
Article:
Lampert, Thomas and O'Keefe, Simon (2010) A Survey of Spectrogram Track Detection Algorithms. Applied Acoustics. pp. 87-100.

https://doi.org/10.1016/j.apacoust.2009.08.007

A Survey of Spectrogram Track Detection Algorithms

Thomas A. Lampert*, Simon E. M. O'Keefe

Department of Computer Science, University of York, Heslington, York, UK, YO10 5DD

Abstract

The detection of tracks in spectrograms is an important step in remote sensing applications such as the analysis of marine mammal calls and remote sensing data in underwater environments. Recent advances in technology and the abundance of data requires the development of more sensitive detection methods. This problem has attracted researchers' interest from a variety of backgrounds ranging between image processing, signal processing, simulated annealing and Bayesian filtering. Most of the literature is concentrated in three areas: image processing, neural networks, and statistical models such as the Hidden Markov Model. There has not been a review paper which describes and critically analyses the application of these key algorithms. This paper presents an extensive survey and an algorithm taxonomy, additionally each algorithm is reviewed according to a set of criteria relating to their success in application. These criteria are defined to be their ability to cope with noise variation over time, track association, high variability in track shape, closely separated tracks, multiple tracks, the birth/death of tracks, low signal-to-noise ratios, that they have no *a priori* assumption of track shape and that they are computationally cheap. Our analysis concludes that none of these algorithms fully meets these criteria.

Key words: Survey, Spectrogram, Acoustic Imaging, Acoustic Signal Detection, Remote Sensing, Vibration Analysis, Frequency Tracking

*Corresponding author. Tel.: +44 (0)1904 432794; fax: +44 (0)1904 432767.

Email addresses: tomal@cs.york.ac.uk (Thomas A. Lampert), sok@cs.york.ac.uk (Simon E. M. O'Keefe)

1. Introduction

The problem of detecting tracks in a spectrogram (also known as a LOFARgram, periodogram, sonogram, or spectral waterfall), particularly in underwater environments, has been investigated since their introduction in the mid 1940s by Koenig *et al.* [1]. The use of automatic detection methods drew increasing attention in the literature during the 1980s, 1990s and early 21st century. Applications are wide and include identifying and tracking marine mammals via their calls [2, 3], identifying ships, torpedoes or submarines via the noise radiated by their mechanical movements such as propeller blades and machinery [4, 5, 6, 7], distinguishing underwater events such as ice cracking [8] and earthquakes [9] from different types of source, meteor detection and speech formant tracking [10]. This paper surveys the variety of methods that have been applied.

The paper begins with a brief overview of the spectrogram creation process to familiarise the reader with the intended application of the algorithms. In the broad sense this "problem arises in any area of science where periodic phenomena are evident and in particular signal processing" [11]. In practical terms the problem can form a critical stage in the detection and classification of sources in passive sonar systems and the analysis of vibration data, the output of which could be the detection of a hostile torpedo or of an aeroplane engine which is malfunctioning. Recent advances in torpedo technology has fuelled the need for more robust, reliable and sensitive algorithms to detect ever quieter engines in real time and in short time frames. Also, recent awareness and care for endangered marine wildlife [12, 2, 13, 3, 14] has resulted in increased data collection which requires automated algorithms to detect calls and determine local specie population and numbers. Therefore, such a survey as this is called for, firstly as there is no such survey present in the literature, and, secondly, weaknesses and strengths in existing algorithms need to be identified to pave the way for future research.

The papers surveyed are from a variety of computer science areas and are concerned with the specific problem of track detection within spectrogram images with application to passive SONAR. Whilst there is a huge amount of literature on acoustic analysis and pattern recognition the intersection of these fields is relatively small, this paper acts as a review of this intersection. The algorithms are evaluated according to criteria, some or all of which are essential for a successful application, namely, their ability to cope with noise variation over time, track association, high variability in track shape, closely

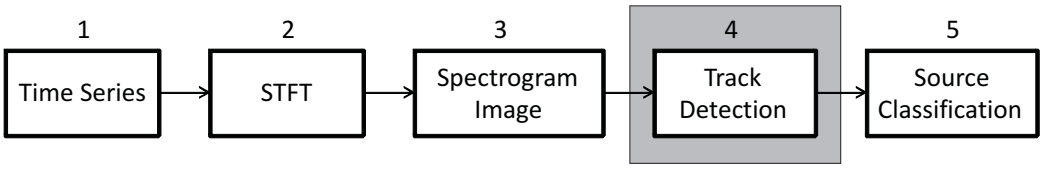

Figure 1: Flow diagram of the spectrogram track analysis process.

separated tracks, multiple tracks, the birth/death of tracks, low signal-to-noise ratio, that they have no *a priori* assumption of track shape and, for real time implementations, that they are computationally cheap. This evaluation is accomplished by inspection of the literature.

The remainder of this paper is organised as follows: Section 2 presents an overview of the problem and a definition of the evaluation criteria. In Section 3 is presented a taxonomy of the reviewed algorithms, and in Section 4 the methods are surveyed and reviewed. A discussion of the main shortfalls of the algorithms with respect to the defined criteria is presented in Section 5, leading to the identification of issues to be addressed in future research. Finally, in Section 6 we draw our conclusions.

2. Analysing Spectrograms

A spectrogram is a visual representation of the distribution of acoustic energy across frequencies and over time. The vertical axis of a spectrogram typically represents time, the horizontal axis represents the discrete frequency steps, and the amount of power detected is represented as the intensity at each time-frequency point.

2.1. Problem Background

Narrowband sound radiated in an underwater environment is exploited in *Passive Sonar* (passive sonar systems do not emit any sound and therefore only sound radiated from the target can be detected by the receiver, Fig. 1, box 1). The short-time Fourier transform (STFT) of the received signal is calculated (Fig. 1, box 2) to determine the power present at each frequency band in a particular time sample (see Fig. 2, top). These Fourier transforms are then collected together and a spectrogram image is built up containing the energy at each time-frequency point (see Fig. 2, bottom).

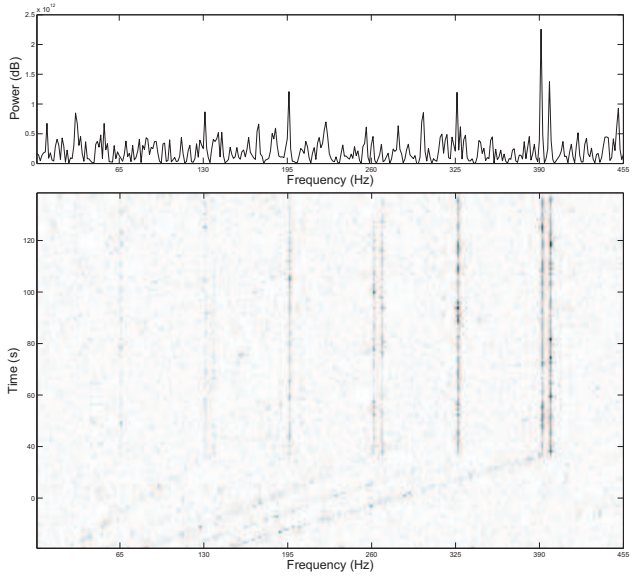

Figure 2: (Top) Fourier transform of the passive acoustic signal at one time step. (Bottom) A spectrogram image where intensity represents signal power (dB). The tracks have a SNR of (from left to right); 1st three 3 dB, 2nd three 6 dB and the last three 9 dB.

Sound sources such as ships and other man-made machines radiate some of their energy as narrowband sound that is dependent upon engine speed [15]. The sources of this radiated sound can be grouped under the classes of internal machinery noise and external propeller noise and produce tracks in a spectrogram that vary in frequency according to the state in which the machine is in. For example, when a source is running at a constant speed this narrowband energy results in time-invariant tracks, as the frequencies emitted do not vary, whereas a source that is accelerating results in tracks that increase in frequency over time. Other sources of radiated narrowband sound that are not dependent on engine speed, the hydrodynamic flow noise and the remainder of the machinery noise, result in constant frequencies regardless of the machine's state. As each type of source emits a particular frequency pattern, it may provide sufficient information for its identification within a spectrogram (Fig. 1, box 5). Urick presents a full discussion on the radiance of acoustic energy from submerged machinery in "Principles of Underwater Sound" [15]. Due to the Doppler effect and the nature of the source's machinery the track is often not time-invariant and therefore

general line detection algorithms are not suitable. However, it still holds that a particular, relative, frequency pattern will be emitted by each source.

The principle source of complexity in the analysis of passive sonar is that all noise from each source in the underwater environment is received. This results in the presence of large amounts of non-uniform background broadband noise in the spectrogram. This noise distorts the tracks, causing them to be broken, particularly at low frequency ranges, and also introduces points of high energy at spurious frequencies. Identifying these from true signals is particularly hard in low Signal-to-Noise Ratio (SNR) conditions.

There exist two distinct approaches to the analysis of time series data; the time domain approach and the frequency domain approach. A discussion of the differences between these two approaches has been presented by Wold [16] and reviews of methods which are applied in the time domain have been presented by Kootsookos [17] and Quinn [18]. This paper is concerned with methods which operate in the frequency domain as, traditionally, this is the domain in which passive sonar data is represented. The transformation of a time domain signal into the frequency domain often allows more efficient analysis to be performed [19]. The transformation also has the effect of quantising a series' noise into the spectrum of frequency bins and therefore the SNR of a time series is enhanced in the frequency domain. However, when constructing a spectrogram image phase information is lost and therefore frequency domain methods should be applied to areas in which the time of measurement commencement is not of importance. The transfer of the signal from the time domain into the frequency domain allows for the application of algorithms from a wide background of research disciplines, as highlighted in this paper, whereas generally time domain analysis is restricted to the signal processing and statistical analysis backgrounds.

There are two methods for measuring the SNR in this problem; either the time domain (or broadband) SNR or the frequency domain SNR. As this review is concerned with the detection of tracks within a spectrogram image the time domain SNR is not a true representation of the problem complexity. In order to convert between the two, full information regarding the STFT process is needed and this is not obtainable for all of the papers reviewed. Therefore, where time domain SNRs are presented the distinction is noted. As an example of the differences between the two measurements for the same signal, a time domain SNR of $-27.01\,\mathrm{dB}$ equates to a frequency domain SNR of $2.99\,\mathrm{dB}$ when a sample rate of $2\,\mathrm{kHz}$ is used (assuming a $1\,\mathrm{Hz}$ bin size STFT).

