Frequency based Algorithm for Robust Contour Extraction of Blue Whale B and D calls

Shyam Kumar Madhusudhana[†], Erin M. Oleson[‡], Melissa S. Soldevilla[‡], Marie A. Roch[†], John A. Hildebrand[‡]
[†] Department of Computer Science, San Diego State University, 5500 Campanile Drive, San Diego, CA 92182-7720
[‡] Scripps Institution of Oceanography, The University of California at San Diego, La Jolla, CA 92093-0205

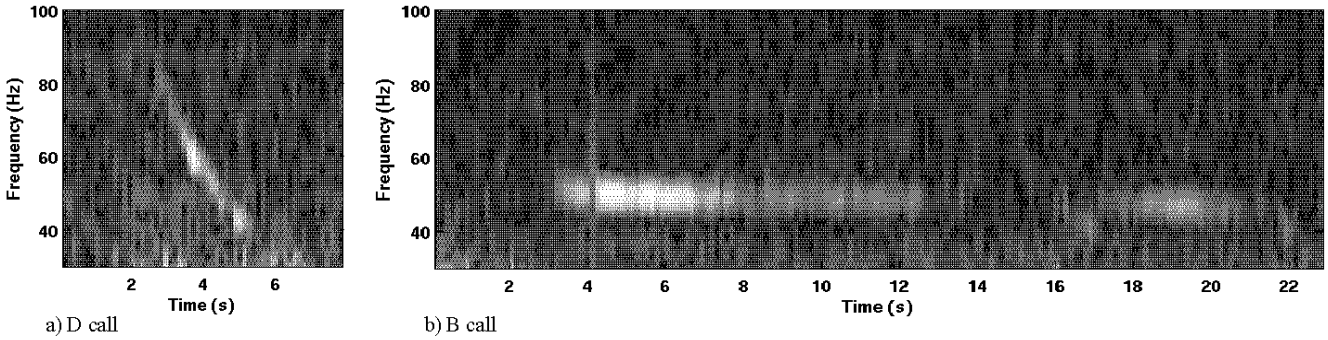
***Abstract*-The sea is home to a myriad of marine animal species, many of which use sound as a primary means of communication, navigation and foraging. Of particular interest are the Blue whales (*Balaenoptera musculus*) of the cetacean family. Massive commercial whaling prior to 1960 brought the species close to extinction and its population still remains very low. Passive acoustic monitoring of baleen whales has recently been used to provide long-term information about their presence and behavior, and provides an attractive complement to traditional visual based monitoring. In this work we present a frequency domain based algorithm developed for extracting the frequency contours of the dominant harmonic in tonal calls of blue whales (B and D calls). The algorithm uses a two pass approach to contour extraction. In the first pass, partial candidate contours are formed, followed by a second pass which uses the partial information to construct complete contours. When evaluated on a one hour labeled recording, the algorithm had 90% recall and 76% precision.**

I. Introduction

Baleen whales are thought to use vocalizations for various purposes – to establish territories, locate conspecifics and to attract mates [1]. Blue whales in the eastern north Pacific are known to produce at least three types of tonal calls and one pulsed call [1]. Of particular interest to us are two classes of tonal calls, namely B and D calls. Adaptation of existing detection algorithms is complicated by the high variability of several call parameters including duration, frequency content, and sweep rate. The blue whale D call contour sweeps steeply downwards within an approximate frequency range of 30 to 95 Hz and lasts 1-4 s [2] (Fig. 1a). D calls are primarily associated with feeding and are produced by both sexes [1]. B calls are quite long (~20 s) with their third harmonic being the most prominent, sweeping downwards within a relatively smaller frequency range of approximately 40 to 55 Hz [3] (Fig. 1b). Although their exact purpose is unknown, they are mostly associated with migration and are known to be produced only by males [1].

Acoustic based methods of monitoring marine mammals are complementary to visual methods and have been used for the purpose of studying behavioral patterns such as migration [3]. However, several factors make it difficult to detect these call types [1] automatically using conventional acoustic approaches. The presence of temporal characteristics and irregularities in these call types are some of the primary factors [2]. Spectrogram correlation [4] has been shown to be effective for tonal B calls, but frequency modulated D calls exhibit high variability in sweep range and slope, making it difficult to generate a suitable kernel for recognition by spectrogram correlation. Some calls have observed dropouts (see for example Fig. 1b between 12 and 17 s). These dropouts were examined in [5] where calls were analyzed to determine if any portion of the call could not be attributed to interference effects resulting from Lloyd's mirror. While interference effects did play a role, in some cases it was likely that variations in the call amplitude were due to variations in the production by the whale. Regardless of the cause, for the purposes of call counting these should be counted as a single call and our strategy is to bridge the contour.

