Analysis of Cetacean Data and Algorithm Development

Ben Bougher
Joe Hood
Ewen MacKillop
Akoostix, Inc.

Prepared by:
Akoostix, Inc.
10 Akerley Blvd - Suite 12
Dartmouth, NS B3B 1J4

Contractor's Document Number: AI CR 2012-001
Contract Project Manager: Joe Hood, (902) 404-7464

PWGSC Contract Number: W7707-4500859801
Technical Authority: James A Theriault, Defence Scientist
Contractor's Publication Date: February 2012

Contract Report
DRDC-RDDC-2017-C115
May 2017

Principal Author

Original signed by Ben Bougher

Ben Bougher

Research Analyst

Approved by

Original signed by Robert A. Stuart

Robert A. Stuart

Head, Technology Demonstration

Approved for release by

Original signed by Calvin V. Hyatt

Calvin V. Hyatt

Chairman, Document Review Panel

Abstract

One of the many challenges faced by cetacean data analysts is the large volume of data they must review in order produce concise results. The objective of this project was to improve on algorithms that support automation of this process. Algorithms intended to produce a list of vocalization events from click-detection data, and to produce an estimate of the number of vocalizing animals based on different time difference of arrival for a vocalization sequence were investigated. During the course of this work, methods for reducing the number of false detections due to multipath and general methods for reducing the number of false detections were also investigated. The algorithm that produces a list of vocalization events examines detection density, attempting to group associated clicks, while discarding random detection events. This is useful because it provides the analyst with a reduced number of events to validate during further analysis, which is beneficial for data where detection is infrequent. The most promising algorithm for counting the number of vocalizing animals results in an image, where animals are represented by lines in the image (traces) and the shape of the line is representative of motion. The method requires frequent vocalization (i.e. a click train) and at least two sensors spaced sufficiently for signal time difference (STD) estimation. The analyst can quickly examine the image and isolate individuals, though multipath can result in overestimation of the animal count. The algorithms developed under this contract were found to be useful and will directly support ongoing development of a practical, efficient, and robust automated cetacean data analysis capability.

Executive summary

Analysis of Cetacean Data and Algorithm Development:

Ben Bougher; Joe Hood; Ewen MacKillop; February 2012.

Introduction or background: This project is part of an ongoing initiative to grow and improve DRDC's cetacean acoustic data analysis capability. The long-term objective is to create software that can be used for efficient, robust, and practical analysis of these data during operations and through post analysis. Software is being developed in the Software Tools for Analysis and Research (STAR), through a collaborative effort including the Department of Fisheries and Oceans (DFO). In this case students and employees of DFO were engaged to support the work.

This project focused on methods for reducing a large volume of data into a concise list of important events and then determining the number of vocalizing animals in each event.

Results: The project was successful, delivering significant improvements to DRDC's cetacean detection and density estimation capability. Improvements include:

- An algorithm that reduces cetacean click detections into dive event summaries. These allow an analyst to efficiently identify areas of interest without having to manually analyze all data or each of the large number of individual detections, which are produced. The method is not useful in cases where much of the dataset contains vocalization.
- An algorithm that results in a correlogram, where animals are represented by lines in the image (traces) and the shape of the line is representative of motion. The method requires frequent vocalization (i.e. a click train) and at least two sensors spaced sufficiently for signal time difference (STD) estimation. The analyst can quickly examine the image and isolate individuals, though multipath can result in overestimation of the animal count.
- An algorithm to discriminate multipath detection from direct path arrivals for click data. Once identified, they are removed from detection lists to improve the quality of the image produced from STD data. This avoids false traces, which result in an overestimation of the number of vocalizing animals.

Significance: The new algorithms can be used by analysts to reduce the required time that it takes to analyze a dataset, while providing additional information on target motion. This efficiency improvement leads to more effective habitat characterization and impact mitigation protocals.

Future plans: Though significant progress was made during this contract, a long list of suggested improvements remains and will continue to be addressed as budget and schedule permit. Additional work is planned to: progress algorithm prototypes into user-friendly software, improve processing efficiency, extend these algorithms to provide more useful information, and improve the efficacy of these algorithms for particular cases to ensure that methods are robust.

Table of contents

List of figures

List of tables

Acknowledgements

Akoostix acknowledges and appreciates the efforts of the staff of the Canadian Forces Maritime Experimental and Test Ranges (CFMETR), Department of Fisheries and Oceans (DFO), and the University of St. Andrews for providing data and answering questions about these data to support analysis and algorithm development. We also acknowledge DFO's contribution of Hilary Moors to support this collaboration. Hilary was very beneficial in helping our team determine where improvements would be beneficial, selecting appropriate data sets, and finding complementary information to understand these data.

1 Introduction

This contractor report documents work performed under contract W7707-4500859801 for Project Authority (PA), James Theriault. The work was conducted between July 2011 and February 2012.

This work is a continuation of ongoing research into efficient and effective methods for monitoring cetacean activity for the purposes of improving our knowledge of their abundance and distribution, along with knowledge of their location with respect to anthropogenic activity (e.g. active sonar operations). These efforts started in 2005, when an initial acoustic cetacean detection capability was developed using Defence Research and Development Canada (DRDC) – Atlantic's Sentinel detector [1][2]. This detector was integrated into the Slocum Glider [3] and a desktop processing and analysis application, named ACDC (acoustic cetacean detection capability) [4]. The detector continues to be studied and improved using real data collected during various experimental trials [5][6][7][8]. These efforts are complemented by research into localization methods [9][10][11] and classification [16].

This contract focused on two objectives:

- Counting the number of animals present during vocalization events (see Section 2.1), and
- Improving click detection and analysis in order to reduce false alarms and distil detection data down to dive events (see Section 2.2)

Significant progress was achieved for both of these objectives. Previous concepts were validated and improved upon, while new algorithms were recommended and validated. This research resulted in a scientific presentation given at the *5th Detection, Classification, Localization and Density Estimation (DCLDE) Workshop* in Portland, Oregon during August 2011 [17]. Data used for this research were taken from acoustic recordings of known sources [18]. A follow on presentation and paper was drafted for the *European Conference on Underwater Acoustics (ECUA)*. These papers relate to animal density estimation.

This report provides a detailed description of the work performed for each of the tasks listed in the Statement of Requirements (SOR). The report is organized as follows:

- Section 2 describes the technical and scientific work,
- Section 3 documents recommendations for future work,
- Section 4 summarizes the results of the project with conclusions and a discussion on planned effort,
- Annex A provides configuration management (CM) information to help users understand which version of the software was used for this work, and
- Annex B provides a brief description of the software used to support this project.

2 Data analysis and algorithm development

This section describes the technical and scientific work performed under this contract. Refer to the related publications and presentations for more detail on the scientific aspects of the work [17][18]. Additional detail related to the algorithms and specific processing settings can be found in the STAR trials09 analysis subversion (SVN) repository. Analyses are organized in a directory structure consistent with the STAR analysis process [19]. The following directories were used for this work:

- *cfmetr_10-1/transient_analysis*: a data set collected at the Canadian Forces Maritime and Experimental Test Ranges (CFMETR) in February of 2010, where a known source was used to replay various vocalization-like signals that were recorded on multiple sonobuoys fitted with GPS. These data were used for the work described in Section 2.1.2.
- *cfmetr_11-1/transient_analysis_3MAR*: a data set collected at CFMETR in March of 2011, where two known sources were used to replay various vocalization-like signals that were recorded on multiple sonobuoys fitted with GPS. These data were used for the work described in Section 2.1.2. Scripts used for the simulated data generation and analysis, described in Section 2.1.1, are also in this directory.
- *trials/q302/dive_event_anal*: a data set collected at the Atlantic Undersea Test and Evaluation Center (AUTEC) in the Bahamas in 2007. It contains acoustic recordings from sonobuoys and a glider. The glider deployment data were used for the dive event analysis described in Section 2.2.3. All scripts used for detection and dive event clustering are in the associated *scripts* and *idlprog* directories.
- *q320/delphinus_30_31*: a data set taken from Delphinus towed array data collected during Canadian Forces Auxiliary Vessel (CFAV) Quest in 2009 at AUTEC. Analysis focused on regions of the data containing beaked whale click trains, which was used for the work described in Section 2.1.3.1.
- *NBW_gully*: a data set collected during a line transect survey, conducted near the shelf edge east of Nova Scotia, using a four-sensor towed array. These data were used for the analysis described in Section 2.1.3.2 and Section 2.2.1.

Details relating to specific algorithm functions are documented in the source code supplied with the STAR software release (see Annex A).

2.1 Density estimation

The primary objective for the density estimation work was to find a method for reducing the amount of effort required for analysts to determine the number of animals vocalizing during detection events. This work made use of algorithms developed during detection analysis tasks, which distilled large numbers of detection events down to dive events (see Section 2.2). The dive events were then analyzed to estimate the number of animals.

The approach and algorithms used for this work evolved throughout the project. This evolution is described herein as a chronological account. Each of the major iterations is described below while further detail and examples are provided in the following subsections.

This research is a continuation of previous contract efforts [10]. At the end of the previous contract an algorithm that was capable of detecting clicks and determining the signal time difference (STD) of a click train between two sensors had been developed [11]. In this form, the algorithm is limited to STD for a single vocalizing animal. This algorithm was adapted to detect multiple animals as follows:

- Clicks are detected using a split-window likelihood detector on energy time series data, generated using a Teager-Kaiser (TK) transformation [12].
- The TK transformation was replaced with an improved detector which uses band-limited summed spectral energy to form a detection function and a gapped split-window likelihood test for detection.
- A synthetic time series is created where a square pulse is inserted at each detection time. One time series is created for each sensor.
- The width of the square pulse is selected to account for uncertainty in the estimated detection time due to sound propagation and data processing.
- A cross-correlation is performed between sensors for multiple overlapping kernels. The resulting correlogram indicates correlation level at each delay and thus, relates to the likelihood that the same click train with the corresponding STD is present on both sensors.
- The number of clicking animals is generally estimated by counting the number of peaks in the correlation function, though the approach for identifying valid peaks evolved during this research.

