

Accurate and Efficient Determination of the Shoreline in ERS-1 SAR Images

David C. Mason and Ian J. Davenport

Abstract—Extraction of the shoreline in SAR images is a difficult task to perform using simple image processing operations such as grey-value thresholding, due to the presence of speckle and because the signal returned from the sea surface may be similar to that from the land. A semiautomatic method for detecting the shoreline accurately and efficiently in ERS-1 SAR images is presented. This is aimed primarily at a particular application, namely the construction of a digital elevation model of an intertidal zone using SAR images and hydrodynamic model output, but could be carried over to other applications. A coarse-fine resolution processing approach is employed, in which sea regions are first detected as regions of low edge density in a low resolution image, then image areas near the shoreline are subjected to more elaborate processing at high resolution using an active contour model. Over 90% of the shoreline detected by the automatic delineation process appear visually correct.

I. INTRODUCTION

THIS PAPER describes a semiautomatic method for accurate determination of the shoreline in ERS-1 synthetic aperture radar images. This has been designed for use in a specific application, namely the construction of a digital elevation model (DEM) of an intertidal zone using a combination of remote sensing and hydrodynamic modeling techniques. However, detection of the shoreline in SAR images is important in a number of applications, as discussed in [1], the method could be used for these, also. In cartographic mapping, for example, it could be used to map the position of the shoreline at times of high and low water.

The difficulties of distinguishing the shoreline in SAR images compared with carrying out the same task using images from visible-band sensors such as the Landsat Thematic Mapper have been pointed out in [1]. The SAR signal returned from a sea surface roughened by wind and waves can equal or exceed the return from the land surface, and the problem is compounded by the presence of speckle and the single-frequency nature of many SAR sensors. Even for experienced interpreters of SAR images, it can be difficult to distinguish the exact location of the shoreline in some areas.

The particular application considered here, the construction of an intertidal DEM, has two main uses. First, it is possible to detect changes in the DEM as a function of time. These may occur as a result of sediment mass transfers due to a major

storm or seasonal changes, or more slowly due to sea-level rise caused by global warming. It allows the calculation of changes in the volumes of banks and salt marshes (see, e.g., [2]), and the display of areas of accretion and erosion for comparison with the predictions of sediment transport models. Second, it can lead to improved hydrodynamic tide-surge models, as in complex estuarial regions the correct modeling of the tide and surge is dependent upon having an accurate topography for wetting/drying areas [3].

The method of DEM construction involves first finding the (x, y) positions of the shoreline from a remotely sensed image using image processing techniques, and registering the boundary to a base map. The heights relative to mean sea level of the registered boundary are then superimposed on the corresponding (x, y) positions. These heights are predicted using a hydrodynamic tide-surge model run for this area with the atmospheric conditions pertaining at the time of image acquisition. From multiple images obtained over a range of tide and surge elevations, it is possible to build up a set of heightened shorelines within the intertidal zone, and from this, a gridded DEM may be interpolated (Fig. 1). In effect, the sea is being used as an altimeter. This approach has also been proposed in [4], and a related technique has been used in [5] for detecting sea level changes.

The use of satellite imagery gives the advantage of being able to construct an intertidal DEM over large areas relatively rapidly and cheaply. SAR sensors are particularly suited to this task because their all-weather day-night capability means that a multitemporal sequence can be built up more easily than by using visible imagery. This is especially true given the dynamic nature of the intertidal zone, as a single large storm can change beach levels significantly. This leads to the need for a statistical approach to the analysis of intertidal heights, with, ideally at each point, a mean height and its variation as a function of time being measured in order for long-term trends to be determined [6]. This statistical approach requires many more satellite images than would be necessary if the beach was a static entity.

The levels of accuracy achieved in the intertidal DEM depend principally on the accuracy of the hydrodynamic model, the local slope of the intertidal zone, and the remote sensor type and resolution. Typically, there will be height uncertainty of a few centimeters or so due to the hydrodynamic model, provided the weather is reasonably calm. Given that a typical intertidal zone slope is $\sim 1:100$, for ERS-1 SAR data, with a pixel side of 12.5 m, an uncertainty of one pixel in shoreline position normal to the sea edge translates to a height uncertainty of about 12 cm, so that satellite resolution

Manuscript received February 7, 1995; revised August 3, 1995. This work was supported under NERC LOIS Special Topic Grant GST/02/777 and NERC Contract F60/G6/12. ERS-1 SAR images were provided by ESA under the ERS-1 Pilot Project Programme.

The authors are with the Natural Environment Research Council Environmental Systems Science Centre (ESSC), Department of Geography, University of Reading, Reading RG6 6AB, U.K.

Publisher Item Identifier S 0196-2892(96)05059-0.

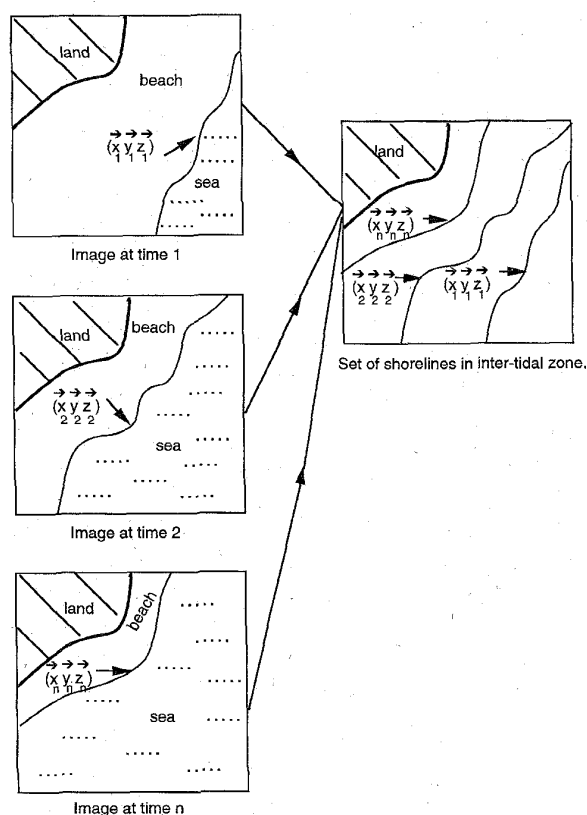


Fig. 1. DEM construction method.

becomes the dominant error source. It is important, therefore, that the method determines the shoreline position as accurately as possible, to the nearest pixel or preferably better.

This paper is concerned with the image processing techniques employed to delineate the shoreline in ERS-1 SAR images. Due to the substantial effort and subjectivity involved in manual delineation of ~ 100 km of shoreline in each image, a semiautomatic method of delineation has been developed. A description of the other aspects of the method has been given in [7], and the DEM's obtained, the accuracies achieved, and a comparison with other methods of intertidal height measurement will be published elsewhere.