2.2. Definition of Evaluation Criteria

The criteria by which the algorithms will be evaluated, some or all of which are essential for a successful application, are defined below (in no particular order):

C1 low SNR - Is reliable detection achieved in a frequency domain SNR below 3 dB, defined as Eq. (4)?

C2 temporal noise variability - Does the method allow for a time-variant noise model?

C3 birth/death tracks - Does the algorithm cope with the initiation and/or termination of tracks at some point within the spectrogram?

C4 multiple tracks - Can the algorithm detect two or more separate tracks that exist concurrently (in the same time frame)?

C5 closely spaced tracks - Can the algorithm distinguish two or more tracks that are separated by one frequency bin?

C6 crossing tracks - Will the algorithm detect and distinguish between multiple tracks that occupy the same point in a spectrogram for one or more consecutive time frames?

C7 high track variability - Does the algorithm detect time-invariant tracks that have high variability?

C8 no *a priori* shape assumption - Is the method free from the assumption of a strict track shape model and therefore can generalise to unknown cases?

C9 track association - Does the method output a series of points that it deems as belonging to the same track?

C10 computationally cheap - Does the algorithm have an on-line computational burden with less than polynomial complexity (not including any training requirements)?

The importance of each criterion depends upon the algorithm's application as each application is concerned with the detection of signals with different characteristics. The dominant signal characteristics of some example applications, along with the criteria which should be met to demonstrate an algorithm's suitability, are identified in Table 1. In addition to these, the

Table 1: Signal characteristics and criteria specific to typical applications of spectrogram track detection algorithms.

Application	**Typical Track Characteristics**	**Criteria Needed**
Whale vocalisation	Short duration, high variability, predictable appearance, initiation and termination observed.	C2 Temporal Noise Variability, C3 Birth/Death Tracks, C4 Multiple Tracks, C7 High Track Variability.
Passive Sonar	Long duration, low SNR, initiation and termination observed.	C1 Low SNR, C2 Temporal Noise Variability,
-Submarine	Low variability.	C3 Birth/Death Tracks, C4 Multiple Tracks, C5 Closely Spaced Tracks, C6 Crossing Tracks, C7 High Track Variability,
-Torpedo	High variability.	C8 No *A Priori* Shape Assumption.
Directly instrumented vibration analysis	Long duration, high SNR.	C4 Multiple Tracks, C5 Closely Spaced Tracks, C6 Crossing Tracks, C7 High Track Variability, C8 No *a priori* Shape Assumption.

need to fulfil the C9 (track association) criterion is dependent upon the type of subsequent processing which will be performed and when on-line detection is needed the C10 (computationally cheap) criterion should be met.

3. Algorithm Taxonomy

Algorithms present in the literature are identified and categorised in Table 2 (in chronological order within subheadings). It should be noted that the majority of research has been conducted within the areas of statistical modelling, image processing and neural networks, with additional contributions from relaxation techniques. Hidden Markov models have, by far, attracted the largest proportion of research interest. Considering the relative size, breadth of techniques and the recent speed of progress in the area of image processing it has received very little attention in the literature.

It should be noted for completeness that additional methods exist, particularly those which are presented in the literature as Master's theses [20, 21], which it was not possible to survey (although they have been included in the taxonomy presented here). However, it is believed that similar techniques from different authors have been reviewed and therefore that the key algorithms are still presented in this paper.

Table 2: Categorisation of spectrogram track detection techniques in chronological order within subheadings.

Approach	Representative Works
Maximum Likelihood	
MLE	Maximum value [22]
	Correlation [23]
	Multi harmonic [24]
Image Processing Techniques	
Likelihood Ratio Test	Morphological operators [25]
Hough Transform	Graph theoretic tracking & heuristic search Hough transform[a] [26]
Multistage Decision	Multistage decision cost function optimisation [27]
Steerable Filter	Gap bridging, region locating & multistage decision process [28, 29]
Two-Pass Split-Window	Broadband subtraction via estimation [5]
Edge Detector	Gaussian filtered spectrogram [30]
Neural Networks	
Supervised Learning	Autoassociative memory & multi-layer perceptron [31]
	Multi-layer perceptron [32]
	Multi-layer perceptron constrained using Ockham's networks [33]
	MNET1 [34]
	MNET2 [34]
	RNET [34]
Unsupervised Learning	Kohonen self-organising map [35]
Statistical Models	
Dynamic Programming	Logarithmic likelihood function [36]
Hidden Markov Model	Viterbi & max amplitude [37]
	Viterbi, "mixed" track & threshold [38]
	Viterbi & "mixed" track [39]
	Viterbi & double threshold [40]
	Viterbi & probabilistic data association [41]
	Parallel, multi model detection [42]
	Forward-backward linking, SNR estimate & track gradient [43]
	Forward-backward linking & SNR estimate [44]
	Viterbi & SNR estimate [44]
	Forward-backward linking & spectrum interpolation [45]
Tracking Algorithms	
Particle Filter	Formant detection [10]
Relaxation Methods	
Relaxation	Relaxation[a] [20]
Simulated Annealing	Simulated annealing[a] [21]
	Simulated annealing [46]
Expert Systems	
Double detection	Double threshold & priority ranking [47]

[a] Master's theses which are not surveyed in Section 4.

4. Spectrogram Track Detection

This section presents the review of the methods from the literature under the categories presented in Table 2. The techniques presented here are specifically those found in the literature which have been applied to the problem of spectrogram track detection in passive sonar systems. As such this is not intended to form a full catalogue of general purpose detection or tracking methods.

Nomenclature. To aid comprehension of the reviewed methods they are presented using the following, consistent, mathematical notation. A time domain sampled signal $x_s(t')$, sampled at a rate of f_s, where $t' = 0, 1, \ldots, T'-1$, is split into N sections, each d seconds in length, $x_s^n(t)$ where $n = 0, 1, \ldots, N$ and $t = 0, 1, \ldots, T-1$ for $T = \lfloor df_s \rfloor$. To construct a spectrogram, first the spectrum of each signal section is calculated using the STFT, defined as

$$F_n(\omega) = \sum_{t=0}^{T-1} x_s^n(t) w(t) e^{-i\frac{2\pi\omega t}{T}}, \qquad 0 < \omega < \frac{T}{2} \tag{1}$$

where $w(t) = 0.54 - 0.46\cos(\frac{2\pi t}{T-1})$ is the Hamming window function and $\omega \in \mathbb{R}$ represents ordinary frequency (Hz). The power of $F_n(\omega)$, defined as

$$P_n(\omega) = \frac{1}{\sum_{t=0}^{T-1} |w(t)|^2} |F_n(\omega)|^2 \tag{2}$$

forms the elements of a spectrogram, such that

$$\mathbf{S} = [s_{ij}]_{N\times M} = \begin{bmatrix} P_0(\omega_0) & P_0(\omega_1) & \ldots & P_0(\omega_{M-1}) \\ P_1(\omega_0) & P_1(\omega_1) & \ldots & P_1(\omega_{M-1}) \\ P_2(\omega_0) & P_2(\omega_1) & \ldots & P_2(\omega_{M-1}) \\ \vdots & \vdots & \ddots & \vdots \\ P_{N-1}(\omega_0) & P_{N-1}(\omega_1) & \ldots & P_{N-1}(\omega_{M-1}) \end{bmatrix} \tag{3}$$

where $i = 0, 1, \ldots, N-1$ is the time frame, $j = 0, 1, \ldots, M-1$ is the frequency bin, $N \in \mathbb{Z}^+$ is the number of previous frames to be retained and $M \in \mathbb{Z}^+$ is the number of frequency bins calculated using the STFT. The signal-to-noise ratio of a spectrogram (frequency domain SNR) is defined as

$$\text{SNR} = 10\log_{10}\left(\frac{\bar{P}_t}{\bar{P}_b}\right) \tag{4}$$

$$\bar{P}_t = \frac{1}{|P_t|} \sum_{(i,j)\in P_t} s_{ij}, \qquad \bar{P}_b = \frac{1}{|P_b|} \sum_{(i,j)\in P_b} s_{ij} \tag{5}$$

where $P_t = \{(i,j)|s_{ij}$ belongs to a track$\}$ and $P_b = \{(i,j)|(i,j) \notin P_t\}$.

4.1. Maximum Likelihood Estimators

Maximum likelihood estimators (MLE) are based upon statistical assumptions regarding the data in question. A statistical test is defined which decides whether a frequency bin contains noise or a track (signal). MLE methods make detections on single spectrogram points and lend themselves to the detection of temporally invariant tracks as no assumptions are made regarding the temporal evolution of a track. However, the simplicity of the detection methods limit their application to high SNR cases. This limitation is overcome with MLE methods based on convolution, which make assumptions regarding the temporal evolution of a track to augment low SNR detection. However, the large search space needed to perform real world detections make them unfeasible.

Rife and Boorstyn [22] state that after STFT output has been obtained, the maximum of the result is the MLE of the estimated frequency, $\hat{\omega}_i$, that is,

$$\hat{\omega}_i = \arg\max_j |s_{ij}|, \qquad i = 0, 1, \ldots, N-1. \tag{6}$$

This is repeated for each observation. Thus, a single frequency is detected within each and every time frame i, and the estimated track is a series of these frequency positions. This method has been applied by Ferguson [48] to the analysis of aircraft acoustics received by an underwater hydrophone.

According to Barrett and McMahon [24], the single frequency case described above, Eq. (6), can be extended to the detection of a single frequency which exhibits harmonics, such that

$$\hat{\omega}_i = \arg\max_j \sum_{l=1}^{m} |s_{i,lj}|^2, \qquad i = 0, 1, \ldots, N-1. \tag{7}$$

These early MLE techniques disregard information describing the distribution of the intensity values attributed to each class, opting to use the maximum instead. This would lead to the method mistaking spurious high power noise for instances of a track. However, an important introduction in the multi-harmonic case is the concept of detecting a fundamental frequency

by integrating information from its harmonics. This information integration should greatly increase the detectability of tracks at low SNRs.

