this case, the skeleton graphs also represented essential structures.

The fourth set consisted of the images of noised characters. In this test, we randomly changed white pixels on a character to black pixels. The random noise is uniformly distributed on a character. Here, we define noise rate $\xi = \nu/\rho$, where ν is the amount of changed pixels and ρ is the original number of pixels on a character. Figure 4D shows the skeleton graphs at different noise level. The skeleton graphs produced by our method was consistent with visual form of characters for $\xi = 0.95$ and 0.99. However, for $\xi = 0.995$, the skeleton graph could not represent the form of characters.

Using our method, we could extract essential structures of a character as a skeleton graph. It is important that the skeleton graphs made from images including a same character have the common essential skeleton. The structure of "A" has the features that are two T-junctions, one cycle, and one acute curve. If our method effectively extracts essential skeleton graphs from various "A" images, the skeleton graphs must have these features. Figure 5A shows the skeleton graphs extracted from not-distorted "A", rotated one, and rotated and distorted one. These skeleton graphs had the common essential features.

However, skeleton graphs generated from handwritten character images or more distorted character images that include a same character may not always have same features. For example, the skeleton graphs generated from the handwritten digits "2" shown in fig. 5B was different from one shown in fig. 5C in spite of the same digit. The skeleton graphs shown in fig. 5B was the typical skeleton of "2". The skeleton graph generated from a printed character image will also have the same structure. The typical structure of the skeleton graph of "2" is one T-junction. While the skeleton graph shown in fig. 5C was not typical because the graph did not have the typical feature that was one T-junction. The skeleton graph of fig. 5C had two T-junctions and one cycle. These results suggest that the skeleton graphs generated from images including a same character may have different features.

4. Conclusion and future works

In this paper, we proposed a method to generate a skeleton graph representing essential features of a character in an image. We generated a skeleton graph from a character image using the GNG method, and then we deleted redundant edges of the skeleton graph using the RNG algorithm. The proposed method has been tested on images including a printed character, a distorted printed-character, a handwritten digit, and a noised character. The experimental results show the effectiveness of the proposed method. The skeleton graph preserved an approximation of the original shapes and had essential features of a character. The topology of the skeleton graph made by our method did not depend on rotation and distortion of a printed character.

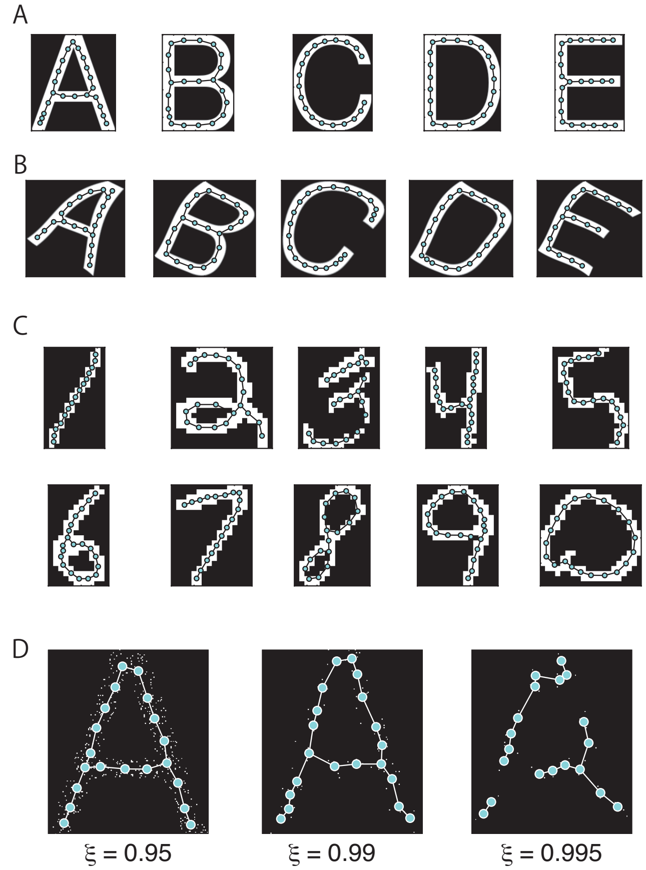

Fig. 4 (A) Alphabets. (B) Distorted and Rotated alphabets. (C) Handwritten digits. (D) Noised alphabets.