In preparation for the current work, DRDC created data sets during the CFMETR 11-1 trial. The acoustic data contains “play-back” of simulated click trains from two known source locations, recorded on a field of seven DIFAR sonobuoys. These data were used to extend and test the previously-developed algorithm for multiple animals. Concurrent with CFMETR 11-1, DRDC developed click train simulations to examine the potential of this approach under varying conditions. These results were provided as input to this contract task.

The first stage of development under this contract was to improve implementation of the click-train simulations, allowing for multiple animals and sensors at fixed locations. The simulation output was connected to the STD estimation algorithm to examine options for extending the approach to multiple animals. During this effort the team researched scientific papers to gather information on variability of inter-click interval, which we learned is an important parameter in the simulation [20]. Initially an automated detector was used to find peaks in the correlation function of each kernel, and a histogram was generated to determine to most probable number of animals. These results were inconclusive, and the algorithm was especially unreliable where animals would start and/or stop vocalizing during the analysis period. An alternate approach was developed where individual correlation functions were converted to scan lines and formed into an image so that consistent delays would form traces that could be visually identified. Each independent trace in the image can be counted to estimate the number of vocalizing animals. Further detail, explaining the findings and algorithm evolution, is provided in Section 2.1.1.

The next stage of development involved testing the STD correlogram approach using the "play-back" data from the CFMETR 10-1 (single source) and CFMETR 11-1 (two sources) trials. These tests were positive, but reinforced the potential implication of multipath on density estimation, which can cause an overestimation of the number of vocalizing animals. These data showed several multipath arrivals for one source, which were parallel to the expected trace. This work is summarized in Section 2.1.2.

The final research iteration used real cetacean data collected in the Caribbean and east of Nova Scotia. The Caribbean data sets were recorded using the Delphinus array, whose sensor spacing proved too close for this approach. The data from Nova Scotia were recorded using an array with two groups of sensors spaced approximately 200 m apart. A number of data segments were analyzed showing clear traces in STD correlograms. Further investigation revealed that some of these traces were due to multipath, though correlation with the correct trace was not apparent due to differences in trace shape. The different shapes can be explained by variable source (animal) depth. Additional algorithm enhancements were performed to mitigate the impact of multipath, providing positive results, which are documented in Section 2.1.3.

2.1.1 Testing and algorithm development using simulated data

The first step during this iteration was to improve the simulator used to generate synthetic click time-series data. The output of the simulation is in the same format as for synthetic time series produced from detection data in the original algorithm. The time series is correlated to estimate STD. Ideally, a distinct peak is generated in the correlation function for each source. The simulation code can be found in the *data_simulator* directory for the STAR-IDL source code, and the configuration scripts can be found in the *trials09*-repository directory *cfmetr_11-1/transient_analysis_3MAR/idlprog/*. The *read_me.txt* file in the *idlprog* directory provides instructions for generating and testing data.

The simulation improvements consisted of re-factoring to break up algorithm responsibilities into software modules and allow for separate definition of simulation parameters. These changes added flexibility in the design, while allowing for creation of multiple simulations using the same software modules. These changes provided additional benefits:

- An arbitrary number of sources can now be defined using STAR-IDL source software objects;
- Each source can produce a unique click train configured by defining the start time, duration, inter-click interval (ICI), and ICI standard deviation;
- Receivers are defined using STAR-IDL receiver software objects; and
- Expected STD calculations are computed using source-receiver geometry and STAR-IDL functions.

Individual scenarios were defined including source and receiver positions, ICI, ICI standard deviation and time of arrival estimation standard deviation for multiple animals. ICI variability is required to provide a unique pattern that can be matched between sensor time series in order to find the correct STD. Increased variability improves algorithm performance, as it reduces the likelihood of aliasing caused when a kernel matches or almost matches different parts of a click train. Variability in estimated time of arrival is caused by environmental conditions (e.g. curved acoustic propagation path) and processing-estimation errors. These errors degrade algorithm performance.

A review of the literature led to the assumption that there is substantial variability in ICI over short time periods [13][14][15], which was reinforced during analysis of real data (see Section 2.1.3). The results of tests on real data also reinforced the assumption that errors related to estimated time of arrival are not significant enough to substantially degrade algorithm performance, when the sensors have sufficient spatial separation. In general, STD correlogram-based localization algorithms are more sensitive to time of arrival estimation errors as sensor spacing is reduced.

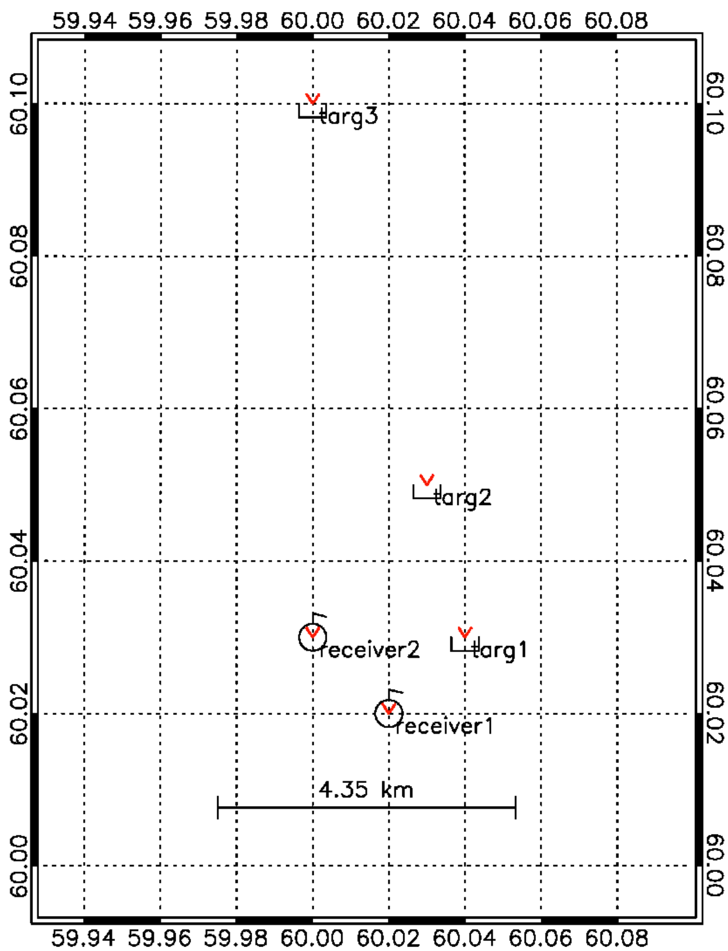

Figure 1: Tactical plot showing the sensor and mammal configuration of the simulation. Sources are indicated by the lower half of a square (unknown submarine symbol) and receivers are indicated by a circle with a line out of the top (sonobuoy symbol).

A simulation was created using three sources (simulated cetacean) and two receivers, with locations shown in Figure 1. For this case, the start and stop time for each vocalizing source was varied to examine this impact on estimation of the number of vocalizing cetaceans. One animal vocalizes for the entire period, another starts 30 seconds into the simulation and stops 100 seconds later, and the third animal starts after 100 seconds and stops 150 seconds later. Each of the three animals was assigned an ICI of 0.4 seconds and ICI standard deviation of 0.05 seconds. Detection performance was assumed perfect. The derived synthetic time series were processed by cross-correlation using a 10-second kernel with 50% overlap. The cross-correlation for all kernels where further analyzed to estimate the number of vocalizing animals.

For the first method, peak detection was performed on the output for each kernel. The number of detected peaks was counted and the peak counts were transformed into the histogram shown in Figure 2. The peak in the histogram suggests that two animals where vocalizing, though the true value is three. This is in part due to the fact that all three animals were only vocalizing simultaneously for 30 seconds of the 5-minnute simulation. If this method produced a perfect result, three cross-correlation peaks would have been detected 12 times, while only five occurrences were found. It is likely that the remaining three-cetacean cases resulted in too many detected peaks, as the cross-correlations become noisy during this period. Note that up to 27 peaks were detected in some cases.