Altes [23] presents a likelihood ratio test based upon the correlation of a spectrogram with an expected, noise free, reference spectrogram $\mathbf{Z}_k = [z_{ij}(\rho_k)]$, such that

$$p(\mathbf{S}|\mathbf{Z}_k) \approx \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[\frac{-z_{ij}(\rho_k)}{\sigma^2} + \frac{s_{ij} z_{ij}(\rho_k)}{\sigma^4} \right] \tag{8}$$

where σ is the standard deviation of the time domain noise which is assumed to be known *a priori*. This process is repeated for K reference signal hypotheses (each with a hypothesised signal parameter of ρ_k) and the maximum response is taken to be the detected signal, $\hat{k} = \arg\max_{1 \le k \le K} [\ln p(\mathbf{S}|\mathbf{Z}_k)]$.

The use of the correlation function allows for the detection of very weak SNR tracks. However, for the method's use in remote sensing applications, where the state and behaviour of the phenomenon under observation are unknown, a very large reference set is needed. Under these conditions the computational burden of this method becomes too great for real-time implementations.

4.2. Image Processing

Image analysis techniques applied to this area treat the spectrogram as an image containing features to be extracted, applying statistical and image processing algorithms to achieve this. Image analysis is a vast research area, and provides a wide range of techniques which could be beneficial to this problem. These are often inspired by human visual perception models, which suggests they might be applicable to this problem as it is accomplished by human operators. Due to the complexity of more advanced methods, however, real-time implementation can often be difficult to achieve.

4.2.1. Likelihood Ratio Test

Abel *et al.* [25] propose a statistical likelihood test to be used for track detection. The probability distribution of a signal (assumed to be Gaussian) is determined along with the distribution of noise probabilities. A likelihood test is defined such that

$$\frac{r_{ij}}{r_{ij}+1} \cdot \frac{s_{ij}}{b_{ij}} \underset{H_B}{\overset{H_N}{\gtrless}} T_\lambda \tag{9}$$

where r_{ij} is the SNR at point (i, j) and b_{ij} is the broadband power at point (i, j), and H_N and H_B are the hypotheses of a pixel containing narrowband and broadband signal (respectively). The result of applying this test is fragmented tracks and isolated false detections. These are repaired using the morphological operators dilation and erosion which expand and contract a track (respectively). In set theory, erosion is defined as $A \ominus B = \{z \in E | B_z \subseteq A\}$ where E is a Euclidean space or an integer grid, $A = \{(i, j) | s_{ij}$ belongs to a track$\}$ in E, B is a structuring element and B_z is the translation of B by vector z. Informally, erosion means to translate the structuring element B to all points in A and take only the points where the structuring element overlaps completely with points in A. Dilation is defined as $A \oplus B = \{z \in E | (\hat{B})_z \cap A \neq \emptyset\}$ where $\hat{B}$ is the symmetric of B. Informally, this means to translate the structuring element to every point in A and take all the points which are covered by the structuring element. Combined and ordered in this way produces 'closing', $A \cdot B = (A \oplus B) \ominus B$, [49] which has the effect of smoothing, eliminating thin protrusions and filling narrow gaps in the tracks. The region grow algorithm is employed to group pixels into a single track. This algorithm recursively groups connected pixels based upon a similarity measure, which, in this case, is that the pixels are part of a track.

The likelihood ratio test is described as being optimal as, for a given probability of a false alarm, the probability of detection is maximised. The background noise is not assumed to be stationary and therefore broadband equalisation is used to estimate r_{ij} on a frame-by-frame basis by taking the trimmed mean over a sliding frequency window. However, over-smoothing may reduce its applicability to the detection of low SNR tracks. This method also requires the use of a threshold which must be determined *a priori*, further limiting its generalisation. In the noisy test image presented in [25] the method appears to cancel a large amount of the background noise whilst preserving the track. However, no quantitative results are presented.

The use of the erosion operator limits this method to approximately stationary tracks because of its assumptions about track shape. Sections of tracks which do not fit the operator B exactly, i.e. tracks which rapidly increase/decrease in frequency, will be eliminated from the resulting detection.

4.2.2. Multistage Decision Process

Di Martino *et al.* [27] present an alternative approach based on feature grouping theory. A new cost function is defined over a track ζ such that

$$\Phi(\zeta) = \frac{\alpha.G(\zeta) + \beta.C(\zeta)}{A(\zeta)} \tag{10}$$

which accounts for the track's amplitude A, where $A(\zeta) = \sum_{(i,j)\in\zeta} s_{ij}$, continuity G, and curvature C. The cost function will decrease if a spectral track is detected and increase otherwise.

The problem is therefore transformed into optimising the cost function along all paths of length N, starting from a given image point. Each time an optimal path is found to go through a point in the image, the point's counter is incremented.

It is claimed in [27] that the computation of the optimal path according to the cost function $\Phi(\zeta)$ is linear in N and the algorithm is amenable to parallel processing. The qualitative result presented in [27], obtained using one spectrogram, reports that the method reduces the noise and that the spectral track "becomes more perceptible". It is stated that the method has been tested on a set of spectrograms with differing SNRs, the results of which show that this method increases track detection and decreases false positive detections (although these results are not presented).

A point to be made regarding the continuity measurement in Eq. (10), which is defined to be proportional to the number of track points which have zero amplitude, is that, in our experience, spectrograms which contain background noise, such as those from sea environments, very rarely have points of zero amplitude. Also, the division by a track's amplitude restricts the detection to relatively high SNR tracks; if the weights are chosen to detect high curvature, high continuity tracks which have high amplitude, tracks which have low curvature, high continuity and low amplitude are likely to be missed. Also, if there are spurious points of high amplitude noise present in the spectrogram, which would have high curvature and low continuity, there is a high probability that these would cause a false positive detection.

4.2.3. Steerable Filter

Di Martino and Tabbone [29] propose an approach using steerable filters. Three steps are defined: the detection process, region locating and track tracing. Smoothing is performed using a Gaussian filter and an energy function $E(\theta) = \ddot{G}(\theta)^2 + H(\theta)^2$ (where $\ddot{G}$ is the second derivative of

the Gaussian and H is its Hilbert transform in the direction θ) is defined to detect edges using steerable filters. The second derivative and mean distance on either side of the detected edges are calculated to determine a region $R_i = \{(i,j)|l_i \leq j \leq r_i\}$, where l_i and r_i are the region's left and right boundaries and i the row index, which encompasses them. Gap bridging is utilised to provide continuity. A multistage decision process (as described in Section 4.2.2) is performed on the original image within the regions detected to extract the spectrogram tracks. This maximises the cost function $\Phi(\mathcal{C})$ defined as

$$\Phi(\mathcal{C}) = \sum_{i=0}^{N-1} A(P_i) - \alpha \sum_{i=2}^{N-1} |l(P_{i-1}, P_i) - l(P_i, P_{i+1})| \tag{11}$$

where $P_i \in R_i$, $A(P_i)$ is the amplitude of P_i, and $l(P_i, P_j)$ is the slope of segment $[P_i, P_j]$. This extracts contours present within the regions R_i. The initial stages of this process (region location) are used to refine the search space within which the multistage decision process optimises thus reducing the computational burden.

It is noted that locating the regions in the proposed way does not guarantee that two tracks have not been merged during smoothing and therefore that only a single track is present within the track tracing search region. Also, the proposed method is not truly unsupervised as a threshold parameter value needs to be manually determined within the track detection stage. The method was tested using spectrograms of varying SNRs[a] $(1.50 - 7.45)$ and varying spatial frequencies [28]. It achieves above 87% detection performance over all SNRs and spatial frequencies and can perform the detection within a 128×128 pixel spectrogram in 36.74 seconds. It is not possible to perform a direct comparison between the SNRs used in this experiment and others as a different SNR measurement is used[a]

The use of the cost function $\Phi(\mathcal{C})$, Eq. (11), provides a balance between the detection of temporally invariant tracks and high SNR tracks. The local nature with which the curvature is calculated prevents the method from linking spurious high amplitude noise responses which are some distance away from the current track, whilst allowing globally fluctuating tracks to

[a]It is assumed that the paper's authors use the same SNR calculation as is presented in their other paper [29] and therefore that these figures are calculated as $\text{SNR} = 10\log_{10}([\bar{P}_t - \bar{P}_b]/\sigma_b)$ where σ_b is the standard deviation of the noise.

be detected. However, if there is a high amplitude noise point within the detected region which is close to the track there is a high probability that it will cause the detected track to deviate from the true location.

4.2.4. Two-Pass Split-Window

Chen *et al.* [5] propose the use of the two-pass split-window (TPSW) to estimate the background broadband noise within a spectrogram image. Once an estimate of this has been calculated, subtracting it from the image should result in a cleaned image containing narrowband tracks. The TPSW algorithm consists of two steps: first a local mean is calculated over a neighbourhood surrounding each bin in the STFT, such that

$$\hat{s}_{ij} = \frac{1}{2W+1} \sum_{l=j-W}^{j+W} s_{il}, \qquad i = 0, 1, \ldots, N-1 \tag{12}$$

where $j = W, \ldots, M-1-W$ and $2W+1$ is the number of bins used to calculate the local mean. The result, $\hat{s}_{ij}$, is clipped and a second, local, mean is calculated upon these (as defined in Eq. (12)).

Although this is a filtering technique, a threshold criterion can be defined upon the TPSW output and a detection made using this. As with any filtering technique, there is a balance to be made between the amount of smoothing and the detectability at low SNRs. In this case, this is controlled with the window size W. As the TPSW is calculated independently for each time step in the spectrogram it has no assumption of track structure. This allows the detection of time-invariant tracks which may be highly irregular in appearance.

4.2.5. Edge Detection

Gillespie [30], proposes an edge detection method which initially smoothes the spectrogram using a Gaussian filter $\mathbf{G}$, such that

$$\mathbf{S}' = \mathbf{S} * \mathbf{G} \tag{13}$$

$$\mathbf{G} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{14}$$

The benefit of smoothing is the prevention of edges breaking up into many parts; the detrimental effect is reduction of the resolution of the spectrogram if the smoothing kernel is too large.

