JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 8, No. 3, September 2014, pp. 149-156

Stroke Width Based Skeletonization for Text Images

Minh Hieu Nguyen, Soo-Hyung Kim, Hyung Jeong Yang, and Guee Sang Lee*
Dept. of Electronics and Computer Eng., Chonnam National University, Gwangju, South Korea
hieunp131@gmail.com, shkim@jnu.ac.kr, hjyang@jnu.ac.kr, gslee@jnu.ac.kr

Abstract
Skeletonization is a morphological operation that transforms an original object into a subset, which is called a 'skeleton'. Skeletonization has been intensively studied for decades and is a challenging issue especially for special target objects. This paper proposes a novel approach to the skeletonization of text images based on stroke width detection. First, the preliminary skeleton is detected by using a Canny edge detector with a Tensor Voting framework. Second, the preliminary skeleton is smoothed, and junction points are connected by interpolation compensation. Experimental results show the validity of the proposed approach.

Category: Human computing

Keywords: Skeletonization; Stroke width; Tensor voting; Text image

I. INTRODUCTION

Image skeletonization is a morphological image processing operation that transforms a component of a digital image into a subset, called a 'skeleton', of the original component. The skeleton usually emphasizes geometrical and topological properties of the shape. It has been used in several applications, such as computer vision, image analysis, and digital image processing.

Consider the binary digital character 'H' in Fig. 1.

Basically, it consists of three connected strokes (two vertical and one horizontal). The topology, which is the important information for object recognition, represents how the three parts are connected. In such a situation, it will be convenient if the thickness is reduced as much as possible (to a skeleton), so that the topological analysis becomes as simple as possible. For example, it would be nice to represent the pattern above as a one-pixel thick white pattern as illustrated in Fig. 2.

The processes or methods to accomplish this task are called skeletonization or thinning techniques, and the resulting patterns are usually referred to as skeletons.

Nowadays, many graphic and vision applications, which include character recognition, signature verification, and fingerprint recognition, are required to represent and understand the shape of a target object. The skeleton is one kind of shape descriptor [1]. It is a compact, medial structure that lies within a solid object [2]. A skeleton of a 2D object consists of 1D elements (e.g., curve, straight line); whereas the skeleton of a 3D object may consist of both 1D and 2D elements (e.g., surface). A skeleton is a lower dimensional object that essentially represents the shape of its target object. Because a skeleton is simpler than the original object, many operations can be performed more efficiently on the skeleton than on the full object.

Skeletonization is divided into two main approaches: iterative and non-iterative. In the iterative techniques, the peeling contour process is iteratively conducted in a parallel or sequential manner; in the parallel way, the whole unwanted pixels are erased after identifying the whole

Open Access http://dx.doi.org/10.5626/JCSE.2014.8.3.149 http://jcse.kiise.org

Received 12 June 2014; **Revised** 22 July 2014; **Accepted** 29 July 2014

*Corresponding Author

pISSN: 1976-4677 eISSN: 2093-8020

Fig. 1. A pattern consisting of three strokes.

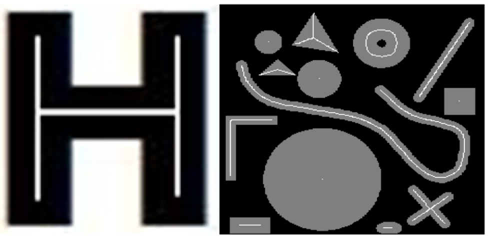


Fig. 2. The skeleton is the thinnest representation of the original pattern that preserves the topology.

wanted pixels. Whereas in sequential techniques, the unwanted pixels are removed in identifying the desired pixels in each iteration as in [3]. Examples of pixel-based techniques are thinning and distance transformation. These techniques need to process every pixel in the image; this can incur a long processing time, and leads to reduced efficiency. In the non-iterative approach, the skeleton is directly extracted without individually examining each pixel, but these techniques are difficult to implement and slow [4].

Some thinning methods suffer from common traditional problems, such as the one pixel width of the skeleton and skeleton connectivity. Besides, distortion in the topology of the shape skeleton is a serious problem in thinning applications [5]. Several techniques failed in preserving the topologic shape [6, 7]. Spurious tails and rotating the text shape are other serious problems that most thinning methods failed to solve [6-8].

