INT. J. REMOTE SENSING, 10 MARCH, 2004,
VOL. 25, NO. 5, 937–958

Automated extraction of coastline from satellite imagery by integrating Canny edge detection and locally adaptive thresholding methods

H. LIU

Department of Geography, Texas A&M University, College Station, Texas 77843, USA; e-mail: liu@geog.tamu.edu

and K. C. JEZEK

Byrd Polar Research Center, The Ohio State University, Columbus, Ohio 43210, USA

(*Received 15 April 2002; in final form 25 March 2003*)

Abstract. This paper presents a comprehensive approach to effectively and accurately extract coastlines from satellite imagery. It consists of a sequence of image processing algorithms, in which the key component is image segmentation based on a locally adaptive thresholding technique. Several technical innovations have been made to improve the accuracy and efficiency for determining the land/water boundaries. The use of the Levenberg-Marquardt method and the Canny edge detector speeds up the convergence of iterative Gaussian curve fitting process and improves the accuracy of the bimodal Gaussian parameters. The result is increased reliability of local thresholds for image segmentation. A series of further image processing steps are applied to the segmented images. Particularly, grouping and labelling contiguous image regions into individual image objects enables us to utilize heuristic human knowledge about the size and continuity of the land and ocean masses to discriminate the true coastline from other object boundaries. The final product of our processing chain is a vector-based line coverage of the coastline, which can be readily incorporated into a GIS database. Our method has been applied to both radar and optical satellite images, and the positional precision of the resulting coastline is measured at the pixel level.

1. Introduction

A coastline is the boundary between land and ocean masses. Knowledge of coastline is the basis for measuring and characterizing land and water resources, such as the area of the land, and the perimeter of coastline. Information about coastline position, orientation and geometric shape is also essential for autonomous navigation, geographical exploration, coastal erosion monitoring and modelling, and coastal resource inventory and management. Conventionally, coastlines were manually identified and traced by cartographers with a pencil on vellum paper overlaying on aerial photographs. Due to the subjectivity and substantial effort involved in manual delineation, an automatic delineation method has been long desired.

Automated coastline extraction from digital image data belongs to the boundary

International Journal of Remote Sensing
ISSN 0143-1161 print/ISSN 1366-5901 online © 2004 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/0143116031000139890

detection problem in the field of computer vision and image processing, in which edge detection and image segmentation are two conventional approaches to the boundary detection. These dual approaches are based on two fundamental observations (Gonzalez and Woods 1992, Pitas 2000): (1) discontinuity: image intensity (grey) values at or near boundaries change abruptly; and (2) similarity: boundaries are located between two relatively homogeneous regions, each with different average intensity values. Technically, edge detection methods emphasize the first property and locate the meaningful intensity discontinuity by using spatial differentiation or edge template operations. Segmentation is conceptually based on the second property. Commonly used segmentation algorithms include thresholding, region growing, and region splitting and merging (Gonzalez and Woods 1992, Parker 1997). Edge detection is relatively simpler to implement than segmentation. However, edge detection methods suffer from the fact that the edge pixels produced by edge detectors are quite discontinuous and seldom characterize a coastline completely (Lee and Jurkevich 1990, Mason and Davenport 1996). To assemble and link edge pixels separated by small breaks is computationally very intensive, even in a crude approximation (Ballard and Brown 1982, Lee and Jurkevich 1990, Pitas 2000). In contrast, image segmentation methods have the advantage in creating a continuous boundary. But, segmentation methods require more post-segmentation processing steps to delineate the boundary pixels, and face difficulties of determining a reliable threshold in thresholding algorithms and formulating homogeneous criteria in region growing, and region splitting and merging algorithms.

Because of the frequent lack of consistent, sufficient intensity contrast between land and water regions and the complexity in distinguishing coastline edges from other object edges, most general-purpose edge detection and image segmentation techniques are inadequate for a coastline extraction task. It is recognized that a comprehensive procedure is required to automate the coastline extraction process (Lee and Jurkevich 1990, Sohn and Jezek 1999). Lee and Jurkevich (1990) presented a coastline extraction technique based on an edge detection algorithm. They employed the Sobel edge operator and created a preliminary edge image for a small portion (512 × 512 pixels) of a SEASAT synthetic aperture radar (SAR) image. To link edge gaps in the derived coastline, they applied a 5 × 5 mean filter twice on the edge image, and subsequently traced the edge pixels. Due to the use of a 5 × 5 mean filter on the edge image, the positional accuracy of the extracted coastline is limited for the purpose of cartographic mapping, as they acknowledged (Lee and Jurkevich 1990). Ryan *et al.* (1991) used an image segmentation method to approach the coastline extraction problem and tested their method on several small portions (256 × 256) of scanned US Geological Survey (USGS) aerial photographs. They applied a neural network method on a texture measure of the images to separate land and water regions. The use of texture influenced the accuracy, and the need for training neural nets also limited its application to a large volume of full-scene image data. Mason and Davenport (1996) employed an edge detection method with a coarse–fine resolution processing strategy and applied their approach to several full-scene European Remote Sensing Satellite (ERS)-1 SAR images. A contrast ratio edge detector (Touzi *et al.* 1988, Mason and Davenport 1996) was used to determine shoreline pixels that have strong edge strength. The edge gaps along the coastline are filled with the use of an interpolation procedure based on active contour models (Kass *et al.* 1987, Cohen 1991, Williams and Shah 1992). Due to the difficulty in setting a global contrast ratio threshold to define coastline edge segments, and the limitations in the active contour modelling to fill the edge gaps, the delineation accuracy of their method is also limited. Sohn and Jezek (1999) presented an alternative approach to automatically mapping the coastline. The key component of their method is image segmentation based on a locally adaptive thresholding algorithm, which was first proposed by Chow and Kaneko (1972) for detecting the boundaries of the left ventricle in a heart image and later employed by Haverkamp *et al.* (1995) to classify sea ice. Sohn and Jezek (1999) applied their coastline extraction approach to both ERS-1 SAR and Satellite Probatoire d'Observation de la Terre (SPOT) images and achieved an adequate accuracy for extracting ice margins for Greenland.

The present research aims to provide a comprehensive algorithmic foundation for extracting a vector-based representation of coastline from image data with minimized human intervention. As in Chow and Kaneko (1972), Haverkamp *et al.* (1995) and Sohn and Jezek (1999), we employ the locally adaptive thresholding algorithm to perform the image segmentation. We also make use of the fact that ocean and land masses are usually large connected regions, as in Ryan *et al.* (1991) and Mason and Davenport (1996). By exploiting the best algorithmic elements of previous studies, we developed an effective, operational method for coastline extraction, with several important technical innovations. The final product is a cartographic line coverage of the coastline, which can be readily incorporated into a Geographical Information System (GIS) database. Our method has been implemented using the C-programming language and applied to both radar and optical satellite images. Visual comparison between the extracted coastlines and the original satellite images shows that the positional precision of the resulting coastline is measured to pixel level.

In the following sections, we will first outline our coastline extraction method. Then, we will describe major image processing algorithms, with the emphasis on our technical improvements. Next, we will demonstrate the application results on both radar and optical satellite images. In the final section, we will present some technical remarks and conclusions.

2. Methodology overview

Our coastline extraction method consists of three groups of image processing algorithms: pre-segmentation, segmentation and post-segmentation. The pre-segmentation processing algorithms aim to suppress image noise and enhance edge elements in the images. The segmentation algorithms partition the input image into homogeneous land and water regions using a locally adaptive threshold. The post-segmentation processing algorithms are designed to differentiate the coastline edges from other object edges, and trace the coastline edge pixels into a vector representation. Our objective is to derive a coastline with a precise geographical location and reliable geometric shape. As the geographical coordinates of the derived coastline are inherited from the source satellite imagery, the input images must be geocoded to assign precise geographical coordinates to image pixels and rectified to remove geometric and terrain distortions in the imagery before applying our coastline extraction method. The geocoding and orthorectification of the source images are beyond the scope of this paper. In our analysis, we assume that the input image has been geo-referenced and terrain-corrected. We also assume that the input image is a panchromatic or single band grey-scale image whose intensity value ranges between 0 and 255. For the sake of clarity, a flow chart is given in figure 1 to outline our algorithms and image processing chain.