Each point (i, j) in the smoothed spectrogram $\mathbf{S}'$ is thresholded by comparison to the background measurement b_{ij}. This background measurement is continuously updated to allow for time-invariant noise conditions and computed independently for each frequency bin, such that

$$b_{ij} = b_{i-1j} + \left(\frac{s'_{ij} - b_{i-1j}}{\alpha} \right) \tag{15}$$

and the spectrogram is thresholded if

$$\frac{s'_{ij}}{b_{i-1,j}} > H \tag{16}$$

where H is the threshold value. In this way detections in subsequent time frames are linked if they are within adjacent or overlapping frequency positions.

This method is applied in [30] to whale call detections and of the 2077 calls detected by humans the method successfully detected 1897 (90%). However, as with all methods which rely on smoothing of the spectrogram, the detection of low SNR tracks can be compromised as they tend to be eliminated in the transformed image.

4.3. Neural Networks

Neural networks are a widely applied classification architecture and a wide variety of neural networks exist, many of which are described in "Neural Networks: A Comprehensive Foundation" by S. Haykin [50]. These models have a proven ability to extract salient features of high-dimensional input spaces, allowing the identification of patterns in complex problems [50] which makes them a strong candidate for applications such as this. A key drawback in the use of neural networks, and any model which employs supervised learning, is the reduction in the model's ability to generalise to unknown cases. In applications such as this, frequency tracks can vary greatly and it is quite probable that a training set will not fully represent the range of variations the model may need to identify. Unsupervised learning methods overcome this limitation by automatic determination of the similarities within data which allows for greater generalisation ability.

4.3.1. Supervised Learning

Di Martino and Tabbone remark that such methods "need a supervised learning set that reduces their utility in real cases" [29]. Kendall *et al.* investigate this by testing several methods for improving the generalisation of

neural networks [33]. In terms of the application this improves the networks ability to detect track structures which were not included in the training data. Several techniques to improve a network's generalisation ability are tested: heuristically changing the number of hidden nodes, weight decay, soft weight-sharing and Ockham's networks.

A hidden node is a unit of a neural network which is neither an input or output unit, these are described as being hidden because their activations are not directly seen by the outside world. The hidden layer (the layer of the neural network which is made up of hidden units) learns to represent the input data in a way which captures salient information. The number of nodes, or even the number of hidden layers, determine the network's ability to represent complex, non-linear, patterns. Having too many hidden nodes can have the side affect of allowing the network to quickly overfit training data - reducing its ability to generalise. Unfortunately, there is no definitive method to determine the number of hidden nodes which is needed to solve a classification problem [51] and so trial and error is often employed.

Weights are applied to the values passed between nodes of the network and represent how much effect the value has on the receiving node's activation. Utilising weight decay helps to avoid overfitting training data by forcing the weights to remain small. This is realised through a simple regularisation function utilised during training, which shrinks the weight's value after they have been updated. This function is defined as

$$C = \sum_i \sum_j (o_j - d_{ij})^2 + \lambda \sum_i w_i^2 \tag{17}$$

where d_{ij} is the desired value of output o_j in the network's output layer, w_i is the network's i^{th} weight and λ can be thought of as a normalising parameter.

Weight-sharing is a technique in which a single weight is shared among several connections in a network, reducing the number of adjustable parameters. This requires good knowledge of the problem background so that it is possible to specify which connections will share weights. Soft Weight-Sharing [52] utilises Gaussian mixture models during training to determine the weight's values and which weights should be linked dynamically. This removes the dependence on the user to fix the weighting links *a priori*.

Ockham's razor states that the best hypothesis is that which requires the smallest number of assumptions. This philosophy is utilised in Ockham's networks to improve the generalisation performance of neural networks in the absence of large amounts of training data [53]. The minimum description

length principle is utilised to attribute a coding length to a network and the classification errors it produces. A cost function is defined such that

$$C = I(x|\Theta) + I(\Theta) \tag{18}$$

where $I(x|\Theta)$ is the description length of the data misfit x, given the chosen model Θ (the input/output values of all the training pairs not correctly classified) and $I(\Theta)$ is the description length of the model itself (the neural network's weights). The network which minimises this cost function is optimal as it has the least combined classification errors and complexity.

The most advantageous of the methods tested in [33] are found to be weight decay and Ockham's networks. Weight decay, constrained by the cost function outlined in Eq. (17), is found to significantly reduce the classification variance on a generalisation set when using a network with one hidden node. For a complex network (eight hidden nodes), correct values of λ not only reduce variance but also provide improvements in the generalisation performance by reducing the network complexity.

The most successful method tested in [33] was Ockham's networks. It was shown that this method provides a generalisation error rate of 16% upon a test set containing 121 instances of 9×9 pixel spectrogram windows (which were independently labelled from the training set). However, the method is very computationally expensive, requiring 24 hrs of computation time for one run. Because of this, no averaging over many trials was performed. It is stated, however, that "given that the genetic algorithm is finding a near global minimum for C it is likely that the variance will be small". As well as improving generalisation performance, the Ockham's network method resulted in the lowest complexity network based on the minimum description length principle.

Kotanzad *et al.* [31] implement a track detection mechanism with the following steps. Initially the spectrogram is thresholded to obtain a binary image. An autoassociative memory (ASM) is employed to eliminate the noise and to reconstruct the received signal. The ASM is trained using a number of clean reference signals which contain a target or no target, of which the closest to the noisy input signal is recalled during evaluation. The output of the ASM is then passed to a multi-layer perceptron (MLP) neural network trained using the backpropagation algorithm to classify the clean data from the ASM as containing a target or not.

It is stated that in an initial study a classification accuracy of 97% was achieved for spectrograms which contain a track, and 100% for noise only

spectrograms. However, these results were obtained using a very small test set, derived by adding Gaussian noise to the training spectrograms and which consisted of 24 spectrograms containing a track and 12 noise only spectrograms. Additionally, the shape of the tracks present in test set were regular and do not vary greatly in appearance. Under these conditions, it is possible that the networks are overfitting the data, explaining the very high classification rates, and that the technique would not generalise well.

Leeming [32] performs a similar investigation solely using an MLP back propagation network which was trained in two ways; the first, to classify a window as containing 0, 1, 2 or greater than 2 tracks, and, the second, whether the MLP can recall a clean picture with no noise from the input data. This is tested using a collection of windows containing, strong time-invariant tracks 10–20 dB above noise, weak time-invariant tracks 4–10 dB above noise and time varying track 7–10 dB above noise.

It is found in [32] that the networks with one hidden layer did not work adequately if there are two or more tracks in the data, however, data containing just 0 or 1 track in each window could be recognised by a single hidden layer ANN. Also, it was possible to remove noise from windows using a network topology of 1 hidden layer, and increasing the number of nodes improved clarity, especially in the time varying track case. Within the networks used to count tracks, increasing the size of the second hidden layer produced no effect, suggesting that the second layer is counting tracks and the first is removing noise (although it is noted that these networks required far fewer nodes in the first hidden layer than the tested window cleaning networks and therefore this distinction is not clear).

The presented experiments demonstrate that this method detects 75% of tracks that are time-invariant within a SNR range of 4–10 dB and 79% of time varying tracks (that is, having a random frequency variation of ± 1 frequency bin per time frame) with SNRs ranging between 7–10 dB (when trained to detect the respective track types). To test the generalisation performance, a network trained to detect time-invariant tracks is tested using the time varying test set and *vice versa*. In this case the performance drops to 69% (trained on time-invariant tracks, tested on time varying tracks) and 43% (trained on time varying tracks, tested on time-invariant tracks).

Adams and Evans implement MNET, a multilayer feedforward NN architecture for track detection inspired by the Hidden Markov Model (HMM) [34] (see Section 4.4.2 for a full discussion of HMM techniques). A method analogous to the forward-backward algorithm is used to allow the output of

each node to be calculated at each time step. The estimated sequence of track locations are then obtained by finding the node with the largest output at each observation time. Two networks are then derived from this architecture; MNET1, which is trained using a supervised learning algorithm, and MNET2, in which parameters are derived analytically from knowledge of the problem structure (a method used by Streit and Barrett [37] and Xie and Evans [38] to determine HMM parameters). In an extension to this, RNET, the nodes representing the HMM states are replaced by an MLP network, and it is trained using a supervised learning algorithm. The addition of the hidden layers creates a non-linear mapping between the input and state output.

The range of frequencies used to train and test these methods was split into eight subranges. Therefore the HMM, MNET1, MNET2 and RNET architectures have eight states corresponding to the track being located in each of these subranges. These architectures are compared against a MLP NN and a HMM using the Viterbi algorithm to track the frequency. The authors find that the HMM outperforms the other methods in tests where SNRs are between 4 and -5.6 dB. RNET achieves the closest performance to the HMM, followed by MNET1, NN then MNET2. However, the operational computational complexity of RNET and both the MNET architectures, $O(NM)$, is lower than that of the HMM, $O(M^2N)$. An advantage of MNET's architecture over the NN is that its number of nodes is tied to the problem formulation and is therefore predefined, whereas the size of a NN needs to be determined by trial and error. Also, compared with the NN, MNET has a smaller network size. This is also true when compared to RNET (which is also smaller than the NN), however, the addition of RNET's hidden layer creates a non-linear mapping from input to output, allowing it to model more complex data and achieve a higher detection rate. A limitation of the experimentation is the coarse frequency resolution into which the spectrograms are subdivided; this limits the method's ability to detect tracks which have small frequency variations and results in networks that have fewer states, simplifying the detection problem.

4.3.2. Unsupervised Learning

Methods using unsupervised learning may show more reliable application to real world cases as they are not trained to detect a specific track structure.

Di Martino *et al.* [35] propose the use of a two layer adapted Kohonen self-organising map, with an input layer of 147 nodes (three nodes for each input

pixel which represent time, amplitude and frequency) and an output layer of 49 nodes (N), applied directly to the spectrogram in an attempt to extract spectral tracks. The map is allowed to converge upon the spectrogram image and a cost function, $\Phi(W)$, is defined to test the convergence for the presence of a track, such that

$$\Phi(W) = \frac{\sum_{i=1}^{N} W_i^A}{N \sum_{i=2}^{N-1} (W_{i-1}^F - 2W_i^F + W_{i+1}^F)^2} \tag{19}$$

where W^F and W^A represent the weights attributed to the connection of the frequency and amplitude input nodes (respectively) to the output layer.