This paper presents a novel approach based on the stroke width of a target object to extract its skeleton. To the best of our knowledge, this is the first time tensor voting and stroke width have been used in skeletonization.

II. PROPOSED METHOD

First, contour from the object is detected by using a Canny edge detector [9] and then tensor voting [10] is applied to estimate the normal vectors of edge pixels in the previous step. Starting from each edge pixel and following its normal vector, its counterpart can be found at the opposite edge. The starting edge pixels and their counterparts form a set of strokes. From these strokes, we have a set of their center points that make up preliminary parts of the skeleton.

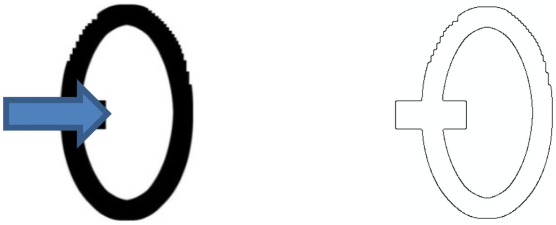


Fig. 3. Contour from Canny.

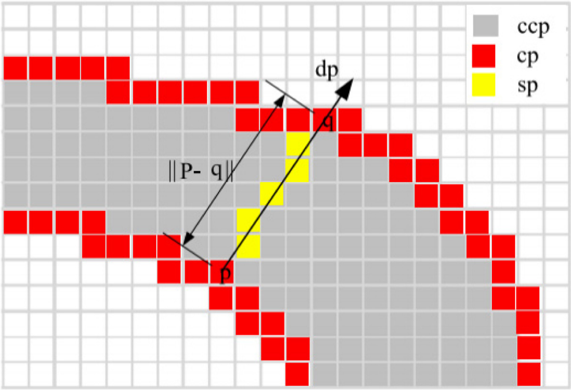


Fig. 4. Stroke width computation.

Second, we smooth the primary skeletons, and connect them through the junction area, by finding the junction points. The junction points are obtained by an interpolation compensation technique.

A. Contour

The contour of the target object is extracted using a Canny edge detector [9] (Fig. 3).

B. Stroke Width

We define a stroke to be a contiguous part of an image that forms a band of nearly constant width as depicted in Fig. 4. We do not assume to know the actual width of the stroke but rather recover it. In Fig. 4, p is a component contour pixel (cp) that lies on the edge. Searching in the direction of the gradient at p leads to finding q, the corresponding pixel on the other side of the stroke. Pixels along the ray form a stroke [11]. These pixels are called stroke width pixels (sp). Candidate component pixels (ccp) lie between the two edges but do not belong to strokes.

C. Normal Vectors of Edge Pixels

1) Using Gradient

In the gradient method, the normal vector of an edge pixel can be approximated by considering only eight surrounding pixels as in Eq. (1). Therefore, when the edge is not smooth enough, i.e., it has staircase elements, normal vectors would be computed incorrectly. Fig. 5(b) illustrates this situation. In this case, the gradient method is no longer suitable for finding normal vectors. This paper proposes to use a Tensor Voting based method to solve this problem.

$$\Delta I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right] \qquad (1)$$

where, I is the grayscale image, $\frac{\partial I}{\partial x}$ is the gradient in the x-direction, and $\frac{\partial I}{\partial y}$ is the gradient in the y-direction.

2) Using Tensor Voting

The Tensor Voting based method estimates the normal vector based on the pixels in a voting field. Pixels in high-density areas of the voting field contribute more significantly than others.