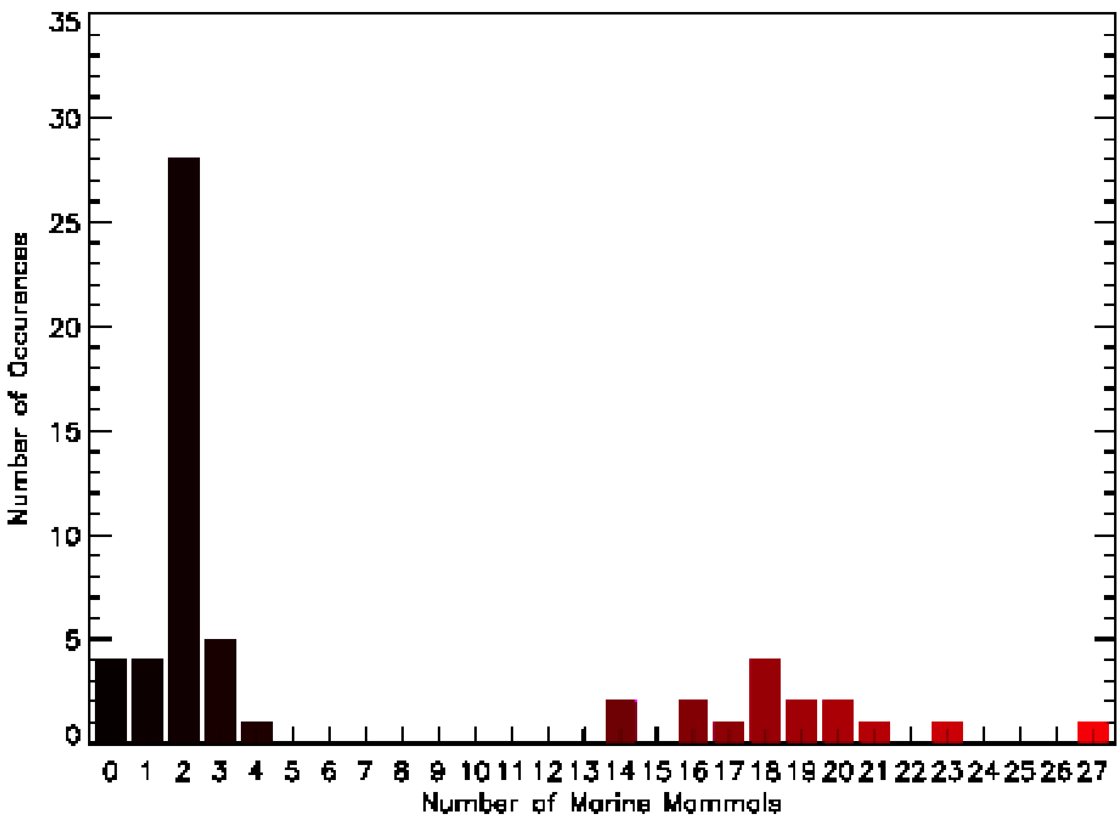

Figure 2: Histogram showing the number of detected cross-correlation peaks for all kernels. The true number of simulated animals is three for this simulation, but the histogram suggests that two animals are likely.

The histogram method was discounted and alternate methods of estimating the number of vocalizing animals was considered. A new approach was conceived where the output of each cross-correlation was converted to greyscale and all output was stacked to create an image similar to a sonogram. The output for this method is shown in Figure 3. The three animals and when the vocalizations start and stop are apparent. It is also clear that the noise level increases with the addition of new vocalizing animals, which contributed to errors for the histogram-based method. The noise is due to the increased potential for partial correlation between synthetic time series, as more clicks are added. Fortunately, the noise remains random and the correct traces, which appear at a consistent delay, are still visible.

The team decided to use the STD correlogram for the remaining analysis. Ideally, appropriate computer algorithms for trace detection could be used or developed to further automate the process.

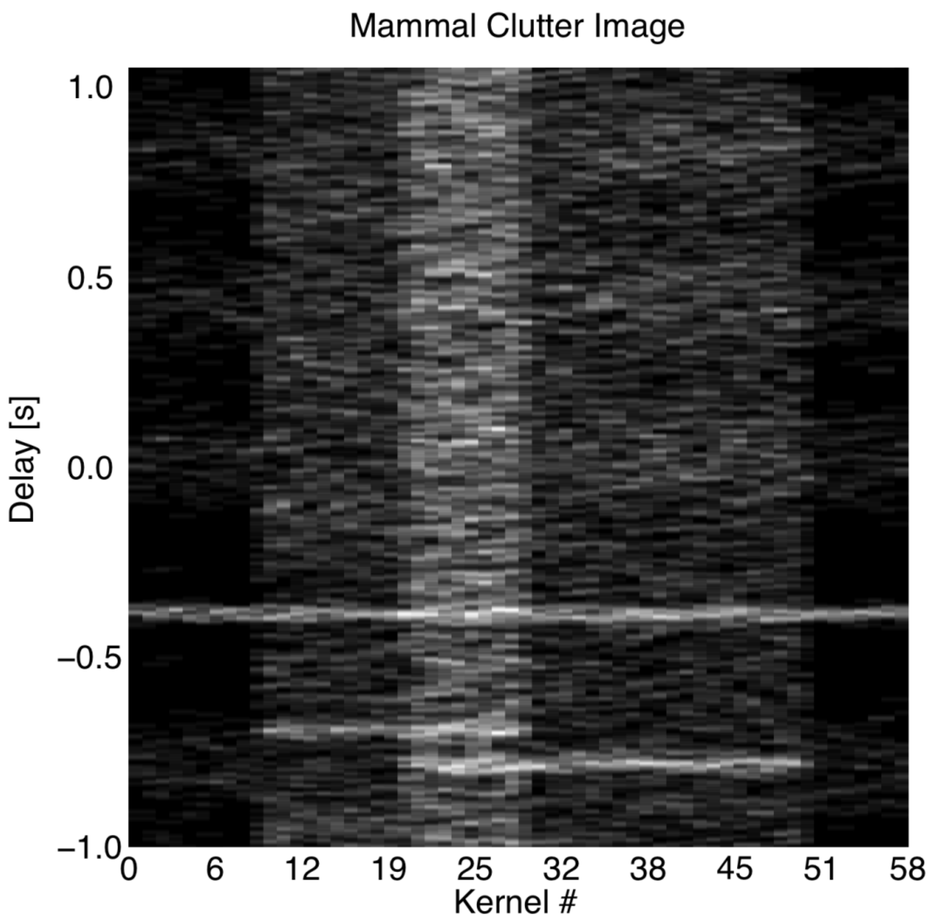

Figure 3: Marine-mammal STD correlogram showing sequential correlation results. Each of the three horizontal lines corresponds to a simulated clicking cetacean. The start and stop of each vocalization is clearly visible on the timeline (Kernel #). Each kernel is 10 seconds long with 5 seconds of overlap.

2.1.2 Testing on “play-back” data recordings

The next research iteration used acoustic data generated from a controlled source. Simulated clicks were transmitted into the ocean from known source locations and recorded on DIFAR sonobuoys, located using GPS units on each buoy. The clicks are pulse-compressed frequency-modulated (FM) waveforms. These were chosen because the processed time series is similar to a click and the FM pulse is more easily produced using available transducers. For these experiments it is important that click duration and ICI variability is similar to that of a cetacean. Click frequency and ambient noise characteristics are much different than for cetacean click recordings, but the algorithm is not sensitive to these inconsistencies. Ambient noise characteristics are different because the data is pulse-compressed (match filtered) during a pre-processing stage. This filters the ambient noise as well. Regardless, the in-water simulation approach produces reasonable data for testing the algorithm.

Detection times were generated using the simple TK detection algorithm and then the detections were processed using the image generation approach discussed in Section 2.1.1. These data produced similar results as for the simulated case, though an additional feature, multipath arrivals, caused additional traces to form in the STD correlogram.

The positions of the sources and receivers for the selected test data are shown in Figure 4. In this case the *Sikanni_src* was moored and transmitted signals with a mean ICI of 1.01 seconds and an ICI standard deviation of 0.05 seconds. The *680_src* was drifting to the NW and transmitted signals with a mean ICI of 1.0 second and an ICI standard deviation of 0.05 seconds.

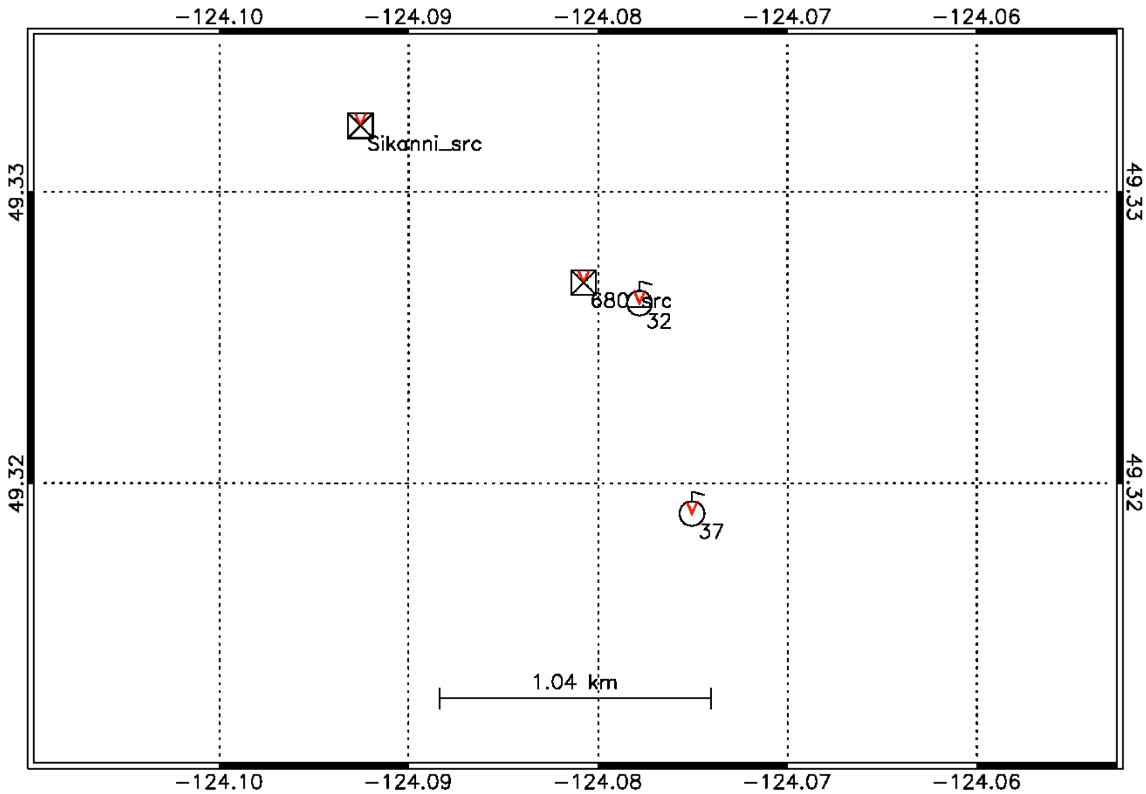

Figure 4: Tactical plot showing the source and receiver geometry used for testing the algorithm with CFMETR 11-1 data. Sources are indicated by a square with an internal X (source symbol) and receivers are indicated by a circle with a line out of the top (sonobuoy symbol).