The method was applied to a spectrogram with a SNR of 2 (calculated as $\text{SNR} = 10 \log_{10}([\bar{P}_t - \bar{P}_b]/\sigma_b)$) and the network's detection resolution was taken to be a 7×7 pixel window in a 70×70 pixel spectrogram. The resulting image has a majority of the noise removed and shows a large response where the track is present in the ground truth data. The track in the original spectrogram is not continuous as noise obscures parts of it. The resolution of the self-organising map causes many of these gaps to be bridged however, this could also result in localisation problems and extend terminated tracks. The formulation of the cost function $\Phi(W)$ allows for the detection of high amplitude, low curvature tracks as its numerator takes a high value and the divisor a low value, equating to a high response. However, when a high amplitude high curvature track is encountered the function will take on a low value, giving a high probability of false negative detections. This would also be the case for low amplitude low curvature tracks, which is a limitation when low SNR track detection is needed.

4.4. Statistical Models

Statistical models determine the optimal path through a number of detections, which include false and true positives, by calculating the path with the maximum likelihood. Hidden Markov Models [54] are well known for their application to this type of problem as they allow for the modelling of an unobservable stochastic process that can be observed through stochastic processes that produce a sequence of observations (in this case the STFT output).

A general limitation of the HMM is the automatic discretisation of an estimated continuous variable [17], in this case the signal's frequency. However, this does not affect its application to this problem as the continuous frequency is discretised during the STFT and the HMM estimates the state within these frequency bins. Another limitation associated with HMMs is the automatic determination of the model's parameters given some training data. An approximation to the solution can be achieved using iterative methods such as the Baum-Welch algorithm [55], the Extended Baum-Welch algorithm [56], which are generalised Expectation-Maximisation algorithms, or gradient techniques [57]. Employing such methods can reduce the generalisation ability of the resulting HMM to track variations which are similar to those present in the training data - a typical problem with supervised learning methods. Anderson *et al.* [58] further discuss issues associated with HMM models.

4.4.1. Dynamic Programming

Scharf and Elliot [36] model a frequency track as a random walk, $z_k = z_{k-1} + \epsilon_k$, and derive a dynamic programming approach for track extraction. The method is described as being applicable to frequency or phase tracking, stating that "the distinction between the two is more imagined than real". A logarithmic likelihood function, l, is defined such that

$$l \sim \frac{1}{2\sigma^2} \sum_{n=0}^{N-1} Re\{e^{(-i\phi_{nk})} P_n(\hat{\omega}_n)\} + \sum_{n=0}^{N-1} \ln p(\hat{\omega}_n|\hat{\omega}_{n-1}) \tag{20}$$

where $\hat{\omega}_n$ is the estimated discrete frequency state, $p(\hat{\omega}_n|\hat{\omega}_{n-1})$ is the transition probability which are chosen to model a notion of physical reality, σ is the standard deviation of the time domain noise and $e^{(-i\phi_{nk})}$ is the phase shift of the STFT, where ϕ_{nk} is the total accumulated phase after nk steps (k is the number of samples in which the phase is assumed to increase at a fixed linear rate). Here σ is fixed and therefore the standard deviation of the noise is assumed to be stationary and known *a priori*. The most likely track is one that maximises l. Dynamic programming is used to determine this by calculating the best path through the observed peaks (a more complete discussion of a related non-linear tracking algorithm is presented in [59]).

The algorithm was tested on two spectrograms with a carrier-to-noise ratio (SNR of a modulated signal) of $-3\,\mathrm{dB}$ (time domain) using 60 time steps of data to calculate the optimal path. They note that even when

STFT peaks are unreliable the method tracks the true frequency. However, it can be observed in the qualitative data presented that, at several points, the tracking diverges from the true frequency.

4.4.2. Hidden Markov Model

Shin and Kil [40] argue that to effectively track a signal any *a priori* knowledge of the signal's behaviour should be used and that Hidden Markov Models allow for this.

4.4.2 a) Single Track. Streit and Barrett [37] demonstrate the use of a HMM spectrogram frequency tracker. In this formulation only the most powerful frequency bin is used in each observation, limiting the method to the detection of single tracks. The inclusion of a zero state allows the tracker to account for the disappearance and initiation of a track, the occurrence of which is detected using a threshold value. Frequency cells composed of a subset, or gate, centred on the previously detected frequency cell (therefore representing the allowed wandering frequency positions) are identified with the states of the hidden Markov chain. Analytic expressions for the basic parameters of the HMM are obtained in terms of physically meaningful quantities. It is shown that the computational complexity of the Viterbi algorithm is $[(n+1)+c_1]^2T$, where c_1 is the complexity (in units equivalent to addition) of computing the measurement PDF (in the case where it is computed for each symbol in the measurement vectors), and the computational complexity of the forward-backward linking algorithm is $[(n+1)+c_2]^2T$, where c_2 is the PDF calculation complexity in units equivalent to multiplication.

The performance of the HMM tracker was qualitatively evaluated for two sets of simulated data demonstrating good detection results in time domain SNRs of $-20\,\mathrm{dB}$ and $-23\,\mathrm{dB}$ with the disappearance and initiation of tracks. The HMM tracker was compared to the dynamic programming method presented by Sharf and Elliot [36] and it was found that their method is equivalent to an HMM using real valued continuous measurement vectors. However, Sharf and Elliot do not include a zero state to account for the absence of a signal. It is noted that the dynamic programming algorithm presented for maximising the likelihood function l, Eq. (20), is equivalent to the Viterbi algorithm.

Paris and Jauffret [44] and Shin and Kil [40] both investigate the use of HMMs applied to this problem. Both compare forms of the Viterbi line detector (a global optimisation scheme) while Paris and Jauffret also test the

forward-backward (F-B) local optimisation algorithm.

Shin and Kil use the smoothed amplitude of the short-term integrator as a feature for the algorithm. Subsequently, a double threshold Viterbi line detector is employed; two thresholds are used to identify which STFT bins are to be linked, reducing the algorithm's computational load. A likelihood function based upon each cell's amplitude and linking distance is used which, as this is based upon amplitude information, allows the algorithm to cope with time varying signal and noise characteristics. They find that below a SNR of -4 dB (time domain) the performance of the Viterbi is weak as false detections become apparent. To compensate for this they propose to extract features from projection spaces other than the spectrogram image and employ feature fusion, optimisation and classification techniques (discussion of this is beyond the scope of this paper). Qualitative results (of the Viterbi detector alone) were presented from one spectrogram image showing that tracks with slow spatial variation are recovered accurately.

Paris and Jauffret propose to integrate SNR estimates into the HMM algorithm to improve tracking performance when the spectrogram SNR is not known *a priori*. Two methods for estimating the SNR of a spectrogram are proposed: a parametric maximum likelihood estimation (MLE) which gives the scaled likelihood, defined as

$$b^s(s_{ij}) \simeq exp\left[\frac{Ms_{ij}}{\sum_{l=0}^{M-1} s_{il}}\right] \tag{21}$$

and a non-parametric probabilistic integration of the spectral power (PISP) approach by taking the normalised spectrogram, such that

$$\bar{s}_{ij} = \frac{s_{ij}}{\sum_{l=0}^{M-1} s_{il}}. \tag{22}$$

Implementing a SNR estimate in this way slightly reduces the computation time associated with the MLE method. Calculating the likelihood of the current observation in terms of its mean allows for detection even if the noise level varies with time.

It was shown that both the Viterbi and the F-B algorithms perform equally well in the experiments, and that estimating the SNR results in no loss of performance (it is also noted that both SNR estimates perform equally well). However, it is stated that the Viterbi algorithm performs many

more comparisons (but fewer multiplications) than the forward-backward algorithm and that PISP is less computationally intensive than MLE. One shortfall of these methods is that they do not take into account the appearance or disappearance of a frequency track or the existence of multiple tracks.

Jauffret and Bouchet [41] outline a probabilistic data association (PDA) method coupled with the Viterbi line extractor. The spectrogram is thresholded resulting in a set of false alarms and a set of true detections. The likelihood of a track in the spectrogram is calculated to be proportional to

$$L(\mathbf{S}_{i*}|y_i) = 1 - P_d + \frac{P_d}{\lambda} \sum_{j=0}^{M-1} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(s_{ij}-y_i)^2}{2\sigma^2}}, \qquad i = 0, 1, \ldots, N-1 \quad (23)$$

where $\mathbf{S}_{i*}$ denotes row i of the spectrogram $\mathbf{S}$, σ is the standard deviation of the Gaussian distribution, y_i represents the the state of the system at time i, P_d is the probability of detection and λ is the probability of false alarm. Several assumptions regarding the nature of the data are made in this calculation which are outlined in the paper. The Viterbi line detector is then used to extract the most likely track from the spectrogram.

This method was shown to reliably detect slowly varying tracks when the SNR is above 4 dB, in both simulated and real world examples. Van Cappel and Alinat comment that "probabilistic data association with severely limited branching factors suffers from various difficulties due to the low SNR and to the variability of track frequencies and amplitudes" [42]. The proposed method also does not account for the birth and death of tracks.

Gunes and Erdol [45] argue that if concentrated noise exists in specific frequency ranges, deriving the observation estimates with respect to the full spectrum will typically lead to unbalanced observation likelihoods. They outline a HMM for the detection of vortex frequency tracks in low SNR conditions which overcomes this limitation by defining an observation likelihood measure based upon the interpolation between local maxima of the spectrum. The spectral estimate's local maxima are determined within each time frame, these maxima form the centres of windows within which interpolation across subsequent time frames is performed. This results in a set of smoothed local maxima which are used to mask the original spectral estimate during the observation likelihood calculation, thus the calculation is determined with respect to a subset of the spectrum.

In [45] the forward-backward linking algorithm was implemented to perform track association. The method was shown to reliably detect tracks within two spectrogram images, one of which exhibits time variant noise irregularly distributed throughout the frequency spectrum and the other Gaussian noise.