The project is part of the Land-Ocean Interaction Study (LOIS), a current U.K. Community Research Programme which will provide an integrated holistic view of how coastal ecosystems work, and how they are likely to respond to future environmental changes [8]. The provision of accurate topography and bathymetry in the coastal zone is an important underpinning activity in LOIS, as many of the component process studies require such data to be available. The images used to illustrate the results of the method have been taken from the main LOIS study area between the Humber and the Wash on the U.K. East Coast.

II. SEMIAUTOMATIC DELINEATION OF THE SHORELINE

Examination of the ERS-1 SAR images confirms the feasibility of determining the shoreline by eye over large stretches

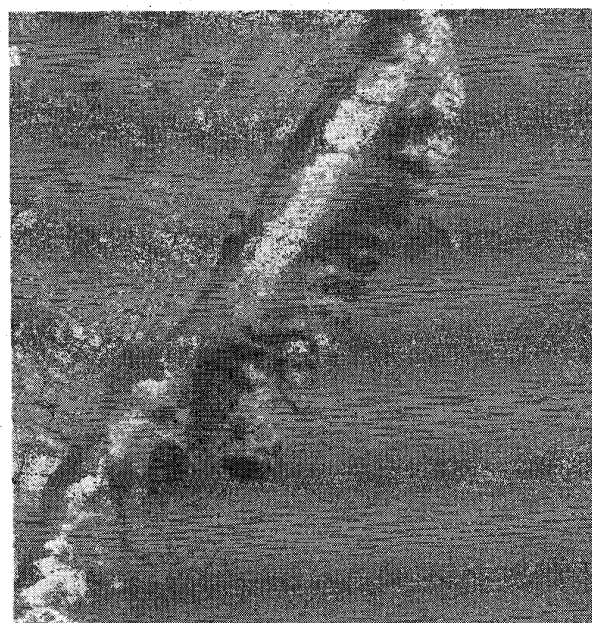


Fig. 2. ERS-1 SAR subimage of Wrangle Flats, Western Wash (1152 \times 1216 pixels) (copyright ESA).

of the imagery (see, e.g., Fig. 2, an image of Wrangle Flats in the Western Wash taken at mid flow tide). On an ebb tide, in particular, a dark band is often apparent next to the sea edge due to wet sand, there acting as a specular reflector away from the satellite. On the other hand, in some areas, it becomes very difficult for the human eye to distinguish this boundary. This is generally due to variations in the radiometric response of different points on the beach, but may occasionally occur because at low wind speeds (below about 3 ms^{-1}), both sea and beach may be too dark to differentiate. As a result, the delineation method attempts to determine just those segments of shoreline in which it has high confidence, rather than trying to distinguish a continuous shoreline terminating only at the image edges. It is left to the interpolation procedure to fill in any gaps using information from this image and others taken at a similar tide/surge state.

Previous work on automatic delineation of the shoreline in SAR images has been carried out by Lee and Jurkevich [1]. They achieved reasonable positional accuracy using their method, but state that refinements are necessary to achieve the accuracy required for geographic mapping. The present method has some similarities with theirs, but also some important differences. As in their approach, we rely on the fact that, after suppression of speckle, the sea is generally much smoother than the land. We also use global information in order to guide local shoreline detection, employing a coarse-fine resolution processing scheme. However, their approach extracts a continuous shoreline, whereas we aim for higher (single-pixel) positional accuracy without necessarily extracting a continuous interface. We also employ a constant false alarm rate edge detector, and make use of the fact that regions of sea are usually large connected regions of similar speckle statistics. The coarse-fine resolution processing

scheme employed also allows complete SAR images to be processed efficiently.

The shoreline extraction procedure involves a sequence of algorithms developed within an image processing system capable of dealing with large images in tiled form [9], [10]. The procedure starts by finding a rough division between land and sea in a coarse resolution processing stage, then processes a narrow strip straddling this division at high resolution in order to locate the shoreline accurately. This approach is similar to that adopted in [11] for shoreline detection in much smaller scenes of visible-wavelength aircraft imagery. The coarse-fine resolution processing strategy significantly reduces computing requirements. Whilst a complete ERS-1 SAR scene contains about 8000×8000 pixels, the intertidal zone is a strip about 500 pixels wide only, so that by limiting high resolution processing to this strip, a saving of $\sim \times 16$ in processing time can be made. As it requires over an hour to process a complete scene using the coarse-fine approach, this represents a significant saving.

2.1. Coarse Resolution Processing Stage

The low resolution coarse processing stage (Fig. 3) detects sea regions as large connected areas of low edge density in the reduced SAR image, extracts boundaries of these regions that do not coincide with the image frame, and generates a set of rectangles along these boundaries within which to perform high resolution processing. The steps in this stage are illustrated using the example images shown in Fig. 4. The input to this stage is a complete ungeocoded ERS-1 SAR PRI scene in tiled-image format [Fig. 4(a)]. The subimage of Fig. 2 was extracted from the image shown.

- 1) The first step involves averaging the original image using a 4×4 -pixel nonoverlapping window. This produces an image $1/16$ the size of the original, thus reducing subsequent processing times. The image also exhibits substantially reduced speckle noise. Whilst the intensity of the sea is similar to that of the land, the sea is now much smoother than the land. The coarse processing performs a texture segmentation of this image and selects regions of sea as large connected regions of low texture contrast that coincide with the image frame.

- 2) The texture measure used is edge strength per unit area [12], which is easily calculated, yet has good discrimination ability, being sensitive to texture contrast and coarseness. Edges are detected by applying a contrast ratio edge detector [13] (see also [14]). The usual edge detectors applied to optical imagery measure differences between pixel values and, because of the multiplicative nature of SAR speckle, produce more false edges within brighter homogeneous regions in SAR data. The contrast ratio edge operator measures the ratio of the average intensities in two nonoverlapping neighborhoods on opposite sides of a point, and is used in both the coarse and fine processing stages because it has a constant false alarm rate in SAR images in which a simple uncorrelated speckle model applies. To ensure valid application of the operator, the averaging in step 1) must be quadratic averaging [13] (in which each pixel value is squared prior to averaging) because the PRI image is an amplitude rather than an intensity image. The edge

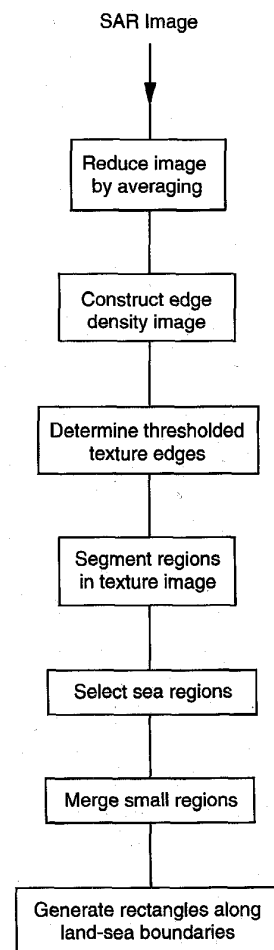


Fig. 3. Steps in the coarse resolution processing stage.

detector is applied over a 3×3 -pixel window in the reduced image, which is equivalent to applying it to two 12×4 -pixel rectangles separated by four pixels along their shorter side in the original image. An image of edge strength per unit area is constructed from the edge image by averaging using an 8×8 -pixel nonoverlapping window. This produces an image which is $1/1024$ th the size of the original SAR image, in which the low texture contrast of the sea is apparent [Fig. 4(b)]. In both these reduction steps, each tile in the input image is processed independently to produce a reduced tile in the output image.