Other noise sources also contribute to the difficulty of call detection and contour extraction. The low frequency components (those below 100 Hz) in the sound of passing ships and the line-induced buzz/hum (50-60 Hz) add to the problems making



a) D call b) B call

Fig 1. Spectrograms showing call contours. Spectrograms produced using 256 ms Hanning window and 1024 point FFT with an overlap of 50%.

[1] Throughout this literature, we will use the term "call" to refer to the vocalizations of interest to this study- namely the blue whale B and D calls.

978-1-4244-2126-8/08/$25.00 ©2008 IEEE

contour extraction more difficult. Also, fin whale (*Balaenoptera physalus*) calls that are downswept and are in the range of 15-50 Hz [2] fall in close frequency ranges of blue whale D calls.

II. Background

An effective and widely used technique for many mysticete calls is spectrogram correlation [4, 6]. The spectrogram of a prototype call is used to form a kernel which is correlated with an audio stream. When the correlation exceeds a user-defined threshold, a call is said to be detected. The technique has been successfully used on B calls (see [3] as an example), but is not capable of recognizing calls that vary significantly from the prototype in duration, shape, or frequency range. An alternative has been proposed for odontocete whistles which are also tonal in nature, but tend to be more complex and overlap due to large groupings of animals. A very brief description of an unpublished semiautomatic algorithm developed by Lammers for odontocete whistles appears in [7]. Given a pair of endpoints, spectral frames are analyzed for non-overlapping contours with evidence of harmonic structure. In [8], a two stage method is proposed where the first stage extracts a variable number of peaks based upon an analysis of spectral variance and forms segments from the connected peaks based on a local search of peak neighborhoods. The second stage assembles contours based upon the fit of contour statistics whose distributions are estimated from a set of training contours.

The spectrogram correlation method has been used effectively for B calls, but variations in the D calls prevent the construction of an effective D call kernel. Lammers' algorithm was tried, but could not really extract these low frequency calls. With non-trivial modifications to heuristics and constraints, the algorithm of [8] could be extended to function with mysticete calls. However, the simpler structure of mysticete calls and sparsity of call overlap make the proposed algorithm a viable alternative which can execute an order of magnitude faster than real time on modest general purpose hardware. The proposed algorithm is based upon ideas used to extract fundamental frequency contours in human speech (e.g. [9]) which rely on dynamic programming.

III. Data Collection

Acoustic recording packages (ARP) [10] were the primary instruments used for autonomous undersea audio data collection. An ARP consists of a self-contained sealed recording package attached to a frame with a tethered hydrophone. The ARP has sufficient storage and battery power to record for over a year and was configured to record audio at a sampling frequency of 1 kHz with 16 bits per sample. This configuration allowed complete sampling of the known blue whale calling repertoire. The Cortez and Tanner Banks in the Southern California Bight, about 180 km west of San Diego, were the primary sites chosen for data collection. They are known feeding grounds for several cetacean species, including blue whales. Acoustic data were recorded at several places around the Cortez and Tanner Banks from August 20, 2000 to February 20, 2004. However, data only from the two sites shown in Fig. 2 were used for analysis as they provided complete seasonal coverage for each year. Further details on the methods used for data collection may be found in [3].

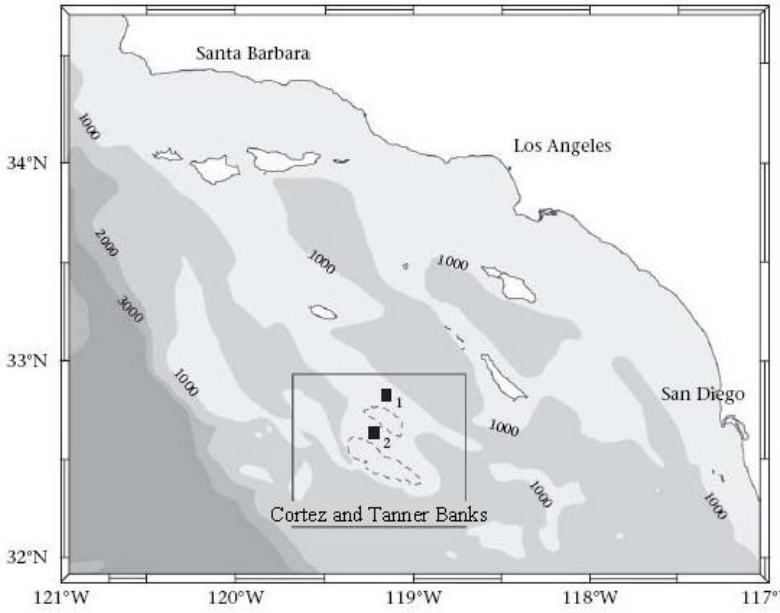


Fig. 2: Southern California Bight bathymetry (in m) showing Cortez and Tanner Banks study site. Site 1 (location: 32°41.3′N, 119°01.9′W; depth: 305 m). Site 2 (location: 32°35.8′N, 119°08.8′W; depth: 215 m). [1]