4.4.2 b) Multiple Tracks. Paris and Jauffret demonstrate a HMM scheme which allows for multiple track detection [43]. The F-B algorithm imposed the constraint that two tracks cannot inhabit the same place in state space by adding the track's rate of frequency change, $\dot{f}$, to the representation of the state y_i, such that

$$y_i = \frac{1}{\Delta f} \begin{bmatrix} f_i \\ \dot{f}_i \end{bmatrix} \tag{24}$$

where f is the state's frequency position and Δf is the frequency resolution used in the STFT.

These modifications also allow two tracks to cross the same point in a spectrogram. The appearance and disappearance of the tracks, which was left unaddressed previously [44], is determined by a sequential test using the mechanism of the F-B algorithm. The tracks are extracted from the spectrogram and their start and end points are calculated using past and future detections.

This technique is not a true general multi track detector as an upper bound on the number of tracks to be found is a parameter of the algorithm. Tests using this algorithm show that it performs well both with known and unknown SNR, with a slight rise in the mean square error with the latter. In a test on a synthetic spectrogram with multiple frequency tracks that were highly corrupted the algorithm recovers them all accurately. When the algorithm is applied to a real spectrogram it again accurately detects the frequency tracks. However, overestimating the number of tracks increases computational workload which would not be desirable in a real time application.

Xie and Evans [38] propose a multi track approach using the Viterbi algorithm which operates on the thresholded output of the STFT. They define a "mixed" track and use the Viterbi algorithm to produce the maximum *a posteriori* "mixed" track estimates. The estimation of the threshold requires good knowledge of the SNR of the signal under scrutiny. They later present further results [39] which remove the need of thresholding and show superior performance over the previous method (although this is at the expense

of increased storage space). To separate the "mixed" tracks into individual tracks it is proposed to use amplitude and transition probability information. If two tracks do not cross then transition information alone is enough; if they do cross then they are assumed to have different constant amplitudes and this, together with state transition information, is used for separation.

Simulation results are presented which show good tracking performance when the track's frequency varies by 5 Hz over approximately 11 hours of data. In these restricted conditions the tracker is able to detect a track at a SNR of −23 dB (time domain).

Van Cappel and Alinat propose an alternative method using multiple HMMs to utilise several frequency track variation models in parallel [42]. It is noted that the solution to track detection must be designed "firstly in taking into account as long as possible observed data blocks (batch processing), secondly in delaying the decisions (knowledge of future) and thirdly in using several frequency line variation models in parallel". A HMM-based track extractor is described which works upon thresholded STFT outputs where the threshold is related to the noise level. A generalised likelihood ratio test is performed using two models in parallel as two standard deviation estimates are used; one accounting for stable tracks and the other for unstable. The maximum of both is taken and the likelihood calculated. Three track models are taken into account, the first a stable track with order 0, the second an unstable track with order 0 and lastly a stable track with order 1. The change from one model to another is triggered by a Bayesian test using the track variation of the recent observations.

Qualitative results are presented for a spectrogram containing tracks which exhibit a large amount of variability. It can be seen that each model has the ability to detect tracks with different characteristics separately and, when combined, the mechanism incorporates the detection attributes of all the models contained.

4.5. Tracking Algorithms

Tracking algorithms such as the Kalman filter [60] form a series of estimates, or predictions, of a system state (in this case the track position). Based upon an existing estimate, the state of the system in the next time frame is predicted; once a measurement becomes available (in this case the STFT output) the estimate is updated according to the observation and the process is repeated. An issue associated with this type of detection method, especially when applied to areas which need quick, accurate detections, is

the latency of detection, i.e. the number of observations that are required to update the *a priori* estimate to accurately locate and track a feature.

4.5.1. Particle Filter

The particle filter is a sequential Monte Carlo method, in which the posterior probability density function (PDF) is represented as a set of particles and associated normalised weights in state space, which generalises the Kalman filter [61]. At each time step particles are drawn from the previously calculated set with probabilities proportional to their weights. The weights of these particles are then updated according to the current observation and used to calculate the Bayesian estimate of the state for the current time step. This is repeated at each time step and has the effect of tracking a state estimate of a non-linear non-Gaussian process, in this case the frequency of a spectrogram track, through time.

Shi and Chang investigate the use of particle filters to extract the formants (peak frequencies of speech signals and therefore tracks) from a spectrogram [10]. Preprocessing converts the spectrogram from log energy to the grey-scale range (0–255). Particle filtering is employed to estimate the state (the frequency) of the k^{th} formant at time t, $\hat{F}_t^{(k)}$, based upon the state estimate in the previous time step, $\hat{F}_{t-1}^{(k)}$, which represents all the previous observations, such that

$$\hat{F}_t^{(k)} = E[F_t^{(k)} | R_t^{(k)}, \hat{F}_{t-1}^{(k)}] \tag{25}$$

where $R_t^{(k)}$ is the formant spectrum region (the observation).

The prediction stage updates the current state to predict the frequency location for the next observation, and, as the next observation becomes available, this prediction is updated. The prior $p(F^{(k)})$ and conditional prior $p(F_t^{(k)}|F_{t-1}^{(k)})$ PDFs are assumed to be Gaussian or products of Gaussians,

$$p(F^{(k)}) \sim \mathcal{N}(F^{(k)}; \mu_{F^{(k)}}, \sigma_{F^{(k)}}) \tag{26}$$

$$p(F_t^{(k)}|F_{t-1}^{(k)}) \sim \mathcal{N}(F_t^{(k)}; F_{t-1}^{(k)}, \sigma_{F_{t|t-1}^{(k)}}) \tag{27}$$

where $\mu_{F^{(k)}}$ and $\sigma_{F^{(k)}}$ are the PDF's mean and standard deviation and are learnt from manually labelled formant tracks. The particle filter algorithm can thus detect the track on a frame by frame basis.

In this form particle filter is applicable to detecting a single track in a spectrogram. However, the paper outlines a method to split the spectrogram

into k non-overlapping regions $R^{(k)}$ and to perform tracking in each region, therefore allowing for multiple tracks to be followed.

The results of the experiments presented in [10] show an average frequency error of 71, 115 and 113 Hz for the first three formants (it should be noted that the tracks in this application cover a larger range of frequencies compared to the very narrow band tracks discussed in other papers). This is a relatively large error, especially for applications which require accurate frequency estimation to perform subsequent source classification.

4.6. Relaxation Methods

Relaxation algorithms such as simulated annealing take their analogy from annealing in metallurgy which involves the heating and controlled cooling of a material to increase the order of its atoms and reduce defects.

4.6.1. Simulated Annealing

Lee [46] applies simulated annealing (SA) to globally optimise a cost function defined upon the SNR over time. The assumption is made that the initial frequency location is known and that the track is constrained to a frequency variance of 0, 1 or -1 frequency bins in each time step. This assumption limits the method's application to cases where it is known *a priori* that the spectrogram contains a track. If this is not the case and the method is applied, a false track throughout the spectrogram will be detected. The cost function is defined as

$$C(i) = \sum_{k=1}^{K} (\alpha \mu_k - s_{ia_k}) \tag{28}$$

where $(a_k)_{k=\overline{1,K}}$ is monotonically increasing sequence such that $a_k = j$ if s_{ij} belongs to a track and $a_k \neq a_t$, $k \neq t$. Term s_{ia_k} is the power of the track at point (i, a_k) and α is a threshold that controls the detection sensitivity, μ_k is the estimate of noise from the previous track or the spectrogram border to the current track, such that

$$\mu_k(i) = \begin{cases} \frac{1}{a_k} \sum_{j=0}^{a_k-1} s_{ij}, & \text{if } k = 1 \\ \frac{1}{a_k - a_{k-1}+1} \sum_{j=a_{k-1}+1}^{a_k-1} s_{ij}, & \text{if } k > 1. \end{cases} \tag{29}$$

The global cost function is defined as $C_T = \sum_{i=0}^{N-1} C(i)$ and maximising the SNR becomes minimising the difference between the noise and signal. The

solution to this, provided by the SA algorithm, guides the solution towards tracks in the spectrogram.

An initial track configuration is generated at random which is then incrementally improved using the SA algorithm. This method was tested using a test set containing −18 to 3 dB SNR (time domain) spectrograms which have a single track at 64 Hz. In these experiments the initial frequency location of the track is known and the detection initiated from this frequency bin. The CPU time required to detect a single frequency track within a 128×128 pixel spectrogram varied from 380 to 572 seconds. Qualitative results are presented which demonstrate reliable detection of time-invariant tracks in most SNRs, with the detection in some cases varying from the true location. Additional experimental results are presented which test the need for accurate *a priori* knowledge of the track's frequency location. The initial state was set to 75 Hz and the experiments repeated with the method successfully recovering the track. However, this experiment was conducted upon a single spectrogram with a very high SNR of 3 dB (time domain).

4.7. Expert Systems

Lu *et al.* [47] employ the use of an expert system and priority ranking to improve the performance of weak track detection and tracking by allowing for a certain degree of learning. The following stages are followed: the broadband component of the STFT output is removed from the signal, a double threshold is taken where the spectrogram is thresholded with a low threshold value and then a second is applied "to make further judgement according to the characteristics of the shape of the frequency line and timing continuity". The detected frequencies are then stored in an expert database and their initial priority ranks are set to zero. The threshold of each entry in the expert database is adjusted and the narrow region encompassing the initial detection is tested according to the characteristics of a typical track. The priority ranking is reduced or increased depending on the outcome of these tests. A track is eliminated when its priority falls below zero, thus false detections are eliminated.

Qualitative results are presented from the application of the method to one synthetic spectrogram containing 4 tracks, the weakest having a SNR of −9.76 dB (time domain), which demonstrate good detection performance. Another qualitative detection within a real world spectrogram is also presented, but these detections are not quantitatively analysed.

5. Discussion

To recapitulate, this paper aims to survey and review algorithms representative of the intersection between the areas of acoustic analysis and pattern recognition for the problem of spectrogram track detection. To accomplish this, a problem statement, set of evaluation criteria, taxonomy of algorithms and a review of each algorithm from within the taxonomy has been presented. This section presents an evaluation of the algorithms with respect to the defined criteria and a discussion of the algorithms' strengths and limitations.