A detailed comparison of texture measures was beyond the scope of this study, and other texture measures may be equally effective at discriminating sea from land. However, the measure used did prove superior to one based on local variance. Also, the discrimination obtained using the contrast ratio edge detector was significantly better than that obtained using a 3×3 -pixel Prewitt edge operator, which calculates the difference between average intensities in the two regions rather than their ratio.

- 3) Texture edges in the texture image are then detected in a second edge detection operation using a 5×5 -pixel window. Again the Prewitt edge detector was compared with the contrast ratio edge detector, and better results were obtained

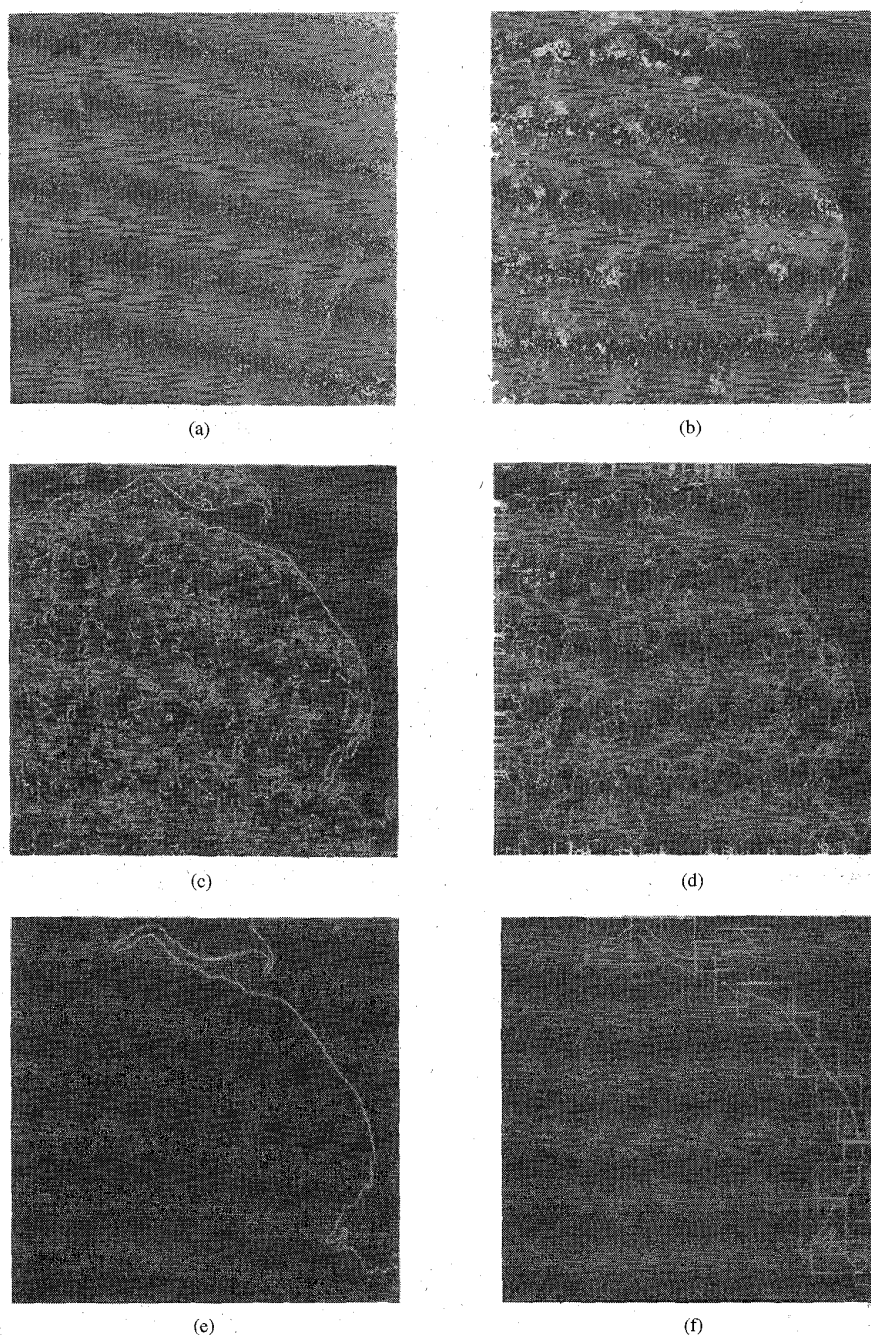


Fig. 4. (a) ERS-1 SAR scene of U. K. Humber/Wash area [subsampled to match the size of images (b–f)]. (b) Image of edge strength per unit area. (c) Thresholded texture edge image. (d) Linked edge image. (e) Boundary of the selected sea region. (f) The set of rectangles generated for high resolution processing.

using the latter. Nonmaxima are suppressed in the resulting texture edge image. The strongest texture edges are found at the shoreline, and these edges are thresholded from the majority of the remaining edges using a hysteresis thresholding algorithm [15] [Fig. 4(c)]. In this, an edge pixel is selected either if it exceeds a threshold T_h above which only the strongest 10% of edge pixels lie, or if it is above a lower threshold T_l ($T_l = 0.8T_h$ here) and is also adjacent to

a previously selected edge pixel. The algorithm is applied iteratively until no further pixels are selected, and has the characteristic that short runs of weaker edge pixels within runs of strong edge pixels are also accepted.

4) Regions are segmented from the thresholded texture edge image. The texture edges are thinned to a single pixel width, then vectorized. As is apparent from Fig. 4(c), in order to form closed regions, gaps between adjacent edge segments must be

filled in. This is achieved by a simple edge-linking process. For each pair of edge segments, the pair of endpoints of the two segments which are closest together are found, and if they are closer than a certain threshold T_d (eight pixels) the endpoint pair is linked together by a straight edge written into the thinned raster image. Further, if any endpoint is within T_d pixels of the image frame, it is joined to the closest point on the image frame [Fig. 4(d)]. Connected components (i.e., regions) are found in the modified thinned raster image, and each region's mean texture is measured.

5) Those regions whose areas exceed 12% of the image area and that touch the image frame are identified. The region within this subset having the lowest mean texture is taken to be a region of sea. In the many images we have processed, there has been only one sea region and this rule has identified it. Fig. 4(e) shows the boundary of the region accepted as sea. It is conceivable that more than one sea region may be present. Others could be identified by comparing the means and variances of each region's texture distribution with those of the region having the lowest mean texture (comparing the means using a t -test and the variances using an F -test), and selecting as additional sea regions those regions having similar low values. This has been successfully simulated using our existing data by dividing the sea region into two parts and checking that both parts are accepted as sea regions, whilst all large land regions are rejected (using 5% significance levels). The texture distribution within larger regions is roughly normal as required by these tests.