IV. The Algorithm

The primary goal in the development of the algorithm is to obtain maximum possible robustness and accuracy. The algorithm is inspired by a widely used pitch tracking algorithm, the robust pitch tracking algorithm (RAPT) [9], which extracts pitch contours from voiced human speech. RAPT uses normalized cross-correlation function (NCCF), a variant of the auto-correlation function (ACF), for obtaining candidate estimates of the true pitch period. The NCCF of the signal is computed and local maxima in the obtained values are identified. In order to speed up computation, RAPT first performs these steps at a significantly reduced sampling rate for all lags, and then again at the original sampling rate for lags in the vicinity of the peaks identified in the first

978-1-4244-2126-8/08/$25.00 ©2008 IEEE

step, thereby producing refined peak location and amplitude estimates. Our algorithm operates in the spectral domain instead, and obtains the estimates of energy and location of peak frequencies using the discrete Fourier transform of the signal. In RAPT, selection among candidate contours is based on a cost function which penalizes candidates with low energy and those that change rapidly. In our algorithm, a two pass dynamic programming approach is employed for selecting the best contour among the candidates.

A. Algorithm Outline

In a first pass, the frequency content of each frame is analyzed for spectral peaks after normalization by spectral means subtraction. As the peaks within the bandwidth of B and D calls are identified, a data structure is created that notes nearby peaks in the previous frame. A second pass builds potential contours from chains of peaks that can be inferred from the peaks and peak predecessors noted in the first pass. Domain specific heuristics are used to accept or reject candidate peaks for inclusion in contours and to accept or reject the contour. Generated contours are smoothed with a moving average process. An optional post processing step permits the bridging of successive contours that meet criteria which would suggest dropout. Fig. 3 provides a high level summary of the algorithm.

For convenience, important constants and variables are summarized in Tables I and II, respectively. Constants were obtained by extensive tuning while testing numerous segments of recordings including a variety of conditions (noise, multiple calls, etc.) which were disjoint from the evaluation data.