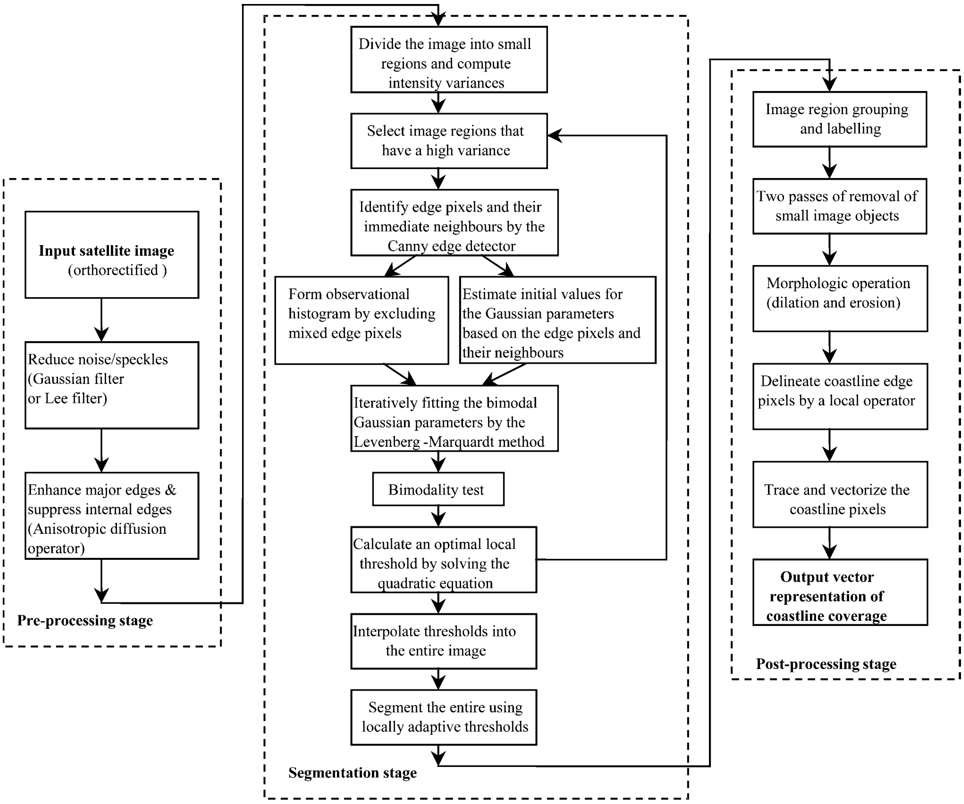

Input satellite image (orthorectified)
Reduce noise/speckles (Gaussian filter or Lee filter)
Enhance major edges & suppress internal edges (Anisotropic diffusion operator)
Pre-processing stage
Divide the image into small regions and compute intensity variances
Select image regions that have a high variance
Identify edge pixels and their immediate neighbours by the Canny edge detector
Form observational histogram by excluding mixed edge pixels
Estimate initial values for the Gaussian parameters based on the edge pixels and their neighbours
Iteratively fitting the bimodal Gaussian parameters by the Levenberg -Marquardt method
Bimodality test
Calculate an optimal local threshold by solving the quadratic equation
Interpolate thresholds into the entire image
Segment the entire using locally adaptive thresholds
Segmentation stage
Image region grouping and labelling
Two passes of removal of small image objects
Morphologic operation (dilation and erosion)
Delineate coastline edge pixels by a local operator
Trace and vectorize the coastline pixels
Output vector representation of coastline coverage
Post-processing stage

Figure 1. Diagram for image processing chain and data flow.

3. Pre-segmentation processing algorithms

There are two primary purposes for preprocessing the image before performing segmentation. The first purpose is to filter the image noise and speckle so as to reduce isolated and noisy edges in the subsequent image segmentation. The second is to enhance the edges along the coastline while suppressing other unimportant edges inside land or ocean mass.

3.1. *Noise removal*

To preserve the precise position of the coastline, an edge-preserving operator is required to remove the image noise. A Gaussian filter is able to filter optical images without blurring the major edge features. The Lee filter (Lee 1986) is capable of reducing radar noise and speckle without degrading the sharpness of the edges. Other filters, such as the Frost filter (Frost *et al.* 1982), Kuan filter (Kuan *et al.* 1985, 1987), Median filter (Rees and Stachell 1997) and Gamma MAP (Maximum a Posteriori) filter (Oliver and Quegan 1998), may be used to replace the Lee filter for suppressing radar speckle while preserving edges.

3.2. *Edge enhancement*

In the preprocessing stage, we also utilize an anisotropic diffusion algorithm (Perona and Malik 1990, Saint-Marc *et al.* 1991, Sohn and Jezek 1999) to enhance strong edges along the coastline and suppress the weak edges and interior variations inside the land or ocean masses.

Figure 2 shows the effect of the pre-segmentation processing. After applying the Lee filter and anisotropic diffusion operation to a SAR image, the image noise and unwanted weak edges are reduced, and strong edges along the coastline are enhanced.

4. Image segmentation using locally adaptive thresholding method

The purpose of image segmentation is to separate the image into its constituent homogeneous regions. The border pixels between segmented land/water regions can then be delineated as the coastlines. In order to reliably separate land objects from the ocean background, a locally adaptive thresholding algorithm is used for image segmentation. If a single global threshold is used for the entire image to determine the land/water boundaries, some local coastline edges will remain undetected due to the heterogeneity of the image intensity contrast, causing the discontinuity of coastline edges in low contrast areas in the image. Our thresholding method sets the threshold value dynamically according to the local characteristics to achieve a good separation between the land and ocean water.

4.1. *Analytically determining local thresholds by fitting a bimodal Gaussian curve*

In our analysis, we divide the entire image into a set of small, overlapping regions. The image regions are a square whose width (w) is a controllable parameter. The value of w should be small enough so that only one or two categories of image pixels exist in the small region. It also should be big enough to ensure reliable statistical analysis of the histogram of the region. The adjustment of

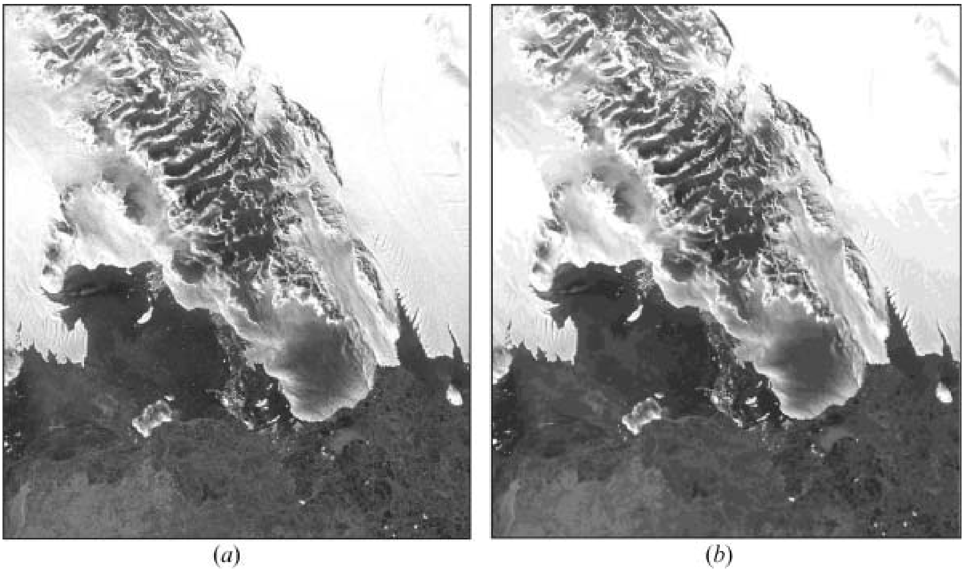


Figure 2. The effect of pre-segmentation processing: (*a*) before applying the Lee filter and anisotropic diffusion operation; and (*b*) after applying the Lee filter and anisotropic diffusion.

this parameter is helpful for tuning the method to a particular coastal environment or to a particular type of image source. Our extensive experiments show that it is appropriate to set it to a value between 25 and 75 pixels. There is 50% overlapping between adjacent image regions. Let *row* and *col* denote the row and column number of the image, then the number of the resulting regions will be *2*col/w* in the horizontal direction and *2*row/w* in the vertical direction. For each small region, we examine the bimodality and analytically determine a local threshold value to separate the land (object) pixels from the water (background) pixels.

If a small region of the image consists solely of the land or the water pixels, the probability distribution of the intensity values will be unimodal. However, if the image region contains a coastline, it implies that the image consists of both land and water pixels. Consequently, intensity values of land pixels and water pixels will be grouped into two dominant modes (lobes) with relatively distinct mean values in the image histogram. The two component distributions commonly overlap each other. As shown in figure 3, the overall histogram for this region generally exhibits two peaks and a valley, namely, a bimodal shape. The problem of determining the boundary reduces to that of assigning each image pixel to a particular component distribution based on optimal threshold value corresponding to the local histogram minimum (the valley point). Given the histogram valley point, the image pixels in the small region can be reliably classified into land and ocean pixels.