6) Small regions adjacent to the main sea regions are then examined to see if they can be merged with the sea regions. This step is included to correct errors that occur in the edge-linking process, which occasionally splits off narrow inlets from the sea regions. To perform merging, it is first necessary to determine the identifiers of the regions adjacent to each sea region using a region adjacency graph [12], as well as the identifiers of the common boundary segments between each adjacent region and the sea region. For a region to be merged, it must have a small area a , a small difference d_t between its mean texture and that of the sea region (normalized by the standard deviation of the sea region texture), a low compactness c ($c = a/p^2$, where p = perimeter), and a weak texture edge (strength t_e) between it and the sea region. These measures are combined into a single figure-of-merit $F = 1/(ad_t t_e c)$ and the region is merged if F exceeds a threshold.

7) The final step in the coarse stage extracts boundaries of sea regions that do not coincide with the image frame, and generates a set of rectangles within which to perform high resolution detection of the shoreline [Fig. 4(f)]. Each shoreline segment not coincident with the image frame is traversed in succession. If for a particular section of shoreline, the width of its containing box in the x direction is greater than w ($w = 22$ coarse pixels), it is assumed that the dominant change in the local coastline is in the x direction, and a fine resolution rectangle is set up by expanding the containing box by an amount 2δ ($\delta = 8$ coarse pixels) in the y direction [e.g., rectangle A in Fig. 4(f)]. Alternatively, if the width of the containing box is greater than w in the y direction, the

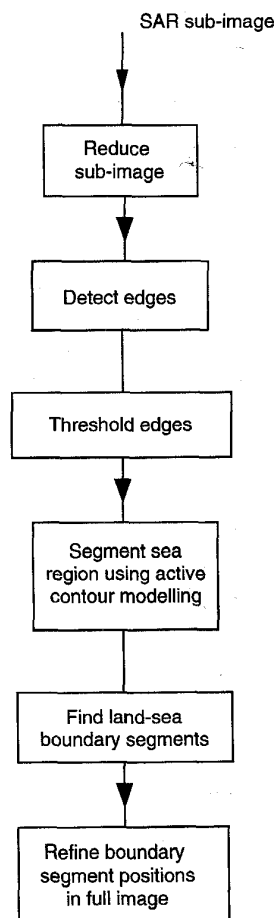


Fig. 5. Steps in the fine resolution processing stage.

fine resolution rectangle is the containing box expanded by 2δ in the x direction [e.g., rectangle B in Fig. 4(f)]. If the segment is a closed loop with a containing box less than w in both dimensions, it is assigned a fine resolution rectangle which is the containing box expanded by 2δ in both directions. If the overlap between adjacent rectangles exceeds a certain threshold, these are replaced with their containing rectangle [e.g., rectangle C in Fig. 4(f)]. This algorithm can lead to a small amount of overscanning so that some positions on the shoreline may be determined more than once when scanning adjacent rectangles.

The coarse processing stage will fail to detect inlets and peninsulas thinner than approximately half the texture edge window width (1 km), and if these are longer than about half the length of the full resolution rectangle (about 3 km or so) their extremes may be lost. In this example, the coarse processing has also failed to detect the river region west of the Humber Bridge [Fig. 4(f)]. These deficiencies could be corrected by employing a more elaborate algorithm in the fine processing stage. However, this is not a serious limitation for the application considered, as the accuracy of the heights from the hydrodynamic model will, in any case, reduce within long narrow river estuaries.

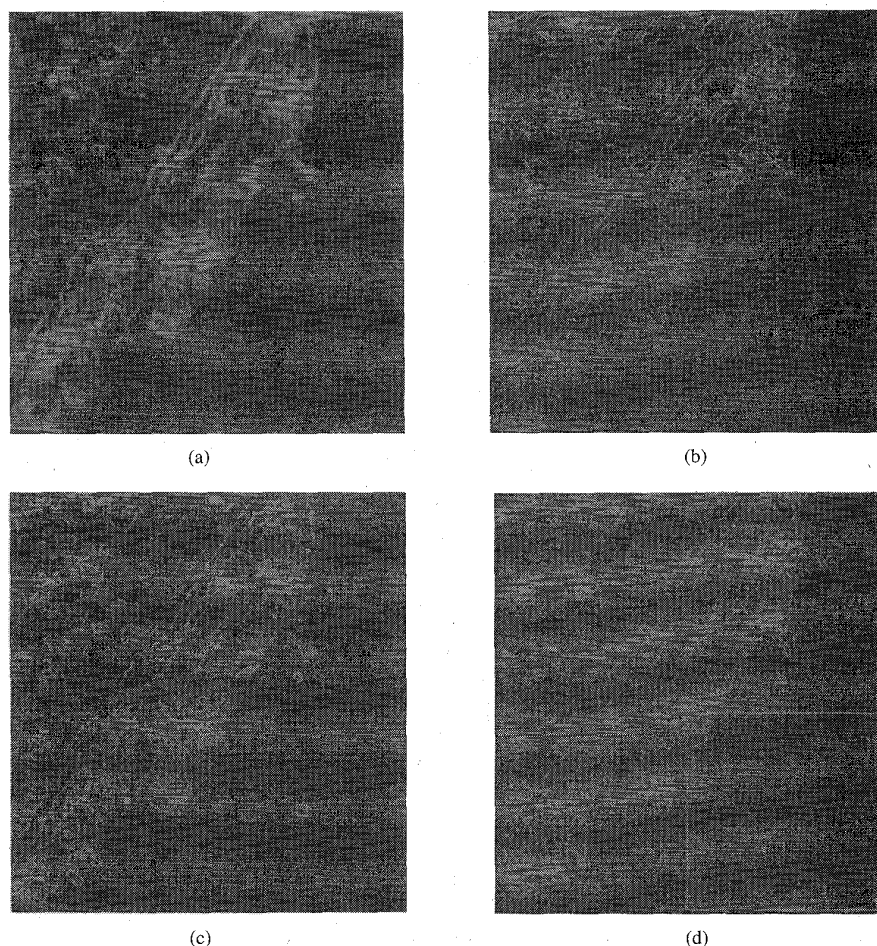


Fig. 6. (a) Contrast ratio edge detection applied to Fig. 2. (b) Edges from hysteresis thresholding. (c) Active contour model "seed" region overlain on edges. (d) Active contour model after 140 iterations.

It would be possible to replace steps 1) to 6) of the coarse processing stage by a manual delineation of the coastline obtained by digitising a map. However, changes that have evolved since the map was produced, such as small sandbanks near the coastline, may then be missed.

