A Pseudo-Distance Map for the Segmentation-Free Skeletonization of Gray-Scale Images

Jeong-Hun Jang and Ki-Sang Hong
Electrical and Computer Engineering Division, POSTECH, Korea
E-mail: {jeonghun,hongks}@postech.ac.kr

Abstract

In this paper, we introduce a new tool, called a pseudo-distance map (PDM), for extracting skeletons from gray-scale images without region segmentation or edge detection. Given an edge-strength function (ESF) of a gray-scale image, the PDM is computed from the ESF using the partial differential equations we propose. The PDM can be thought of as a relaxed version of a Euclidean distance map. Therefore, its ridges correspond to the skeleton of the original gray-scale image and it provides information on the approximate width of skeletonized structures. Since the PDM is directly computed from the ESF without thresholding it, the skeletonization result is generally robust and less noisy. We tested our method using a variety of synthetic and real images. The experimental results show that our method works well on such images.

1. Introduction

A skeleton plays a vital role as a compact shape descriptor. The skeletonization of shapes has been studied by many researchers for a long time since Blum first introduced its concept [1]. Skeletonization methods proposed up to date can be categorized largely into four approaches: topological thinning [2], analytical computation of medial axes [3], medial axis extraction from a distance map [4, 5], and shock detection from an evolving boundary contour [6, 7]. The common requirement of the above approaches is that the boundary contour of a shape must be determined before its skeletonization. This requirement makes it difficult to use conventional skeletonization methods to extract skeletons from gray-scale images. If one tries to skeletonize gray-scale images using conventional methods, the images should be segmented first into meaningful regions using region segmentation techniques, or the boundaries of the regions should be extracted directly with edge detection techniques. However, since region segmentation and edge detection always involve thresholding operations, it is unavoidable to lose some useful information provided by the original image which seems to be necessary to extract a reliable skeleton. For example, in the case of edge detection, gradient magnitude information is completely lost during the thresholding operation and edge detectors usually produce edge gaps, which cause a noisier skeleton than expected. Therefore, more sophisticated pruning algorithms are required to extract a reliable and meaningful skeleton, which complicates the problem [8].

There are several papers dealing with the problem of skeletonizing gray-scale images without their segmentation. Tari *et al.* proposed a skeletonization method, where a skeleton is defined as a set of local maximum curvature points of level curves of an edge-strength function that is computed from a gray-scale image [9]. Although their method is theoretically interesting, the skeleton tends to be ill-formed where the edge-strength function is almost flat, and it is typically not connected, as pointed out in [7]. Moreover, the method does not provide any information regarding the width of detected structures, which makes the extracted skeleton less useful.

Pizer *et al.* suggested a concept called *cores*, which are obtained by integrating ridges of a medialness function in scale space [10]. Lindeberg took a similar approach to get medial axes directly from gray-scale images [11]. The major drawback of this approach is that the connectivity between medial axes of large-scale objects and those of small-scale ones is not good. Furthermore, according to their experimental results, the produced skeletons are often noisy.

Chung and Sapiro modified the continuous-scale erosion equations to extract skeletons without presegmenting gray-scale images [12]. However, their method is restrictive in that it is applicable to only those images whose background is almost homogeneous and darker (or brighter) than foreground objects to be skeletonized over the entire domain of the images. It also gives no information on the width of the skeletonized structures.

In this paper, we propose a new method for the skeletonization of gray-scale images, which is robust and capable of providing information on the thickness of the skele-

0-7695-1143-0/01 $10.00 (C) 2001 IEEE

tonized structures. The core of our work is the computation of a *pseudo-distance map* from the edge-strength function of a gray-scale image using the variational method. An edge-strength function v is a smooth function that approaches one at the shape boundary and decays rapidly to zero while receding from the boundary. A pseudo-distance map f can be thought of as a relaxed version of a conventional Euclidean distance map. The value of f is almost equal to zero where edge strength is relatively large and the function has nearly constant slopes at the points with small edge strength except the positions where two opposite slopes meet, which correspond to the skeleton of an input image. Given the edge-strength function v, the function f is obtained by numerically solving the partial differential equations (PDEs) which are derived from the energy functional we propose. After obtaining the function f, the ridges of f are extracted with a conventional ridge detector.