To automate the determination of the threshold value, we need to model the bimodal histogram and computationally derive the valley point for each image region, rather than visually picking the valley point from the shape property of the histogram. Our assumption is that the bimodal distribution is a mixture of two Gaussian (normal) distribution functions (figure 3) (Haverkamp *et al.* 1995, Sohn and Jezek 1999). The intensity values of the image pixels are regarded as random numbers drawn from one of two normal distributions. Each component Gaussian (normal) distribution can be statistically defined by two parameters: mean and standard deviation. A mixture of two normal distributions $\phi_1(x)$ and $\phi_2(x)$ has a

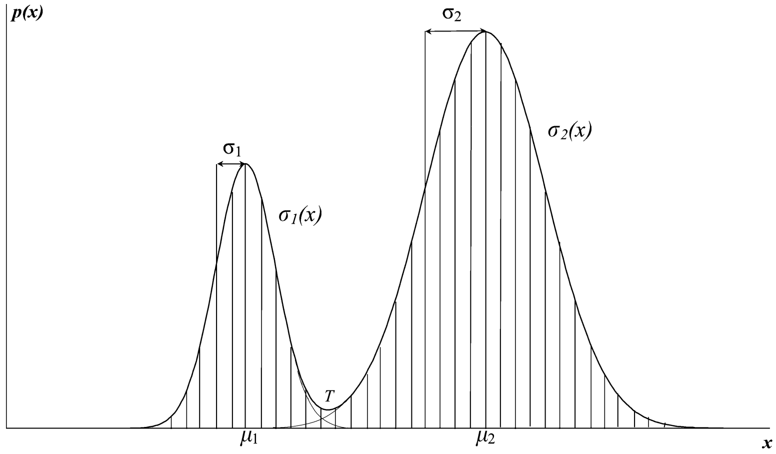


Figure 3. Two components of the bimodal Gaussian curve.

probability density function $p(x)$ given by the following equations:

$$\phi_1(x) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(x-\mu_1)^2}{2\sigma_1^2} \right] \tag{1}$$

$$\phi_2(x) = \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(x-\mu_2)^2}{2\sigma_2^2} \right] \tag{2}$$

$$p(x) = p_1\phi_1(x) + p_2\phi_2(x) \tag{3}$$

$$p_1 + p_2 = 1 \tag{4}$$

$$p(x) = \frac{p_1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(x-\mu_1)^2}{2\sigma_1^2} \right] + \frac{1-p_1}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(x-\mu_2)^2}{2\sigma_2^2} \right] \tag{5}$$

where μ_1 and μ_2 are the mean values of two component normal distributions, σ_1 and σ_2 are the standard deviations about the means, and p_1 is the coefficient of the mixture, representing the theoretical fraction of area occupied by water (background) pixels in the image region. As shown in equation (5), the mixture density function is characterized by five unknown parameters. If these five parameters are determined, the optimal threshold (the valley point in the histogram) can be analytically determined by solving a quadratic equation.

4.2. *Using the Lenvenberg-Margardt method to iteratively fit the Gaussian curve*

With an observed histogram $h(i)$ for each image region, the five parameters can be calculated. Mathematically, it is a problem of nonlinear function minimization, in which the parameters of the model are adjusted to yield best-fit parameters. A merit function is defined to measure the agreement between the observed histogram and the bimodal Gaussian curve with a particular choice of parameter values. In the least-squares fit method, the sum of squares errors χ^2 is used as the merit function:

$$\chi^2(p_1,\mu_1,\sigma_1,\mu_2,\sigma_2) = \sum_{i=0}^{255} \left[\frac{p_1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(i-\mu_1)^2}{2\sigma_1^2} \right] + \frac{1-p_1}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(i-\mu_2)^2}{2\sigma_2^2} \right] - h(i) \right]^2 \tag{6}$$

Optimal estimates for the five parameters of the bimodal Gaussian curve minimize the merit function χ^2.

The nonlinear minimization problem needs to be approached iteratively. Chow and Kaneko (1972) used a hill climbing method employing the conjugate gradients. This method is unstable and fails to converge upon a solution in some situations (Chow and Kaneko 1972, Press *et al.* 1992). Haverkamp *et al.* (1995) and Sohn and Jezek (1999) used a stochastic approximation algorithm (Kashyap and Blaydon 1968, Young and Coraluppi 1970) for fitting the Gaussian parameters. The stochastic approximation algorithm has a slow convergence rate (Young and Coraluppi 1970). In our implementation, we introduce the Levenberg-Marquardt method (Press *et al.* 1992), which is an elegant combination of the inverse Hessian matrix (the second derivative matrix) algorithm and the steepest descent algorithm. When the parameter estimations are far from the minimum, the steepest descent algorithm is used. As the minimum is approached, it smoothly switches to the inverse Hessian matrix algorithm. By exploiting the advantages of both the steepest descent method and the inverse-Hessian matrix method, the Levenberg-Marquardt method makes the fitting process rapidly converge to a reliable solution.

We derived the first and second partial derivatives of the merit function $\chi^2(p_1,\mu_1,\sigma_1,\mu_2,\sigma_2)$ with respect to each of the five parameters for specifying the components of the Hessian matrix in the minimization algorithm. According to the Levenberg-Marquardt method, we developed the iterative procedure for fitting bimodal Gaussian parameters. One round of iteration is described as follows:

1. Given the observed histogram $h(i)$ and initial estimates for the five parameters $\mu_1(0)$, $\mu_2(0)$, $\sigma_1(0)$, $\sigma_2(0)$, and $p_1(0)$, compute $\chi^2(p_1(0),\mu_1(0),\sigma_1(0),\mu_2(0), \sigma_2(0))$ using equation (6).
2. Set λ to a modest value of 0.001.
3. Solve the simultaneous linear equations (7) for Δp_1, $\Delta\mu_1$, $\Delta\sigma_1$, $\Delta\mu_2$ and $\Delta\sigma_2$ by the Gauss-Jordan elimination method (Press *et al.* 1992).

$$\begin{cases} (1+\lambda)\frac{\partial^2\chi^2}{\partial p_1\partial p_1}\Delta p_1+\frac{\partial^2\chi^2}{\partial\mu_1\partial p_1}\Delta\mu_1+\frac{\partial^2\chi^2}{\partial\sigma_1\partial p_1}\Delta\sigma_1+\frac{\partial^2\chi^2}{\partial\mu_2\partial p_1}\Delta\mu_2+\frac{\partial^2\chi^2}{\partial\sigma_2\partial p_1}\Delta\sigma_2=-\frac{\partial\chi^2}{\partial p_1} \\ \frac{\partial^2\chi^2}{\partial p_1\partial\mu_1}\Delta p_1+(1+\lambda)\frac{\partial^2\chi^2}{\partial\mu_1\partial\mu_1}\Delta\mu_1+\frac{\partial^2\chi^2}{\partial\sigma_1\partial\mu_1}\Delta\sigma_1+\frac{\partial^2\chi^2}{\partial\mu_2\partial\mu_1}\Delta\mu_2+\frac{\partial^2\chi^2}{\partial\sigma_2\partial\mu_1}\Delta\sigma_2=-\frac{\partial\chi^2}{\partial\mu_1} \\ \frac{\partial^2\chi^2}{\partial p_1\partial\sigma_1}\Delta p_1+\frac{\partial^2\chi^2}{\partial\mu_1\partial\sigma_1}\Delta\mu_1+(1+\lambda)\frac{\partial^2\chi^2}{\partial\sigma_1\partial\sigma_1}\Delta\sigma_1+\frac{\partial^2\chi^2}{\partial\mu_2\partial\sigma_1}\Delta\mu_2+\frac{\partial^2\chi^2}{\partial\sigma_2\partial\sigma_1}\Delta\sigma_2=-\frac{\partial\chi^2}{\partial\sigma_1} \\ \frac{\partial^2\chi^2}{\partial p_1\partial\mu_2}\Delta p_1+\frac{\partial^2\chi^2}{\partial\mu_1\partial\mu_2}\Delta\mu_1+\frac{\partial^2\chi^2}{\partial\sigma_1\partial\mu_2}\Delta\sigma_1+(1+\lambda)\frac{\partial^2\chi^2}{\partial\mu_2\partial\mu_2}\Delta\mu_2+\frac{\partial^2\chi^2}{\partial\sigma_2\partial\mu_2}\Delta\sigma_2=-\frac{\partial\chi^2}{\partial\mu_2} \\ \frac{\partial^2\chi^2}{\partial p_1\partial\sigma_2}\Delta p_1+\frac{\partial^2\chi^2}{\partial\mu_1\partial\sigma_2}\Delta\mu_1+\frac{\partial^2\chi^2}{\partial\sigma_1\partial\sigma_2}\Delta\sigma_1+\frac{\partial^2\chi^2}{\partial\mu_2\partial\sigma_2}\Delta\mu_2+(1+\lambda)\frac{\partial^2\chi^2}{\partial\sigma_2\partial\sigma_2}\Delta\sigma_2=-\frac{\partial\chi^2}{\partial\sigma_2} \end{cases} \quad (7)$$

4. Evaluate $\chi^2(p_1(0)+\Delta p_1,\mu_1(0)+\Delta\mu_1,\sigma_1(0)+\Delta\sigma_1,\mu_2(0)+\Delta\mu_2,\sigma_2(0)+\Delta\sigma_2)$ using equation (6).
5. If $\chi^2(p_1(0)+\Delta p_1,\mu_1(0)+\Delta\mu_1,\sigma_1(0)+\Delta\sigma_1,\mu_2(0)+\Delta\mu_2,\sigma_2(0)+\Delta\sigma_2)\geqslant\chi^2(p_1(0),\mu_1(0),$ $\sigma_1(0),\mu_2(0),\sigma_2(0))$, increase λ by a factor of 10 and go back to step 3. If $\chi^2(p_1(0)+\Delta p_1,\mu_1(0)+\Delta\mu_1,\sigma_1(0)+\Delta\sigma_1,\mu_2(0)+\Delta\mu_2,\sigma_2(0)+\Delta\sigma_2)<\chi^2(p_1(0),\mu_1(0),$ $\sigma_1(0),\mu_2(0),\sigma_2(0))$, decrease λ by a factor of 10, and then update the initial estimates, namely, replace $p_1(0),\mu_1(0),\sigma_1(0),\mu_2(0),\sigma_2(0)$ by $p_1(0)+\Delta p_1,\mu_1(0)+$ $\Delta\mu_1,\sigma_1(0)+\Delta\sigma_1,\mu_2(0)+\Delta\mu_2,\sigma_2(0)+\Delta\sigma_2$. Then go back to step 3 and perform the next round of iteration for refining the estimates for the five parameters.