2.2. Fine Resolution Processing Stage

More elaborate processing is then applied at higher resolution to the image within each rectangle comprising the shoreline "mask" (Fig. 5). To speed processing, all the steps except the final one are applied to reduced-resolution versions of each subimage and its derivatives. Each subimage is reduced by averaging (again quadratically) using 2×2 -pixel nonoverlapping windows to achieve a $\times 2$ reduction in subimage side. The steps in the fine resolution processing chain are illustrated below by applying them to the subimage shown in Fig. 2 [rectangle C of Fig. 4(f)].

1) Edges are detected in the reduced subimage by applying the contrast ratio edge operator over a single window size of 11×11 (reduced) pixels, which was found empirically to produce substantial edge strengths at the shorelines (Fig. 6(a), in which strong edges are bright). In contrast, the Prewitt

edge detector acting over the same window size gave less discrimination between shoreline edges and edges in the sea region.

2) An adaptive edge threshold calculation is then performed in order to separate pixels within sea regions from pixels at the shoreline. The calculation is carried out within a mask region containing an almost pure sample of sea pixels. This mask is the intersection of the sea region found in the coarse scan with the domain of the fine resolution rectangle. Prior to intersection, the sea region is shrunk to remove areas near the shoreline. The resulting mask may still contain strong spatially correlated edges due to small sandbanks or ships not detected in the coarse scan. These false sea edges are again suppressed by eliminating from the mask all edges above a fixed large edge threshold, together with adjacent edges within a distance of half the contrast ratio edge detector window width. On the assumptions that all pixels within the mask are sea pixels, that sea pixels generally have lower edge strengths than shoreline pixels, and that sea regions exhibit a characteristic form of edge strength distribution having mean μ and standard deviation σ , empirical upper and lower edge thresholds are set at $T_{he} = \mu + 5.5\sigma$ and $T_{le} = \mu + 3\sigma$, respectively. This produces a low false alarm rate of typically 1 in 10^4 .

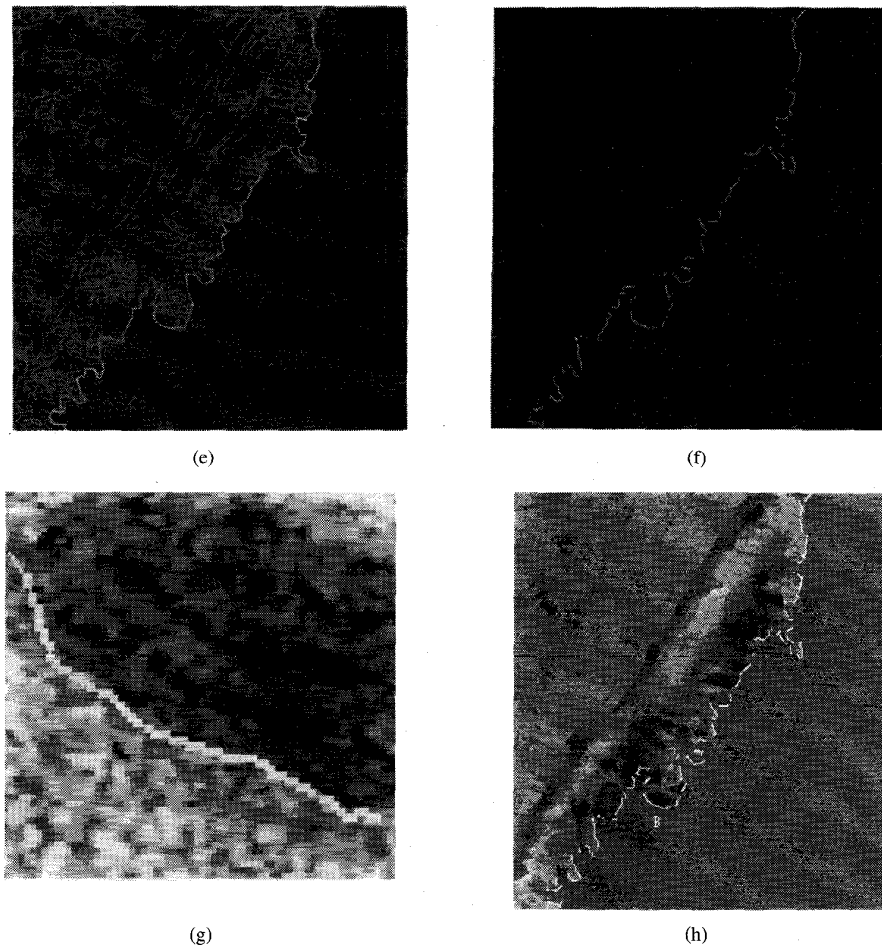


Fig. 6. (Continued.) (c) Edges linked by active contour model. (f) Shoreline segments. (g) Detail of the refined shoreline at B in Fig. 6(h). (h) Refined shoreline segments overlain on Fig. 2.

sea pixels within the mask above the upper threshold. After suppressing nonmaxima, hysteresis thresholding is applied to the edge image using these thresholds [Fig. 6(b)].

3) The next problem is to extract from the set of thresholded edges that subset comprising shoreline edges. A common way to do this (indeed that used in the coarse scan) is to link together adjacent edges to form closed regions, then to segment the subimage and extract sea regions as large connected regions of low edge density touching the subimage frame. This amounts to a general segmentation of the subimage and is, in fact, unnecessary, as it fails to exploit knowledge of the location of the sea that has been obtained from the coarse processing. An alternative method is employed that uses this information to limit processing to the sea region. This uses a continuous curve model of the sea region, and tries to localize the portion not intersecting the image frame on the subset of edges which are shoreline edges. The use of a closed contour gives the ability to span gaps in the shoreline edges.

The use of active contour models or "snakes" to extract features of interest in images was introduced in [16], and developed further in, e.g., [17]–[20]. The contour is represented by the vector $\mathbf{v}(s) = (x(s), y(s))$ where s is arc length. The

energy function is

$$E_{\text{snake}}^* = \int E_{\text{int}}(\mathbf{v}(s))ds + \int E_{\text{image}}(\mathbf{v}(s))ds + \int E_{\text{con}}(\mathbf{v}(s))ds$$

where E_{int} represents the internal energy of the contour, E_{image} the energy due to image edges, and E_{con} the energy due to the external constraints. Contours are sought which correspond to local minima of this function. The internal energy comprises components due to internal tension and stiffness forces

$$E_{\text{int}} = (\alpha|\mathbf{v}_s(s)|^2 + \beta|\mathbf{v}_{ss}(s)|^2)/2 \quad (1)$$

where the first-order derivative term has larger values when there is a large spacing between adjacent points on the curve, and the second-order derivative term will be larger where the curve is bending rapidly.