Our method usually gives robust and less noisy skeletonization results due to the full utilization of edge strength information and some regularization effect of the variational formulation. Furthermore, since our method uses only the central finite-difference scheme to solve the PDEs without requiring more elaborate schemes such as level-set methods [13], it is easy and straightforward to implement our algorithm.

This paper is organized as follows. In Section 2, we first formulate PDEs for computing a pseudo-distance map in one dimension and extend them to the two dimensional case. In addition, we suggest a useful technique for stabilizing and speeding up the convergence of the proposed PDEs. In Section 3, various edge-strength functions that can be considered are introduced, and a method for extracting skeletons from pseudo-distance maps is also introduced. Section 4 shows the results obtained by applying our method to a variety of synthetic and real images. Conclusions are given in Section 5.

2. Pseudo-Distance Map

2.1. Euclidean distance map

Since our work is motivated by the conventional Euclidean distance map, we feel the necessity to explain it briefly before proceeding with our own work. A Euclidean distance map (EDM) is defined as an image where each pixel is assigned the value of the distance to the nearest object boundary pixel. The importance of the EDM is the fact that its local maxima (i.e., ridge pixels) correspond to the skeleton of the object from which the map originates.

Worthy of note is the connection between the EDM and the eikonal equation. Consider a curve evolving with the PDE

$$\frac{\partial \mathbf{C}}{\partial t} = F(x,y)\mathbf{N}, \tag{1}$$

where $\mathbf{C}$ is a parameterized representation of the curve, $\mathbf{N}$ is the unit inward (or outward) normal to the curve, and F $(F > 0)$ is a speed function which depends only on the position (x, y). If $T(x, y)$ represents the time at which the curve crosses a point (x, y), it can be shown that $T(x, y)$ satisfies the *eikonal equation* [7, 13]

$$\|\nabla T\| F = 1, \quad T = 0 \text{ on } \Gamma, \tag{2}$$

where Γ is the initial location of the curve. The connection between the EDM and the eikonal equation is the fact that the EDM can be interpreted as the solution T of Equation (2) with $F = 1$ if Γ corresponds to the object boundary. There are several methods for solving Equation (2), which include, for example, Sethian's fast marching method [13]. But what if the boundary location is not exactly given and what is given to us is the strength of the boundary at each pixel position? Can we compute something similar to T in such a situation? This question is the motivation of our work.

2.2. Computing a pseudo-distance map in one dimension

We begin by formulating an energy functional that will be minimized to obtain a pseudo-distance map (PDM) in one dimension. The functional will be extended to the two dimensional case in Subsection 2.3. Assuming that an edge-strength function $v(x)$ is given, which ranges from 0 to 1 and monotonically increases as an edge at x gets stronger, the functional is given by

$$E(f) = \int \underbrace{\alpha v[f^2 + f_x^2 + (b - f_{xx})^2]}_{(1)} + \underbrace{\beta(a^2 - f_x^2)^2}_{(2)}\, dx, \tag{3}$$

where α, β, a, and b are positive constants and f is a PDM to be computed. The functional works as follows. If v is small, the minimization of E will be dominated by the term (2), which means that the slope of f will approach a or $-a$. If v is large, the term (1) will also affect the minimization of E and the constraints on f will work; that is, the magnitudes of f and f_x should become zero and f_{xx} should approach the positive constant b. These constraints make f have local minima of nearly zero where v is large. Therefore, the overall shape of f will become similar to that of an EDM after the minimization of E.

The function f that minimizes the functional E can be computed by the *variational method* if the initial form of f is appropriately given. In the variational method, PDEs corresponding to a given energy functional are obtained using the calculus of variations [14] and the PDEs are discretized to compute their numerical solutions. Note that f_{xx} in Equation (3) causes the fourth-order derivative of f in the corresponding PDE, which tends to make the PDE noise-sensitive. To avoid the use of f_{xx}, we introduce a new function g which approximates f_x [15] and incorporate it with Equation (3):

$$\begin{aligned} E(f,g) &= \int \alpha v[f^2 + f_x^2 + (b-g_x)^2] + \beta(a^2 - f_x^2)^2 \\ &+ (g-f_x)^2 \, dx. \end{aligned} \tag{4}$$