The parameter λ controls the switching between the steepest descent algorithm and the inverse Hessian matrix algorithm. When λ is large, the steepest descent algorithm is used. When λ approaches 0, the inverse Hessian matrix algorithm is used. The convergence condition for stopping the iteration is that a change in the parameters only decreases χ^2 by a negligible amount specified by the user. Once the convergence condition is satisfied, the final fitted parameters can be computed by setting $\lambda=0$ and inversing the matrix consisting of the coefficients of the left side of equation (7). The covariance matrix of the standard errors can also be calculated at this stage for analysing the goodness-of-fit for each parameter.

4.3. *Integrating the Canny edge detector to improve the estimation of Gaussian parameters*

As described above, the bimodal Gaussian curve fitting process is based on the observed histogram of each small image region. To determine an optimal threshold, the most important aspect is to reliably identify the valley point in the observed histogram (figure 3). It is intuitively evident that the chance of obtaining a good threshold should be considerably enhanced if the two peaks of the observed histogram are tall, narrow and symmetrical, and if these peaks are clearly separated by a deep valley. It is the case if the land region and water region in the image are homogeneous and have distinctive intensity values. However, in many cases the intensity values of the land region and the water region may slowly vary or may contain image noise and speckle. As a result of sampling or limited resolution of sensor, the edge pixels at or near the boundary between land and water regions often appear as a transitional, mixed zone, which may extend across some number of pixels. Owing to the contribution of the mixed pixels, the separation of histogram lobes will be blurred. This adversely influences the calibration of bimodal Gaussian parameters, therefore giving an unreliable estimate for the local threshold (Ahuja and Rosenfeld 1978, Rosenfeld and Davis 1978, Weszka and Rosenfeld 1979, Narayanan and Rosenfeld 1981).

To achieve an observed histogram with two distinguished lobes, the Canny edge detector (Canny 1986, Parker 1997) is employed to identify the mixed (edge) pixels in the image region. The Canny edge detector works in a multi-stage process. First, the image is smoothed by Gaussian convolution, then two-dimensional first derivatives are computed, the gradient magnitude (edge strength) and gradient direction are calculated. The first-order derivative of an image $f(x,y)$ at location (x,y) is defined as the two-dimensional vector:

$$G[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{8}$$

The absolute gradient magnitude is given by:

$$|G[f(x,y)]| = \left| G_x^2 + G_y^2 \right|^{1/2} \tag{9}$$

The gradient direction (edge orientation) is defined as:

$$\theta = \arctan \frac{G_y}{G_x} - \frac{3}{4}\pi \tag{10}$$

Next, non-maximal suppression process is applied to the gradient magnitude (edge strength) image to identify the local maxima. Only pixels with edge strength larger than their two adjacent pixels in the gradient direction are identified as edge candidates and others are set to zero. Non-maximal suppression results in one-pixel wide edge segments. To remove false edge segments caused by noise and fine texture, a hysteresis tracking process is further applied with two thresholds in which all candidate edge pixels below the lower threshold are labelled as non-edges and all pixels above the low threshold that can be connected to any pixels above the high threshold through a chain of edge pixels are labelled as edge pixels.

Our purpose in using the Canny edge detector is to remove the mixed pixels from the formation of the observation histogram. By expanding (dilating) the edge pixels identified by the Canny edge detector into their immediate neighbourhood, a transitional buffer zone is created around the edge pixels. All pixels inside this buffer zone are excluded from the formation of the observed histogram. As demonstrated in figure 4, removal of mixed, transitional edge pixels from the observed histogram formation has effectively sharpened the valley between the peaks of the histogram. This ensures that local thresholds could be more accurately determined.

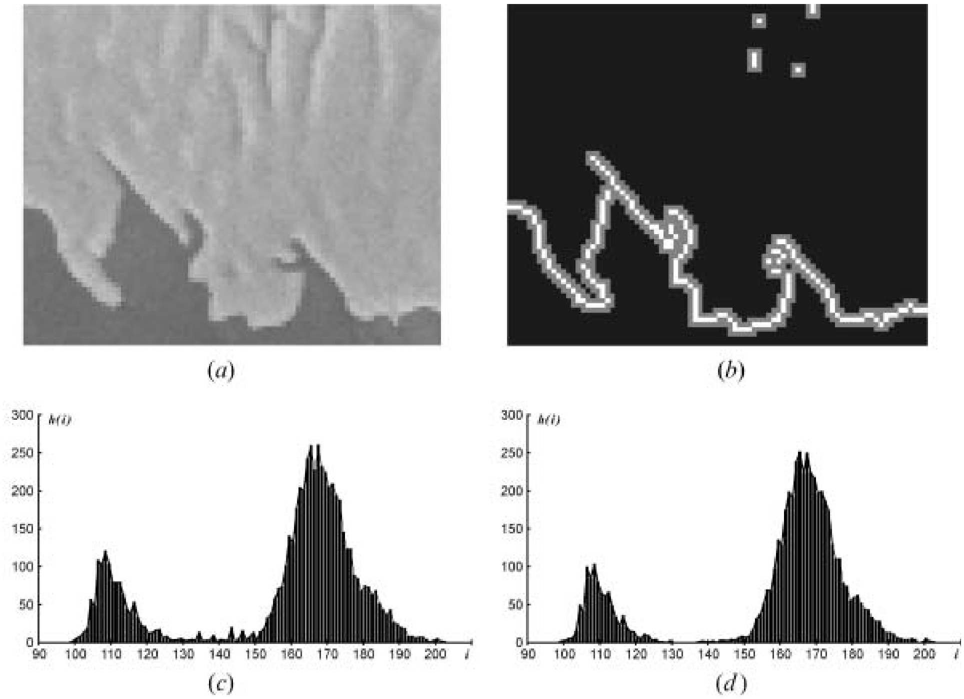


Figure 4. Sharpen observed histogram by removing mixed edge pixels: (*a*) a sample image region; (*b*) edge pixels (white) and their immediate neighbours (grey) identified by the Canny edge detector; (*c*) image histogram of the original image region with spikes in the valley between two lobes; and (*d*) sharpened image histogram after removing the pixels in the transitional buffer zone in (*b*).

The observed histogram can be further smoothed using a one-dimensional low-pass filter:

$$h_s(i) = \frac{h(i-2)+2h(i-1)+3h(i)+2h(i+1)+h(i+2)}{9} \tag{11}$$

The smoothed version $h_s(i)$ is less susceptible to image noise than is the original $h(i)$.

4.4. *Using the Canny edge detector to provide a reliable basis for generating initial values for Gaussian parameters*

Fitting a nonlinear Gaussian curve on the observed histogram is an ill-conditioned minimization problem, and the global minimization is not guaranteed (Chow and Kaneko 1972, Press *et al.* 1992). However, it can be accomplished if reasonably good initial values for the five parameters are used. In Chow and Kaneko (1972), Haverkamp *et al.* (1995) and Sohn and Jezek (1999), the initial values of the five parameters were estimated based on the global mean of the image region. The histogram of the image region was dissected into two parts using the global mean value as the dividing point. Then, the mean and standard deviation were calculated for each part of the histogram as the initial estimates for two component distributions. This procedure is appropriate only if the image region is equally divided by land pixels and water pixels. However, in most cases the land area is smaller or larger than the water area in an arbitrarily selected image region.

In this situation, their procedure will generate biased, unreliable initial estimates for the parameters, slowing the convergence of iterative fitting process or even failing to converge to a solution.

To create reliable initial values for the Gaussian parameters, we developed a new procedure based on the transitional buffer zone identified by the Canny edge detector. As the immediate neighbours on both sides of the edge pixels are included, the buffer zone contains an approximately equal number of nearby land and water pixels as well as transitional edge pixels (figure 4(*b*)). We use the mean value of this buffer zone, instead of the entire image region, as the dividing point to separate the histogram of the image region into two parts. The mean and standard deviation are calculated for each part of the histogram as the initial estimates for $\mu_1(0)$, $\sigma_1(0)$, $\mu_2(0)$, $\sigma_2(0)$, and the fraction of the pixels in the first part of the histogram is used as the initial value for the coefficient of mixture $p_1(0)$. Our procedure is independent of the relative size between land (object) and water (background) areas in the image region, and creates much more reliable initial estimates than the previously used method. As shown in figure 5(*a*), the conventional method produced very poor initial estimates for the image region shown in figure 4(*a*), which failed the convergence of iterative fitting process. Our new procedure creates reliable initial estimates (figure 5(*b*)), with which optimal estimates for the five Gaussian parameters are achieved only after three iterations (figure 5(*c*)).

4.5. *Bimodality test*

To ensure the reliability of threshold estimation, we only select an image region whose histogram has appreciable bimodality for threshold computation. The bimodality is measured by the valley-to-peak ratio:

$$\delta = \frac{\underset{\mu_1 < i < \mu_2}{Min}\{\hat{p}(i)\}}{Min\{\hat{p}(\mu_1), \hat{p}(\mu_2)\}} \tag{12}$$

where $\hat{p}$ is the Gaussian curve fitted in the previous steps. In our applications, the bimodality criteria are set as: (1) the valley-to-peak ratio $\delta < 0.8$; and (2) the means of the two component distribution must differ by more than three grey levels: $\mu_2 - \mu_1 > 3$.