In the current implementation, the sea region model is initialized using knowledge from the coarse scan. A "seed" snake is constructed around the boundary of the coarse scan pixel that is farthest from the coarse shoreline and the rectangle frame in the intersection of the coarse scan sea region and the

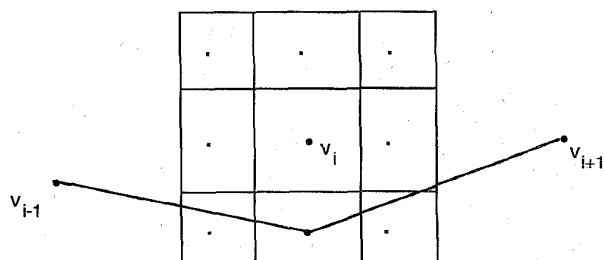


Fig. 7. Positions at which the energy function is computed (after [19]).

current fine resolution rectangle [Fig. 6(c)]. In the absence of external and image forces, the internal forces of the contour will make the snake contract upon itself. To overcome the internal forces, an external pressure force is used to inflate the snake in a similar manner to a balloon. The pressure force is modified in the vicinity of edges. When the snake intersects edges, it is stopped if the intersection is with a length of edge that is sufficiently substantial to overcome the pressure force, but passes over short isolated edge segments. This "seed" approach differs from that of a multiscale snake [21], which would fail to delineate offshore sandbanks too small to have been detected in the coarse processing stage.

The implementation is a modified form of the iterative algorithm of [19], which is considerably faster than other methods. During each iteration, a neighborhood of each snake point v_i ($i = 0, \dots, n$) is examined, and the energy function is computed for the current location of v_i and each of its eight neighbors (Fig. 7). The location having the smallest value is chosen as the new position of v_i . The curvature term in (1) is estimated by the method given in (5) of [19], which has the desirable property that the rate of change of curvature continues to increase as curvature increases, so that it becomes impossible to exert sufficient force to make the angle contained by three adjacent snake points very small. Values of $\alpha = 0.1$ and $\beta = 0.2$ were found to give satisfactory results over a wide range of images. To stop the snake intersecting itself, a two-dimensional accumulator array is used [20].

Active contour models are normally applied to unthresholded edge images, but in this case, a binary edge image is being used. This is because the requirement of the hysteresis thresholding procedure that any edge pixel accepted above T_{le} must be connected to a stronger edge pixel above T_{he} means that many fewer edge pixels above T_{le} are accepted in the sea region than would be the case if simple thresholding using T_{le} was being applied. As a result, these edge pixels cannot act as noise edges which trap the snake before it reaches the shoreline edges. The behavior of the snake thus becomes more predictable.

In order to make the snake expand, an external pressure force is applied at snake points away from image edges. In a similar manner to [20], the forces due to image edges and the external pressure are combined into a single pressure force which is modified depending on whether or not the snake point coincides with an edge. The pressure force is

$$F = (j \cdot n_i)G$$

where n_i is the unit vector normal to the curve at point v_i , j is the vector describing the proposed move of v_i from its current position ($j = (k, l)$ with $-1 \leq k, l \leq 1$), and

$$G = 1 - 2(e_{00} - e_{kl})$$

where e_{kl} is the edge strength (0 or 1) at (k, l) in the 3×3 neighborhood (Fig. 7). Thus, away from image edges, where $e_{00} = e_{kl} = 0$, $G = 1$, and the pressure force makes the snake expand rapidly. If the snake point is in the vicinity of an edge, such that $e_{00} = 0$ and $e_{kl} = 1$, the pressure forcing the snake point onto an edge becomes maximal, with $G = 3$. However, if the snake point coincides with an edge ($e_{00} = 1$), and one attempts to move it away from the edge ($e_{kl} = 0$), G becomes -1 , the direction of the pressure force reverses, and the snake is attracted back to the edge. In addition, if a snake point is on an edge and one attempts to move it to a second point on the edge, G is again set to -1 to avoid snake points being moved along edge segments and bunching together at their ends.

Adjacent snake points become separated as the snake expands, and additional points must be added to preserve spatial detail. Similarly, if two snake points become too close together one or other may be removed to reduce processing requirements. A point is also removed if it has a high negative curvature associated with it, as this invariably indicates a short "noise" edge upon which this single snake point is sticking. A reparametrization step is included at the end of each iteration [20].

Fig. 6(c) shows the initial seed snake, Fig. 6(d) the snake after 140 iterations, and Fig. 6(e) the final snake after 450 iterations.

4) Shoreline segments of high confidence are then identified. The snake intersects image edges only at the snake points. In order to identify the subset of edges which have been "linked" by the snake, the next step extracts strong image edges near the snake that have similar orientation to the snake. All edge pixels accepted by hysteresis thresholding that are within two pixels of the snake and within 60 degrees of its local orientation are selected. This is achieved using a distance transform in which each pixel in the transform image has associated with it not only the distance to the nearest point on the snake, but also the orientation of the snake at this point. Those edge segments selected are thinned and vectorized. Each vector segment is then traversed to test whether the mean intensity m at a distance of half the contrast ratio edge detector window width away on its seaward side is similar to that of the sea region it bounds, i.e., whether $(m - \mu)/\sigma' < 2$, where μ' and σ' are the mean and standard deviation of the whole sea region. Edge segments passing this test are accepted as shoreline segments [Fig. 6(f)].

5) Finally, the positions of the shorelines identified in the half-resolution subimage are refined using the full resolution subimage. This is necessary not only because of the increase in resolution, but also because at concavities and convexities in the shoreline having a radius of curvature comparable with the contrast ratio edge detector width, the edge detector gives a maximum response which may be shifted from the true position by one or two pixels.

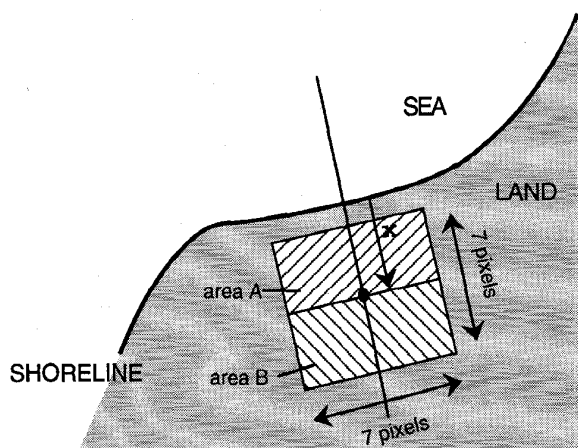


Fig. 8. Refinement of shoreline positions.

First, the true edge position at each point on a shoreline segment is determined precisely, treating all shoreline pixels independently. At each point, the edge strengths obtained by applying a contrast ratio edge detector perpendicular to the sea edge and centered at pixels along a line perpendicular to this edge (line x in Fig. 8) are determined. The detector width is reduced in size from that used in step 1), being seven pixels along the shoreline and seven pixels perpendicular to it. Initially, the position of the minimum contrast ratio within three pixels of the original edge location is found. If this minimum is less than 0.6, its position is taken as the true edge location. If not, the minimum within an extended search distance of twenty pixels is found, and its position becomes that of the true edge if the minimum is less than 0.6. If no such point is found, the original edge position is retained.