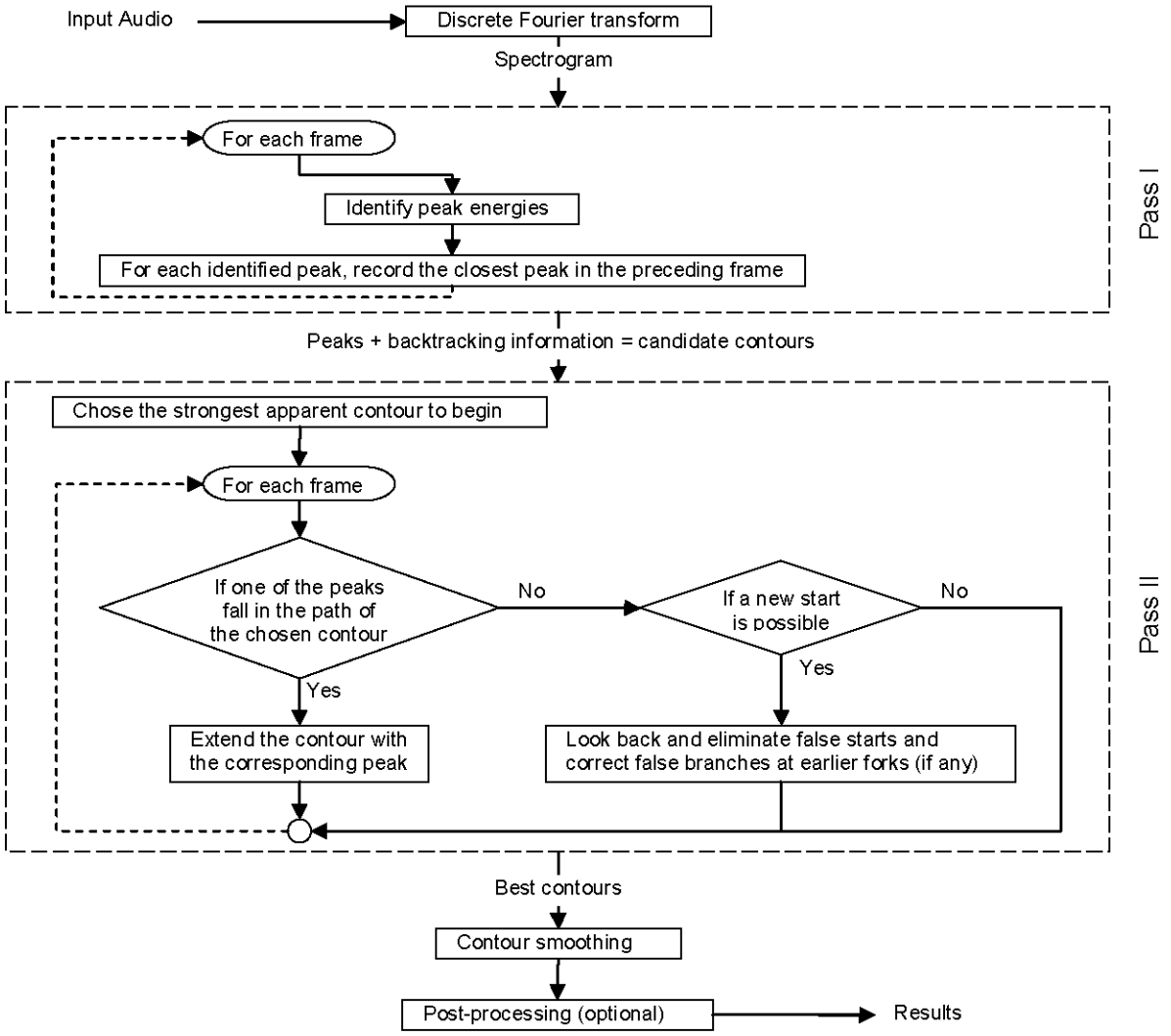


Fig 3. Overview of the proposed algorithm.

978-1-4244-2126-8/08/$25.00 ©2008 IEEE

TABLE I
CONSTANTS USED IN THE ALGORITHM

Constant	Meaning	Value
CAND_TR	Eliminate peaks whose energy falls beneath CAND_TR% of the strongest peak	80
N_CANDS	Max number of peaks to retain at each frame	10
FREQ_WIN	Limit search for adjacent peaks in previous frame to FREQ_WIN Hz from current peak	8
PRR	Peak Reject Ratio. Peaks smaller than PRR% of *MaxPeaksMean* are rejected.	90
PRR_S2	Peak Reject Ratio stage 2, allows more lenience for accepting slightly more weaker peaks	70
PP_GAP	Upper limit on separation between consecutive contours (s), used in post-processing	1.5

TABLE II
SYMBOLS USED IN THE ALGORITHM

Symbol	Meaning
$\Phi_{i,k}$	Energy for frame i at frequency bin k, after normalization
$PeakIdxs_{i,k}$	Location of the k^{th} peak in the i^{th} frame, in the normalized power spectrum (Hz)
$Peaks_{i,k}$	Energy at the k^{th} peak in the i^{th} frame, in the normalized power spectrum
$\beta_{i,k}$	Index to the peak in $PeakIdxs_{(i-1)}$ that is closest to $PeakIdxs_{i,k}$
MaxPeaksMean	Mean of the highest peaks from every frame

B. Data Preparation / Input Transformation

Audio data is processed in 60 s segments which are long enough to completely contain the longest call type (B call, ~20 s). Consecutive segments are considered with a 50% overlap, in order to avoid losing portions of calls that lie along the boundaries of segments. The segment length is chosen so that the overlapping region is long enough to completely contain at least one B call (~20 s). Another reason for considering such long segments is for suppressing an undesirable effect of normalization, which will be discussed later. For each segment, the power spectrum is computed from the discrete Fourier transform of the signal taken in frames of 256 ms zero-padded to 1024 samples. A frame length of 256 ms has been chosen in order to have better frequency resolution. All operations are limited to a frequency bandwidth of 30 to100 Hz that completely covers all variations of B and D calls. The prominence of line-induced noise and of the sound of passing ships, in the spectrum is reduced through the use of spectral means subtraction using mean estimates from short (3 s) initial and trailing sections of the segment. The drawback to this approach is that when these sections contain parts of actual calls, they may reduce the prominence of other calls in the segment thereby affecting accuracy. However, since the calls have a frequency sweep, there is negligible effect on any single frequency bin. In addition, this drawback can be further suppressed by choosing longer segments (~60 s).

C. Dynamic Programming

Contour tracking is performed over two forward passes across frames of the normalized power spectra. In the first pass, the data required for backtracking is prepared. For each spectral frame Φ_i, local maxima are identified as candidates for inclusion in a contour. The peak locations are refined by parabolic interpolation on the three samples of Φ_i defining every local maximum. Peaks whose energy exceeds a certain percentage (CAND_TR) of the highest peak for the frame are noted. If a large number of such peaks are obtained, the list is pruned such that only the highest N_CANDS peaks remain. The peak locations (in Hz) for the i^{th} frame are stored in $PeakIdxs_i$ and the corresponding energy values are stored in $Peaks_i$.