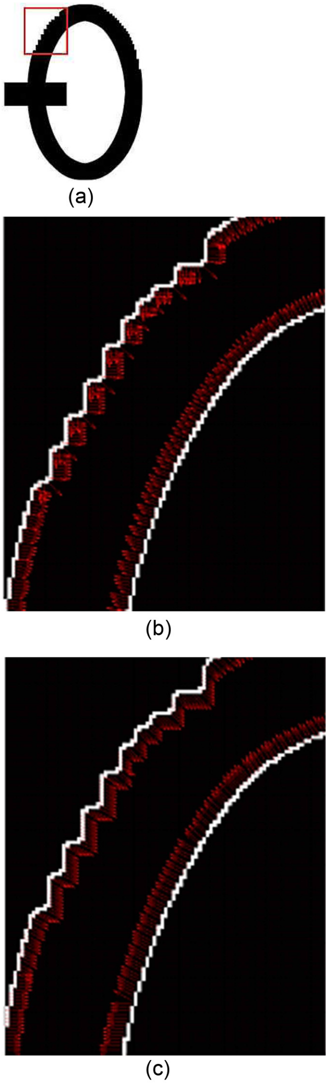


Fig. 5. Normal vectors of edge pixels. (a) Text with rough edges, (b) normal vectors by gradient methods, and (c) normal vectors by tensor voting.

First, edge pixels are extracted. Coordinates of these edge pixels are input into the Tensor Voting framework [10]. After the voting process, tangent vectors are obtained from the stick tensor. The normal vector is roughly perpendicular to the tangent vector. Fig. 5(c) shows normal vectors that are results from the tensor voting method. In this situation, it is better than the gradient method. Compared to Fig. 5(b), the normal vectors around curves are well computed.

D. Preliminary Skeleton Extraction

Contour from the object is detected by using Canny edge detector, and then tensor voting is applied to estimate the normal vectors of edge pixels in the previous step. Starting from each edge pixel and following its normal vector, its counterpart can be found at the opposite

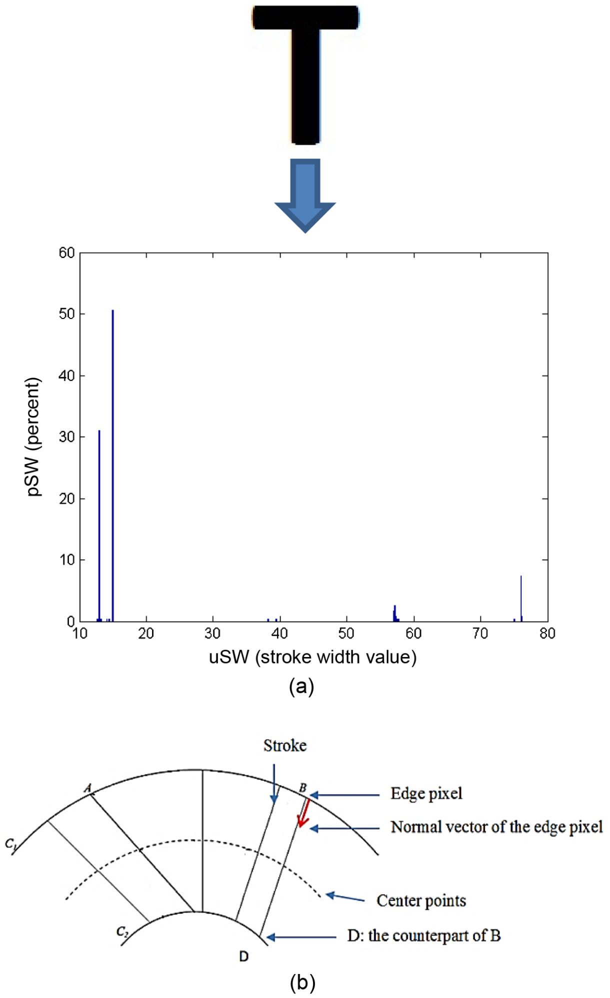


Fig. 6. (a) Stroke width distribution, (b) preliminary skeleton extraction.

edge. The starting edge pixels and their counterparts form a set of strokes. Then we have the stroke width distribution. The weighted prevalent stroke width (WPSW) is computed by Eq. (2).

$$\mathrm{WPSW} = \sum_{i=1}^{n} \mathrm{uSW}_i * \mathrm{pSW}_i \tag{2}$$

where, uSWi represents one stroke width value in the distribution and pSW is the corresponding percent of that value over the total.

