Morphological Image Processing Techniques Applied To Detection of Correlogram Tracks

Ben Rosen and Luc Vincent*

November 5, 1993

Atlantic Aerospace Electronics Corporation
470 Totten Pond Road
Waltham, Massachusetts 02154
617-890-4200 (Voice)
617-890-0224 (FAX)

U.S. Navy Journal of Underwater Acoustics, Vol. 44, No. 2., April 1994.
(UNCLASSIFIED).

*Presently at Xerox Imaging Systems, 9 Centennial Drive, Peabody MA 01960.

Abstract

Image processing (IP) techniques are applied to problems of detection and classifications of submarines in low received signal-to-noise (SNR) situations. While incoherent integration techiques, applied to persistent low SNR signals, improves detectibility, these often fail when significant signal dynamics are present. Results on implementing computationally-efficient morphological image processing techniques for passive broadband correlogram line detection and classification, sponsored by the Naval Air Warfare Center, Aircraft Division Warminster, are described here. Low SNR signals, received at several sensors of an array of sensors, are cross-correlated. Despite correlation gain, with no strong correlation peaks available for detection, morphology-based IP techniques can detect and track. We develop the ideas of track-before-detect: using IP to identify curvilinear features (tracks) in images and accepting or rejecting tracks, not peak intensity points, as potential detections. This basis of this approach is to minimize a generalized distance function. The algorithms were tested on simulated and real Navy-supplied data. This report describes the nature of the algorithms but does not attempt to quantify their performance. The application of this technology is motivated by the need of sonar operators in a multi-sonobuoy tactical scenario to rapidly view multiple displays and is useful as an automatic detection or operator-alert mechanism.

1 Introduction

The passive detection and localization of undersea targets is a difficult problem especially when the target radiates very low levels of acoustic energy. The problem of localization is further complicated when the environment exhibits multipath propagation and/or when there are multiple targets. Both omni-directional and DIFAR sensors are used to receive narrowband and broadband acoustic energy. At a sonar operator's disposal is an assortment of processing capabilities such as, for example, narrowband and broadband sonar displays. The narrowband display might be based upon a short-time Fourier transform followed by a display of the time-evolution of the Fourier coefficients (such as a Lofargram). A broadband display could be a short-time cross-correlation followed by a display of the time-evolution of the correlation function (i.e. a correlogram). In response to the results of increased noise quieting efforts, the sonar system designer might undertake several strategies such as (1) increasing the density of sensors in a given area (2) making use of the improved gains obtainable by spatial processing and (3) developing sensors with increased sensitivity. In any of these scenarios, the demands placed upon sonar operators grow and they might have difficulty keeping up with the increased amount of input data and resultant information flow. Therefore, either automatic detection, and/or operator-cueing aids will be important additions to any future sonar information processing system.

The passive sonar detection problem divides into two components: (1) detection and estimation of target signal parameters such as bearing, time-delay of arrival, and significant frequency components and (2) mapping detections into those belonging to individual underwater objects which can then be localized and tracked (data association). The algorithms which are used in passive source localization are generally nonlinear functions of the measured signal parameters and localization errors are often very sensitive to errors in the measured signal parameters.

Improvements made in submarine noise quieting have increased the importance of using signal processing methodologies which take into account all the radiated acoustic energy emitted by a submarine, such as Full Spectrum Processing. For broadband signals, considerable processing gain can be attained by cross-correlating signals from pairs of sensors, a gain on the order of the processing-time-bandwidth product ($T \cdot BW$). While the methodologies described herein were motivated by the problems which arise in cross-correlating data from air-deployed sonobouys, the techniques are certainly applicable to other systems such as correlated segments of a split-beam correlator array.

2 Detection Scenario

A representative sonar scenario that illustrates the problems discussed in this paper is shown in Figure 1. An array of sonobuoys is deposited in a V-shaped formation in the vicinity of a submerged target. The overall extent of the array is typically on the order of kiloyards and typically consists of dozens of sonobuoys. Sonobouy signal are transmitted to an overflying aircraft and examined by a sonar operator. In this example, energy from a submerged target is strong enough to be registered at several sonobuoys. Other acoustic signals which may be received are biological interference, nearby and distant shipping noise, ice-generated and seismic noise. The job of the sonar operator is to detect and discriminate likely target signals from noise and interference.

Consider the following very-simplified model of the received signals at two sonobuoys from which a

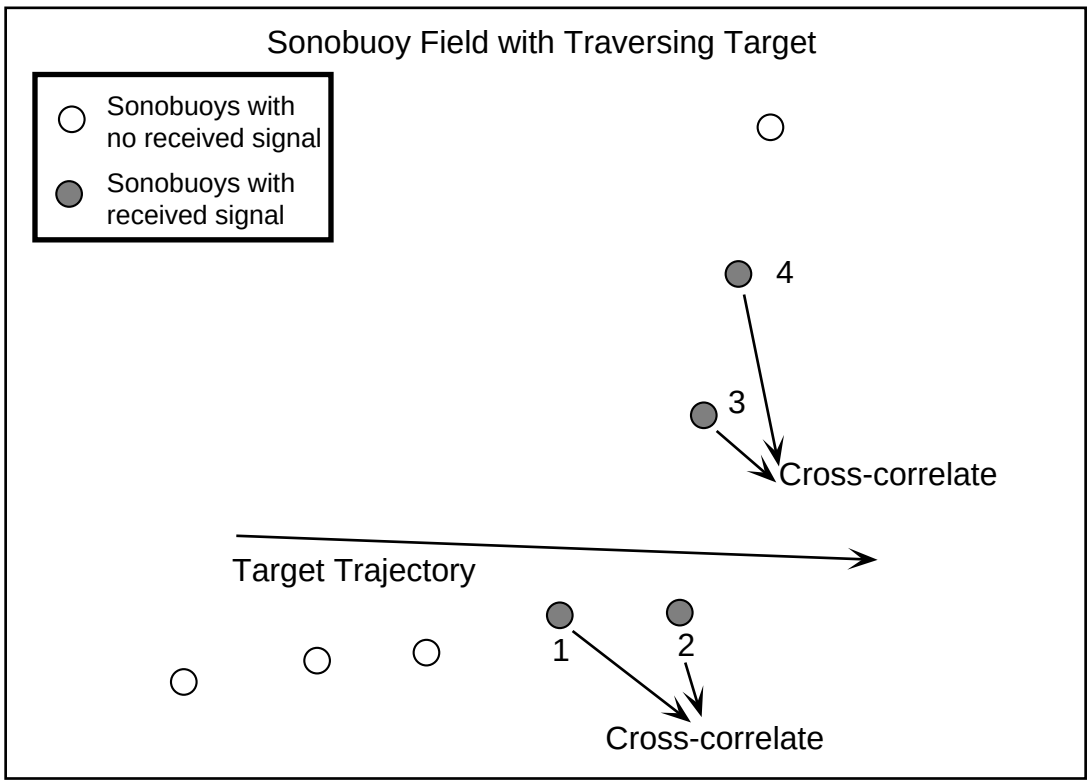


Figure 1: Sonobuoy target tracking scenario

correlogram display is to be generated. The signal $s(t)$ is assumed to be a zero-mean Gaussian random waveform in a background of zero-mean Gaussian random noise. For an infinite, homogeneous medium, the signals received at the two sonobuoys are:

$$r_1(t) = s(t) + n_1(t) \tag{1a}$$
$$r_2(t) = As(t - D) + n_2(t) \tag{1b}$$

where A is the relative amplitude of the received signal at sensor 2, D is the delay in time-of-arrival at sensor 2 with respect to sensor 1, which satisfies $|D| \leq L/c$ where L is the distance between sensors and c the speed of sound. The noises, $n_1(t)$ and $n_2(t)$, assumed to be independent, identically distributed, are uncorrelated between sensors. (While the assumptions of Gaussian signal and noise are oversimplifications, they provide a useful means of generating simple test images upon which development of the image processing techniques can proceed; these assumptions should certainly be discarded when assessing the operating performance of the detection algorithms developed here). The received signals are discretized and segmented into data frames of duration $T = Mt_s$ with t_s being the sampling interval. The discrete correlation operation is performed yielding the correlation function

$$\hat{R}(n) = \frac{1}{M - |n|} \sum_{k=0}^{M-|n|-1} r_1(k) r_2(k + n) \tag{2}$$

for $|n| \leq M$.