The results of processing these data with a synthetic click width of 1.0 ms and 10.0 ms are shown in Figure 5 and Figure 6 respectively. Multipath effects are apparent for both of the sources when the smaller click width is used. The source-receiver motion is also apparent because the STD varies throughout the sample period. This slow change is indicated by the slope of the traces. The weaker third trace is likely caused by reflection of the signals transmitted from the *Sikanni_src* off of a bottom feature, while the diamond pattern in the background noise of Figure 6 is likely due to the beating[1] of the two ICI, which has a period of 100 seconds (20 kernels).

[1] The amplitude of two mixed signals will vary at a frequency that is the difference of the two signal frequencies. This is known as beating and the rate is called the beat frequency.

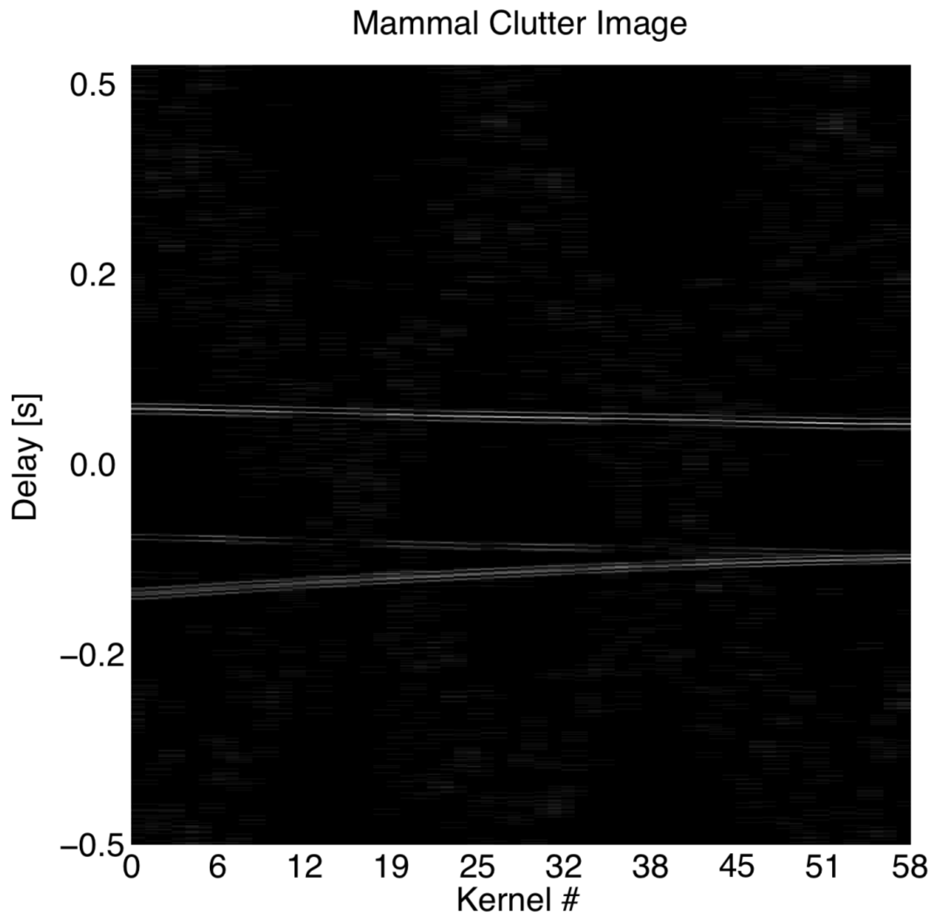

Figure 5: STD correlogram generated using a synthetic click width of 1.0 ms. Multipath signals are indicated by weaker lines close to the primary trace.

Visual analysis of either figure would result in a conclusion that either two or three sources are transmitting. An experienced analysis, with enough observation time, is likely to determine that the third trace is related to the trace near 0.1 s delay. An analyst could also inspect the raw data using a time-series graph or spectrogram and perform a manual STD analysis for a short segment of data to conclude that only two sources are present. The result of this testing was therefore positive; an analyst could efficiently count the number of transmitting sources using the proposed STD correlogram.

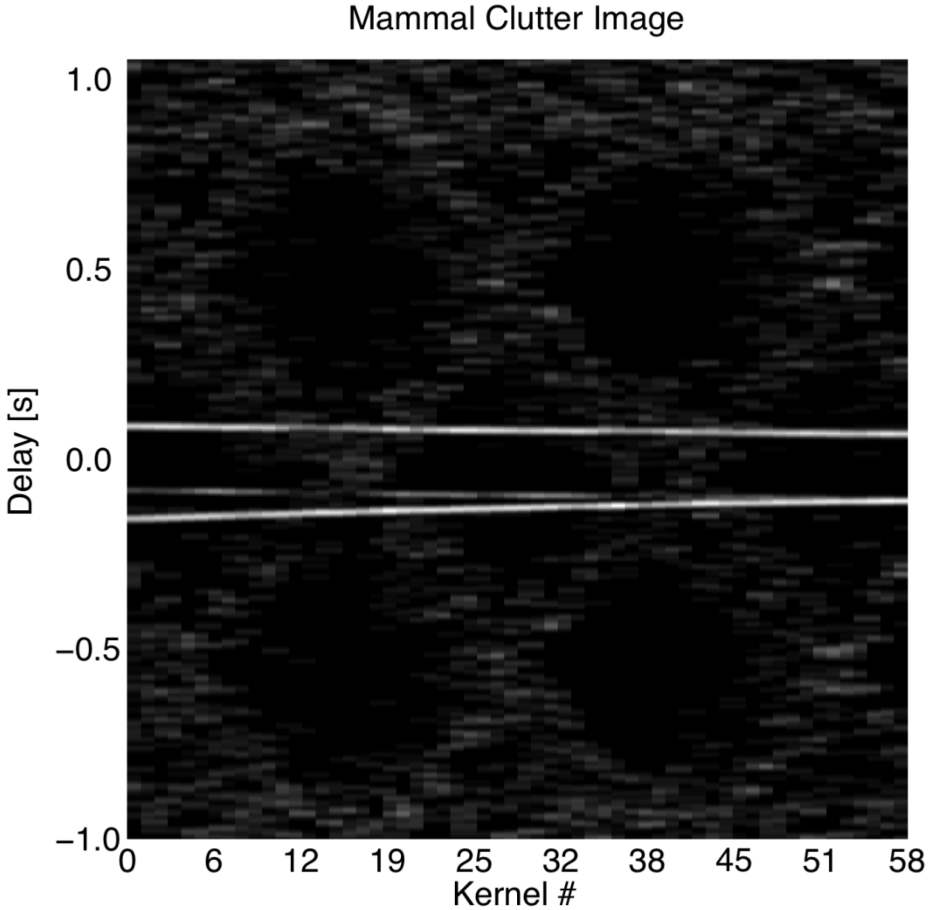

Figure 6: STD correlogram generated using a synthetic click width of 10.0 ms. Multipath arrivals are blended into a single peak. The diamond pattern in the background noise is likely due to the beating of the two ICI, which has a period of 100 seconds (20 kernels).

2.1.3 Testing on real data

The next research iteration involved attempting to generate STD correlograms for real data. First, data from the Delphinus array was processed. However, the sensor spacing was not adequate for this approach. In response to this, data collected using an array owned by the University of St. Andrews was selected. These data were supplied via DFO with permission from the university. In this case, the array is composed of two pairs of hydrophones with the pairs spaced 200 m apart, while the hydrophones in each pair are 25 cm apart. For this data, the array was towed with 400 m of cable scope. This resulted in one hydrophone pair 200 m behind the vessel and another 400 m out.

2.1.3.1 Delphinus array data processing

The density estimation algorithm was applied to a subset of the Delphinus towed-array data set from CFAV Quest cruise Q320. Data from the three HF sensors were correlated to make density-estimation graphs. The sensors were found to be too close together (max 3.7 meters) to provide the resolution required for the approach to be effective. In this case an approach using coherent correlation on the raw time series may provide a useful result.

2.1.3.2 University of St. Andrews array data processing

The St. Andrews array contains sensors with up to 200 m spacing, which allows for about 0.13 seconds of delay. These data recordings included bottlenose whale click trains and corresponding visual detection data, making this an ideal data set for algorithm evaluation. These data sets were processed using the dive event detection approach described in Section 2.2. Dive events exceeding 1000 s and occurring during visual sightings were selected for processing.

Initial attempts at processing the data using the TK-based detector produced poor results (not shown). This was attributed to the high sampling rate (high bandwidth) and relatively weak signals. The detector was replaced with an improved band-limited detector that uses the processing depicted in Figure 7 to improve detection performance. This method provided better detection data for the algorithm. Detection bands for sperm whale, beaked whale, and dolphin clicks were configured. Specific bands and resolutions are contained in the *scripts* directory and *NAD/target_files/* directories for the NBW gully trial. In addition to using the improved detector, a sliding median normalizer was applied to the STD correlogram to enhance the traces.