Equation (4) gives rise to two PDEs

$$\begin{aligned} f_t &= \alpha[(vf_x)_x - vf] + 2\beta f_{xx}(3f_x^2 - a^2) \\ &+ (f_{xx} - g_x), \qquad (5) \\ g_t &= \alpha[(vg_x)_x - bv_x] + (f_x - g). \qquad (6) \end{aligned}$$

We implemented the above PDEs using central finite-difference approximations [13] and tested them with the initial value of f set to $1-v$. Unfortunately, according to our experiments, the convergence of the PDE in Equation (5) was not good; that is, it was very slow and often stuck to unwanted local extrema. To speed up the convergence and avoid falling into the local extrema, we added a new term to Equation (5):

$$\begin{aligned} f_t &= \alpha[(vf_x)_x - vf] + 2\beta f_{xx}(3f_x^2 - a^2) \\ &+ (f_{xx} - g_x) + \gamma\phi f_{xx}, \end{aligned} \tag{7}$$

where γ is a positive constant and

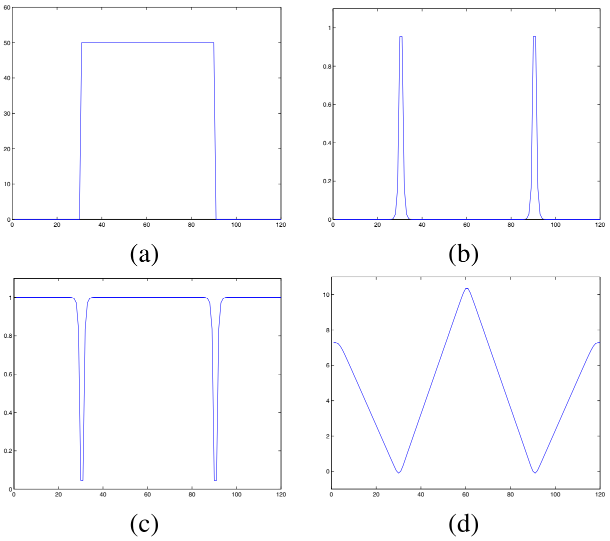
$$\phi = \begin{cases} 1 & \text{if } \{f < 0\} \text{ or } \{f_x \approx 0,\ f_{xx} > 0,\ v \approx 0\}, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

During the iterative computation of f, the new term activates only when $f(x)$ has a negative value at x or has a local minimum where v is very small. The two conditions, in fact, illustrate the cases which should not be allowed during the formation of a PDM. The new term makes f escape from them by smoothing it out where they occur. Our actual implementation of ϕ is as follows:

$$\phi = \begin{cases} 1 & \text{if } \{f < \epsilon_1\} \text{ or } \{|g| < \epsilon_2,\ g_x > \epsilon_3,\ v < \epsilon_4\}, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where $\epsilon_1 \leq 0,\ \epsilon_2 > 0,\ \epsilon_3 > 0,\ \epsilon_4 > 0$, and their absolute values are very small.

An illustrative example is given in Figure 1. The input signal of Figure 1(a) has two step edges at $x = 30$ and 90. The edge-strength function of Figure 1(b) was obtained from the input signal using Ambrosio and Tortorelli's method, whose details are given in Subsection 3.1. The result of Figure 1(d) was obtained by setting $\alpha = 0.1$, $\beta = 1$, $a = 0.35$, $b = 3a$, $\gamma = 3$, $\epsilon_1 = -0.1$, $\epsilon_2 = 0.05$, $\epsilon_3 = 0.01$, $\epsilon_4 = 0.05$, and $\Delta t = 0.1$. Note that we can compute the approximate distance between the two edges from the value of f at the center. In Figure 1(d), f is equal to 10.35 at the center, from which the distance is computed by $10.35/a \times 2 = 59.14$, which is approximately equal to the true distance, 60.



(a) (b)

(c) (d)

Figure 1. (a) Input signal. (b) Edge-strength function v. (c) Initial $f = 1 - v$. (d) f after 60,000 iterations.

At first glance, our method seems to involve many parameters to be adjusted. However, considering the role of each parameter, one can see that the parameters γ, ϵ_1, ϵ_2, ϵ_3, ϵ_4, a, and b are independent of given edge-strength functions. Therefore, once the parameter values selected are proven to be suitable for convergence, they are also suitable for other edge-strength functions. The effect of the parameters α and β will be discussed in Section 4.