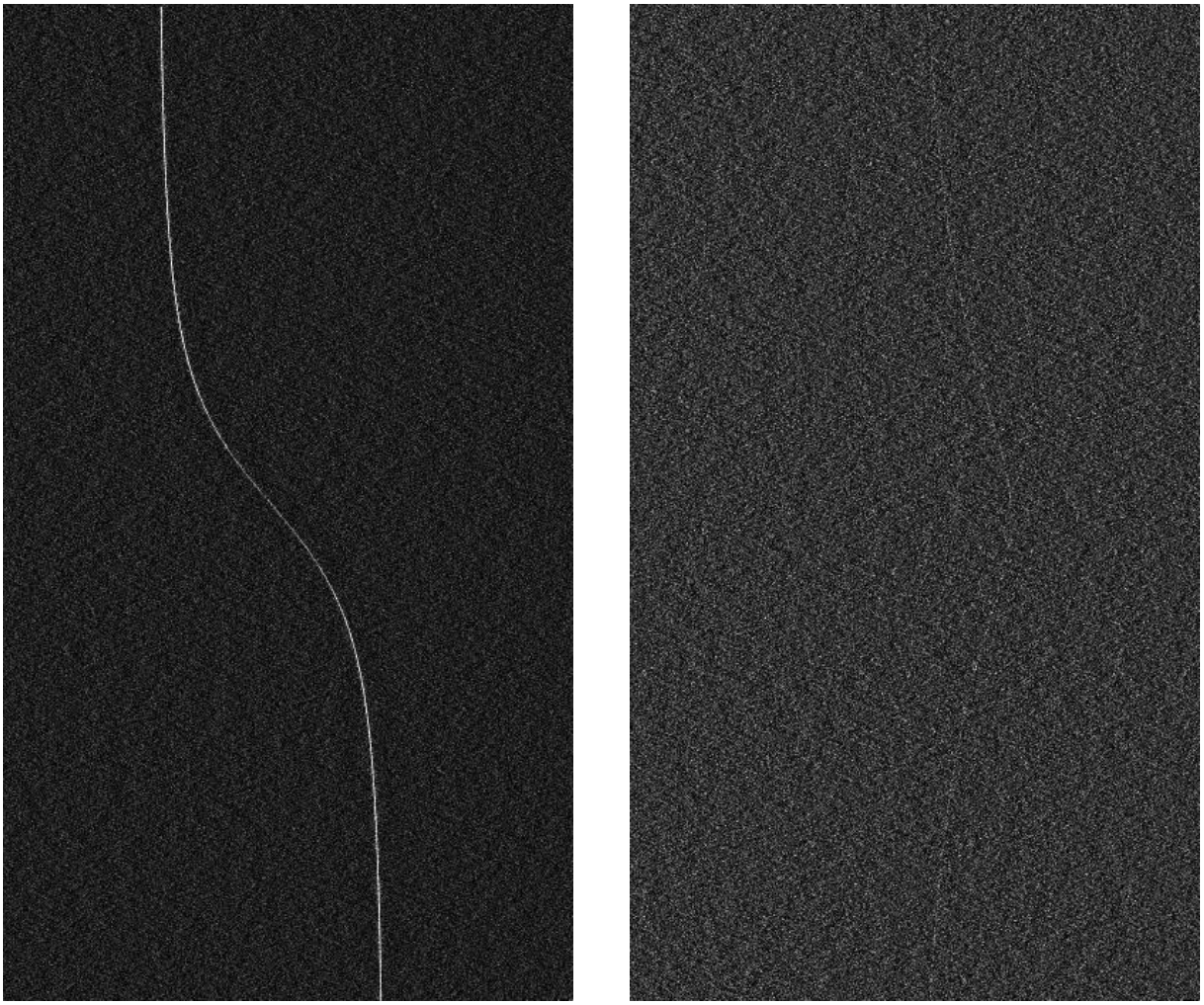
When the segmented correlation functions are stacked, a correlogram image is formed. The horizontal location of each pixel corresponds to the discrete time-delay (τ_i)-bin and its vertical location corresponds to time t_k. Figure 2a is a simulated correlogram generated from broadband Gaussian signals received at sensors 1 and 2 arriving from the close-in target in Figure 1. Figure 2b shows the correlogram produced by cross-correlating the weaker signal (farther away) received at sensor 3 with sensor 4. The received sensor-level SNR's in this scenario were chosen to be -6 dB (ref: 1 Hz BW) at each sensor of pair 1–2 and -14 dB at sensor pair 3–4. A broadband target in the correlogram domain shows up as a thin curved line with the width being inversely proportional to the bandwidth of the signal. The signal levels in each figure were not chosen to be realistic but chosen to illustrate the different IP techniques discussed in the next section.

Pairs of sonobuoy can be used to estimate the direction of arrival of the acoustic energy. For a distant source (range much greater than L), the direction of arrival is closely related to the time-delay-of-arrival of the signal at the pair of sensors (τ).

$$\cos\theta = \frac{c\tau}{L} \tag{3}$$

where θ is the angle-of-arrival with respect to the baseline between the pair of sonobuoys. An estimator of the time-delay of arrival is given by the τ-value corresponding to the peak of the correlation function. In fact, this is the maximum-likelihood estimator [5, 4] for sufficiently high signal-to-noise ratio. For low SNR cases, when noise peaks may be larger than the correlation peak, extant algorithms, such as ADAC and PDADAT [2], are capable of detecting and associating low correlation peaks and forming tracks in the correlation domain.

For fixed targets, increasing the processing time T increases the processing gain, and thus targets may be detected at lower SNR. Theoretically the processing time could be extended without limits



(a) *(b)*

Figure 2: Sample track images (a) high SNR track (b) low SNR track.

for a fixed target and thus continue to lower the SNR required for detection. Even in this case there will be a limit imposed by the presence of some correlated noise in the surrounding environment. Beyond that point, no additional processing gain may be realized. However, for moving targets (i.e. sources for which τ varies as a function of time), the maximum allowable value of T should generally be limited to the time required for τ to change by an amount equal to the resolution with which it can be measured. This resolution is inversely proportional to the bandwidth of the source signals

$$\frac{1}{BW} \geq \Delta\tau = \dot{\tau}_{max} T \tag{4}$$

$$\frac{L}{c} \ll T \leq \frac{1}{BW\dot{\tau}_{max}} \tag{5}$$

where $\dot{\tau}_{max} = \left(\frac{d\tau}{dt}\right)_{max}$ is the maximum expected range of change of the delay. $1/BW$ is the resolution with which τ can be estimated from the correlation function. Targets in the near field of the sonobuoy array exhibit strong dynamics which will weaken the correlation function. The larger the bandwidth, the shorter the maximum processing-time-bandwidth product for a fixed correlation-rate of change, because larger bandwidths make the gain more susceptible to Doppler degradation. Any correlation tracking algorithm must be robust under these conditions.