For example, in Fig. 6(a), from the stroke width distribution, we have uSW = {12.64, 13, 13.03, 13.15, 13.34, 14.14, 14.31, 15, 38.27, 39.39, 57, 57, 57.14, 57.21, 57.42, 57.55, 57.70, 75, 76, 76, 76.16, 76.23} and pSW = {0.0043, 0.3013, 0.0087, 0.0043, 0.0043, 0.0043, 0.0043, 0.5065, 0.0043, 0.0043, 0.0131, 0.0043, 0.0262, 0.0087, 0.0043, 0.0043, 0.0043, 0.0043, 0.0698, 0.0043, 0.0043, 0.0043}. By applying Eq. (2), we have WPSW = 22.64; this is the prevalent stroke width throughout the input image.

After that, only strokes that have width roughly equivalent to the weighed prevalent stroke width are kept. From these valid strokes, we have a set of their center points that make up preliminary parts of the skeleton.

Algorithm 1. Preliminary Skeleton Extraction

Step 1. Find edge pixels in a grayscale image using the Canny method.
Step 2. Estimate normal vectors of edge pixels using gradient and tensor voting methods.
Step 3. Compute the weighted prevalent stroke width using the stroke width calculation in Eq. (2).
Step 4. Get valid strokes, which have their ends as edge pixels and width roughly equivalent to the weighted prevalent stroke width.
Step 5. Extract center points of valid strokes using the Euclidean distance.

Since the proposed method starts from a contour of the object, the existence of edge pixels is crucial for extracting the target skeleton. In the cases of junctions or crossing areas, some edges are missing so that the skeleton cannot be extracted by examining the stroke width alone. The reason is that as the distance between the starting pixel in one edge and its counterpart in another edge (they form a stroke) increase, stroke widths in these areas are considerably larger than the stroke width that is prevalent over the entire image. Strokes in junction areas are discarded because they exceed by far the weighed prevalent stroke width. In this paper, an interpolation compensation technique is used to deal with this problem.

E. Interpolation Compensation

To find junction points that are missing from the preliminary skeleton, we develop an interpolation technique [12]. The key of this technique is to position the junction skeleton points in the crossing area based on three detectable types of characteristic points: terminal point, divider point, and corner point. The junction skeleton point can be estimated in two ways: 1) it is the centroid of the polygon, which is composed of the above three types of characteristic points and 2) the position of the junction skeleton point can be calculated by the least square method with the characteristic points.

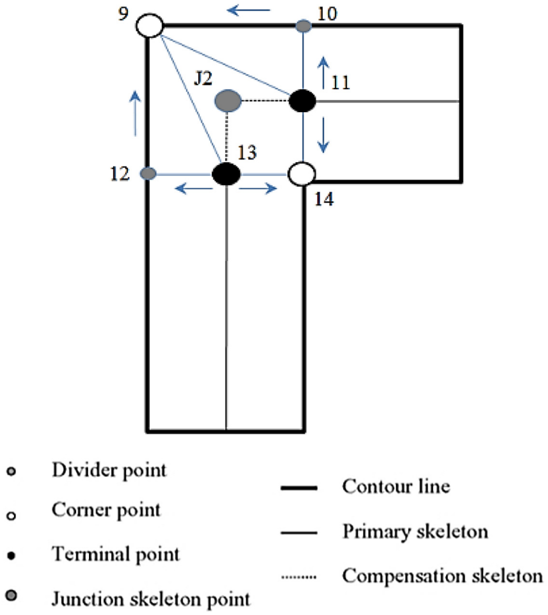


Fig. 7. Interpolating a missing junction point.

The corner points are detected by using Harris corner detection [13] as above. The terminal point is the start or end point of the preliminary skeleton. The divider point is the intersection point between the stable and the unstable contour segments. Along the gradient direction of the terminal point (for example, points 11 and 13 in Fig. 7), the divider points (points 10 and 12 in Fig. 7) can be easily detected. They are illustrated in Fig. 7.

Algorithm 2. Interpolation Compensation

Step 1. Detect terminal points from the preliminary skeleton.
Step 2. Find divider points by following the gradient (positive and negative) direction of each terminal point. Along its corresponding unstable contour segment, another divider (or corner) point can be met.
Step 3. Group all characteristic points related to the terminal point into a subgroup.
Step 4. Merge two subgroups that share some characteristic points into a bigger group. Repeat this process until all groups are stable.
Step 5. Estimate the centroid of each polygon that is formed by each isolated group above.