2.3. Extension to two dimensions

The extension of Equation (4) to two dimensions is straightforward:

$$\begin{aligned} E(f,g,h) &= \int \alpha v[f^2 + f_x^2 + f_y^2 + (b-g_x)^2 \\ &+ \frac{1}{2}(g_y + h_x)^2 + (b - h_y)^2] \\ &+ \beta(a^2 - \|\nabla f\|^2)^2 \\ &+ (g-f_x)^2 + (h-f_y)^2 \, dxdy, \end{aligned} \tag{10}$$

where

$$g \approx f_x \quad \text{and} \quad h \approx f_y. \tag{11}$$

Consequently,

$$f_{xx} \approx g_x,\ f_{yy} \approx h_y, \text{ and } f_{xy} \approx \frac{1}{2}(g_y + h_x). \tag{12}$$

The corresponding PDEs are

$$\begin{aligned} f_t &= \alpha(\nabla \cdot v\nabla f - vf) \\ &+ 2\beta[(\|\nabla f\|^2 - a^2)\nabla^2 f \\ &+ 2(f_x^2 f_{xx} + 2f_x f_y f_{xy} + f_y^2 f_{yy})] \\ &+ (\nabla^2 f - g_x - h_y), \qquad (13) \\ g_t &= \alpha\{2[v(g_x - b)]_x + [v(g_y + h_x)]_y\} \\ &+ 2(f_x - g), \qquad (14) \\ h_t &= \alpha\{[v(g_y + h_x)]_x + 2[v(h_y - b)]_y\} \\ &+ 2(f_y - h). \qquad (15) \end{aligned}$$

We also added a new term $\gamma\phi\nabla^2 f$ to Equation (13) for faster and reliable convergence, where ϕ is given by

$$\phi = \begin{cases} 1 & \text{if } \{f < \epsilon_1\} \text{ or } \{|\det(\mathbf{H})| < \epsilon_2, \\ & \quad \text{trace}(\mathbf{H}) > \epsilon_3, v < \epsilon_4\}. \\ 0 & \text{otherwise,} \end{cases} \qquad (16)$$

where $\epsilon_1 \leq 0$, $\epsilon_2 > 0$, $\epsilon_3 > 0$, $\epsilon_4 > 0$, and their absolute values are very small. The matrix $\mathbf{H}$ is a Hessian matrix of f which is given by

$$\mathbf{H} = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix} \approx \begin{bmatrix} g_x & \frac{1}{2}(g_y + h_x) \\ \frac{1}{2}(g_y + h_x) & h_y \end{bmatrix}. \qquad (17)$$

The condition $\{|\det(\mathbf{H})| < \epsilon_2,\ \text{trace}(\mathbf{H}) > \epsilon_3\}$ in Equation (16) is provided for detecting valleys (i.e., local minima) of f. Note that two eigenvalues λ_1 and λ_2 ($|\lambda_1| \leq |\lambda_2|$) of a Hessian matrix of $f(x,y)$ correspond respectively to the minimum and maximum second-order directional derivatives (i.e., curvatures) of f at (x,y). Therefore, at the valleys of f, the following condition is usually satisfied:

$$\det(\mathbf{H}) = \lambda_1 \cdot \lambda_2 \approx 0 \text{ and } \text{trace}(\mathbf{H}) = \lambda_1 + \lambda_2 \approx \lambda_2 > 0. \qquad (18)$$

3. Gray-Scale Image Skeletonization

3.1. Edge-strength functions

In this subsection, a variety of edge-strength functions (ESFs) available are introduced. The simplest form would be

$$v = g(\|\nabla G_\sigma * I_0\|), \qquad (19)$$

where G_σ is a Gaussian kernel of size σ, I_0 is an input gray-scale image, and g is a monotonically increasing function which maps $[0,\infty]$ to $[0,1]$. The parameter σ controls the overall scale of the ESF. It can be shown that convolving an image with a Gaussian kernel is equivalent to performing isotropic linear diffusion with an initial state set to I_0. Perona and Malik proposed an anisotropic diffusion equation for edge-preserving smoothing [16]:

$$I_t = \nabla \cdot c(\|\nabla I\|)\nabla I \text{ with } I = I_0 \text{ at } t = 0, \qquad (20)$$

where c is called a conductance function. They suggested two conductance functions

$$\begin{aligned} c(x,y,t) &= \exp\left(-\frac{\|\nabla I(x,y,t)\|^2}{K^2}\right) \text{ and} \qquad (21) \\ c(x,y,t) &= \frac{1}{1 + \|\nabla I(x,y,t)\|^2/K^2}. \qquad (22) \end{aligned}$$