Most correlogram track detection techniques are local in that they detect exceedences of individual pixels above a threshold and generate target tracks by connecting temporal sequences of detections. We present here a global method where the track itself is detected rather than its constituent points. It looks at a large portion of the tau-time plane and determines that track or path through the plane which, in a crude sense, has a signifcant integrated intensity. Strictly speaking, the method minimizes a *generalized distance function*. The minimum distance algorithm (MDA) method is robust when it comes to signal fading due to Doppler. The MDA approach is successful when the signal-to-noise ratio is large but also works well at low SNR cases. An important advantage of this technique is that it is self-initiating; no apriori knowledge of starting locations of correlation lines is needed. The algorithm belongs to a branch of image processing which is called morphological filtering and deals with nonlinear shape-based image filtering. A complete discussion of the generalized distance function is given below. There are other global optimization methods for detecting signal tracks; one of particular interest is the application of simulated annealing methods to frequency line tracking [8]. This method could easily be adapted to correlation peak tracking just as the methods described here could be applied to frequency line tracking [1].

Since the MDA algorithm detects a track or line in an image, it does not assume any dynamical model for the line. As mentioned, detections are made of tracks, not individual correlation peaks. This is an example of a track-before-detect principle. In order to impart dynamical information, the line detector is followed by a state estimation algorithm which estimates a smooth time-dependent state vector, $\tau(t)$ and $\dot{\tau}(t)$. For this reason, every track detected by the MDA algorithm is the input to a state estimation algorithm which serves several purposes. It

- Estimates track states $\tau(t)$ and $\dot{\tau}(t)$,
- Prunes non-physical tracks such as those which are short-lived or those having $c\tau > L$ where L is the sensor spacing,
- Associates two or more disjoint tracks,

- Detects target maneuvers.

Combining estimated time-delays from at least two pairs of sensors produces a geographic position using the assumed known geographic positions of the relevant sensors. The state vectors are passed into a geographic tracking algorithm which combines state data from different pairs of sensors in order to localize the target. At this stage further smoothing and target track pruning may also be invoked. While a complete end-to-end processor was developed at Atlantic Aerospace, in this article, the primary emphasis is placed on the presention of the image processing aspects of the target detection problem. The remaining steps are adaptations of well-known techniques and need not be described herein.

3 Track Extraction Using Generalized Distance Functions

3.1 Introduction

Let us now consider the track detection problem from an image processing point of view: the track images (correlograms) that this study is concerned with are characterized by *high noise levels* and *thin* tracks. Thinness here is defined in the image processing sense: the width of the track (typically between 1 and 5 pixels) is much smaller than its linear extent. Despite the ambient noise, some tracks clearly stand out from the background (see for example Figure 2a). Others have extremely low contrast and are sometimes even hard to locate visually. Such is the case illustrated by Figure 2b.

Tracks with high contrast, such as in Figure 2a, are very easy to detect: any thresholding algorithm such as the morphological *top hat* [12] whereby an estimated background is subtracted from the original image, would give very good results. This is because the tracks are brighter than the background at any location along them. The *top hat* algorithm belongs to a class of morphological image processing algorithms which generate residual images. Specifically, the *top hat* operation on an image is the difference between an image and its morphological "opening". An algorithm like the *top hat* is used to locate pixel peaks and ridges based on their local sharpness and would therefore easily find the track. The *top hat* algorithm can be considered as a generalized image processing version of a traditional correlogram threshold detector.

Note that such a binarization algorithm would have to be followed by some simple noise cleaning. Moreover, the extracted tracks would have to be reduced to one pixel thickness using skeletonization techniques such as the one described in [13].

In such simple cases, traditional signal processing algorithms, detecting maxima on each line of the correlogram and tracking them from line to line, would work as well. However, the case of Figure 2b is more tricky. Based on a local examination of the pixel values in a small window, it is often impossible to decide where the track actually goes. Figure 3 is a rectangular sub-window taken from Figure 2b. Although it is relatively large (100 x 100 pixels), it is almost impossible to visually detect any track in it. A human observer, as well as a computer program, would easily be fooled. Only at larger scales can one reliably locate the track.

Hence, it appears that more *global* information needs to be used for the process of track extraction, not only local information (and in particular, not only information from a single correlogram line). In other words, even for a human operator, it is impossible to detect some tracks if the correlogram

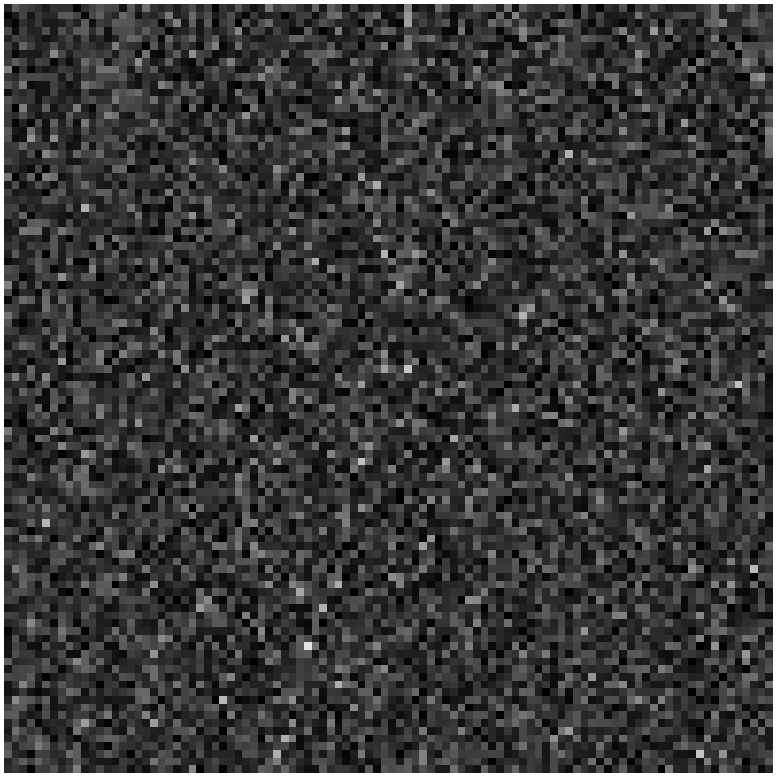


Figure 3: 100×100 pixels window taken from the center-right of Fig. 2b.

is not long enough, i.e., if the image does not have enough lines. The lower the SNR, the more lines are needed in order to detect some of the tracks. Therefore, in order for a detection algorithm to be as robust as possible, it should not decide whether a pixel belongs or not to a track by examining pixel values in a small window around this pixel: *all* the image pixels should participate in this decision. This is where image processing techniques, intrinsically bi-dimensional, will turn out to have a definite advantage compared to traditional line-based algorithms.