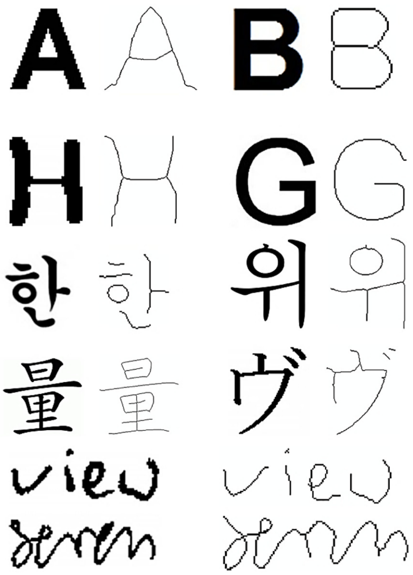


Fig. 8. Some experimental results.

Table 1. Results from the methods discussed

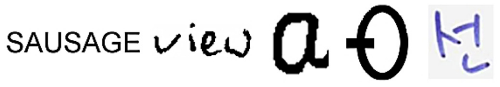
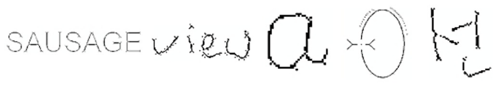
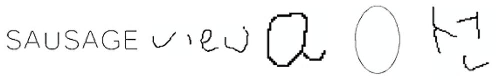
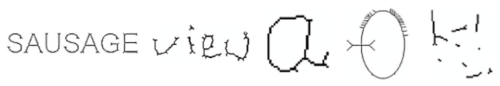
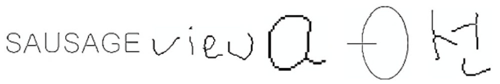
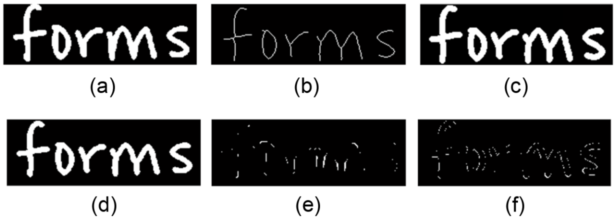
Method	Image
	
Distance transform -based	
Voronoi diagram -based	
Blum (Matlab function)	
Proposed method	

Table 2. Comparison with benchmarks

Method	Precision	Recall	F-measure
Blum [14]	97.82	75.31	85.10
Distance transform-based [15]	98.73	72.81	83.81
Voronoi diagram-based [16]	90.03	82.97	86.35
Proposed method	**85.34**	**86.25**	**85.79**



(a) (b) (c)

(d) (e) (f)

Fig. 9. Evaluation method. (a) Original image, (b) skeleton, (c) reconstructed image, (d) duplicated area, (e) missing area, and (f) error area.

III. EXPERIMENT AND COMPARISON

In this section, highlight results of the proposed method are presented and discussed. The algorithm is tested on a variety of inputs—isolated letters, handwritten words, and finally, graphics and symbols, regarding skeleton continuity, shape preservation, and a one-pixel-wide skeleton. The goal is to verify if the performance complies with the expectations. There are 106 images in the dataset.

A. Experimental Results

As seen in Fig. 8, the proposed method preserves the letter shapes. It does a good job of maintaining the original connectivity in the output skeletons, which is essential for many applications.

B. Comparison

Table 1 shows sample results of the proposed approach in comparison with methods mentioned in the Introduction section.

Table 2 presents a comparison between the proposed approach and other methods by quantitative evaluation. The image reconstructed from the skeleton is compared with the original image in terms of duplicated area, missing area, and error area (Fig. 9).

- Duplicated area: overlapped parts of reconstructed image (from skeleton) and original image.
- Missing area: parts that exist in the original image, but are missing in the reconstructed image.
- Error area: parts that exist in the reconstructed image, but are missing in the original image.

$$Precision = \frac{Duplicated\ area}{Duplicated\ area + Missing\ area}$$

$$Recall = \frac{Duplicated\ area}{Duplicated\ area + Error\ area}$$

$$F-measure = \frac{(1+\beta^2)(Precision \times Recall)}{\beta^2(Precision + Recall)}$$

In our work, β^2 is set to 1. This is also known as the F1-measure with recall and precision being evenly weighted.