4.6. *Computing local threshold by solving a quadratic equation*

For each image region whose histogram has appreciable bimodality and passed the bimodality test, the optimal threshold value T is computed by solving the following quadratic equation:

$$AT^2 + BT + C = 0 \tag{13}$$

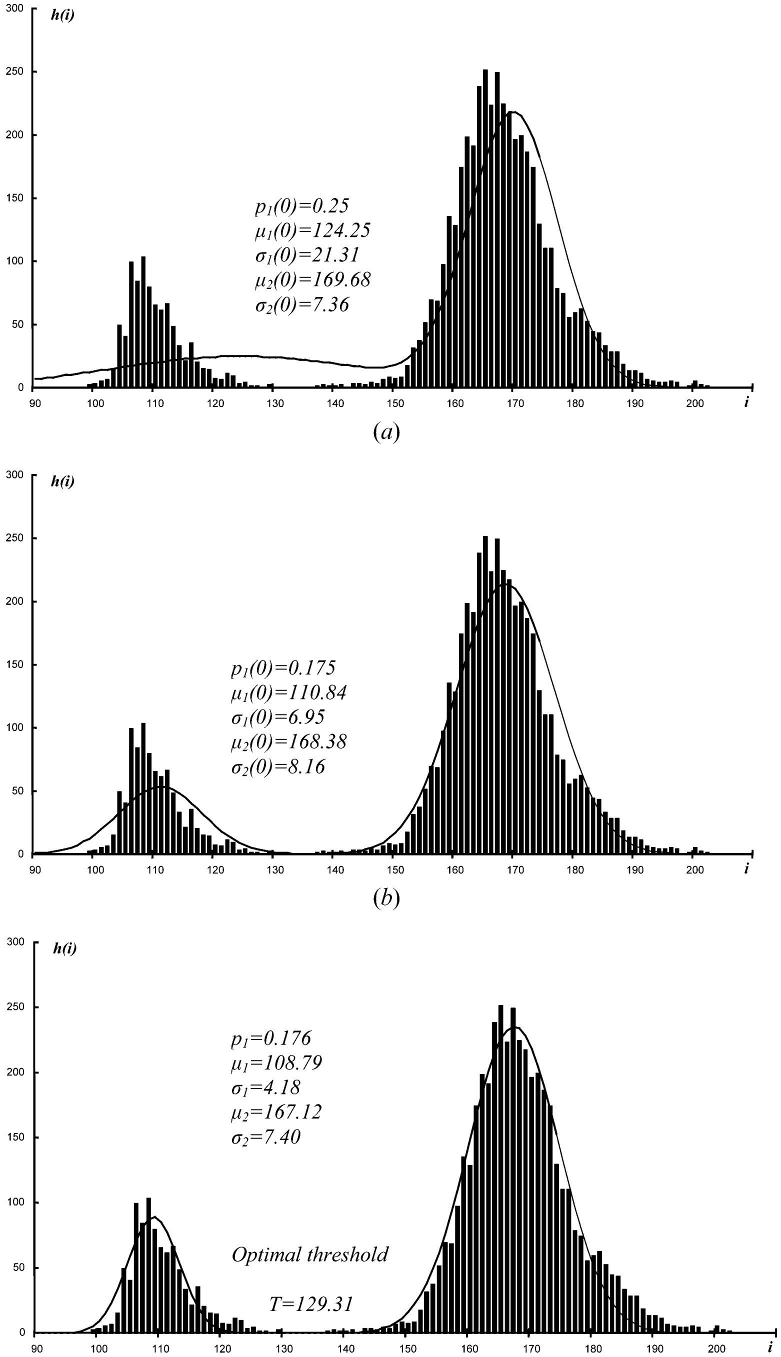
where

$$A = \sigma_1^2 - \sigma_2^2$$

$$B = 2\left(\mu_1\sigma_2^2 - \mu_2\sigma_1^2\right)$$

$$C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln\frac{\sigma_2 p_1}{\sigma_1 p_2}$$

The above quadratic formula is derived from the method of maximum likelihood for the value of threshold *T*, which minimizes the probability of misclassification (Chow and Kaneko 1972, Gonzalez and Woods 1992).



(*a*)

(*b*)

(*c*)

Figure 5. Determination of initial values for Gaussian parameters: (*a*) initial bimodal Gaussian curve with parameters calculated by conventional method; (*b*) initial bimodal Gaussian curve with parameters calculated by our new method; and (*c*) best-fitted bimodal Gaussian curve by the Levenberg-Marquardt method after three iterations.

4.7. *Local threshold interpolation and image segmentation*

In Haverkamp *et al.* (1995) and Sohn and Jezek (1999), threshold values calculated for different regions were clustered into two groups—upper thresholds and lower thresholds—and then the image pixels were separated into three classes using the upper and lower thresholds. Our experiments show that clustering thresholds into two groups and subsequently segmenting the image into three classes tends to create spurious coastline edges and adversely influences the accuracy of the coastline position.

In our implementation, we directly interpolate the threshold values computed from previous steps without clustering them into two groups. The interpolation method that we used is the inverse distance weighted (IDW) method, which is similar to the one used in Haverkamp *et al.* (1995). The threshold for a region that fails the bimodality test is calculated through the weighted average of the thresholds of neighbouring regions that have passed the bimodality test, and the weight is determined according to the normalized inverse distance. As a result of interpolation, each image pixel (i, j) has a locally adaptive threshold value $T_{i,j}$.

A binary image $g(i,j)$ is created after the thresholding operation:

$$g(i,j) = \begin{cases} 255, \text{ } if \text{ } f(i,j) > T_{i,j} \\ 0, \text{ } if \text{ } f(i,j) \leqslant T_{i,j} \end{cases} \tag{14}$$

where $f(i,j)$ is the intensity value of the image pixel at (i,j), and $T_{i,j}$ is the locally adaptive threshold. The pixels with a higher intensity value than the local threshold are coded as 255 (land pixels), while the pixels with a lower intensity value than the local threshold are coded as 0 (water pixels).

5. Post-segmentation processing

5.1. *Region grouping and labelling*

Digital image segmentation produces a binary image output $g(i,j)$, which consists of various connected image regions. Any contiguous regions of land or water in $g(i,j)$ can be grouped into a higher level of representation—image objects (figure 6). Each image object is then labelled by a unique identification number and characterized by its size, position and geometric shape. Analysis of the geometric properties of image objects renders the capability of differentiating the true coastline edge pixels from other object edge segments.

The region grouping and labelling algorithm implemented in this research is based on a 'grassfire' concept. The image is scanned in a row-wise manner, and a 'fire' is set at the first pixel of an image object. The fire propagates to all pixels belonging to the four- or eight-neighbourhood of the current pixel. The propagation is continued recursively until all pixels of the image object are 'burnt'. The 'fire' is extinguished after all objects in the image are labelled. During the grouping and labelling process, the areal size and other properties of each labelled object are calculated.

5.2. *Two passes of selective removal of small image objects*

Based on the image objects extracted from the segmented image, heuristic human knowledge about the size and continuity of land and ocean masses is applied to separate true coastline edges from other object boundaries. The fundamental observation is that both land and ocean masses are large, continuous image objects. In the first pass of processing, the water pixels coded as 0s in $g(i,j)$

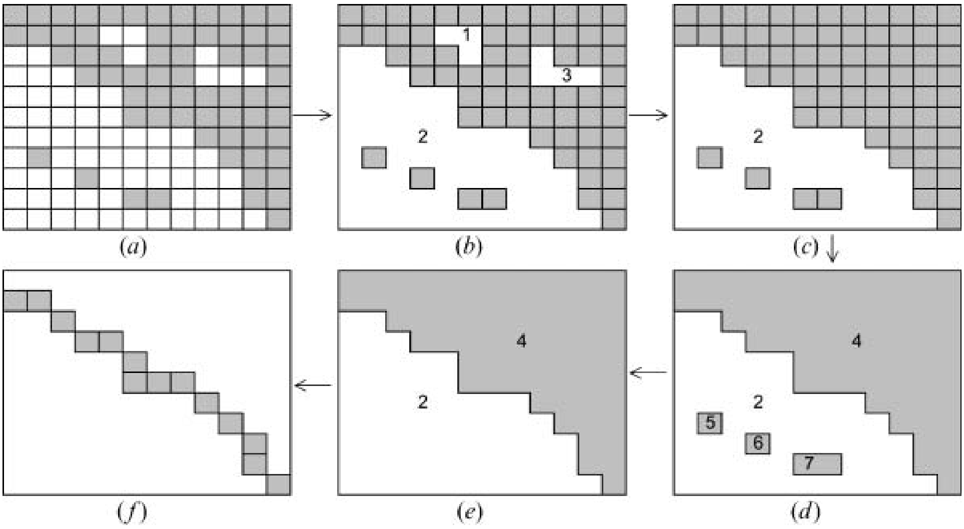


Figure 6. Image region grouping and labelling: (*a*) segmented image with land pixels (grey) and water pixels (white); (*b*) water pixels (white) are grouped into three image objects 1, 2 and 3; (*c*) small image objects 1 and 3 are removed; (*d*) land pixels (grey) are grouped into four image objects 4, 5, 6 and 7; (*e*) small image objects 5, 6 and 7 are removed, and only two big objects—2 (ocean water) and 4 (land)—are left; and (*f*) coastline edge pixels are delineated.

are grouped and labelled as individual image objects (figure 6(*b*)). The small image objects identified in this pass often correspond to wetlands, lakes, stream, shadows or image noise in the land area. Since the boundaries of these small image objects are not true coastline, they are dissolved into the land area (figure 6(*c*)). In the second pass of processing, the land pixels coded as 255s are grouped and labelled into individual image objects. Small, scattered image objects identified in the ocean are mainly due to floating ice, icebergs, ships, other ocean facilities or image noise. Similarly, these isolated small image objects identified with a selected area threshold can be fused into the ocean area (figure 6(*e*)). After two passes of removal of residual, isolated image objects, only two large continuous land and ocean objects are left. This procedure effectively eliminates unwanted objects whose boundaries are not the coastline, and therefore minimizes the editing work for cleaning up the final coastline product.