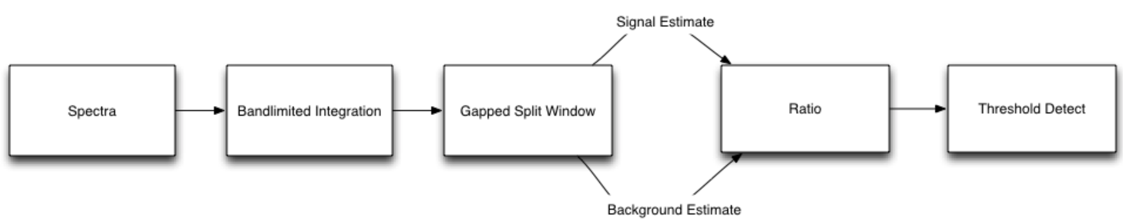

Figure 7: Processing block diagram of detector used for St Andrews array data. The signal estimate is divided by the background estimate to produce the detection function, which is tested against the threshold to determine detection.

Processing for the dive events occurring between 20:40:10Z and 22:40:10Z are shown in Figure 8. A large number of traces are visible, with eight simultaneous traces occurring in the region of kernels 350 through 400. None of the traces are parallel, so analysts initially assumed that the traces corresponded to eight vocalizing animals.

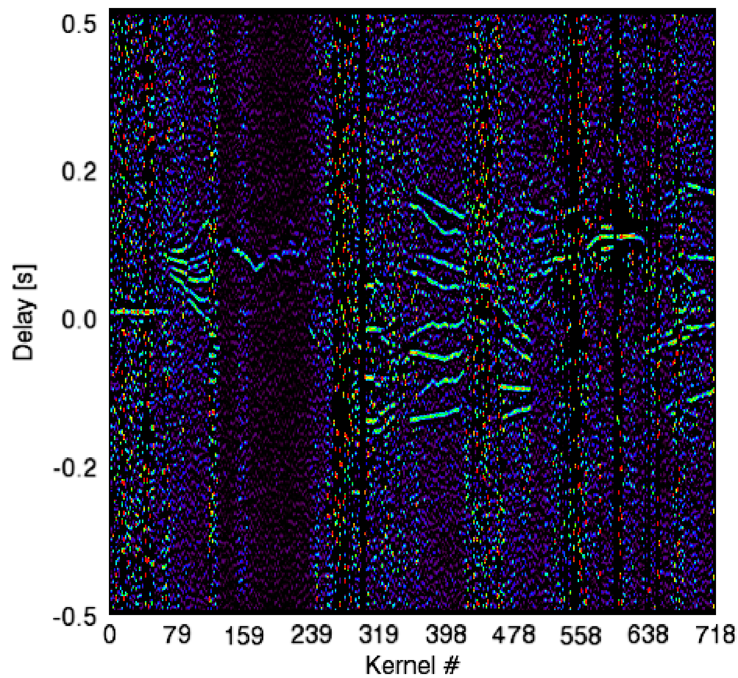

Figure 8: STD correlogram displaying data for the St. Andrews array from 20:40:10Z to 22:40:10Z. 20-second kernels with 50% overlap were used for cross-correlation processing.

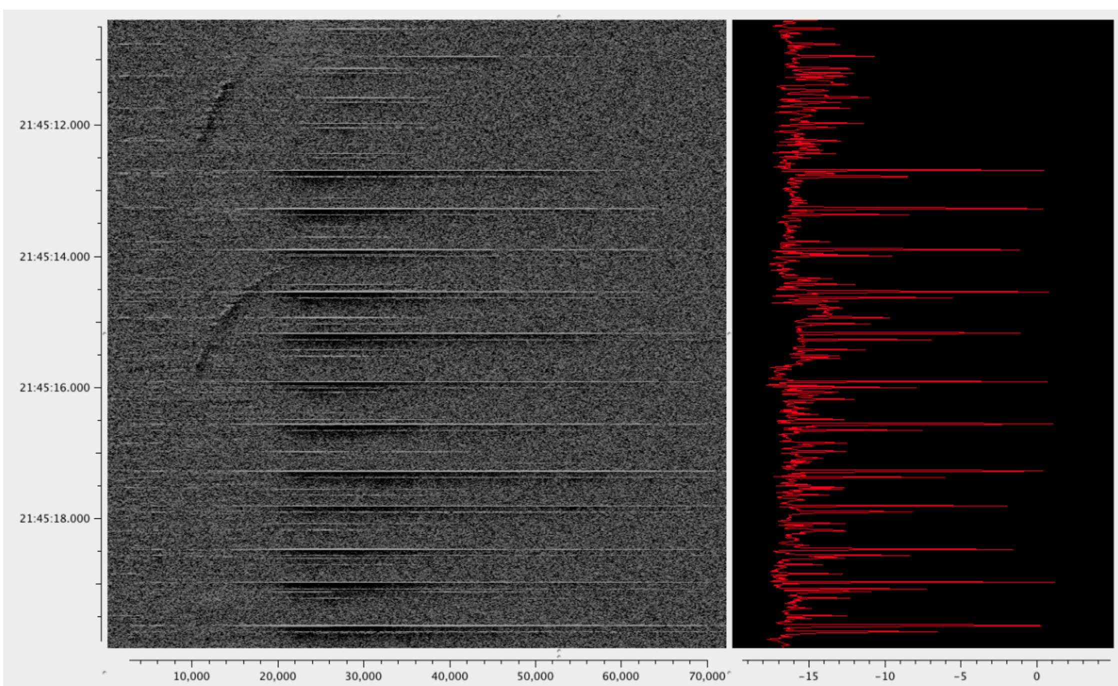

Figure 9: Sonogram and ETI for mammal clicks on St. Andrews recorder 1 that correspond to a data segment when eight traces are detected near the center of Figure 8.

Further analysis revealed that the data were much more complex, as shown in the sonograms in Figure 9 and Figure 10. In this case there are distinct multipath arrivals at different delays for each sensor. This is likely due to the different depth of the fore and aft sensor pair (sensor depth data is not available for detailed analysis).

Manual STD analysis between the data shown in Figure 9 and Figure 10 proved that a single vocalizing animal with a different multipath arrival on each sensor can produce four distinct traces on the STD correlogram. If the animal's depth varies throughout the vocalization the delay between the direct- and multipath-arrival will vary, resulting in traces that are not parallel. Analysts are therefore likely to overestimate the number of vocalizing animals using just a STD correlogram for analysis.

Further analysis of the sonograms shown in Figure 9 and Figure 10 revealed that the direct- and multipath-arrival have different characteristics. Specifically, the multipath-arrival is more extended in time. A simple classification module was created, using a variation of the KM signal-duration-measurement algorithm [21] to estimate the click duration. Detection with a duration exceeding a predetermined threshold (1.0 ms) were classified as multipath and discarded. The duration threshold was selected by computing a histogram, shown in Figure 11, for the KM-computed duration of all clicks and picking an appropriate point on the curve. The KM-based classifier is processing intensive, but significantly improved the results, as shown in Figure 12.

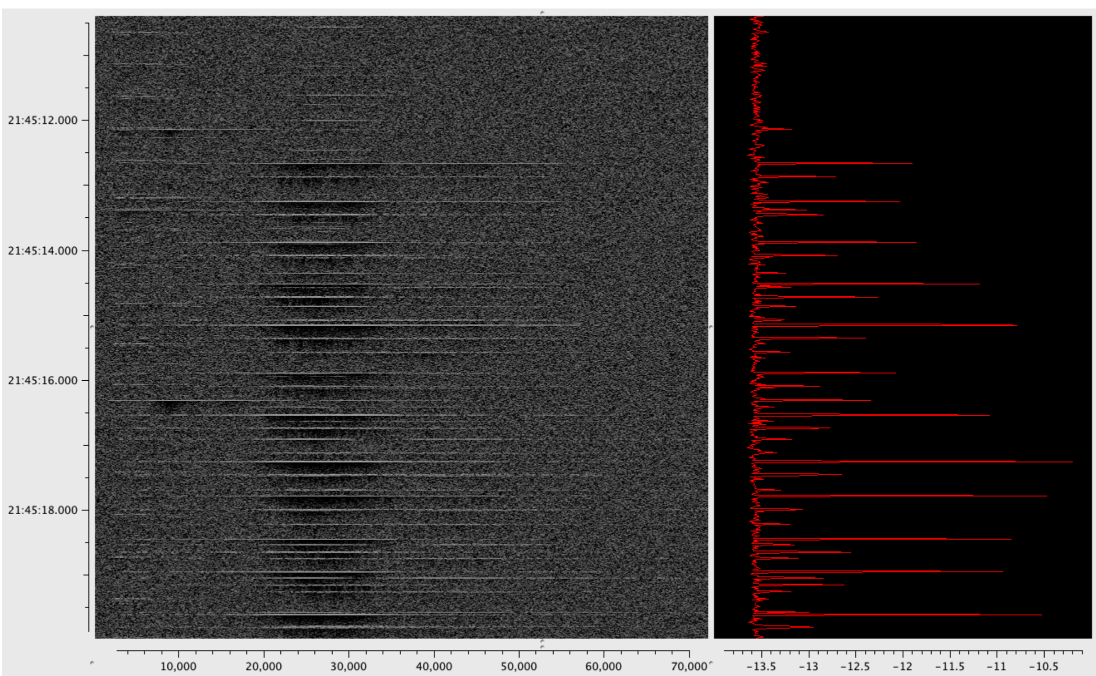

Figure 10: Sonogram and ETI for mammal clicks on St. Andrews recorder 4 that correspond to a data segment when eight traces are detected near the center of Figure 8.