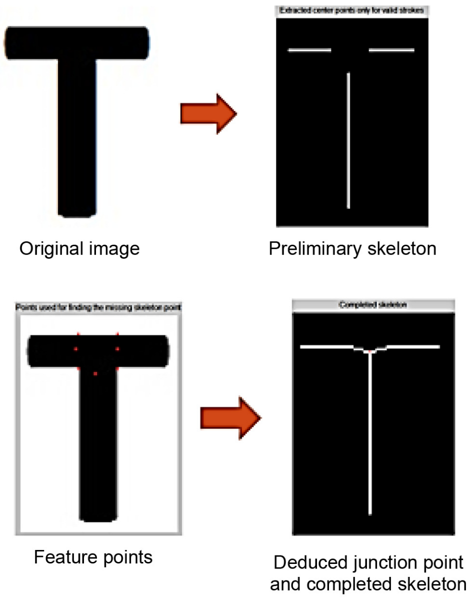


Fig. 10. From the input to the output.

C. Strengths and Weakness

The proposed method has the following strengths:

- It extracts skeletons without spurs or undesirable branches (Fig. 10).
- It is non-iterative and works even when the edge is not smooth enough, i.e., it has staircase elements.
- It is tailored for printed characters, where the stroke width is consistent.
- It is a grayscale skeletonization algorithm that can solve the problem of shape distortion by binarization (as seen in Fig. 11).

It also has the existing weaknesses (Fig. 12):

- It is a 2D skeletonization method.
- It is sensitive to the stroke width of the target object.
- It fails to produce desirable results in cases of complex shapes.

IV. CONCLUSION

This paper proposed a novel approach based on the stroke width of a target object to extract its skeleton. First, we detect the contour from the object and use normal vectors, which are estimated by a Tensor Voting based technique to extract strokes between edge pixels. From these strokes, we have a set of their center points that make up preliminary parts of the skeleton. Second, we smooth the primary skeletons and connect them through junction areas by finding the junction points. The junction points are obtained by an interpolation compensation technique. The experimental result proves that tensor voting and stroke width can be used to extract the skeletons of 2D objects. For the purpose of extracting skeletons to reconstruct the original image, the performance of the proposed method can be matched with popular skeletionization methods. In future research, the accuracy for complex shape objects needs to be improved.

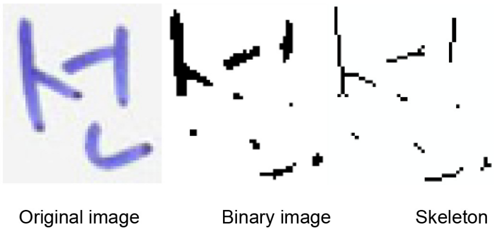


Fig. 11. Shape distortion by binarization.

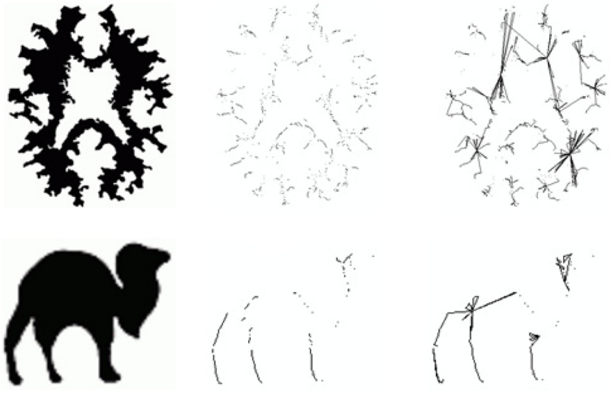


Fig. 12. Current failed cases.

AKNOWLEDGMENTS

This work was supported by a National Research Foundation of Korea grant (No. 2013-056480 and 2013-006535) funded by the Korea government (MEST). This research was also supported by a Ministry of Science, ICT & Future Planning of Korea grant (No. NIPA-2014-H0301-14-1014) under the Information Technology Research Center support program supervised by the National IT Industry Promotion Agency.