Once all frames have been processed, *MaxPeaksMean*, the mean of the highest peaks from all the frames, is computed as

$$MaxPeaksMean = \frac{\sum_{\text{i = all frames}} \max(Peaks_i)}{\text{Number of frames}}. \quad (1)$$

Fig. 4 shows information from a representative first pass on a synthetic data set. Each peak is denoted with an asterisk. If any peak in the $(i\text{-}1)^{th}$ frame lies within a frequency distance of FREQ_WIN Hz from the k^{th} peak in the i^{th} frame, its index in $Peaks_{(i-1)}$ is stored in $\beta_{i,k}$ (see Fig. 4b) which can be thought of as representing a possible contour path from frame i back to frame $(i\text{-}1)$. In case of multiple previous peaks lying in the range, the highest peak is chosen. When no previous peaks lie in the range, $\beta_{i,k}$ is set to null, indicating that any possible contour passing through frame i frequency k would terminate at this point. The values of $\beta_{i,k}$ for $i = 1$ are all set to null. The first pass terminates with β containing backward chains of potential contours. With *PeakIdxs* showing the frequency and β the connections between peaks, we can observe a form of connectivity across peaks from

978-1-4244-2126-8/08/$25.00 ©2008 IEEE

consecutive frames. These connected sets of peaks shown in Fig. 4a comprise the set of candidate contours that will be examined in the next pass.

The second pass identifies the best contour within a segment through the application of rules and heuristics. The algorithm progresses in the direction of increasing time, incrementally following the best contour, among the candidates available at each frame, as described in the next section. When conditions are met for adding a peak to the best contour, we let its index within the list of peaks for that frame ($PeakIdxs_i$) be denoted as *LastBest*, which is used to track the location of the best contour[2] in the current frame. *LastBest* is initialized to null. Whenever we start to follow a new contour (including the first time), *LastBest* will be set to the strongest peak in the frame under the assumption that the peak of highest energy is most likely to be part of the next contour. This assumption may lead to a false start, which is addressed later in this section. In addition, when a new contour is started, a flag called *NewStart* is also set, which is used to note that the next output should be marked as the first frame of a new contour.

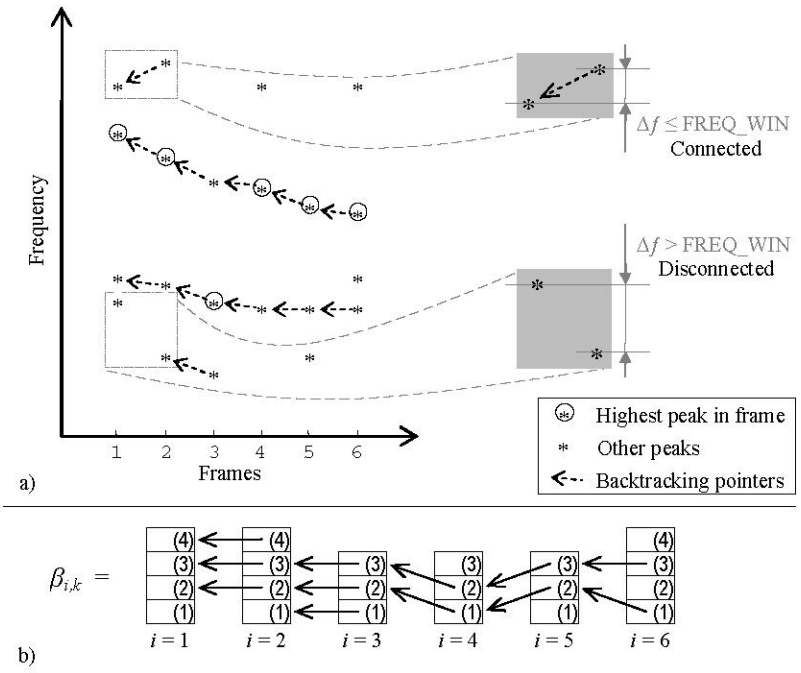


Fig 4. Snapshot of a first pass on synthetic data. a) Plot showing the location of peaks in each frame. The backward arrows from each peak point to the closest best from the previous frame. Zoomed regions demonstrate the use of FREQ_WIN for establishing connectedness. b) The corresponding β structure populated as described. The k^{th} peak in the i^{th} frame is identified with a (k) inside the corresponding block. The backward arrows indicate the index stored in each $\beta_{i,k}$.

As each frame is examined, potential contours are incrementally generated. These contour elements (partial results) are stored as they are generated in an array of 3-tuples which we denote as *fx*. Each tuple contains (*i*, *peak_location, start_flag*) where:

- *i* indicates the contour passes through the i^{th} frame
- *peak_location* is the contour's frequency in frame *i*,
- *start_flag* is an indicator function which is set when the entry denotes the start of a new contour.

For notational convenience, fx_t will denote the t^{th} tuple, and $fx_{t,m}$ the m^{th} item of tuple fx_t. *endIdx* refers to the index of the last tuple of *fx* with *endIdx* of 0 denoting the empty *fx*. The operation of adding a tuple to *fx* will be described as extending the contour by (*i, peak_location, start_flag*).