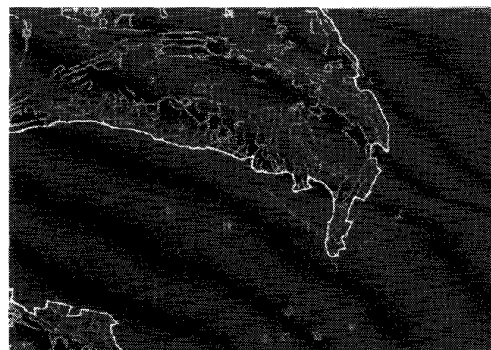
A fitting procedure is then used to remove any discontinuities in the resulting shoreline positions. A moving quadratic fit is performed at each point on a shoreline segment using the positions of the point and its three neighbors on either side. Prior to the fit, the points are transformed so that the x -axis of the coordinate system used for fitting passes through the positions of the two furthest neighbors. After fitting the reverse transformation is applied. Any point which is more than one pixel away from its fitted position is replaced by the fitted position, otherwise its position remains unchanged. Fig. 6(g) shows the refined shoreline in detail in the small region at B in Fig. 6(h). Fig. 6(h) shows the refined shoreline segments overlain on the original image of Fig. 2.

III. RESULTS AND DISCUSSION

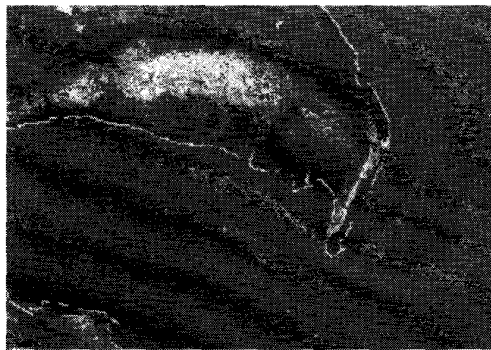
Figs. 9 and 10 show two further examples of detected shorelines in subimages taken at different tidal states or positions in the LOIS study area. Fig. 9 is an image of Spurn Head taken at low tide. Fig. 10 is another image of Wrangle Flats taken at a lower tide than that of Fig. 2, and is included to show how the active contour modeling copes with sandbanks. Figs. 2, 9, and 10 are a representative sample from a much larger set of images which have been processed by the method. This set includes 50 images of the Wrangle Flats area taken at different tidal states. The same parameter settings were



(a)



(b)



(c)

Fig. 9. (a) Spurn Head (1344×928 pixels) (copyright ESA). (b) Edges linked by active contour model. (c) Refined shoreline segments overlain on original.

used for all images. Over 90% of the shoreline determined appears visually correct. The remainder may be corrected using a sophisticated manual editing facility. Editing options include deletion of incorrect shoreline segments and insertion of new ones. This is the only stage in the delineation procedure at which user interaction is required.

Delineation errors arise mainly due to the difficulty of setting a contrast ratio edge threshold which selects all distinguishable shoreline segments, and to limitations in the active contour modeling. Some small errors are apparent in Figs. 9 and 10, for example, in Fig. 10, along the northern part of the northeast sandbank, and the delineation of the linear feature

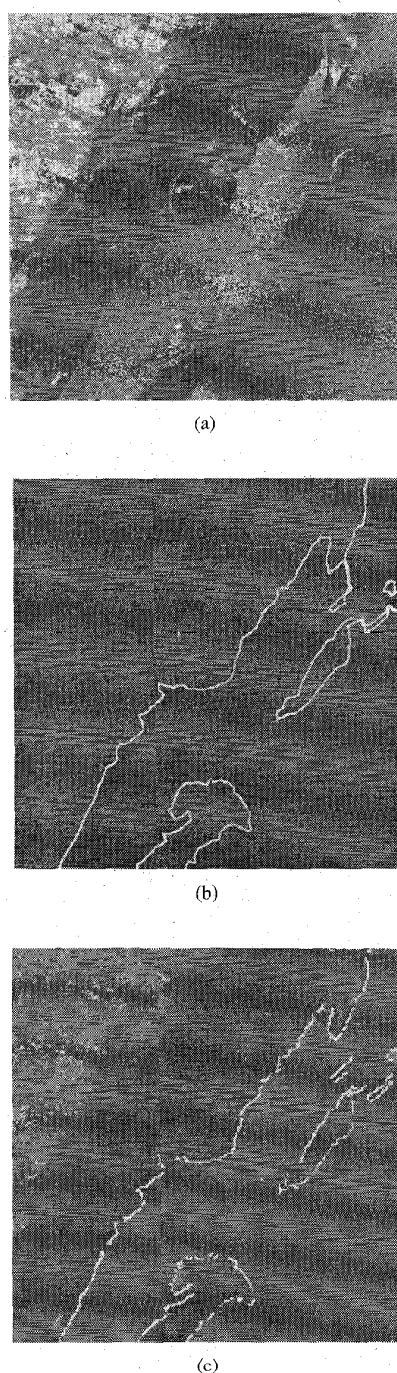


Fig. 10. (a) Wrangle Flats at low tide (1024×1024 pixels) (copyright ESA). (b) Edges linked by active contour model. (c) Refined shoreline segments overlain on original.

in the sea. The results could probably be improved at the expense of increased computation by using more sophisticated processing and/or higher-level information in areas of interest. For example, the simulated annealing algorithm of White [22], although requiring a large computational effort if applied over a complete SAR scene, could be applied as an adaptive smoothing filter in the neighborhood of the shoreline between the coarse and fine resolution processing stages.

A further problem is whether the "visually correct" shoreline segments determined from the image actually correspond to true shoreline positions. SAR processing of a moving sea surface can give rise to ambiguities in position in a number of ways [23]. For example, calculations indicate that a breaking wave with a component of velocity in the radar's range direction may be displaced by several pixels in the azimuth direction, so that if a very nearshore breaking wave of sufficient scattering cross section was actually travelling at 45 degrees to the radar, its image could possibly be displaced onto the beach. In the present system, potential problems due to motion effects are alleviated because waves are not taken into account in the current hydrodynamic model, and to minimize elevation errors due to wave setup [24], only images acquired during periods of low wave activity are used to construct the DEM. To confirm the accuracy of the shoreline positions estimated from the imagery, ground measurements were made at time of satellite overpass during a number of field visits. The shoreline positions determined from the images were found to agree with the measured ground positions to within a pixel or so.

The time required to process a complete ERS-1 SAR scene is 1–2 h on a Sun SparcStation 10/50 using C-coded algorithms. The computationally intensive processes are the averaging and edge density calculation in steps 1) and 2) of the coarse resolution processing stage, and the contrast ratio edge detection and active contour modeling in the fine resolution processing stage.