The ESF of $I(x,y,t)$ can be given by

$$v = 1 - c(x,y,t). \qquad (23)$$

Other anisotropic diffusion equations can be found in [17, 18].

Ambrosio and Tortorelli [19] proposed an ESF which is obtained by minimizing the functional

$$\begin{aligned} E(I,v) &= \int \mu(1-v)^2\|\nabla I\|^2 + \eta(I - I_0)^2 \\ &+ \frac{\sigma}{2}\|\nabla v\|^2 + \frac{v^2}{2\sigma}\, dxdy, \qquad (24) \end{aligned}$$

where I is a smoothed image and v is an ESF. The corresponding PDEs are given by

$$\begin{aligned} I_t &= \nabla \cdot (1-v)^2\nabla I - \frac{\eta}{\mu}(I - I_0), \qquad (25) \\ v_t &= \nabla^2 v - \frac{v}{\sigma^2} + \frac{2\mu}{\sigma}(1-v)\|\nabla I\|^2. \qquad (26) \end{aligned}$$

Note that the larger μ/η is, the more details of the ESF are smoothed out. Other ESFs based on a variational approach can be found in [20, 21].

3.2. Skeleton extraction from PDMs

We made use of Steger's ridge detection method to extract a skeleton from a PDM [22]. Steger's method was originally designed for detecting ridges in gray-scale images, where input images are first smoothed with a Gaussian filter whose kernel size depends on the width of ridge structures to be detected. We omitted this step when applying the method to a PDM, since it has almost the same cross-sectional shape (i.e., triangular shape) on top of the ridge structures regardless of their widths.

4. Experimental Results

In this section, we show several examples of ESFs and skeletonization results obtained from synthetic and real images. The ESF of Figure 2(b) was obtained with Ambrosio

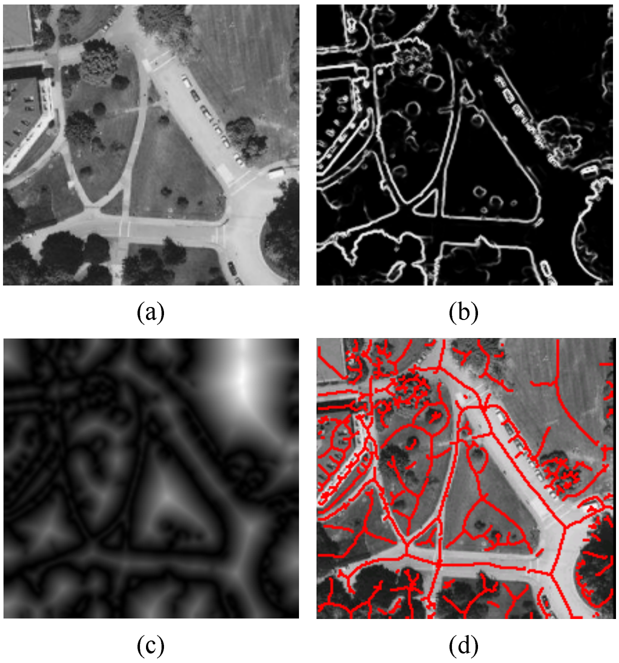


Figure 2. (a) Input image. (b) ESF obtained with Ambrosio and Tortorelli's method. (c) PDM obtained with $\alpha = 0.2$ and $\beta = 1.5$. (d) Ridges extracted from the PDM of (c).

and Tortorelli's method, and the PDM of Figure 2(c) was obtained from the ESF with $\alpha = 0.2$, $\beta = 1.5$, $\gamma = 2.5$, $a = 0.35$, $b = 3a$, $\epsilon_1 = -0.3$, $\epsilon_2 = 0.01$, $\epsilon_3 = 0.01$, $\epsilon_4 = 0.1$, and $\Delta t = 0.05$. From the result of Figure 2(d), one can see that the skeleton was successfully extracted except where two or more structures with big differences in thickness join together. The detection failure is due to the fact that when a very thin structure meets a very thick structure, the ridge strength of the thin structure becomes negligible around the junction, which disables a ridge extractor from detecting the ridges of the thin structure near the junction.