D. Heart of the Second Pass

Each frame is considered in the context of either extending an existing contour or possibly creating a new one. We begin by examining the extension of contour. To extend the contour into the i^{th} frame, there must exist a candidate peak of sufficient energy with a back pointer which points to a peak at frame *i*-1 which lies on the current contour. More formally, this occurs when the following conditions are met:

i) LastBest ≠ null — indicates if a contour is being followed at all
ii) $\exists\ k$ such that $\beta_{i,k} = Peaks_{(i-1),LastBest}$ — $k \in$ {indices to all stored peaks in the i^{th} frame}
iii) $Peaks_{i,k} \geq MaxPeaksMean \times$ PRR — for *k* identified in previous condition

If multiple peaks meet the criteria, we select the peak which is closest in frequency to the previous frame's best candidate peak ($Peaks_{(i-1),LastBest}$). This decision is made to favor smoothness in the contour. We denote the index to such a peak in $Peaks_i$ by ρ.

If contour continuity cannot be established, the strongest peak in the current frame is considered to be a candidate for a new contour. In this case, *LastBest* is set to the strongest peak and *NewStart* is set to true, as described earlier, in preparation for examining the next frame, and the algorithm continues with the next frame. When contour continuity can be established, *LastBest* is set equal to ρ and the algorithm proceeds with determining if the current frame extends a contour or if there is a possibility for the start of a new contour at the current frame, as described below.

A new contour may have started when no contour is currently being followed as indicated by the *NewStart* flag and the previous contour (if any) ended two or more frames ago. This occurs when both conditions below are met:

i) *NewStart* is set
ii) *endIdx* = 0 (no previous contour) or $fx_{endIdx,1} < i$-1

[2] Henceforth we use just the term contour to refer to the (best) contour being followed. Wherever ambiguities arise, they will be resolved explicitly.

978-1-4244-2126-8/08/$25.00 ©2008 IEEE

If the above conditions are not satisfied, the algorithm skips to examining conditions at the next frame after extending the current contour by (i, $PeakIdxs_{i,\rho}$, 0) as the peak at $Peaks_{i,\rho}$ satisfies continuity of an existing contour. Otherwise, it attempts to confirm the possibility a new start. If $fx_{endIdx,2}$ and $PeakIdxs_{i,\rho}$ are within FREQ_WIN Hz range of each other and they are both separated by only one or two frames, then it may not be a true new start but one longer contour with small segmentation which can be realized by joining the two ends. In such a case, we will not treat this as a new start. These segmentation gaps may have been caused by an absence of peaks at those locations in the intermediate frames, due to the presence of too many higher peaks at other locations or due to the strategy of selecting closer peaks over stronger ones employed at an earlier frame. Joining the two ends involves extending the contour by (i-1, intermediate-peak-location, 0) followed by (i, $PeakIdxs_{i,\rho}$, 0). The intermediate peak location is obtained utilizing the back pointers stored in β and *PeakIdxs*. In case of a two frame gap, a pair of intermediate peaks (one from each frame) which exhibit minimum average distortion in their locations are picked. Then the contour is extended by (i-2, intermediate-peak-1, 0), (i-1, intermediate-peak-2, 0) followed by (i, $PeakIdxs_{i,\rho}$, 0) and the algorithm continues with examining conditions at the next frame after resetting *NewStart*.

If the conditions for a possible new start were met, the algorithm checks if the last recorded contour was a false start or if a wrong branch was followed at an earlier choice point (if any) in the contour. Determining and rectifying these two types of errors are described below. If any of the checks return true, appropriate steps are taken to rectify the corresponding error, and then the contour is extended by (i, $PeakIdxs_{i,\rho}$, 0). Otherwise, the contour is extended by (i, $PeakIdxs_{i,\rho}$, 1) and *NewStart* is reset. This marks the end of examining conditions at a single frame in the second pass. Repeating the above process for each frame completes the second pass with raw estimates of the contours present in the audio segment.

A false start at an earlier point can be determined as follows. If the last recorded contour ended at $(i\text{-}1)^{th}$ frame, at this point it is certain that the end of that contour is not within FREQ_WIN distance of $PeakIdxs_{i,\rho}$ since we have already checked for this condition. The error can be rectified by looking backwards one previous frame at a time replacing the second element of the $(endIdx\text{-}n)^{th}$ tuple in fx with the location of that peak in the $(i\text{-}n)^{th}$ frame which lies in the backward extension of the new contour; for n = 1, 2, 3,… up to the end of the backward extension. A backward extension is the portion of the candidate contour before the i^{th} frame that can be traced with the help of β. See Fig. 5a for an example. The replacement of the values at the n^{th} backward frame is made only when the value at the corresponding peak in the backward extension is greater than that at $fx_{(endIdx\ -\ n),1}$ and also greater than PRR% of *MaxPeaksMean*. The peak values are obtained from $Peaks_{i,k}$. Whenever the condition fails, the backward loop is broken and the third element of the tuple in fx, corresponding to the last replaced peak is set to 1 to indicate the start of the new contour there. This way we get rid of the falsely recognized contours caused by the presence of transient noise around the actual starts of true contours.