5.1. Algorithm Evaluation

The reviewed algorithms have been evaluated with respect to specific criteria which are prerequisites for a reliable and successful spectrogram track detection algorithm. These criteria have been defined in Section 2.2 and the results are summarised in Table 3.

5.2. Technique Limitations

In addition to the benefits of each technique and the insight into the nature of the data which the study of these methods gives us, we also identify several fundamental limitations of the techniques which have been presented.

Smoothing of the spectrogram using spatial filtering techniques cannot guarantee that two close tracks have not been merged. It can also cause instances where a detected track has been shifted from the true location through the use of such a filter. These problems carry over to methods employing some form of resolution reduction as a preprocessing stage.

Di Martino *et al.* describe problems which follow from using multiple hypothesis testing methods [27], the first being the "number of possible solutions which grows up when the search depth increases" and therefore "thresholding during the search is necessary in order to avoid the combinatory explosion". Also that "the decision process is local and so very sensitive to initialisation".

Thresholding and likelihood estimates are statistically powerful and simple methods. However, when the SNR of a spectrogram is low the probability density functions overlap considerably. Consequently, a low threshold value will result in a high true positive rate but will also detect many false positives. Conversely, if the threshold value is set to a low value the resulting detection will contain few false positives but false negatives start to be the

Table 3: Analysis of spectrogram track detection algorithms ('-' denotes the inability to make a judgement regarding the criteria for a specific method due to lack of information).

Algorithm	C1 Low SNR	C2 Temporal Noise Variability	C3 Birth/Death Tracks	C4 Multiple Tracks	C5 Closely Spaced Tracks	C6 Crossing Tracks	C7 High Track Variability	C8 No *A Priori* Shape Assumption	C9 Track Association	C10 Computationally Cheap
Maximum Likelihood										
Single frequency [22]	N	Y	Y	N	N	N	Y	Y	Y	Y
Multi harmonic [24]	N	Y	Y	N	N	N	Y	Y	Y	Y
Correlation [23]	-	N	Y	Y	Y	Y	Y	N	N	N
Image Processing Techniques										
Likelihood ratio & morphological operators [25]	-	Y	Y	Y	Y	Y	N	Y	Y	-
Multistage decision process [27]	-	Y	Y	Y	Y	Y	Y	Y	N	Y
Steerable filter & multistage decision [28, 29]	N	Y	Y	Y	N	N	Y	Y	N	Y
Two-pass split-window [5]	N	Y	Y	Y	N	N	Y	Y	N	Y
Edge detector [30]	N	Y	Y	Y	N	Y	Y	Y	N	Y
Neural Networks										
ASM and MLP [31]	N	-	N	N	N	N	N	N	Y	-
Multi-layer perceptron [32]	N	-	Y	Y	Y	-	N	N	N	-
MLP using Ockham's networks [33]	N	-	Y	Y	N	Y	Y	N	N	Y
Kohonen self-organising map [35]	N	Y	Y	Y	N	N	N	Y	N	-
MNET1 [34]	N	-	N	N	N	N	N	N	Y	Y
MNET2 [34]	N	-	N	N	N	N	N	Y	Y	Y
RNET [34]	Y	-	N	N	N	N	N	N	Y	Y
Statistical Models										
Dynamic programming [36]	-	N	N	N	N	N	N	Y	Y	-
Viterbi & max amplitude [37]	-	N	Y	N	N	N	Y	Y	Y	-
Viterbi, "mixed" track & threshold [38]	-	N	Y	Y	Y	Y	N	Y	Y	-
Viterbi & "mixed" track [39]	-	N	Y	Y	Y	Y	N	Y	Y	-
Viterbi & double threshold [40]		Y	Y	Y	Y	N	N	Y	Y	-
Viterbi & PDA [41]	N	Y	N	N	N	N	N	Y	Y	-
Parallel, multi model detection [42]	-	N	Y	Y	-	-	Y	Y	Y	-
F-B linking, SNR estimate & track gradient [43]	Y	Y	Y	Y	-	Y	Y	Y	Y	N
F-B linking & SNR estimate [44]	Y	Y	N	N	N	N	Y	Y	Y	N
Viterbi & SNR estimate [44]	Y	Y	N	N	N	N	N	Y	Y	N
F-B linking & spectrum interpolation [45]	-	Y	N	N	N	N	Y	Y	Y	-
Tracking Algorithms										
Particle filter [10]	-	Y	Y	Y	N	N	Y	Y	Y	N
Relaxation Methods										
Simulated annealing [46]	-	Y	N	Y	Y	N	N	Y	Y	N
Expert Systems										
Double threshold & priority ranking [47]	-	-	Y	Y	Y	Y	Y	Y	N	Y

drawback. Another drawback of these techniques is the constant variation of the noise distribution present in real world noise environments. This problem then lends itself to machine learning techniques which are adaptive to the environment.

Although the reviewed RNET and MNET neural network architectures do not account for multiple tracks, track crossing and track birth/death, their HMM counterparts are able to. Due to their close similarity to HMM formulations, these properties should be easily transferred to their implementations.

The representative work of probabilistic data association (coupled with the Viterbi line detector) and dynamic programming assume that one track is present at any one time frame of the spectrogram. This limitation has been overcome with methods implementing hidden Markov models, some of which incorporate information regarding the current FFT observation into the likelihood measurement which allows them to account for time varying signal-to-noise ratio levels. However, many of the implementations which are shown to work in low SNR conditions are tested using tracks which are relatively stationary (typical variations are 1 Hz over minutes/hours of data). Anderson notes "the transition and measurement probabilities are derived effectively on the assumption that the actual tracks are piecewise constant, which is not at all the case" [58]. If the track varies too greatly the probabilities will not be able to represent the behaviour accurately and therefore the track will not be extracted to the accuracy needed for source classification. The representation of a probability distribution function as a set of particles, as in particle filtering, allows the modelling of non parametric system state distributions which can be dynamic due to particle population re-sampling at each iteration. However, this introduces added computational burden as many particles are needed to produce a good approximation and each of these needs to be updated at each iteration (along with their associated weights). With regard to the proposed HMM solutions, each perform specific aspects of the desired properties however, not one algorithm combines all of the desirable features to fully realise a viable solution.

The representative work based upon simulated annealing assumes that the initial track position is known. Although experiments have shown that it need not be known accurately to result in the correct detection of a track, it is unclear how much error is allowed for the method to work effectively. This also limits the method's application to spectrograms in which a track is known to exist.

The fundamental SNR limit of current techniques seems to be in the region of 2–4 dB in the frequency domain for tracks which exhibit low shape variation (this is derived by converting time domain SNR levels using assumptions of common spectrogram derivation parameters). This is not sensitive enough for some applications.

There appears to be a theoretical division in the literature present in this field. A number of methods concentrate on detecting the presence of a track within a window of data, and therefore conduct classification, whereas the remainder concentrate on detecting the presence of a track at a specific pixel location, and therefore conduct track detection. The practical effect of this divide is that classification mechanisms are applicable, and most often used, to 'clean' spectrograms, that is, to present the operator with a reduced complexity task where noise is suppressed and difficult to see features are highlighted. On the other hand, a reliable track detection mechanism replaces the need for such an operator all together, allowing the output to be directly passed to higher level decision mechanisms (be it an operator or computational system) for further processing.

6. Conclusions

It is hard to present a direct performance comparison of the outlined techniques as there is a large variation in the type of results presented in the literature. Several papers lack quantitative results, favouring qualitative analysis of one or two spectrograms instead. Additionally, where quantitative results exist, there is a lack of consistency in the type of data which each technique is tested upon. These inconsistencies include; testing upon synthetic data, real world data or both, the type of structure variation that tracks exhibit and the SNRs (even the measure of SNR) and noise environment present in the data set. This greatly inhibits the ability to form any direct comparison of results between papers describing different techniques.

The representative work from hidden Markov models and image processing techniques demonstrate applicability to this problem (albeit from different directions), each of the reviewed solutions demonstrate the ability to achieve one or more of the defined criteria. However, it seems that there has been no effort to combine all of these properties into one viable solution and therefore there is still room for improvement in order to meet the challenges posed by present applications.

This survey has been concerned with surveying track detection methods applied to spectrogram images. Techniques exist that include phase information derived from the FFT and these are not reviewed here. For further reading the following is recommended [62, 58, 63, 64].

Acknowledgements

We are grateful to Tatjana Aleksić (Department of Mathematics, University of Kragujevac, Serbia) for the mathematical treatment within this paper. The research has been supported by QinetiQ Ltd.[1] and the Defence Science and Technology Laboratory (DSTL)[2], with special thanks to Jim Nicholson[1] and Duncan Williams[2] for guiding the objectives.

References

[1] W. Koenig, H. K. Dunn, L. Y. Lacy, The sound spectrograph, J. Acoust. Soc. Am. 18 (1) (1946) 244–244.

[2] R. P. Morrissey, J. Ward, N. DiMarzio, S. Jarvis, D. J. Moretti, Passive acoustic detection and localisation of sperm whales (*Physeter Macrocephalus*) in the tongue of the ocean, Applied Acoustics 67 (2006) 1091–1105.

[3] D. K. Mellinger, S. L. Nieukirk, H. Matsumoto, S. L. Heimlich, R. P. Dziak, J. Haxel, M. Fowler, C. Meinig, H. V. Miller, Seasonal occurrence of north atlantic right whale (*Eubalaena glacialis*) vocalizations at two sites on the scotian shelf, Marine Mammal Science 23 (2007) 856–867.

[4] J. M. de Seixas, W. S. Filho, J. B. O. S. Filho, D. O. Damazio, N. N. Moura, A compact online neural system for classifying passive sonar signals, in: Proc. of the International Conference on Signal Processing Applications and Technology, 1999, pp. 1–5.

[5] C.-H. Chen, J.-D. Lee, M.-C. Lin, Classification of underwater signals using neural networks, Tamkang J. of Science and Engineering 3 (1) (2000) 31–48.

[6] W. S. Filho, J. M. de Seixas, L. P. Caloba, Principle component analysis for classifying passive sonar signals, in: Proc. of the IEEE International Symposium on Circuits and Systems, Vol. 3, 2001, pp. 592–595.