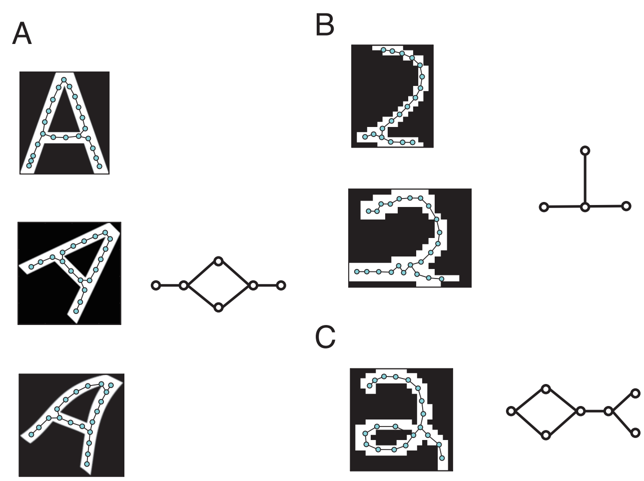

Fig.5 Relation between skeleton graphs and topology of the graphs. In the cases of A and B, the skeletons generated from images of same characters had same topology. In the case of C, the skeleton had different topology from one in B.

However, skeleton graphs generated from images including same character did not always have same features, for example handwritten characters. Furthermore, the skeleton graphs generated from images including different characters may have same topology. For example, the skeleton graphs made from “e” and “p” have one cycle and one T-junction, and the topology of these graphs is same. In this case, character recognition is not achieved using only topology of the skeleton graphs. To achieve character recognition, location of nodes, the number of nodes will be required.

In this study, we made a skeleton graph from a binary image. However, our method can be applied to directly making a skeleton graph from a gray scale image itself. To extract a skeleton graph from a gray scale image itself, the probability of selection of pixels on a character depends on the intensity of pixels in the GNG process.

In future work, we shall develop a character classification method using similarity of skeleton graphs because the skeleton graphs extracted from images including same character have similar features.

References

[1] A. Datta, S. K. Parui, and B. B. Chaudhuri, “Skeletonization by a topology-adaptive self-organizing neural network”, Pattern Recognition, 34, 2001, pp. 617-629.
[2] E. Fatemizadeh, C. Lucas, and H. Soltanian-Zadeh, “Automatic landmark extraction from image data using modified growing neural gas network”, IEEE Transactions on Information Technology in Biomedicine, 7, 2003, pp. 77-85.
[3] H. Frezza-Buet, “Following non-stationary distributions by controlling the vector quantization accuracy of a growing neural gas network”, Neurocomputing, 71, 2008, pp. 1191-1202.
[4] B. Fritzke, “A growing neural gas network learns topologies”, Advances in Neural Information Processing Systems, 7, 1995, pp. 625-632.
[5] K. Fukushima, “Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”, Biological cybernetics, 36, 1980, pp. 193-202
[6] J. García-RodríGuez, A. Angelopoulou, J. M. García-Chamizo, A. Psarrou, S. O. Escolano, and V. M. Giménez, “Autonomous growing neural gas for applications with time constraint: Optimal parameter estimation”, Neural Networks, 32, 2012, pp. 196-208.
[7] Y. Holdstein and A. Fischer, “Three-dimensional surface reconstruction using meshing growing neural gas (mgng)”, The Visual Computer, 24, 2008, pp. 295-302.
[8] L. Jia and Z. Hou, “An improved cost-efficient thinning algorithm for digital image”, International Journal of Computer Science Issues, 10, 2013, pp. 190-197.
[9] T. Kohonen, “The self-organizing map”, Proceedings of the IEEE, 78, 1990, pp. 1464-1480.
[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition”, Proceedings of the IEEE, 86, 1998, pp. 2278-2324.
[11] J. Rocha and T. Pavlidis, “Character recognition without segmentation”, IEEE transaction on Pattern Analysis and Machine Intelligence, 17, 1995, pp. 903-909
[12] G. T. Toussaint, “The relative neighbourhood graph of a finite planar set”, Pattern Recognition, 12, 1980, pp. 261-268.