5.3. *Morphological dilation and erosion*

The morphological operation of dilation immediately followed by erosion produces the closing operator (Parker 1997). The use of the closing operation tends to generalize the jagged boundaries of image objects, making the coastline morphologically smoother.

5.4. *Delineating the boundary pixels of image objects into raster image of coastline*

A coastline is defined as the boundary of land image objects. Each land pixel in the image objects is scanned by a 3 × 3 neighbourhood window to examine its four immediate neighbours (horizontal and vertical). If one or more neighbours of the land pixel belong to water (background) pixels, this land pixel is then flagged as boundary pixel and coded with a value of 255. Otherwise, the pixel will receive a value of 0 as a background pixel (figure 6(*f*)). In this way, all land pixels immediately adjacent to the water pixels are extracted to form the land/water boundary:

$$b(i,j)=\begin{cases} 255, \textit{if}\ g(i,j)=255 \text{ and } (g(i-1,j), \text{ or } g(i+1,j), \text{ or } g(i,j-1), \text{ or } g(i,j+1)=0) \\ 0, \quad \text{otherwise} \end{cases} \qquad (15)$$

where $b(i, j)$ is the raster image of extracted coastline.

5.5. *Line tracing and vectorization*

Edge tracing is the process of following the edge pixels and recording their coordinates into a list of vector line segments. We employ a recursive algorithm for edge tracing. With a starting pixel, the tracing direction for the next step is defined based on the principle of minimizing angular change. The algorithm checks the eight-connected neighbours of the starting pixel, and seven tracing directions are allowed (figure 7(*a*)). The first candidate successor is the edge pixel that lies in the direction defined by the previous and current edge pixel. If such a successor does not exist, the algorithm searches other possible successors that result in a minimum angular change in the tracing direction. The tracing continues recursively until no new candidate pixel is available. This tracing procedure tends to produce a smooth vector boundary with a low curvature (figure 7(*b*)). We record the coordinates of traced boundary pixels as a series of vector line segments in ArcInfo Ungenerate format. Based on the Ungenerate file, an ArcInfo vector line coverage can be created to represent the final coastline.

6. Application examples

Two examples are given in this paper to demonstrate the effectiveness and performance of the processing chain.

6.1. *Extracting coastline of Antarctica from Radarsat SAR images*

The SAR images used in this example were acquired by the Canadian Satellite Radarsat-1 using C-band sensor during the first Antarctic Imaging Campaign in 1997 (Jezek 1999). The raw SAR data were processed at the Alaska SAR Facility into four-look intensity images. The SAR imagery has experienced a rigorous

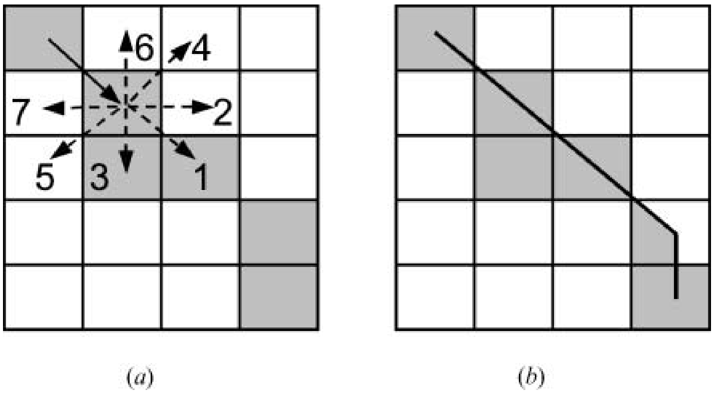


Figure 7. Tracing and vectorizing edge pixels (grey cells): (*a*) seven possible tracing directions, prioritized in 1, 2, …, 7 order; and (*b*) vectorized coastline.

geocoding and orthorectification process with well-distributed ground control points and mathematical modelling of imaging geometry at Byrd Polar Research Center (Jezek 1999) and therefore provides a planimetrically accurate image basis for our algorithms. The Polar Stereographic map projection was selected for the geocoding process. By applying our method to the SAR images, we have successfully extracted a complete, high-resolution coastline for the entire Antarctic continent.

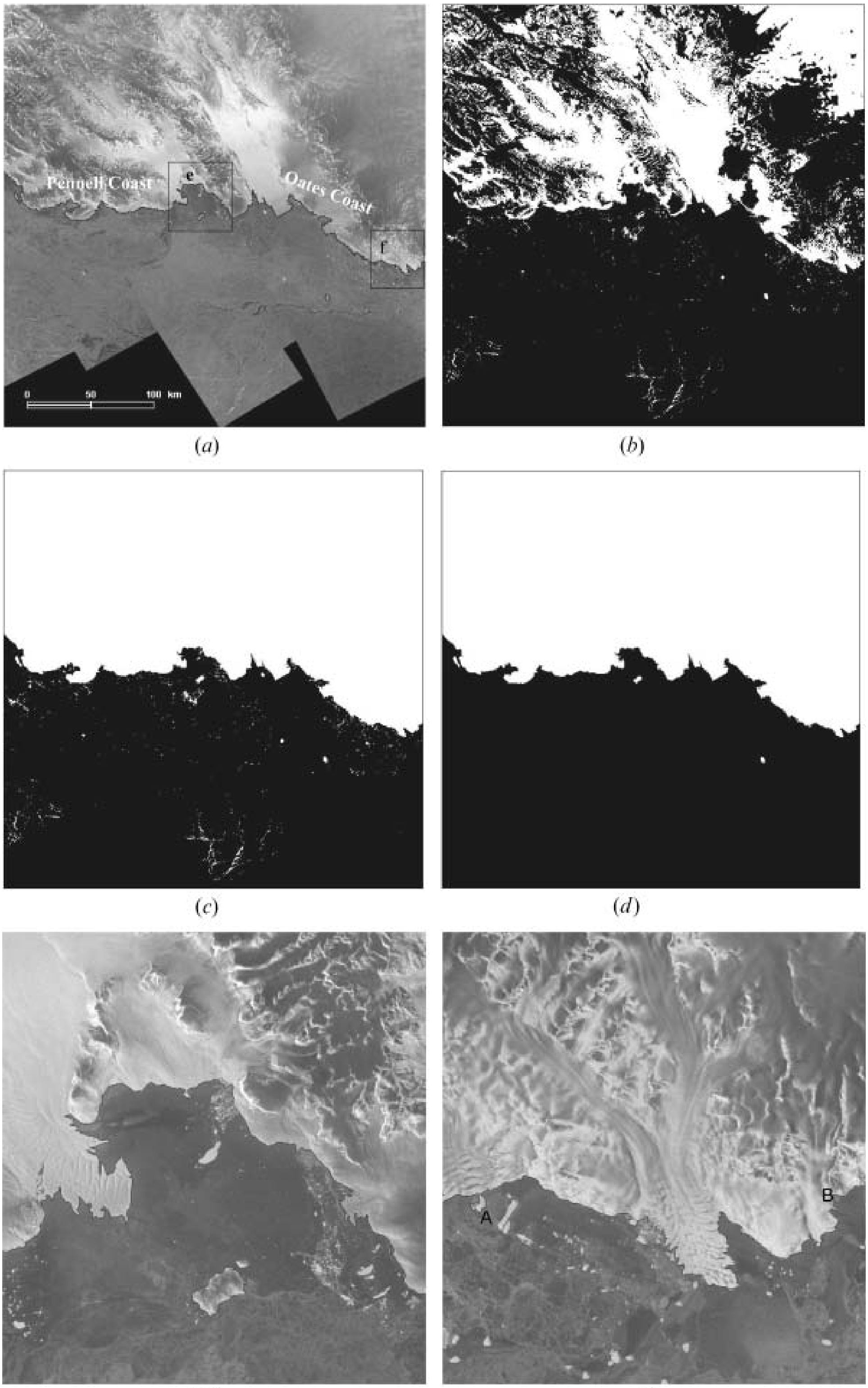
Figure 8 illustrates the processing sequence applied to a SAR image with 100 m resolution. It is located along the Pennell Coast of Northern Victoria Land, Antarctica. The image has 4096 × 4096 pixels, covering an area of 409.6 km × 409.6 km. First, a 5 × 5 Lee filter is applied, and then the anisotropic diffusion operator is applied to the SAR image. After applying the Lee filter and anisotropic diffusion operator, the image noise and speckle is considerably reduced, intensity variations and weak edges inside the land or the ocean are effectively suppressed, and the coastline edges are enhanced.