Comparison of the present method with that of [1] is not straightforward because the two are intended for somewhat different applications. Development of our algorithm has been strongly driven by the need to construct an intertidal DEM over a large area from shorelines obtained at different tide/surge states. In [1], the shoreline extracted is a continuous curve, whereas the present method determines the shoreline only where it is deemed to be distinct. Detailed examination of the images indicates that the latter is a more realistic course of action. In the discussion of the method of [1], it is stated that further refinement of the shoreline position could be achieved by searching the pixel neighborhood in the direction perpendicular to the detected shoreline. This occurs in our final edge refinement step using a contrast ratio edge detector applied to the full resolution image, allowing single-pixel accuracy to be achieved. Finally, the present method is designed to process complete SAR scenes efficiently, whereas that of [1] was applied to 512×512 -pixel scenes only.

IV. CONCLUSION

A semiautomatic method of detecting the shoreline in ERS-1 SAR images has been presented which combines both accuracy and speed. This adopts a hierarchical processing approach, with the coarse resolution processing stage feeding knowledge of the approximate location of the sea to a full resolution processing stage which is only required to perform a partial segmentation of the image using an active contour model. Although the method has been developed for ERS-1

SAR images, it should be possible to modify it to work with other SAR sensors.

ACKNOWLEDGMENT

The authors wish to thank C. Blight for useful discussions on active contour modeling, and R. Flather and B. McCartney for discussions regarding the application.

REFERENCES

- [1] J. S. Lee and I. Jurkevich, "Coastline detection and tracing in SAR images," *IEEE Trans. Geosci. Remote Sensing*, vol. 28, pp. 662-668, 1990.
- [2] A. W. Pringle, "Holderness Coast erosion and the significance of Ords," *Earth Surface Processes Landforms*, vol. 10, pp. 107-124, 1985.
- [3] R. A. Flather and K. P. Hubbert, "Tide and surge models for shallow water—Morecambe Bay revisited," in *Modeling Marine Systems*, A. M. Davies, Ed. Boca Raton, FL: CRC Press, vol. 1, 1989, pp. 136-166.
- [4] B. N. Koopmans and W. Yiman, "ERS-1 imaging radar over the Waddensca-N, Netherlands," in *Proc. First ERS-1 Symp.*, Cannes, France, Nov. 4-6 1992, pp. 861-865.
- [5] J. R. Jensen, D. J. Cowen, J. D. Althausen, S. Narumalani, and O. Weatherbee, "The detection and prediction of sea level changes on coastal wetlands using satellite imagery and a geographic information system," *Geocarto Int.*, vol. 4, pp. 87-98, 1993.
- [6] A. H. Brampton, "Coastline monitoring," Hydraulics Research, Wallingford, U.K., Report IT 345, Feb. 1990.
- [7] D. C. Mason, I. Davenport, R. A. Flather, B. McCartney, and G. R. Robinson, "Development of an inter-tidal digital elevation model for the U.K. Land-ocean interaction study using ERS-1 data," in *Proc. First ERS-1 Pilot Programme Workshop*, Toledo, Spain, June 22-24 1994, pp. 337-342.
- [8] LOIS Science Plan, NERC, Swindon, U.K., 1992.
- [9] D. C. Mason, C. Oddy, A. J. Rye, S. B. M. Bell, M. Illingworth, K. Preedy, C. Angelikaki, and E. Pearson, "Spatial database manager for a multi-source image understanding system," *Image Vision Comput.*, vol. 11, no. 1, pp. 25-34, 1993.
- [10] G. Sawyer, D. C. Mason, N. Hindley, D. G. Johnson, I. H. Jones-Parry, C. J. Oddy, T. K. Pike, T. Plassard, A. J. Rye, A. de Salabert, B. Serpico, and A. Wielogorski, "MuSIP Multi-Sensor Image Processing system," *Image Vision Comput.*, vol. 10, no. 9, pp. 589-609, 1992.
- [11] T. W. Ryan, P. J. Sementilli, P. Yuen, and B. R. Hunt, "Extraction of shoreline features by neural nets and image processing," *Photogrammetric Eng. Remote Sensing*, vol. 57, no. 7, pp. 947-955, 1991.
- [12] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*. London, U.K.: Chapman and Hall, 1993.
- [13] R. Touzi, A. Lopes, and P. Bousquet, "A statistical and geometrical edge detector for SAR images," *IEEE Trans. Geosci. Remote Sensing*, vol. 26, pp. 764-773, 1988.
- [14] R. G. Caves, "Automatic matching of features in SAR data to digital map data," Ph.D. dissertation, University of Sheffield, U.K., 1993.
- [15] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 679-698, 1986.
- [16] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Int. J. Comput. Vision*, vol. 1, pp. 321-331, 1987.
- [17] A. A. Amini, S. Tehrani, and T. E. Weymouth, "Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints," in *Proc. 2nd Int. Conf. Comput. Vision*, 1988, pp. 95-99.
- [18] L. D. Cohen, "On active contour models and balloons," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 211-218, 1991.
- [19] D. J. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *CVGIP: Image Understanding*, vol. 55, no. 1, pp. 14-26, 1992.
- [20] J. Ivins and J. Porrill, "Statistical snakes: active region models," in *Proc. British Machine Vision Conf.*, York, U.K., Sept. 1994, pp. 63-72.
- [21] D. Geiger, A. Gupta, L. A. Costa, and J. Viontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 294-302, 1995.
- [22] R. G. White, "A simulated annealing algorithm for radar cross-section estimation and segmentation," *SPIE Int. Conf. Applications Artificial Neural Networks V*, Orlando FL, Apr. 4-8 1994, pp. 2243-2225.
- [23] F. T. Ulaby, R. K. Moore, and A. K. Fung, *Microwave Remote Sensing*. Reading, MA: Artech House, 1986, vol. 3, pp. 1722-1726.
- [24] B. A. King, M. W. L. Blackley, A. P. Carr, and P. J. Hardcastle, "Observations of wave-induced set-up on a natural beach," *J. Geophys. Res.*, vol. 95, no. C12, pp. 22289-22297, 1990.



David C. Mason received the B.Sc. and Ph.D. degrees in physics from the University of London, U.K., in 1963 and 1968, respectively.

He was with the U.K. Medical Research Council as a Staff Scientist automating human chromosome karyotyping from 1969 to 1978, and then on military image processing with Plessey Electronic Systems Research from 1978 to 1983. Since 1984, he has been the Deputy Director of the NERC Environmental Systems Science Centre, carrying out spatial data handling techniques research for the environmental sciences. His research interests include the automated extraction of information from remotely sensed images, the design of databases for very large images and maps, hierarchical spatial data structures, and higher-dimensional geographic information systems.



Ian J. Davenport received the B.Sc. degree in physics with astrophysics from the University of Kent, U.K., in 1990, and the Ph.D. degree in astronomy from the University of London, U.K., in 1994.

His previous research projects include an initial study into the Cassini Huygens space probe and the development of ultra-cooled X-ray detector systems. Since 1994, he was a Research Fellow with the NERC Environmental Systems Science Centre in coastal zone research.