Manual analysis of the short data segment in Figure 9 and Figure 10 indicated that three animals were vocalizing, but there may have been one or two others. Only two animal vocalizations were strong and consistent, while the others were found by carefully accounting for clicks that matched as one image was shifted over the other. This was a slow painstaking process. Considering this analysis and the traces shown in Figure 12 it is likely that some of these animals were missed and that the true number of vocalizing animals would be underestimated.

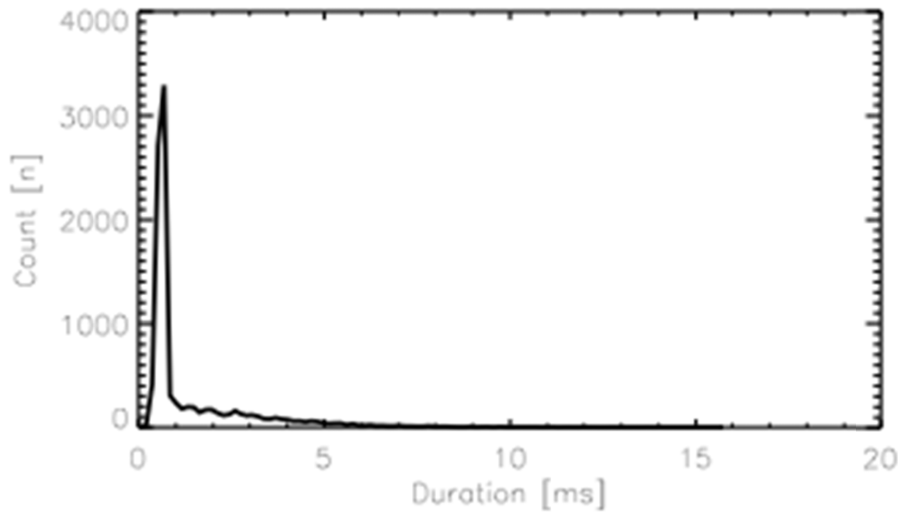

Figure 11: Histogram of measured click duration used for determining the direct-path discrimination threshold.

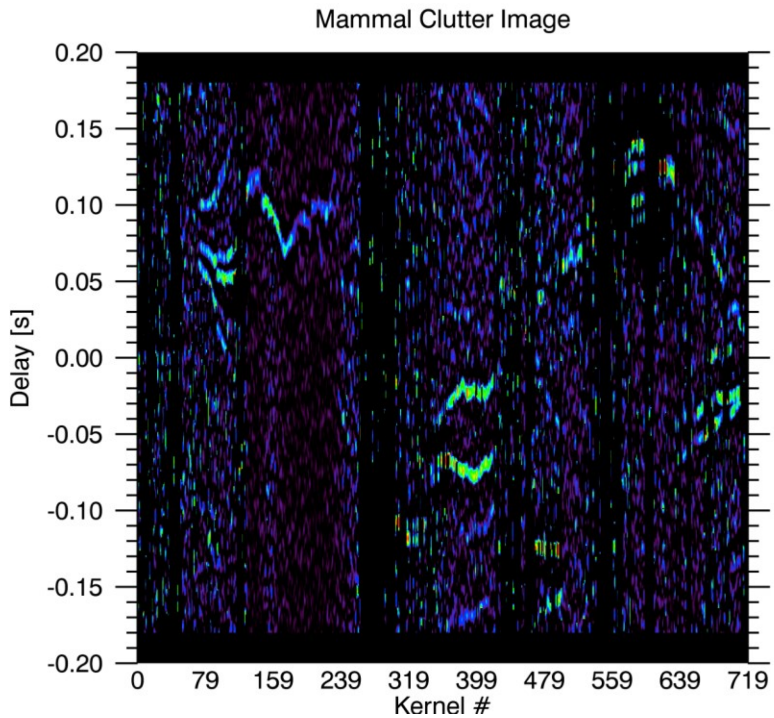

Figure 12: STD correlogram displaying data from 20:40:10Z to 22:40:10Z with multipath removed. 20-second kernels with 50% overlap were used for cross-correlation processing. There are fewer traces than for the same data processed without the multipath removal algorithm in Figure 8.

2.2 Dive event analysis

The objective of the task was to exploit knowledge of cetacean click trains produced during foraging, in order to group individual cetacean clicks into click trains and then dive events. Typically a cetacean will produce numerous click trains during a single foraging dive (dive event). Each click train contains a series of closely-spaced clicks with a pseudo-periodic ICI, which is typically less than one second, while pauses between click trains are significantly longer.

The current click detector marks individual clicks, resulting in a large number of indicators for one dive event, while the dive event itself is of greatest interest. The click detector also generates spurious false alarms and detection for isolated clicks, in which analysts are less interested because they are difficult to verify and if included, would significantly increase the false alarm rate.

Researcher hypothesized that a post-processing stage could be developed to analyze detection sequences and mark dive events. If successful, this would significantly reduce the number of events to verify/analyze, while reducing false alarms.

The post-processing stage groups individual click detections into click trains based on a minimum allowed spacing between individual detections. The click train can then be filtered based on its duration, which must be greater than a configurable length. This operation filters out what are likely spurious false alarms and isolated clicks. The filtered click trains can then be grouped into dive events based on a minimum spacing between click trains. Thus, the detector output represents an entire dive event and or a region of many false alarms, which reduces the number of decisions that an operator or analyst must perform.

This relatively-simple dive-event algorithm was applied to three data sets to examine performance and determine of the algorithm might be useful:

- The algorithm was applied to the University of St. Andrews array data described in Section 2.1.3.2. The data set was taken from the very active Gully marine protected area (MPA) located near the edge of the continental shelf, east of Sable Island. Here, detection was near continuous making it difficult to assess algorithm performance. Instead, the algorithm was used to find extended regions of continuous detection across two hydrophones for density algorithm testing.
- The algorithm was assessed using towed array data from Q320, collected by the Dutch Delphinus array on 30 Jan 2009 at the AUTEC range. These data contain manually annotated truth data of beaked whale clicks and was used for a high level detection performance assessment of the dive event algorithm.
- ROC analysis was performed using data taken from glider deployments conducted during Q302 in Exuma Sound and on the AUTEC range. These data include manually annotation of beaked whale clicks and self-noise that can be used to evaluate automated processing.

Initial settings for the dive-event grouping algorithm were determined using general knowledge of click-detector performance and cetacean click properties. Dive events were formed as follows:

- Individual detections occurring less than five seconds apart were grouped. If there were three or more detections grouped together, a click train was formed. Detections not meeting the criteria for a click train were discarded. The five second window was chosen to allow for gaps in detection caused by low SNR clicks, directivity changes, and propagation effects.
- Dive events were formed using click trains that were longer than one second and spaced less than sixty seconds apart.

The settings listed above were used for testing the algorithm, though further optimization could be performed after analysis on more data. Firm definitions for click trains and dive events have not yet been selected, and will differ for different species. More analysis is necessary to generate robust dive-event configurations.

2.2.1 St Andrew’s towed array data

The St Andrew’s array data was recorded in a very active MPA, causing nearly continuous detection. The dive event analysis algorithm was applied to the data set, but due to the abundance of true detections it did not significantly reduce the amount of data that might be presented to an analyst for verification and detailed analysis. A timeline showing data recordings and dive event groupings is shown in Figure 13. Long duration detection events occurring on both channel 1 and channel 4 were used for the density analysis described in Section 2.1.3.2.

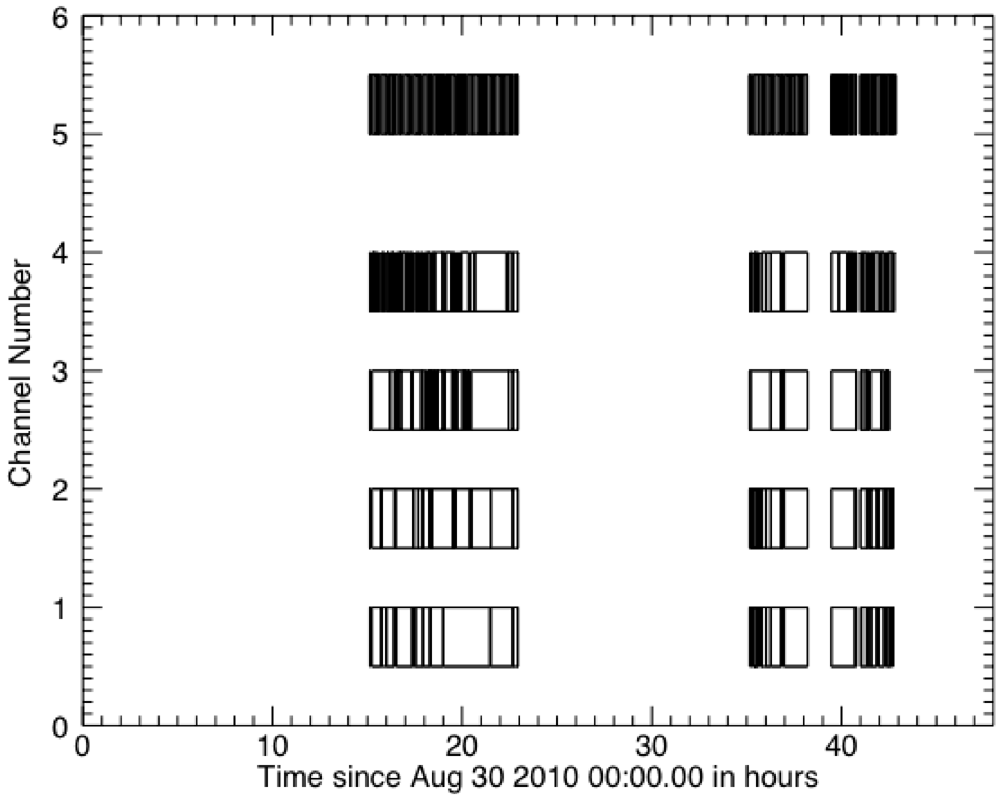

Figure 13: Timeline showing dive events for the St. Andrews data set. Channels 1 – 4 show detected dive events and Channel 5 represents times where recording occurred. Each data file and dive event is represented by a black box. Data files were very short and in many cases dive events were very short.