3.2 Experiments with the watershed transformation

With this in mind, we formulate the problem of track detection in terms of *global image operators,* which do not assign a value to a pixel based solely on the values of the pixels in a neighborhood around it. Such operators are known to be extremely powerful in image analysis, particularly for picture segmentation. Some examples are the *Skeleton by Influence Zones (SKIZ)* or the *Watershed Transformation*[3, 18]. The latter is especially useful in complex image segmentation tasks [9, 17, 15]. If one regards a grayscale image as a topographic relief, where the altitude of a pixel is proportional to its gray-level, features such as basins, crest-lines, peaks, etc, can be characterized on this image; the watershed transformation extracts the highest crest-lines of an image that separate a specific set of *markers* [15]. It draws its name by analogy with the concept of a topographic watershed line, which divides a region into various *catchment* or *drainage* basins.

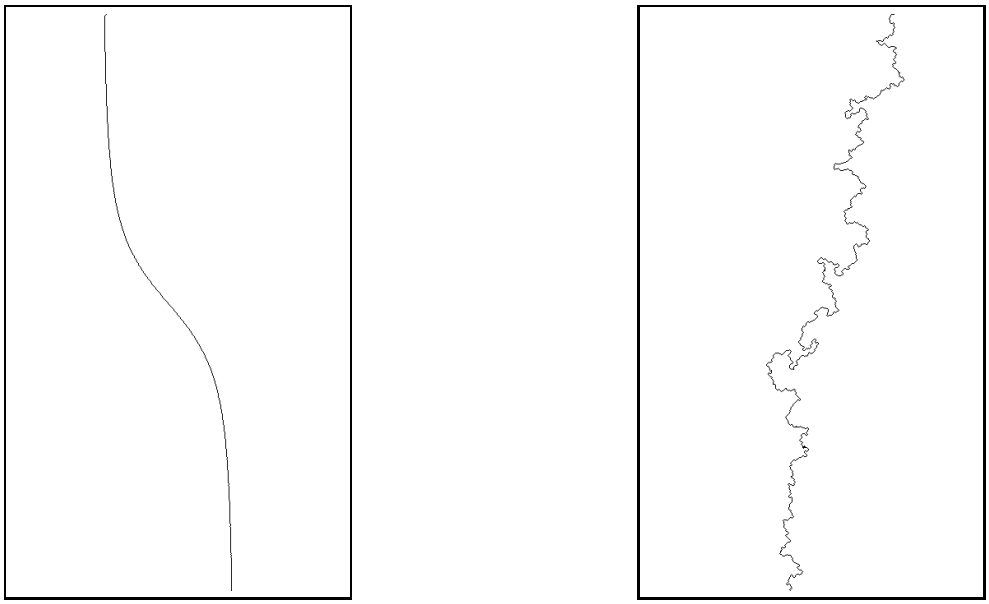
In the present case, tracks can be interpreted as crest-lines in our correlograms, and one could use *watersheds* to extract these tracks as the "highest (i.e. brightest) crest-lines separating the left side of the image from the right side". Just as for the *top hat* algorithm mentioned above, this method works well with highly contrasted tracks, as illustrated by Figure 4a. However, for the watershed transformation, the height of a crest-line is related to the average gray-level (altitude) of the pixels along it. Under high noise levels such as that observed in Figure 2b, the extracted crest-lines would be extremely tortuous, wind around and go through the largest possible number of noise pixels with high values. The watershed algorithm is therefore not a reliable track extractor, as shown in Figure 4b.

3.3 Generalized distance functions and paths of minimal cost

In order to correctly take global image information into account, the present section formulates the problem of track extraction in terms of an optimality criterion. One of the problem with *watershed track extraction* is that the length of the track is not taken into account. A better optimality criterion for track extraction should consider:

- the pixel values along the track (to maximize)
- the length of the track (to minimize)
- the raggedness of the track (to minimize)

Let I denote a discrete two-dimensional image containing potential target tracks. It is discrete since it contains a finite number of pixels $\{p\}$. Let $I(p)$ denote the positive intensity value assigned to each pixel and $max(I) = max(I(p))$ be the maximum intensity over the image. Without confusion,



(a) *(b)*

Figure 4: Watershed track extraction on the images of (a) Figure 2a and (b) Figure 2b.

we use the same symbol I to denote an image as well as intensity of the set of pixels which constitute the image. Each pixels takes intensity values between 0 and $max(I)$. An *edge* is a pair of adjacent pixels (p, q) and a *path* is the non-unique set of *edges* connecting two pixels. We can define the *cost* of a *pixel*, an *edge*, and a *path* in I as follows:

Definition 1 *The* cost *of a pixel p in the image I is defined as:*

$$cost_I(p) = \max(I) - I(p).$$

Definition 2 *The* cost *of an edge in the image I is defined as the sum of the costs of p and q:*

$$cost_I(p, q) = cost_I(p) + cost_I(q).$$

Definition 3 *The cost of a path P between any two pixels p and q in I is the sum of the costs of its successive edges.*

We seek paths of minimal cost in a given class of allowable paths. Intuitively, the paths of minimum cost correspond to paths of maximum integrated intensity. One of the constraints on the paths is that correlogram tracks $\tau(t)$ are single-valued functions of time. To find such paths in a computationally-tractable manner, we constrain the set of possible paths in a manner we will now describe. Let us define the unit vectors between adjacent pixels as $\vec{u}_0$, $\vec{u}_1$,..., $\vec{u}_7$, as illustrated in Figure 5.

The track extraction task will be performed as follows:

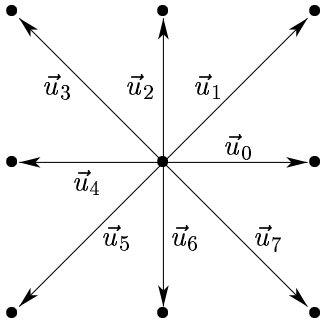


Figure 5: Definition of unit vectors between adjacent pixels.

Find the paths of minimal cost between the top row T and the bottom row B of the image I, such that the edge $\vec{pq}$ between any two adjacent pixels p and q of this path is constrained to be of the form $\vec{pq} = \vec{u}_5$ or $\vec{pq} = \vec{u}_6$ or $\vec{pq} = \vec{u}_7$.

Such paths are called *paths of minimal cost*. This criterion assures that the individual pixel intensities along such paths are high (i.e., have low *cost*) and that the lengths of the paths remain small. The extracted paths are relatively straight while going through pixels of high value.

The condition on edge orientation is not strictly necessary, but is added for computational efficiency. Another way to formulate it is to say that at any location along a track, we assume that the absolute value of the angle between the track and the vertical direction is less than or equal to some maximum, chosen in this case to be 45^o. The maximum angle is actually determined by the dynamics of the target and the target-sensor geometry.

Note that the work reported in this paper has already been generalized to the robust extraction of tracks even without this assumption (this will be the topic of further publications). However, the present edge orientation assumption has the advantage of significantly reducing the complexity of the track extraction algorithm. Moreover, it also guarantees a certain smoothness to the extracted tracks.