The skeletons of Figures 3(a) and (b) are obtained from the same ESF of Figure 2(b) by setting α to 0.02 and 2.0 respectively with the other parameter values fixed. It is obvious from the results that as α/β gets larger, more detailed structures appear, but resulting skeletons become noisier.

Figure 4(a) shows the skeletonization result of a synthetic character image corrupted by additive Gaussian noise. If we know the thickness of the character strokes in advance, it is easy to extract only the character portions of the skeleton like Figure 4(b) using information on the thickness of skeletonized structures provided by a PDM. Other skeletonization examples are shown in Figure 5.

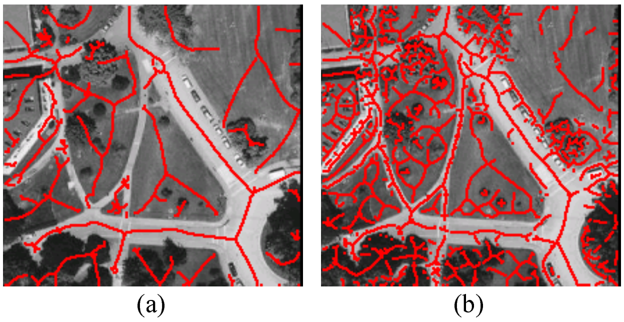


Figure 3. Other skeletons obtained from the same ESF of Figure 2(b) with different values of α: (a) $\alpha = 0.02$. (b) $\alpha = 2.0$.

5. Conclusions

In this paper, we proposed a new method for extracting skeletons from gray-scale images. The heart of our work is a pseudo-distance map, which is directly obtainable from the edge-strength function of a gray-scale image without thresholding it. Therefore, the PDM enables us to avoid the risk caused by the thresholding operation which usually gives rise to the loss of useful information about region boundaries. It also provides information on the width of skeletonized structures. Therefore, it can be said that the PDM contains information about regions as well as their skeletons. The implementation of our method is simple because it can be done only by replacing derivatives in the PDEs with corresponding finite-differences. The main drawback of our method is a long computation time, which makes it difficult to use the method in real-time applications. However, we expect that a parallel-computing machine can reduce the computation time significantly because the computation for solving the PDEs is easily paral-

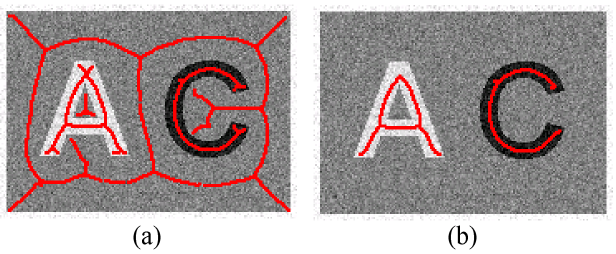


Figure 4. (a) Skeletonization result of a noisy character image. (b) Skeleton of the characters obtained using information on the thickness of the character strokes.

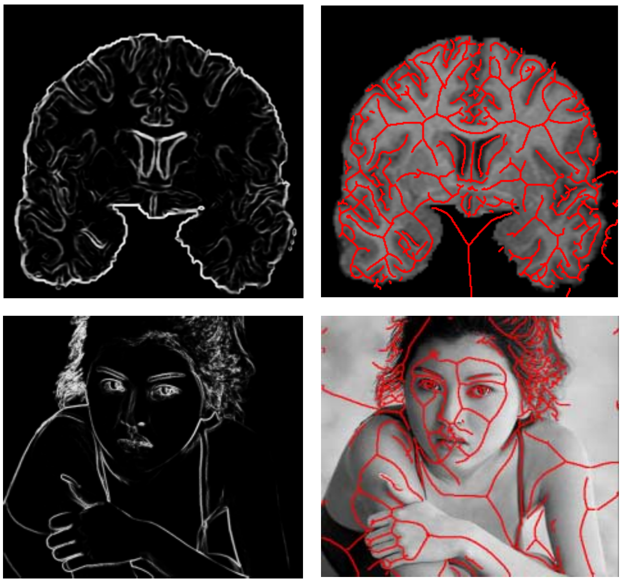


Figure 5. Other skeletonization examples.

lelizable.