An incorrect branch taken at an earlier fork can be rectified as follows. If the new contour has a backward extension, it is traced backwards one frame at a time while replacing the second element of the $(endIdx\text{-}n)^{th}$ tuple in fx, as described before, until the fork is reached which can be realized when the point at the $(endIdx\text{-}n)^{th}$ tuple in fx equals that peak in the $(i\text{-}n)^{th}$ frame which lies in the contour's backward extension. See Fig. 5b for an example. The strategy employed here is that the longer candidate contour wins. Hence, the decision on replacement is not based on comparison of the heights of the peaks in consideration. A replacement is made when the value at the corresponding peak in the backward extension is greater than PRR_S2% of *MaxPeaksMean*. PRR_S2 is chosen here in order to allow slightly smaller peaks also to play a role by aiding in the tracing of a longer contour. Again, whenever the condition fails, the backward loop is broken but this time without setting the flag in the last element of the corresponding tuple in fx since we do not have a new start here.

E. Contour Smoothing

After the end of the two pass dynamic programming section, we obtain a set of raw disjoint contours whose coordinates are available in the first two elements of the tuples in fx. The quadratic interpolation used in refining the locations of peaks provides estimates of the actual peak locations. The locations of peaks obtained this way may be slightly off from their true locations. Hence, the obtained peaks in consecutive frames which are considered to form the identified contour may be scattered around their corresponding true locations which would have otherwise defined the true

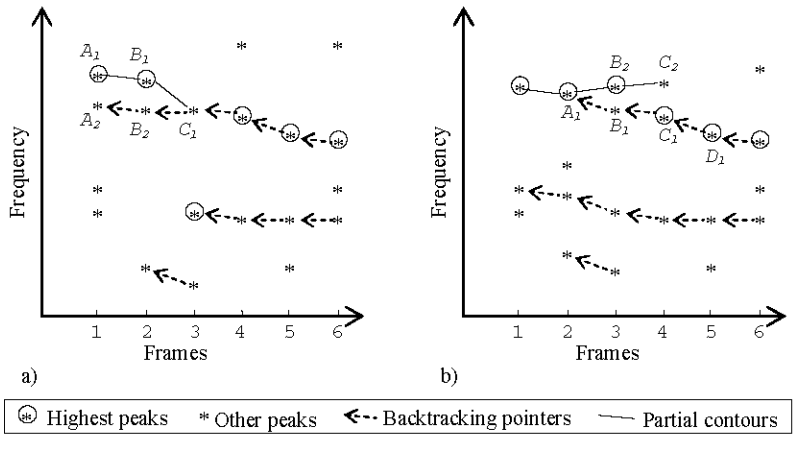


Fig 5. a) Example of a false start: A_1 and B_1 are initially selected as they are the strongest peaks in the two frames. At frame 3, C_1 is selected as it is within FREQ_WIN Hz of B_1. However backpointers from frame C_1 indicate that B_2 and A_2 would produce the better estimate of the contour. b) Example of an incorrect branch at frame 2: At fork A_1, B_2 is selected as it is stronger and closer. However at frame 5 when the original path ends, D_1 is examined and we realize that a branch from A_1 to B_1 would produce a longer contour.

978-1-4244-2126-8/08/$25.00 ©2008 IEEE

contour. As a result of this scattering, the identified contour is prone to jitter. Smoothing of this contour reduces jitter and helps us realize a much closer estimate of the true contour in the call. Every identified contour's ordinate sequence (available in the second element of the tuples in *fx*) is passed through a 3-point moving average filter to obtain jitter-free equivalents.

F. Post Processing

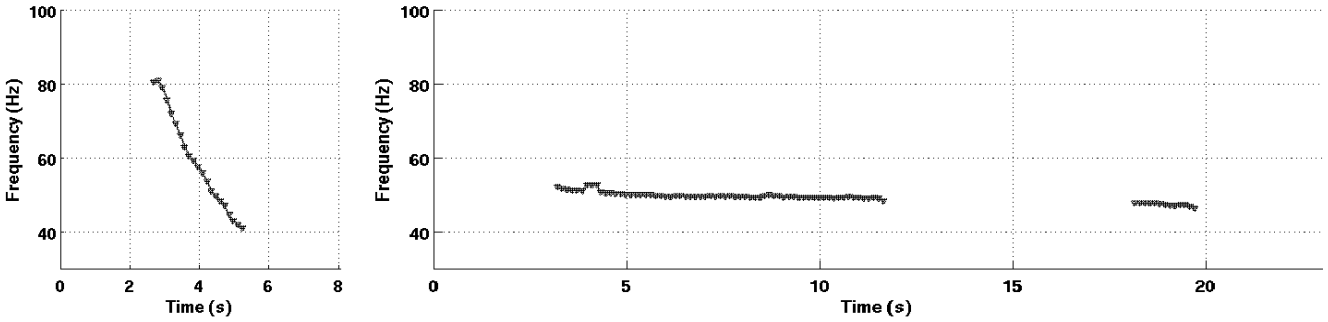
This is an optional step. If a pair of consecutive contours have a time separation less than PP_GAP (i.e., the end of the first contour and the start of the next contour are less than PP_GAP apart) and the corresponding interior ends are within FREQ_WIN distance of each other, then they may be joined together by linear interpolation of the points in the gap in order to realize, as a single long contour, the complete call with broken sub-contours. This is particularly helpful with the rather long B calls. Typically the contours of these calls appear to be segmented for several reasons, as described earlier. The technique, however, has drawbacks where it may end up joining together the contours of two different calls. In any case, this step involves the expansion of *fx* from the middle in order to accommodate the new points. Once the additional peaks are accommodated, the value in the third element of the tuple in *fx*, corresponding to the start of the second contour considered is set to zero to indicate that it is now not the start of a contour.