A practical algorithm for finding the minimum path uses *generalized distance functions*. We first define a distance between the top of the image and any point p within the image as well as a corresponding distance between the bottom and any point p. Let T denote the set of pixels of I constituting the uppermost line of the image, and B be the set of pixels in the bottom line of I. Then

Definition 4 *The* downward generalized distance function $d^-(I)$ *assigns to each pixel p of I the minimal value of the costs of all paths between T and p with the constraint that all edges in the path are in the directions of vectors $\vec{u}_5$, $\vec{u}_6$ or $\vec{u}_7$.*

Definition 5 *The* upward generalized distance function $d^+(I)$ *assigns to each pixel p the minimal value of the costs of all paths between p and B with the constraint that all edges in the path are in the directions of vectors $\vec{u}_1$, $\vec{u}_2$ or $\vec{u}_3$.*

It can then be shown that if a pixel p belongs to a path of minimal cost between T and B, then $d^-(I)(p) + d^+(I)(p)$ is equal to the generalized distance $d(T, B)$ between T and B. Conversely, if a pixel p is such that $d^-(I)(p) + d^+(I)(p) = d(T, B)$, then it belongs to a path of minimal cost between T and B. Therefore, we have the following result:

Pixels belonging to paths of minimal length between T and B are exactly those pixels p for which the value of $d^-(I)(p) + d^+(I)(p)$ is minimal over I.

We point out that the upward and downward generalized distance functions assign distance values to each pixel in the original image and, hence, can be regarded as images themselves.

3.4 Track Extraction Algorithm

The overall *Minimum Distance Algorithm* for reliable track extraction is therefore the following (see also Figure 6):

1. Compute the downward generalized distance function image $d^-(I)$,
2. Compute the upward generalized distance function image $d^+(I)$,
3. Perform a pixelwise addition of the two images: $J = d^-(I) + d^+(I)$,
4. Extract the minimal value m of J,
5. Keep only pixels whose value in J is equal to m. These constitute the desired track pixels.

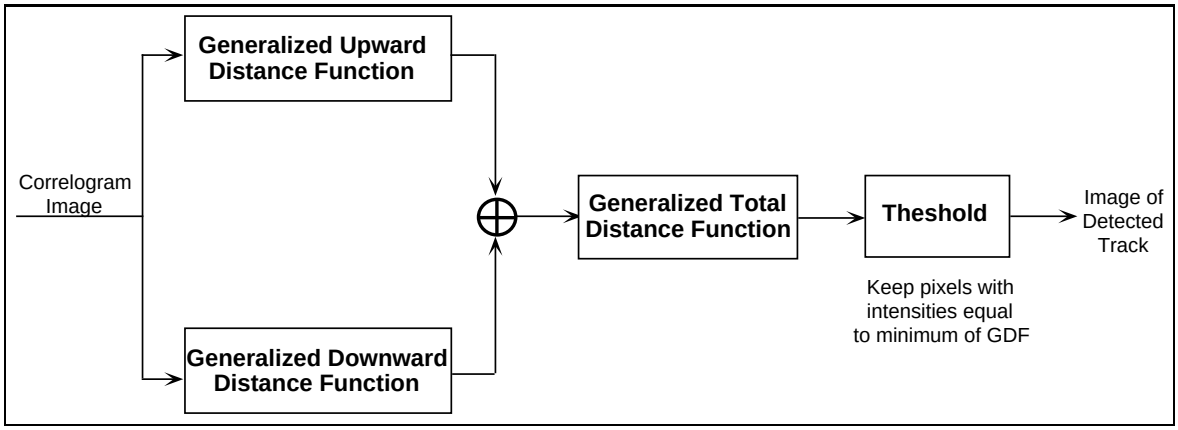
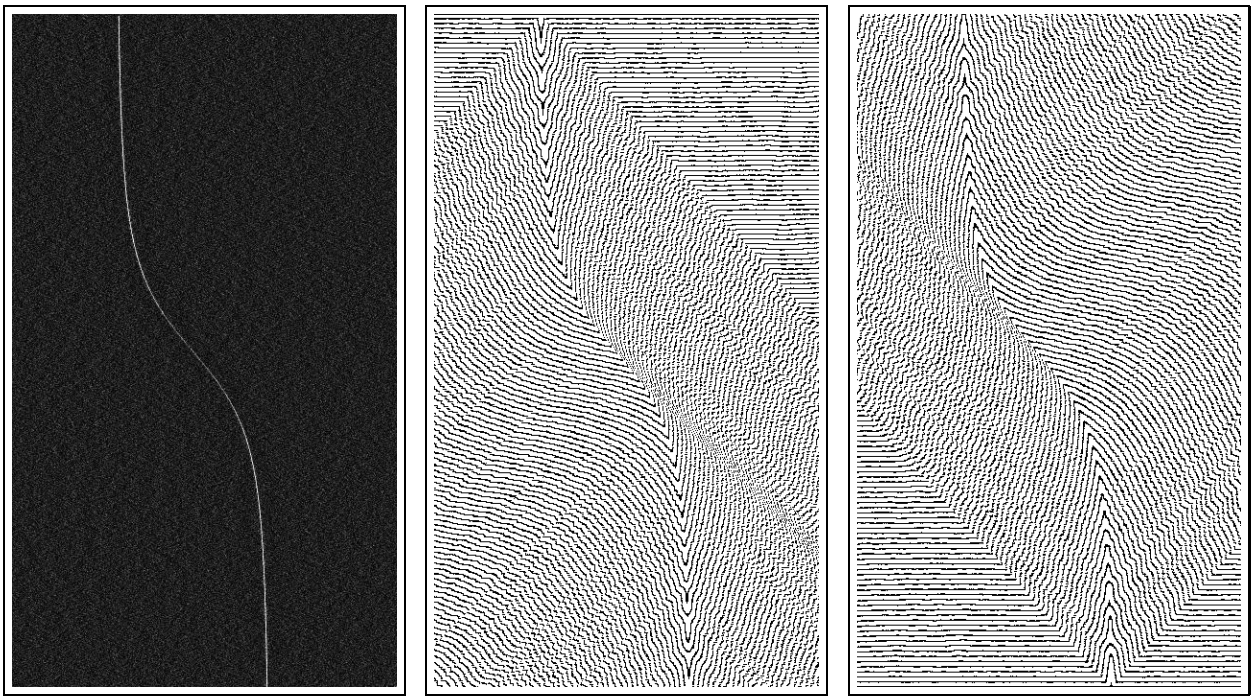