2.2.2 Delphinus towed array data

The Q320 Delphinus dataset contained strong beaked whale detection events with significant gaps between events, making it good candidate for demonstrating the dive-event-grouping approach. The algorithm was applied to these data by grouping. The algorithm performance is presented as timelines in Figure 14 and Figure 15, showing how dive events, click trains, and automatic detections relate to manual annotations and recording times. As anticipated, the data was reduced to a manageable number of detection events without sacrificing a significant number of true detections. One group of valid detections, shown in Figure 16, was missed because the click train was shorter than the one second threshold. It is reasonable to exclude the short detection to avoid excessive false alarms.

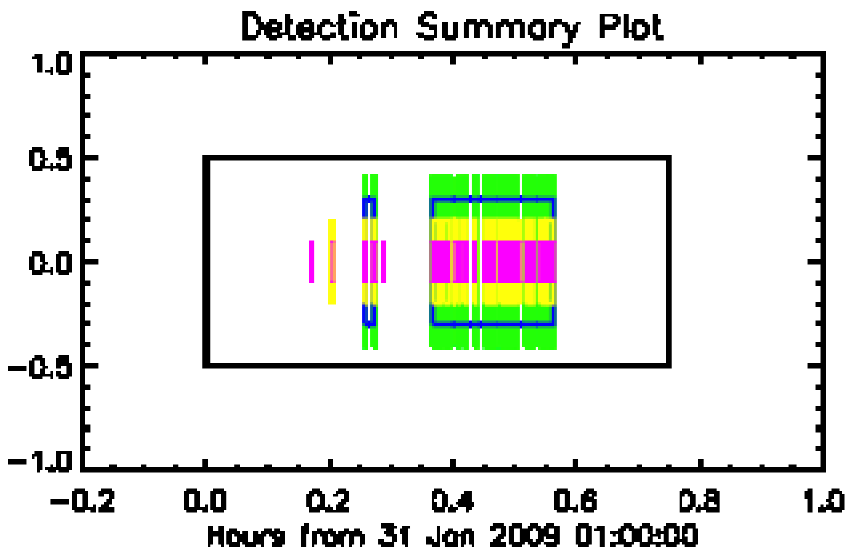

Figure 14: Timeline of detection and recording for 31 Jan 2009 Delphinus array data. Recording time is represented by the large black box, manual detections are green, automatic detections are pink, click trains are yellow and dive events are blue. The algorithm grouped all valid detections into two dive events, while filtering several spurious false detections.

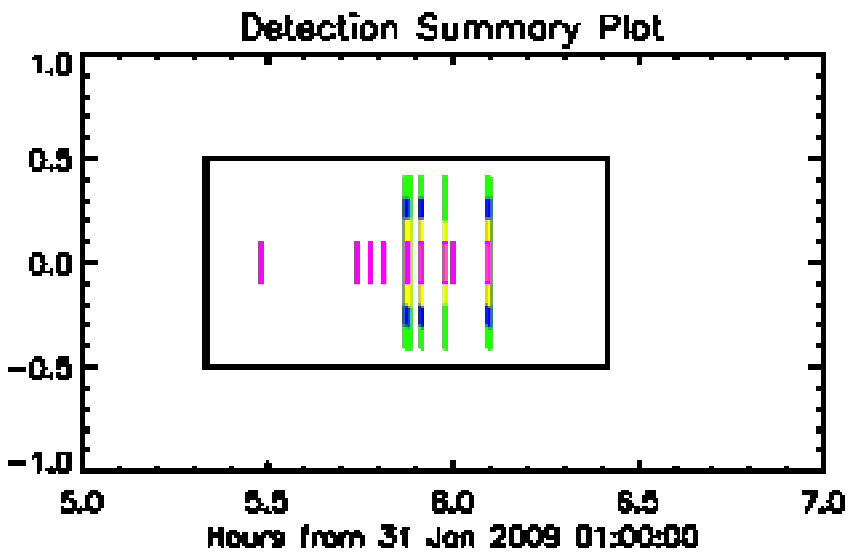

Figure 15: Timeline of detection and recording for 31 Jan 2009 Delphinus array data. Recording time is represented by the large black box, manual detections are green, automatic detections are pink, click trains are yellow and dive events are blue. In this case, one region of valid detections was filtered because it failed the duration test.

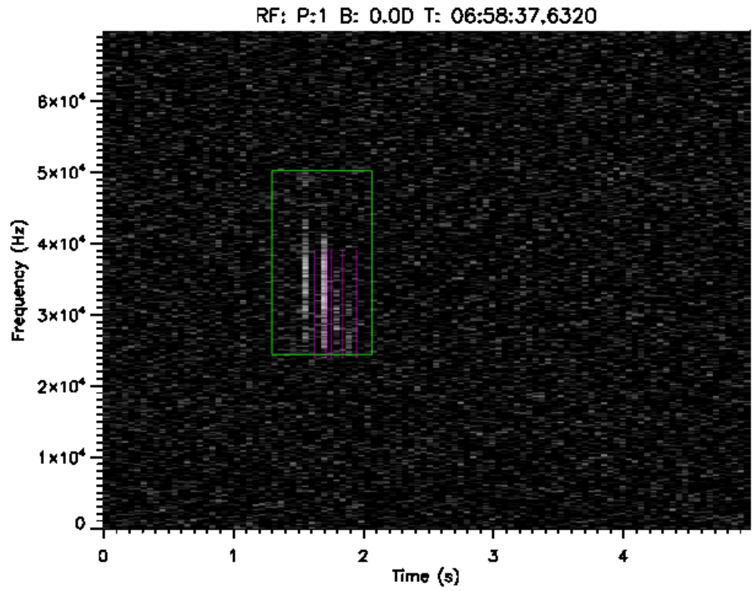

Figure 16: Spectrogram for the missed detection group. The detections, shown in pink, were filtered as they form a train less than 1 second in duration. This is likely a reasonable outcome.

2.2.3 Q302 Slocum Glider data

The Q302 Slocum Glider data set includes glider deployments that occurred between 30 January 2007 and 4 February 2007. These data include manual annotations of sperm whale, beaked whale, and glider self noise. The dive event grouping algorithm was applied to automatic detection results for beaked whale clicks. A receiver operating characteristic (ROC) analysis was performed to examine detection performance for each stage of the algorithm.

The detection data and algorithm output are presented in summary graphs where regions of missed detection, confirmed detection, and false alarms can be analyzed on a timeline. Detection-summary graphs are shown for the three glider deployments in Figure 17, Figure 18, and Figure 19. Manual annotations closer than 20 seconds were joined together in order to make the truth data more useful for dive events and click train analysis. The glider data were susceptible to false dive events caused by mechanical glider noise, shown in Figure 20. Although the data was not filtered by the algorithm, it would now be easier for an analyst to examine the dive event, quickly identifying it as due to glider noise and then ignore the entire detection group.

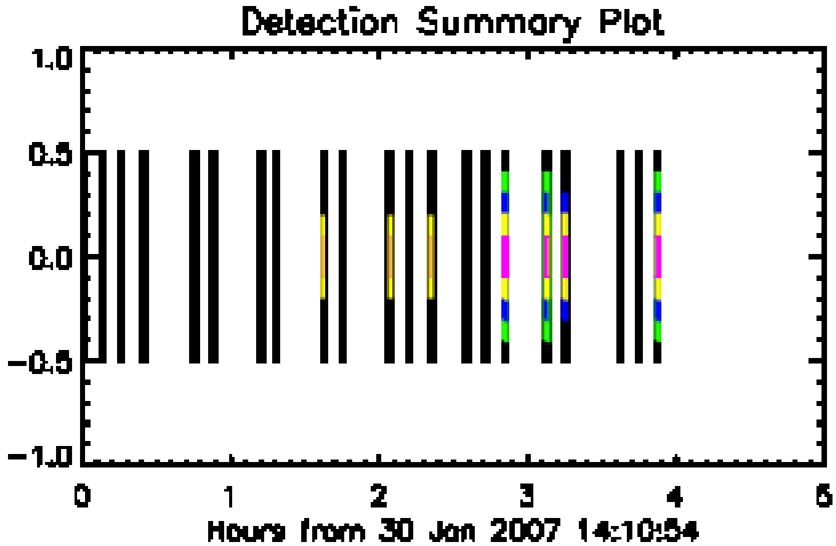

Figure 17: Detection-summary graph for the first Q320 glider deployment. Recording times are indicated by black boxes, manual detections are green, automatic detections are pink, click trains are yellow and dive events are blue. In this case, one recording event containing glider noise was classified as a dive event.