V. Results and Performance

When evaluated on a randomly chosen segment of the recordings, the algorithm has a recall of 90%, detecting the contours of 56 calls out of the 62 known calls in a 60 minute sample. The precision was 76%, with 18 false positives in sections of data with fairly strong channel noise. The algorithm was also observed identifying contours of other tonal call types, including calls from other species. Some of the false positives were caused in part by the presence of fin whale calls. Since the algorithm is optimized to function in the [30Hz, 100Hz] bandwidth, the downswept fin whale calls that are produced in the lower boundary of this bandwidth affect the performance to some extent. The task of rejecting such detections is considered that of a classifier, which is beyond the scope of this work.

Fig. 6 shows the extracted contours for the D and B calls shown in Fig. 1. The dots in the contours indicate the frequencies at the corresponding frames that form the contour. Consecutive dots are joined by straight lines. Post-processing was turned on while extracting these contours. However, since the two segments in Fig. 6b are too far apart, we can see that they have not been joined by linear interpolation.

The algorithm is implemented in Matlab and runs at a real-time factor of over 40x on an AMD Athlon 1.19GHz machine, making it suitable for use as a component in an on-site acoustic monitoring program. Future work will focus on the development of a classifier.



a) D call b) B call

Fig 6. Call contours extracted using the proposed algorithm, for the audio used in producing the spectrograms shown in Fig. 1.

Acknowledgments

The authors would like to thank Mark Lammers for examining the viability of using the algorithm of [7] for the contour extraction. Funding for data collection was provided by Frank Stone and Ernie Young from CNO-N45, Robert Holst from the Navy's Strategic Environmental Research and Development Program and Bob Gisiner and Ellen Livingston of ONR. Data was collected with the help of the SIO Whale Acoustic Lab, in particular, Sean Wiggins, Allan Sauter, Chris Garsha, and Graydon Armsworthy.

References

[1] E. M. Oleson, "Calling behavior of blue and fin whales off California," San Diego: University of California, 2005.

[2] A. Širović, "Blue and fin whale acoustics and ecology off Antarctic Peninsula," San Diego: University of California, 2006.

[3] E. M. Oleson, S. M. Wiggins, and J. A. Hildebrand, "Temporal separation of blue whale call types on a southern California feeding ground," *Animal Behaviour*, vol. 74, pp. 881-894, 2007.

978-1-4244-2126-8/08/$25.00 ©2008 IEEE

[4] D. K. Mellinger and C. W. Clark, "Recognizing transient low-frequency whale sounds by spectrogram correlation," *The Journal of the Acoustical Society of America*, vol. 107, pp. 3518-3529, 2000.
[5] C. L. Berchok, D. L. Bradley, and T. B. Gabrielson, "St. Lawrence blue whale vocalizations revisited: Characterization of calls detected from 1998 to 2001," *The Journal of the Acoustical Society of America*, vol. 120, pp. 2340-2354, 2006.
[6] D. K. Mellinger and C. W. Clark, "Methods for automatic detection of mysticete sounds," *Marine and Freshwater Behaviour and Physiology*, vol. 29, pp. 163-181, 1997.
[7] M. O. Lammers, W. W. L. Au, and D. L. Herzing, "The broadband social acoustic signaling behavior of spinner and spotted dolphins," *The Journal of the Acoustical Society of America*, vol. 114, pp. 1629-1639, 2003.
[8] X. C. Halkias and D. P. W. Ellis, "Call detection and extraction using Bayesian inference," *Applied Acoustics*, vol. 67, pp. 1164-1174, 2006.
[9] D. Talkin, "A Robust Algorithm for Pitch Tracking (RAPT)," in *Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds. Amsterdam, The Netherlands: Elsevier, 1995, pp. 495–518.
[10] S. Wiggins, "Autonomous acoustic recording packages (ARPs) for long-term monitoring of whale sounds," *Marine Technology Society Journal*, vol. 37, pp. 13-22, 2003.

978-1-4244-2126-8/08/$25.00 ©2008 IEEE