Figure 6: Processing Flow Diagram of Minimum Distance Algorithm

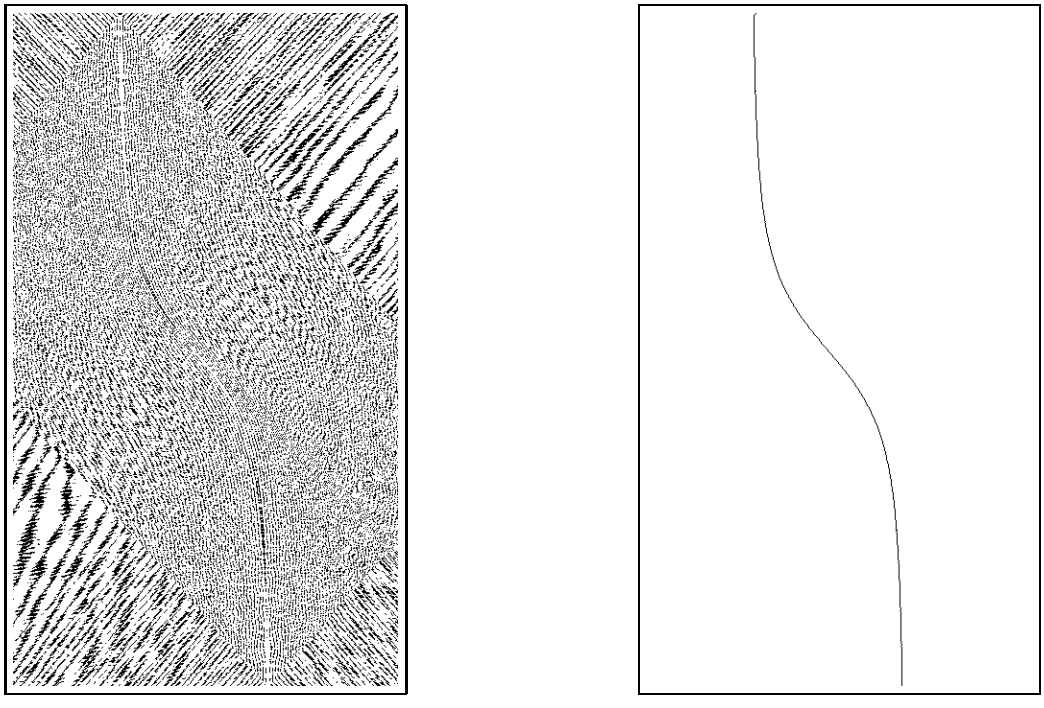
Similar techniques, using simpler distance functions, have been used in [16, 6] to extract candidate fracture lines in images of porous materials. A detailed description of this algorithm goes beyond the scope of this paper, but we mention that the algorithm is of *sequential type* [10, 11, 14] and proceeds in a single raster (respectively, anti-raster) image scan, as opposed to using two scans for traditional sequential distance function algorithms. Distance values are progressively propagated downward (respectively, upward) until the entire image frame has been scanned. The algorithm is fast and does not require any random access to image pixels. It is therefore particularly suited to hardware implementations (sequential distance function algorithms have already been implemented on VLSI's, see [7]).

An illustration of the complete algorithm is shown in Figure 7. We used the image in Figure 2a as illustrative because its intermediate images are easier to understand. Note that the generalized distance functions are displayed as contour plots rather than as pixel displays to show the minimum paths very clearly. The output of this algorithm on Figure 2b is shown in Figure 8. The algorithm was not perturbed by the extremely high noise levels and yielded an accurate result.

This algorithm can easily be adapted to the extraction of several tracks, although higher accuracy is obtained in images with one single track. Once a first track has been extracted, its pixels are removed from the original image, and the extraction of the path of minimal cost in the new image



original image *downward distance* *upward distance*



sum of distance functions *extracted track*

Figure 7: Track extraction algorithm applied to Fig: 2a.

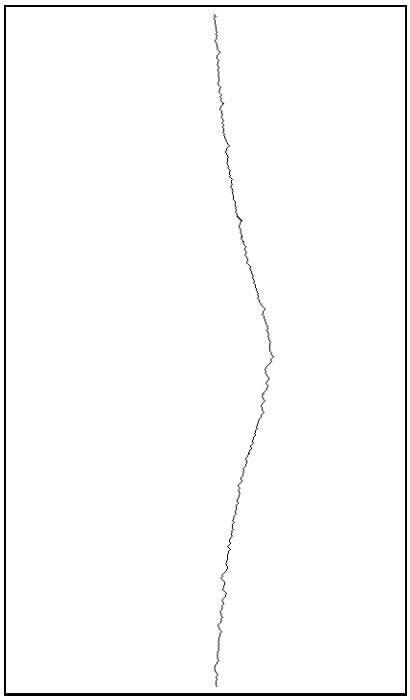


Figure 8: Track extracted from Fig. 2b.

proceeds in the new image. No matter what kind of image I is given as input to the algorithm, a path of minimal cost will be found. It is therefore essential to define an *a posteriori criterion* for deciding whether an extracted track is valid or not. Such a decision can be made based on the average track intensity compared to that of the image. The smoothness of the extracted track may also be an important factor, the assumption being that smooth paths of minimal costs have a higher probability of corresponding to real tracks. Some of the issues involved in detecting multiple tracks are discussed below.

3.5 Implementation Issues for a Real-Time Processor

The MDA algorithm, as described above, was implemented in an end-to-end demonstration system which detected, tracked and sorted out, multiple correlation tracks from multiple pairs of sensors, ending up with a geographic target track for each vehicle. In order to implement the procedure in a real-time detection system, the tau-time domain must be segmented into finite time segments. The time duration of each segment is a tradeoff between providing a large enough image to allow the method to work and small enough to obtain a timely detection. In addition, in a realistic scenario, there will often be more than one track and often these tracks will cross each other. The method performed robustly under these conditions. As mentioned above, for any image, whether it contains a correlation line or not, there will always be some path with a minimum generalized distance. This means that that MDA algorithm must be controlled by an additional detection criterion. Our approach was to compare some test statistic calculated for each detected track against a threshold. One of the first test statistics considered was to compare the mean track intensity with the mean intensity of an arbitrary path of the same length in the image. Because of the usual large number of

points which typically make up a path, the integrated intensities can be considered to be Gaussian distributed and any threshold setting proportional to the noise variance can be expected to establish a constant probability of false alarm.

Unless some means is available to remove a previously detected path, the MDA algorithm will continue to return to it. The simplest method to prevent this is to null out a neighborhood of a detected track from the image and is described here. When any line is detected, the detected track image is used to create a mask to remove that track from the original image. Figure 7 is an example of such a mask. This mask consists of ones where the track was detected and zeros everywhere else. This mask is then morphologically *dilated* in the tau-direction (horizontally) to expand the line. (*Dilation* is an operation whereby the value of any image pixel is replaced by the maximum of the image in a neighborhood of the pixel. For binary images, i.e. black/white, *dilation* corresponds to an OR operation over the neighborhood of the pixel). The neighborhood is defined by a morphological *kernel*. The kernel was chosen to be a single-pixel high and width based upon the expected width of the correlation track. (The MDA algorithm always generates a track which is one pixel wide and hence always narrower than any real correlation line). Very wide correlation line might occur, for example, when there is continuous multipath, such as in a ducted environment. If the dilation kernel is not wide enough, portions of a detected line may remain in the image and be detected again. The mask is then inverted and a pixel-by-pixel multiplication with the original image is performed. This resultant image, minus the first detected track, is passed into the MDA algorithm again. This process is repeated until the detected track fails the threshold test. Figure 9 illustrates the complete multiple-track processing algorithm.

The next problem considered was how to handle intersecting tracks which may occur when two sources are present (correlation lines arising from discrete multipath can also intersect). This is a critical issue which is still under development. When the first track is detected, a neighborhood of the track is set to zero as before. At the intersecion point of two tracks, the second track will become discontinuous. If the first removed track is thin, the MDA method can cross the gap and successfully generate a path. If the first path is thick, it may not continue and two tracks segments will be generated instead of one. This usually presents no problem since the track state estimator will associate the disjoint tracks. However, when the removed correlation track is very wide, another approach may be more optimum and is being investigated. This method replaces each point of the removed track, not with zeros, but with a local estimate of the mean intensity of any potential crossing tracks.