References

[1] H. Blum, "A Transformation for Extracting New Descriptions of Shape," *Models for Perception of Speech and Visual Form* (W. Wathen-Dunn, Ed.), MIT Press, pp. 362-380, 1967.

[2] T. Y. Zhang and C. Y. Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns," *Communications of the ACM*, Vol. 27, No. 3, pp. 236-239, 1984.

[3] R. L. Ogniewicz and O. Kübler, "Hierarchic Voronoi Skeletons," *Pattern Recognition*, Vol. 28, No. 3, pp. 343-359, 1995.

[4] C. Arcelli and G. S. di Baja, "Ridge Points in Euclidean Distance Maps," *Pattern Recognition Letters*, Vol. 13, pp. 237-243, 1992.

[5] G. Malandain and S. Fernández-Vidal, "Euclidean Skeletons," *Image and Vision Computing*, Vol. 16, pp. 317-327, 1998.

[6] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "Shapes, Shocks, and Deformations I: The Components of Two-Dimensional Shape and the Reaction-Diffusion Space," *International Journal of Computer Vision*, Vol. 15, pp. 189-224, 1995.

[7] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker, "The Hamilton-Jacobi Skeleton," *Proc. International Conference on Computer Vision*, Vol. 2, pp. 828-834, 1999.

[8] J. H. Jang and K. S. Hong, "Detection of Curvilinear Structures Using the Euclidean Distance Transform," *Proc. IAPR Workshop on Machine Vision Applications*, pp. 102-105, 1998.

[9] Z. S. G. Tari, J. Shah, and H. Pien, "Extraction of Shape Skeletons from Grayscale Images," *Computer Vision and Image Understanding*, Vol. 66, No. 2, pp. 133-146, 1997.

[10] S. M. Pizer, D. Eberly, and D. S. Fritsch, "Zoom-Invariant Vision of Figural Shape: The Mathematics of Cores," *Computer Vision and Image Understanding*, Vol. 69, No. 1, pp. 55-71, 1998.

[11] T. Lindeberg, "Edge Detection and Ridge Detection with Automatic Scale Selection," *International Journal of Computer Vision*, Vol. 30, No. 2, pp. 117-154, 1998.

[12] D. H. Chung and G. Sapiro, "Segmentation-Free Skeletonization of Gray-Scale Images via PDE's," *Proc. International Conference on Image Processing*, Vol. 2, pp. 927-930, 2000.

[13] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 1999.

[14] B. K. P. Horn, *Robot Vision*, MIT Press, pp. 469-474, 1986.

[15] M. Proesmans, E. Pauwels, and L. van Gool, "Coupled Geometry-Driven Diffusion Equations for Low-Level Vision," *Geometry-Driven Diffusion in Computer Vision* (B. M. ter Haar Romeny, Ed.), Kluwer Academic Publishers, pp. 191-228, 1994.

[16] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 7, pp. 629-639, 1990.

[17] B. Jähne, H. Hauβecker, and P. Geiβler, *Handbook of Computer Vision and Applications*, Academic Press, Vol. 2, pp. 423-450, 1999.

[18] A. Bovik (Ed.), *Handbook of Image and Video Processing*, Academic Press, pp. 433-447, 2000.

[19] L. Ambrosio and V. M. Tortorelli, "On the Approximation of Functionals Depending on Jumps by Quadratic, Elliptic Functionals," *Boll. Un. Mat. Ital.*, 1992.

[20] J. Shah, "A Common Framework for Curve Evolution, Segmentation and Anisotropic Diffusion," *Proc. Computer Vision and Pattern Recognition*, pp. 136-142, 1996.

[21] S. Teboul, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Variational Approach for Edge-Preserving Regularization Using Coupled PDE's," *IEEE Transactions on Image Processing*, Vol. 7, No. 3, pp. 387-397, 1998.

[22] C. Steger, "An Unbiased Detector of Curvilinear Structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 2, pp. 113-125, 1998.