REFERENCES

1. P. Morrison and J. J. Zou, “An effective skeletonization method based on adaptive selection of contour points,” in *Proceedings of the 3rd International Conference on Information Technology and Application*, Sydney, Australia, 2005, pp. 644-649.
2. T. Ju, M. L. Baker, and W. Chiu, “Computing a family of skeletons of volumetric models for shape description,” *Com-*

puter-Aided Design, vol. 39, no. 5, pp. 352-360, 2007.

3. G. Nemeth and K. Palagyi, “Topology preserving parallel thinning algorithms,” *International Journal of Imaging Systems and Technology*, vol. 21, no. 1, pp. 37-44, 2011.
4. K. Saeed, M. Tabedzki, M. Rybnik, and M. Adamski, “K3M: a universal algorithm for image skeletonization and a review of thinning techniques,” *International Journal of Applied Mathematics and Computer Science*, vol. 20, no. 2, pp. 317-335, 2010.
5. W. R. Quadros, K. Shimada, and S. J. Owen, “Skeleton-based computational method for the generation of a 3D finite element mesh sizing function,” *Engineering with Computers*, vol. 20, no. 3, pp. 249-264, 2004.
6. Z. Guo and R. W. Hall, “Fast fully parallel thinning algorithms,” *CVGIP: image understanding*, vol. 55, no. 3, pp. 317-328, 1992.
7. M. Ahmed and R. Ward, “A rotation invariant rule-based thinning algorithm for character recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1672-1678, 2002.
8. Y. Y. Zhang and P. S. P. Wang, “A parallel thinning algorithm with two-subiteration that generates one-pixel-wide skeletons,” in *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, 1996, pp. 457-461.
9. J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.
10. G. Medioni, M. S. Lee, and C. K. Tang, *A Computational Framework for Segmentation and Grouping*, Amsterdam: Elsevier, 2000.
11. B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010, pp. 2963-2970.
12. X. You and Y. Y. Tang, “Wavelet-based approach to character skeleton,” *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1220-1231, 2007.
13. C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proceedings of the 4th Alvey Vision Conference,* 1988, pp 147-151.
14. H. Blum, “A transformation for extracting new descriptors of shape,” in *Proceedings of a Symposium on Models for the Perception of Speech and Visual Form*, 1967, pp. 362-380.
15. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision (Volume 1)*, Reading: Addison-Wesley, 1991.
16. R. Ogniewicz and M. Ilg, “Voronoi skeletons: theory and applications,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Champaign, IL, 1992, pp. 63-69.

Minh Hieu Nguyen

Minh Hieu Nguyen received his B.S. degree in Software Engineering from FPT University, Vietnam in 2011. He is currently a M.S. student in the Department of Electronics and Computer Engineering in Chonnam National University, South Korea. His research interests are in image processing, computer vision, and text extraction.

Soo-Hyung Kim

Soo-Hyung Kim received his B.S. degree in Computer Engineering from Seoul National University in 1986, and his M.S. and Ph.D. degrees in Computer Science from the Korea Advanced Institute of Science and Technology in 1988 and 1993, respectively. From 1990 to 1996, he was a senior member of the research staff in the Multimedia Research Center of Samsung Electronics Co., Korea. Since 1997, he has been a professor in the Department of Computer Science, Chonnam National University, Korea. His research interests are in pattern recognition, document image processing, medical image processing, and ubiquitous computing.

Hyung Jeong Yang

Hyung Jeong Yang received her B.S., M.S. and Ph.D. from Chonbuk National University, Korea. She is currently an associate professor at the Department of Electronics and Computer Engineering, Chonnam National University, Gwangju, Korea. Her main research interests include multimedia datamining, pattern recognition, artificial intelligence, e-Learning, and e-Design.

Guee Sang Lee

Guee Sang Lee received his B.S. degree in Electrical Engineering and M.S. degree in Computer Engineering from Seoul National University, Korea in 1980 and 1982, respectively. He received his Ph.D. degree in Computer Science from Pennsylvania State University in 1991. He is currently a professor of the Department of Electronics and Computer Engineering in Chonnam National University, Korea. His research interests are mainly in the field of image processing, computer vision, and video technology.