The entire image is divided into overlapping image regions, each with 32 × 32 pixels. Twenty per cent of image regions that have a high variance are selected for analytically determining optimal local thresholds. By integrating the Canny edge detector and the Levenberg-Marquardt method, the five bimodal Gaussian parameters are iteratively fitted. For most image regions, the convergence is achieved within 4–7 iterations. The optimal local thresholds for these regions are interpolated throughout the entire image. The entire image is subsequently segmented using the locally adaptive threshold. The segmented image is shown in figure 8(*b*).

At the post-segmentation processing stage, the water pixels in the segmented image are first grouped and labelled into individual image objects, and the resulting small image objects are removed, as shown in figure 8(*c*). Then, the ice/land pixels are grouped and labelled into individual image objects, and those small image objects are removed as shown in figure 8(*d*). After two passes of selective removal of small noisy image objects, only true continuous land and ocean image objects are left. The extracted coastline pixels are then traced, and their coordinates are recorded as a vector-based ArcInfo Ungenerate file. Finally, a vector coverage of the coastline is created in ArcInfo format. To gain an appreciation of the precision of the algorithms, we display two enlarged portions of the derived coastline with the original SAR image as backdrop. As shown in figure 8(*e*) and (*f*), subtle coastal features, such as small inlets and ice tongues, are accurately delineated. The visual examination shows that the extracted coastlines closely match those obtained from human interpretation of the original images. The average accuracy of coastline position is down to one pixel. Due to the rigorous orthorectification of the source SAR imagery, our assessment shows that the absolute accuracy of the geographical position of the derived coastline is better than 130 m.

During the systematic processing of radar images for Antarctica, some

Figure 8. Automated coastline extraction from Radarsat SAR imagery (100 m): (*a*) extracted coastline draped on the original SAR image; (*b*) segmented image using locally adaptive thresholding method; (*c*) after applying image region grouping and removing small objects that are classified as water objects; (*d*) after applying image region grouping and removing small objects that are classified as land/ice objects; (*e*) coastline draped on the enlarged SAR image; and (*f*) two delineation errors are marked at the locations A and B.



(*a*)

(*b*)

(*c*)

(*d*)

(*e*)

(*f*)

delineation errors were detected for our automated method. Radar imagery is not affected by clouds or darkness, making it ideal for coastline detection in polar regions. Nevertheless, difficulties and problems remain (Lee and Jurkevich 1990, Mason and Davenport 1996). In the case of Antarctica, the presence of sea ice, fast ice, icebergs and wet snow zones in the coastal regions complicates the coastline detection and extraction. For instance, when a large floating sea ice was directly attached to the ice and snow covered land area, the extracted coastline mistakenly extruded out into the ocean. Such an example is shown at location A in figure 8(*f*). When a rock exposure area or a very smooth glacial floor extends to the ocean water, the tracing was misled into the inland area to form artificial inlets or false openings along the coastline. Such an example is shown at location B in figure 8(*f*). These errors are fixed in an ArcInfo GIS environment. With the ArcInfo graphic display tools, the vector-based coastline coverage is displayed on top of the original source satellite image. The quality of the derived coastline is inspected and verified with the zoom-in, zoom-out and pan tools. With the graphic editing tools of ArcInfo, incorrect coastline segments can be deleted and new ones can be inserted.

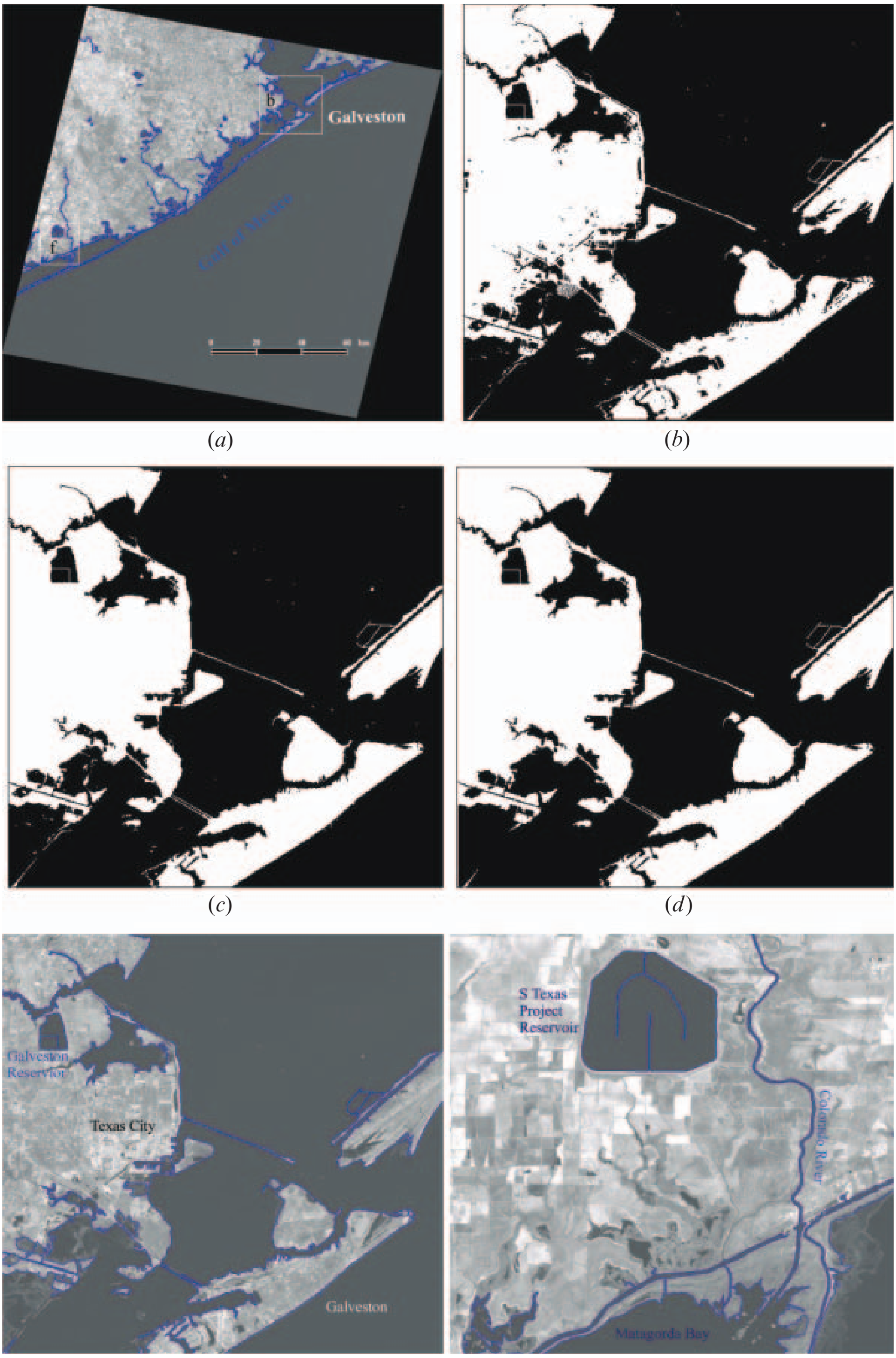
6.2. *Extracting the coastline of the Gulf of Mexico from Landsat 7 ETM+ images*

Another example demonstrates the applicability of our coastline extraction method to optical satellite images. The input image is a Landsat 7 Enhanced Thematic Mapper Plus (ETM+) image over the Galveston Bay of the Gulf of Mexico, near Houston, Texas. The infrared channel (band 5) of the image is used for the processing (figure 8(*a*)). The image has 7088 × 7545 pixels with 30 m spatial resolution and has been georeferenced to the UTM map projection. A 5 × 5 Gaussian filter, instead of Lee filter, is applied to the infrared image. For the post-segmentation processing, a small threshold is used to remove small, spurious image objects after the grouping and labelling operation. Islands and lakes with a relatively large size are kept. Other processing steps and parameter settings are similar to those used for processing the radar images. As shown in figure 9, peninsulas, small inlets, islands, lakes, reservoirs, rivers and piers in the reservoirs are precisely marked out. An excellent match of the extracted coastline with the original image substantiates the applicability of our method to optical satellite images for coastline detection and delineation.

7. Discussion and conclusions

Coastline mapping seems to be a simple application of remote sensing data, but in practice automated extraction of the land/water boundaries is more difficult than one would expect. Because of the frequent lack of consistent, sufficient intensity contrast between land and water regions and the complications of distinguishing coastline from other object boundaries, coastline extraction is a difficult task with most general-purpose edge detectors or image segmentation techniques. In this research, we present a comprehensive technique for automating coastline extraction

Figure 9. Automated coastline extraction from Landsat 7 ETM+ image (band 5, 30 m): (*a*) extracted coastline draped on the entire original Landsat image; (*b*) segmented image for the enlarged area b using locally adaptive thresholding method; (*c*) after applying image region grouping and removing small objects that are classified as water objects; (*d*) after applying image region grouping and removing small objects that are classified as land objects; (*e*) coastline draped on the enlarged area b; and (*f*) coastline draped on the enlarged area f.


b
Galveston
f
Gulf of Mexico
Galveston Reservoir
Texas City
Galveston
S Texas Project Reservoir
Colorado River
Matagorda Bay

(*a*) (*b*)

(*c*) (*d*)

(*e*) (*f*)

from satellite imagery. Our method consists of a sequence of image processing algorithms written in the C language.