4 Experimental Results

In this section, we present the results of the proposed track extraction techniques for a two-target simulation and for a real correlogram image. The first image in Figure 10a is the result of a simulation of the correlation tracks of two vehicles. One is a strong emitter which exhibits a single correlation track. A local examination of the strong track in Figure 10a shows that the correlation peaks stand out well above the noise peaks. This corresponds to the kind of track considered in Figure 2a. The propagation path from the quiet source shows three paths due arising from multipath propagation. An examination of the correlation peaks in a local neighborhood of the quiet tracks shows they do not exceed the local noise peaks. This scenario corresponds to the kind of track considered in Figure 2b. The Minimum Distance Algorithm, when applied to this image, detected the strong line

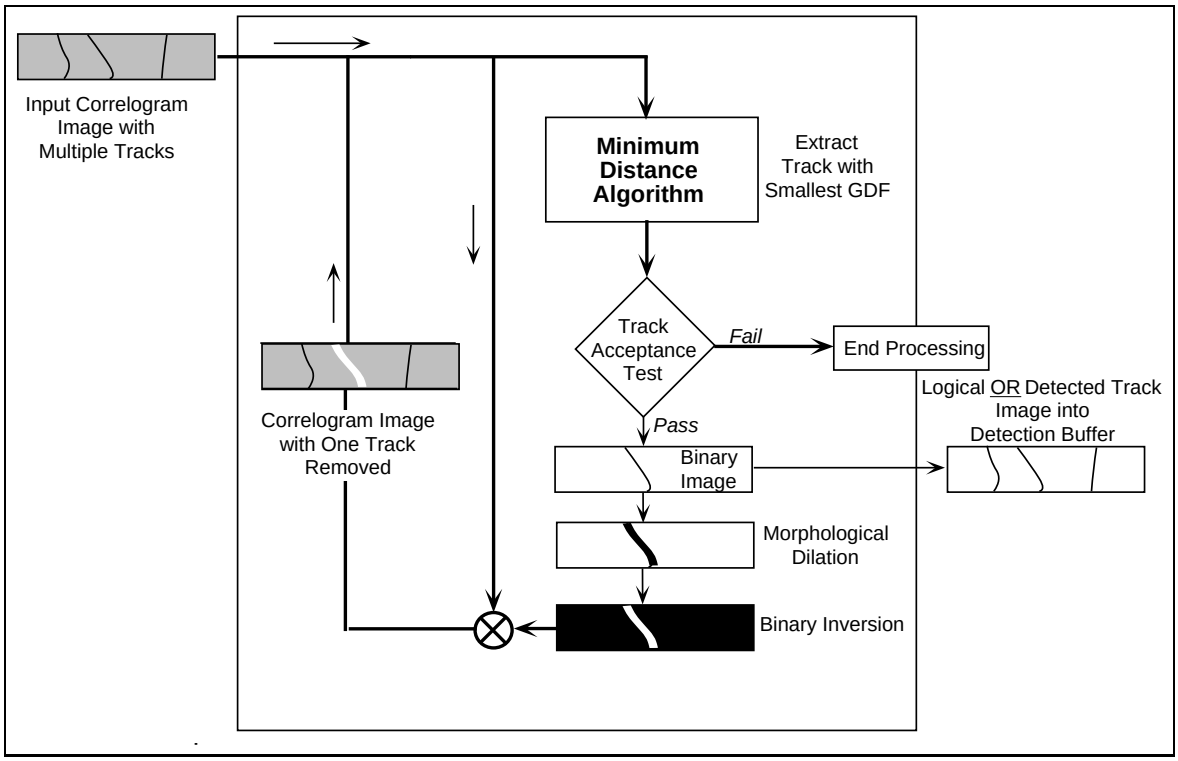


Figure 9: Processing Flow Diagram of Multiple Track Detection

first, removed it, then successively detected the three lines of the quiet source in succession of their relative intensity (see Figure 10b). There is also a very faint vertical line in the middle of the image which is an artifact of the algorithm which generated the correlogram from simulated sensor data. The MDA method readily detected this line. The next image, Figure 11, is a real correlogram which shows the correlation peaks from two surface ships. The first, on the right-hand side of Figure 11a, is a set of very intense correlation lines showing considerable multipath propagation. The peaks of the four most prominent correlation lines are clearly shown in Figure 11b. On the left-hand side of Figure 11a are three very feint discrete multipath tracks due to a second surface ship. The MDA algorithm detected the peaks, as can be seen in Figure 11b.

5 Summary and Future Work

Image Processing techniques, especially line detection algorithms based on generalized distance functions, have proved themselves a viable method to detect and associate low-level linear structures in a correlogram image. These structures correspond to coherent correlation of signals received at two sensors from the same acoustic source. The method depends upon the minimization of a global function of the image data, the generalized distance function. This function is related to the path through an image which exhibits the highest integrated intensity. The method works at very low SNR, where traditional line-based methods fail completely, and even when the human observer has trouble detecting the track. Robust techniques were developed to detect multiple lines in an image as well as correlation lines that intersect. The operation of these techniques on simulated and real data was demonstrated.

We have already improved the method introduced in this paper by eliminating the condition of maximal angle between the track and the vertical direction. Other future work efforts will concentrate on relating the SNR in the correlogram to the minimal number of image scan lines required to detect tracks with a given probability of detection. We would then be able to process the correlograms in smaller "strips" and use the tracks detected in the previous strip to initiate the search for tracks in the current strip. Other future work efforts will concentrate on improving criteria to decide whether an extracted minimal path is indeed a track and on improving track detection by preprocessing the correlogram image, by using alternative ways to value the edges between neighboring pixels, and by imposing a smoothness condition for the tracks.

6 Acknowledgements

This work was performed under contract N62269-90-C-0573, as part of the Full-Spectrum Processing and Chaos Project, RJ14C61, under the Air Undersea Warfare Surveillance Block, AW3A. The sponsor is Mr. T. Goldsberry, ONR code 4510. The Project is managed by Lee Allen, NAWCADWAR, code 5033.