[7] S. Yang, Z. Li, X. Wang, Ship recognition via its radiated sound: The fractal based approaches, J. Acoust. Soc. Am. 11 (1) (2002) 172–177.

[8] J. Ghosh, K. Turner, S. Beck, L. Deuser, Integration of neural classifiers for passive sonar signals, Control and Dynamic Systems - Advances in Theory and Applications 77 (1996) 301–338.

[9] B. P. Howell, S. Wood, S. Koksal, Passive sonar recognition and analysis using hybrid neural networks, in: Proc. of OCEANS '03, Vol. 4, 2003, pp. 1917–1924.

[10] Y. Shi, E. Chang, Spectrogram-based formant tracking via particle filters, in: Proc. of IEEE ICASSP, Vol. 1, 2003, pp. I–168–I–171.

[11] B. G. Quinn, Estimating frequency by interpolation using Fourier coefficients, IEEE Trans. Signal Process. 42 (5) (1994) 1264–1268.

[12] J. Ward, M. Fitzpatrick, N. DiMarzio, D. Moretti, R. Morrissey, New algorithms for open ocean marine mammal monitoring, in: Proc. of OCEANS 2000, Vol. 3, 2000, pp. 1749–1752.

[13] S. E. Moore, K. M. Stafford, D. K. Mellinger, J. A. Hildebrand, Listening for large whales in the offshore waters of Alaska, Bioscience 56 (2006) 49–55.

[14] I. R. Urazghildiiev, C. W. Clark, Acoustic detection of north atlantic right whale contact calls using spectrogram-based statistics, The Journal of the Acoustical Society of America 122 (2007) 769–776.

[15] R. Urick, Principles of Underwater Sound, 3rd Edition, McGraw-Hill, New York, 1983.

[16] H. O. A. Wold, Forecasting by the chain principle, Time Series Analysis (1963) 471–497.

[17] P. J. Kootsookos, A review of the frequency estimation and tracking problems, Tech. rep., Systems Engineering Department, Australian National University (1993).

[18] B. G. Quinn, E. J. Hannan, The Estimation and Tracking of Frequency, Cambridge University Press, 2001.

[19] D. R. Brillinger, Time Series: data analysis and theory, International Series in Decision Processes, Holt, Reinhart and Winston Inc., New York, 1975.

[20] Y. H. Yang, Relaxation method applied to lofargram, Master's thesis, Naval Postgraduate School Monterey CA, U.S.A. (June 1990).

[21] T.-S. Chen, Simulated annealing in sonar track detection, Master's thesis, Naval Postgraduate School Monterey CA, U.S.A. (December 1990).

[22] D. C. Rife, R. R. Boorstyn, Single-tone parameter estimation from discrete-time observations, IEEE Transactions on Information Theory 20 (1974) 591–598.

[23] R. A. Altes, Detection, estimation, and classification with spectrograms, The Journal of the Acoustical Society of America 67 (1980) 1232–1246.

[24] R. F. Barrett, D. R. A. McMahon, ML estimation of the fundamental frequency of a harmonic series, in: Proc. of ISSPA 87, Brisbane, Aurstralia, 1987, pp. 333–336.

[25] J. S. Abel, H. J. Lee, A. P. Lowell, An image processing approach to frequency tracking, in: Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing, Vol. 2, 1992, pp. 561–564.

[26] V. A. Brahosky, A combinatorial approach to automated lofargram analysis, Master's thesis, Naval Postgraduate School Monterey CA, U.S.A. (June 1992).

[27] J.-C. D. Martino, J. P. Haton, A. Laporte, Lofargram line tracking by multistage decision process, in: Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing, Vol. 1, IEEE, 1993, pp. 317–320.

[28] J.-C. D. Martino, S. Tabbone, Detection of lofar lines, in: C. Braccini, L. DeFloriani, G. Vernazza (Eds.), Proc. of the 8th Int. Conference on Image Analysis and Processing (ICIAP), Springer, Berlin, 1995, pp. 709–714.

[29] J.-C. D. Martino, S. Tabbone, An approach to detect lofar lines, Pattern Recognition Letters 17 (1) (1996) 37–46.

[30] D. Gillespie, Detection and classification of right whale calls using an 'edge' detector operating on a smoothed spectrogram, Canadian Acoustics 32 (2004) 39–47.

[31] A. Khotanzad, J. H. Lu, M. D. Srinath, Target detection using a neural network based passive sonar system, in: International Joint Conference on Neural Networks, Vol. 1, 1989, pp. 335–440.

[32] N. Leeming, Artificial neural nets to detect lines in noise, in: International Conference on Acoustic Sensing and Imaging, 1993, pp. 147–152.

[33] G. D. Kendall, T. J. Hall, T. J. Newton, An investigation of the generalisation performance of neural networks applied to lofargram classification, Neural Computing and Applications 1 (2) (1993) 147–159.

[34] G. J. Adams, R. J. Evans, Neural networks for frequency line tracking, IEEE Transactions on Signal Processing 42 (4) (1994) 936–941.

[35] J.-C. D. Martino, B. Colnet, M. D. Martino, The use of non supervised neural networks to detect lines in lofargram, in: Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing, Vol. 2, IEEE, 1994, pp. 293–296.

[36] L. L. Scharf, H. Elliot, Aspects of dynamic programming in signal and image processing, IEEE Trans. on Automatic Control AC-26 (5) (1981) 1018–1029.

[37] R. L. Streit, R. F. Barrett, Frequency line tracking using hidden Markov models, IEEE Transactions on Acoustics, Speech and Signal Processing 38 (4) (1990) 586–598.

[38] X. Xie, R. Evans, Multiple target tracking and multiple frequency line tracking using hidden Markov models, IEEE Transactions on Signal Processing 39 (12) (1991) 2659–2676.

[39] X. Xie, R. J. Evans, Multiple frequency line tracking with hidden Markov models - further results, IEEE Transactions on Signal Processing 41 (1993) 334–343.

[40] F. B. Shin, D. H. Kil, Full-spectrum signal processing using a classify-before-detect paradigm, Journal of the Acoustical Society of America 99 (4) (1996) 2188–2197.

[41] C. Jauffret, D. Bouchet, Frequency line tracking on a lofargram: An efficient wedding between probabilistic data association modelling and dynamic programming technique, in: Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers, Vol. 1, IEEE, 1996, pp. 486–490.

[42] D. V. Cappel, P. Alinat, Frequency line extractor using multiple hidden Markov models, in: OCEANS '98 Conference Proceedings, Vol. 3, 1998, pp. 1481–1485.

[43] S. Paris, C. Jauffret, A new tracker for multiple frequency line, in: Proc. of the IEEE Conference for Aerospace, Vol. 4, IEEE, 2001, pp. 1771–1782.

[44] S. Paris, C. Jauffret, Frequency line tracking using HMM-based schemes, IEEE Trans. on Aerospace and Electronic Systems 39 (2) (2003) 439–450.

[45] T. Gunes, N. Erdöl, HMM based spectral frequency line tracking: Improvements and new results, in: Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing, Vol. 2, 2006, pp. 673–676.

[46] C.-H. Lee, Simulated annealing applied to acoustic signal tracking, in: E. R. Dougherty, J. T. Astola, C. G. Boncelet (Eds.), Proceedings of the SPIE, Nonlinear Image Processing III, Vol. 1658 of Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, 1992, pp. 344–355.

[47] M. Lu, M. Li, W. Mao, The detection and tracking of weak frequency line based on double-detection algorithm, in: Int. Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, 2007, pp. 1195–1198.

[48] B. G. Ferguson, Time-frequency signal analysis of hydrophone data, IEEE Journal of Oceanic Engineering 21 (4) (1996) 537–544.

[49] R. C. Gonzalez, R. E. Woods, Digital Image Processing (3rd Edition), Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[50] S. Haykin, Neural Networks : A Comprehensive Foundation, 2nd Edition, Prentice Hall, Upper Saddle River, N.J., 1999.

[51] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, Wiley-Interscience Publication, 2000.

[52] S. J. Nowlan, G. E. Hinton, Simplifying neural networks by soft weight-sharing, Neural Computation 4 (4) (1992) 473–493.

[53] G. D. Kendall, T. J. Hall, Improving generalisation with Ockham's networks: minimum description length networks, in: Third International Conference on Artificial Neural Networks, 1993, pp. 81–85.

[54] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.

[55] L. E. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, The Annals of Mathematical Statistics 41 (1) (1970) 164–171.

[56] D. Kanevsky, T. N. Sainath, B. Ramabhadran, D. Nahamoo, Generalization of extended baum-welch parameter estimation for discriminative training and decoding, in: Proceedings of the 9th Annual Conference of the International Speech Communication Association, 2008, pp. 277–280.

[57] L. R. Rabiner, S. E. Levinson, M. M. Sondhi, An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition, Bell System Technical Journal 62 (4) (1983) 1035–1074.

[58] B. D. O. Anderson, B. James, R. C. Williamson, Frequency line tracking, extended Kalman filters and some HMM problems, in: Proceedings of the Workshop on Hidden Markov Models for Tracking, 1992, pp. 1–8.

[59] L. L. Scharf, D. D. Cox, C. J. Masreliez, Modulo-2π phase sequence estimation, IEEE Trans. on Information Theory 26 (1980) 615–620.

[60] R. Kalman, A new approach to linear filtering and predication problems, Trans. of the ASME-Journal of Basic Engineering 82 (Series D) (1960) 35–45.

[61] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking, IEEE Transactions on Signal Processing 50 (2) (2002) 174–188.

[62] R. F. Barrett, D. A. Holdsworth, Frequency tracking using hidden Markov models with amplitude and phase information, IEEE Transactions on Signal Processing 41 (1993) 2965–2976.

[63] D. R. A. McMahon, R. F. Barrett, An efficient method for the estimation of the frequency of a single tone in noise from the phases of discrete Fourier transform, Signal Processing 11 (2) (1986) 169–177.

[64] D. R. A. McMahon, R. F. Barrett, Generalization of the method for the estimation of the frequencies of tones in noise from the phases of discrete Fourier transforms, Signal Processing 12 (4) (1987) 371–383.