The crucial step governing the positional accuracy of the extracted coastline is image segmentation. The accuracy of the image segmentation, in turn, depends upon the reliability and correctness of the local threshold analytically determined by fitting a bimodal Gaussian curve. This research has made several technical innovations to improve the accuracy and efficiency of the algorithms for achieving a reliable, optimal threshold. The Levenberg-Marquardt method is introduced for the first time for iteratively fitting a nonlinear bimodal Gaussian curve to analytically determine optimal local thresholds. The Canny edge detector is integrated into the Gaussian curve approximation process. The removal of mixed pixels from the formation of observed histogram and the estimation of the initial values of bimodal Gaussian parameters based on the edge pixels and their immediate neighbourhood have greatly increased the quality of the Gaussian parameter estimations and also accelerated the convergence rate of the iterative fitting process. This in turn improves the accuracy and efficiency of local threshold computation.

The underlying assumption for the locally adaptive thresholding algorithm is that a small image region that contains a coastline can be characterized by a mixture of two Gaussian distributions. Since thresholds are determined analytically according to local, rather than global, characteristics of the observed histograms, they account for changing geophysical characteristics along lengthy reaches of the coastline. As a result, our method generated the coastline with a high accuracy, approaching the spatial resolution of the imagery.

A series of post-segmentation processing steps greatly automated the extraction of vector-based coastline and minimized the manual editing of the final vector products. Particularly, grouping and labelling image regions into individual image objects enable us to utilize heuristic human knowledge about the size and continuity of land and ocean masses and hence to eliminate object boundaries other than true coastline. The vectorized coastline products can be viewed and further edited in a GIS environment with reference to the original satellite images. The batch processing of the Radarsat SAR images for Antarctic coastline demonstrates that our method can significantly remove the burden of conventional manual delineation and achieve a high positional accuracy of the coastline that is difficult for the human 'on-screen' digitization method.

Despite the successful applications of our method on both radar and optical satellite images, it should be noted that the quality of the image sources remains an important factor for coastline extraction. The success of our method still depends upon whether considerable contrast exists between water and land masses. To a lesser degree it also depends on the homogeneity of the water or land mass. During the systematic processing of radar images for Antarctica, we detected some delineation errors. These errors can be easily corrected in a GIS environment by deleting incorrect coastline segments and/or inserting new ones with graphic editing tools. In addition, the GIS-based editing tools are also used for merging the coastlines extracted from adjacent image tiles.

The relative accuracy is evaluated in this research based on the comparison between the algorithm derived coastline and the coastline visually interpreted from the original satellite images. Extensive visual comparison shows that the relative accuracy of the algorithm-derived coastline is within one image pixel, compared with the human visual interpretation of the coastline features. In a real-world application, the absolute accuracy of the geographical position of the derived coastline is essential. It should be emphasized that the absolute accuracy is influenced not only by the coastline extraction method, but also by the georeferencing accuracy of the source images. To derive a coastline with precise absolute geographical coordinates and correct geometric shape, the source images used to extract the coastline must be geocoded and orthorectified before applying our coastline extraction algorithms.

Although our coastline delineation algorithms are operational and effective, some further enhancements could be made in a future version. For example, we only utilized the size and continuity of the image objects identified by the grouping and labelling operation for the purpose of differentiating the true coastline from other object boundaries. Other attributes of the image objects, such as shape, texture and relative position, can also be used in the post-segmentation processing for determining the true water/land boundaries. The results could probably be improved at the expense of a slightly increased computation.

Acknowledgments

This work was supported under a National Aeronautics and Space Administration (NASA) grant NAG5-10112 and the National Science Foundation (NSF) grant No. 0126149. SAR data were processed and provided by the Radarsat: Antarctic Mapping Project of the Byrd Polar Research Center. The Landsat image was purchased using a research enhancement grant from the College of Geosciences, Texas A&M University. The authors wish to thank Hong-Gyoo Sohn for helpful discussions on the local dynamic thresholding algorithm.

References

AHUJA, N., and ROSENFELD, A., 1978, A note on the use of second-order gray-level statistics for threshold selection. *IEEE Transactions on Systems, Man, and Cybernetics*, **8**, 895–898.

BALLARD, D., and BROWN, C., 1982, *Computer Vision* (Englewood Cliffs, NJ: Prentice-Hall).

CANNY, J., 1986, A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, 679–698.

CHOW, C. K., and KANEKO, T., 1972, Automatic boundary detection of the left ventricle from cineagiograms. *Computer and Biomedical Research*, **5**, 388–410.

COHEN, L. D., 1991, On active contour models and balloons. *CVGIP: Image Understanding*, **53**(2), 211–218.

FROST, V. S., STILTES, J. A., SHANMUGAN, K. S., and HOLTZMAN, J. C., 1982, A model for radar images and its application to adaptive digital filtering of multiplicative noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **4**, 157–166.

GONZALEZ, R. C., and WOODS, R. E., 1992, *Digital Image Processing* (Reading, MA: Addison-Wesley).

HAVERKAMP, D., SOH, L. K., and TSATSOULIS, C., 1995, A comprehensive, automated approach to determining sea ice thickness from SAR data. *IEEE Transactions on Geoscience and Remote Sensing*, **33**, 46–57.

JEZEK, K. C., 1999, Glaciologic properties of the Antarctic Ice Sheet from spaceborne Synthetic Aperture Radar observations. *Annals of Glaciology*, **29**, 286–290.

KASHYAP, R. L., and BLAYDON, C. C., 1968, Estimation of probability density and distribution functions. *IEEE Transactions on Information Theory*, **14**, 549–556.

KASS, M., WITKIN, A., and TERZOPOULOS, A., 1987, Snakes: active contour models. *International Journal of Computer Vision*, **1**, 321–331.

KUAN, D. T., SAWCHUK, A. A., STRAND, T. C., and CHAVEL, P., 1985, Adaptive noise smoothing filter for images with signal-dependent noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **7**, 165–177.

KUAN, D. T., SAWCHUK, A. A., STRAND, T. C., and CHAVEL, P., 1987, Adaptive restoration of images with speckle. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **35**, 373–383.

LEE, J. S., 1986, Speckle suppression and analysis for Synthetic Aperture Radar images. *Optical Engineering*, **25**, 636–643.

LEE, J. S., and JURKEVICH, I., 1990, Coastline detection and tracing in SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, **28**, 662–668.

MASON, D. C., and DAVENPORT, I. J., 1996, Accurate and efficient determination of the shoreline in ERS-1 SAR Images. *IEEE Transactions on Geoscience and Remote Sensing*, **34**, 1243–1253.

NARAYANAN, K. A., and ROSENFELD, A., 1981, Image smoothing by local use of global information. *IEEE Transactions on Systems, Man, and Cybernetics*, **118**, 826–831.

OLIVER, C., and QUEGAN, S., 1998, *Understanding Synthetic Aperture Radar Images* (Norwood: Artech House).

PARKER, J. R., 1997, *Algorithms for Image Processing and Computer Vision* (New York: John Wiley & Sons).

PERONA, P., and MALIK, J., 1990, Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 629–639.

PITAS, I., 2000, *Digital Image Processing Algorithms and Applications* (New York: John Wiley & Sons).

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., and FLANNERY, B. P., 1992, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd edn (Cambridge: Cambridge University Press).

REES, W. G., and STACHELL, M. J. F., 1997, The effect of median filtering on synthetic aperture radar images. *International Journal of Remote Sensing*, **18**, 2887–2893.

ROSENFELD, A., and DAVIS, L. S., 1978, Iterative histogram modification. *IEEE Transactions on Systems, Man, and Cybernetics*, **8**, 300–302.

RYAN, T. W., SEMENTILLI, P. J., YUEN, P., and HUNT, B. R., 1991, Extraction of shoreline features by neural nets and image processing. *Photogrammetric Engineering and Remote Sensing*, **57**, 947–955.

SAINT-MARC, P., CHEN, J., and MEDIONI, G., 1991, Adaptive smoothing: a general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 514–529.

SOHN, H. G., and JEZEK, K. C., 1999, Mapping ice sheet margins from ERS-1 SAR and SPOT imagery. *International Journal of Remote Sensing*, **20**, 3201–3216.

TOUZI, R., LOPES, A., and BOUSQUET, P., 1988, A statistical and geometrical edge detector for SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, **26**, 764–773.

WESZKA, J. S., and ROSENFELD, A., 1979, Histogram modification for threshold selection. *IEEE Transactions on Systems, Man, and Cybernetics*, **9**, 38–52.

WILLIAMS, D. J., and SHAH, M., 1992, A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, **55**, 14–26.

YOUNG, T. Y., and CORALUPPI, G., 1970, Stochastic estimation of a mixture of normal density functions using an information criterion. *IEEE Transactions on Information Theory*, **16**, 258–263.