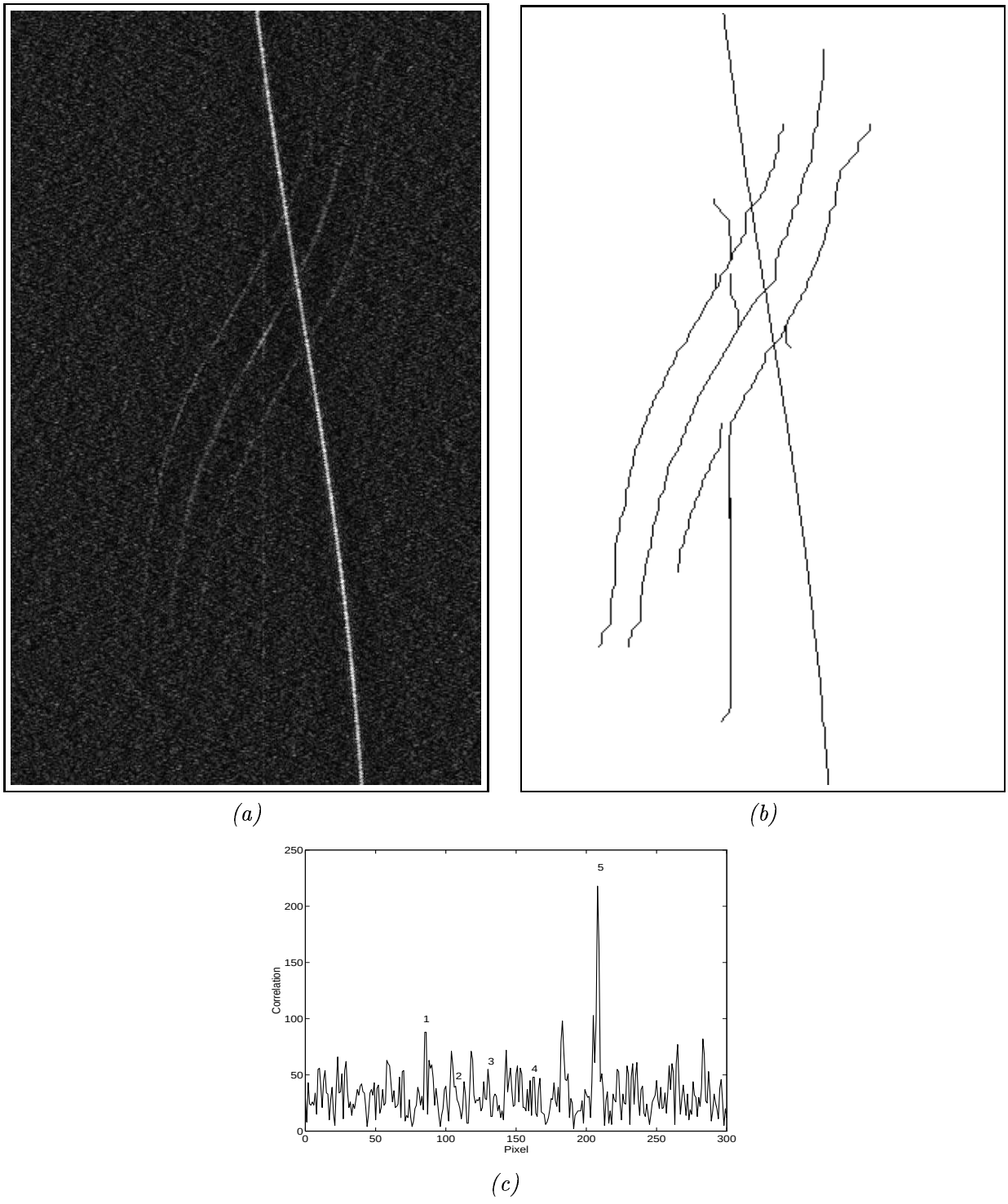
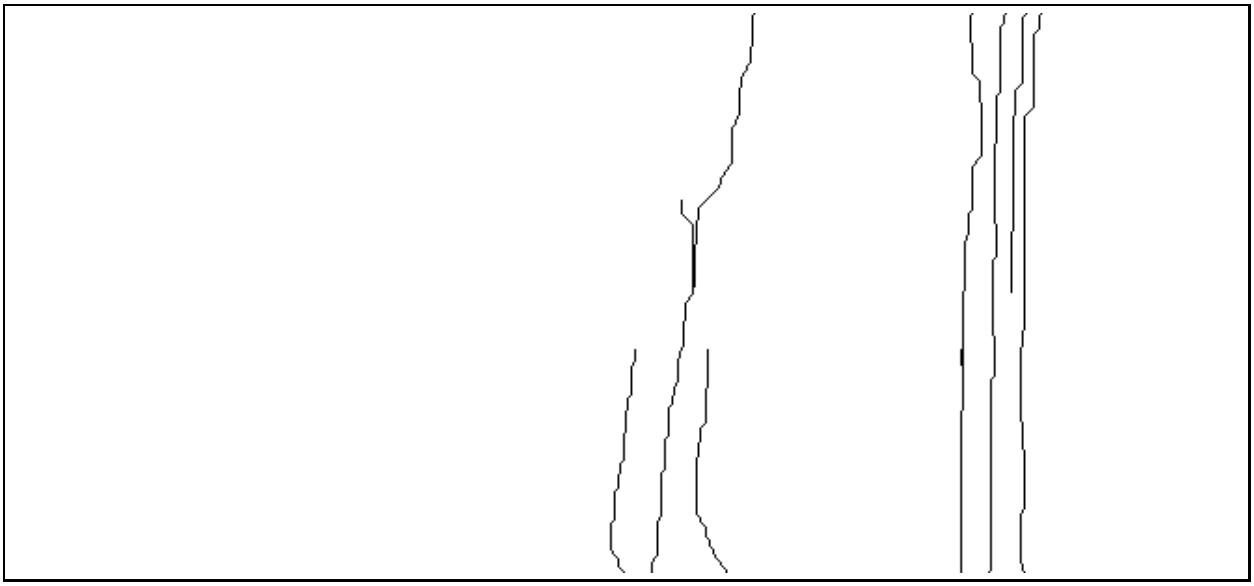


Figure 10: Multiple crossing tracks: (a) correlogram image, (b) detected tracks image, and (c) typical cross-section of correlogram (a).

(a)



(b)

Figure 11: Detection of surface ship correlation tracks: (a) correlogram, (b) detected tracks

References

[1] R. Barret, A. Steele, and R. Streit. Frequency line tracking algorithms. In Y. Chan, editor, *Underwater Acoustic Data Processing*. Kluwer, Boston, 1989.

[2] R. E. Bethel and R. Rahikka. Optimum time delay detection and tracking. *IEEE Trans. Aerosp. Electronic Systems*, 26:700–711, Sept. 1990.

[3] S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In *International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, Rennes, France, 1979.

[4] G. Carter. Time delay estimation. In H. Urban, editor, *Adaptive Methods in Underwater Acoustics*. D. Reidel, Dordrecht, 1985.

[5] J. Hassab. *Underwater Signal and Data Processing*. CRC Press, Boca Raton, 1989.

[6] D. Jeulin, L. Vincent, and G. Serpe. Propagation algorithms on graphs for physical applications. *JVCIR*, 3(2):161–181, June 1992.

[7] J.-C. Klein and R. Peyrard. PIMM1, an image processing ASIC based on mathematical morphology. In *IEEE's ASIC Seminar and Exhibit*, pages 25–28, Rochester NY, 1989.

[8] C.-H. Lee. Simulated annealing applied to acoustic signal detection. In *SPIE Vol. 1658, Nonlinear Image Processing III*, pages 344–355, San Jose CA, Feb. 1992.

[9] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1:21–46, Sept. 1990.

[10] A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *J. Assoc. Comp. Mach.*, 13(4):471–494, 1966.

[11] A. Rosenfeld and J. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1:33–61, 1968.

[12] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

[13] L. Vincent. Efficient computation of various types of skeletons. In *SPIE Vol. 1445, Medical Imaging V*, pages 297–311, San Jose, CA, 1991.

[14] L. Vincent. Morphological algorithms. In E. R. Dougherty, editor, *Mathematical Morphology in Image Processing*, pages 255–288. Marcel-Dekker, Inc., New York, Sept. 1992. Chapter 8.

[15] L. Vincent and E. R. Dougherty. Morphological segmentation for textures and particles. In E. R. Dougherty, editor, *Digital Image Processing Methods*, pages 43–102. Marcel-Dekker, New York, 1994. Chapter 2.

[16] L. Vincent and D. Jeulin. Minimal paths and crack propagation simulations. In *5th European Congress For Stereology*, pages 487–494, Freiburg im Breisgau FRG, Sept. 1989. Acta Stereologica. Vol. 8/2.

[17] L. Vincent and B. Masters. Morphological image processing and network analysis of corneal endothelial cell images. In *SPIE Vol. 1769, Image Algebra and Morphological Image Processing III*, pages 212–226, San Diego, CA, July 1992.

[18] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(6):583–598, June 1991.