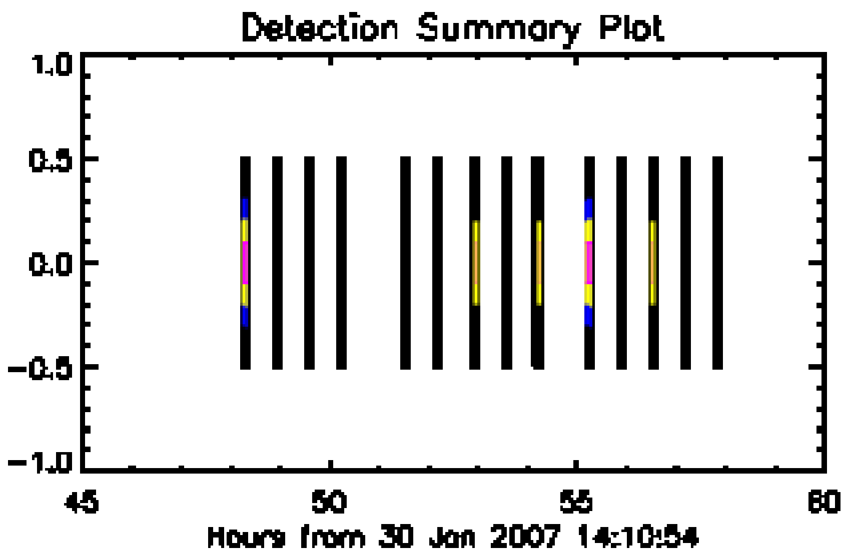

Figure 18: Detection-summary graph for the second Q320 glider deployment. Recording times are indicated by black boxes, manual detections are green, automatic detections are pink, click trains are yellow, and dive events are blue. In this case two recording events containing glider noise were classified as dive events.

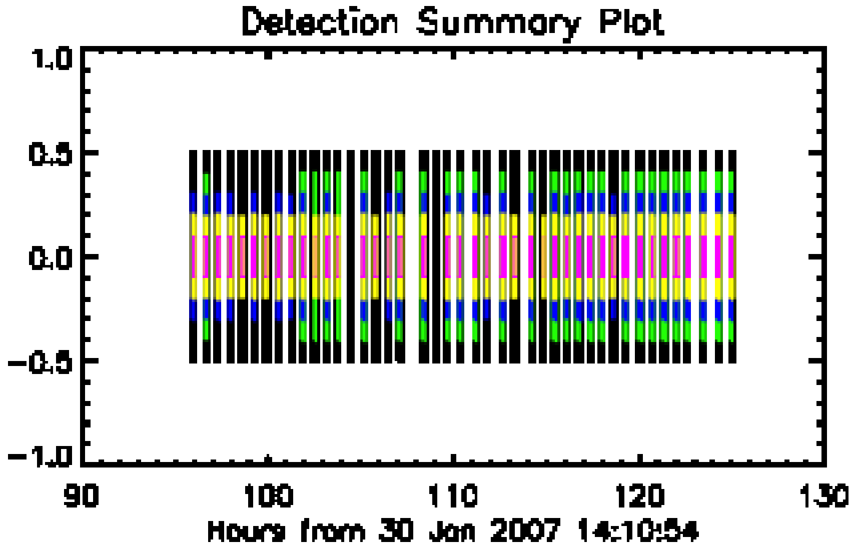

Figure 19: Detection-summary graph for the third Q320 glider deployment. Recording times are indicated by black boxes, manual detections are green, automatic detections are pink, click trains are yellow, and dive events are blue. Six of the nine false dive events were triggered by glider noise, one was triggered by sperm whale clicks covering the detection frequency-band, one was triggered by an unknown signal in the detection band, and one was triggered by spurious ocean noise.

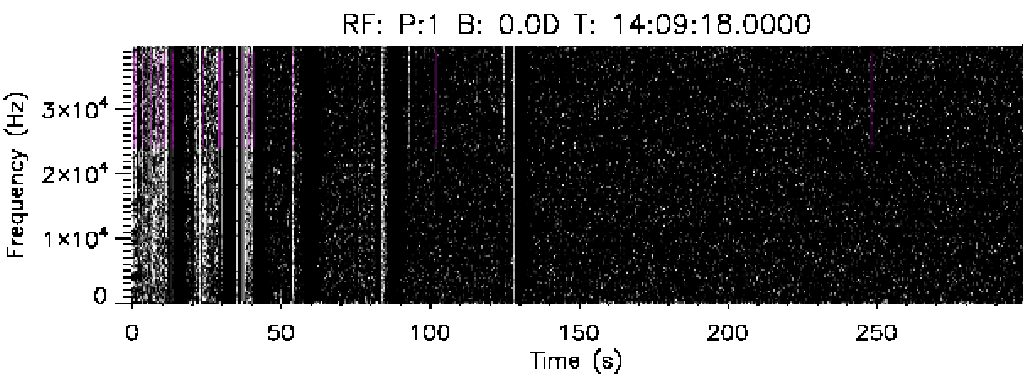

Figure 20: Example of false detections caused by mechanical glider noise. The detections, shown in pink, would be grouped into a false dive event.

ROC curves, shown in Figure 21, were generated for automatic detections, filtered click trains, and dive events. Each point on the curve was generated using a different detection threshold. Any detection contained inside of a manual annotation was deemed a confirmed detection, any click train overlapping with a manual annotation was considered a confirmed click train, and any dive event overlapping with a manual annotation was considered a confirmed dive event. Cases where the false alarm rate is less for a lower threshold (inversions in the ROC curve) are caused by long events breaking up for higher threshold and some of the now-shorter events being classified as false alarms. Figure 22 shows a region where noise and positive detections are grouped into one event that may break off at higher detection thresholds. More ROC curves could be generated by varying grouping and filtering parameters, which may yield curves with better performance.

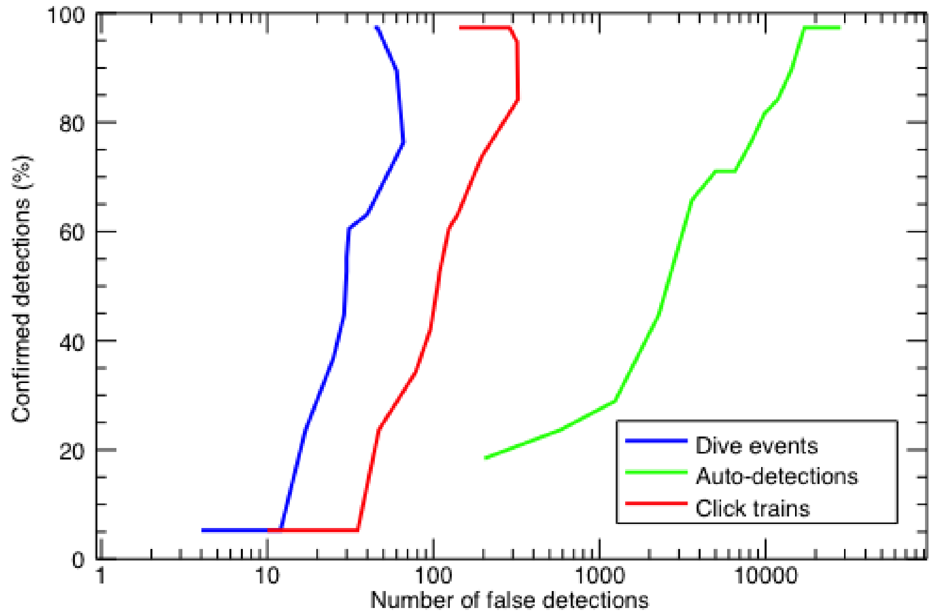

Figure 21: ROC curves showing detector performance for automatic detection, click trains, and dive events.

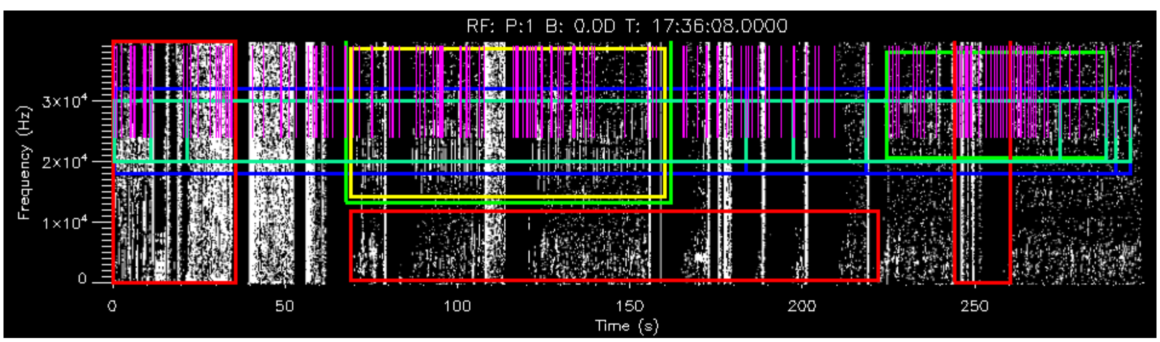

Figure 22: Glider data from 3 Feb 2007 showing that false alarms from neighbouring noise may be included in a valid dive event. The red boxes mark annotated structured noise, yellow boxes mark annotated unknown clicks, green boxes mark annotated beaked whales, pink boxes mark detections, teal boxes mark click trains and blue boxes mark dive events. If a higher detection threshold is applied, the noise-based detections will separate from the dive event and be classified as a false alarm. This can cause inversions in the ROC curve, which are not typical.

3 Recommendations

Ideas for improvement of both the software and algorithms are frequently generated during a contract. This section documents those recommendations, capturing both a high-level description of the suggestion and a rough order-of-magnitude (ROM) estimate of the required effort. Each section contains a summary table itemizing the ideas and the related ROM. Often a reference to a Jira ticket, where more detailed information is available, is also provided.

ROM estimates are generated after a preliminary review of the requirements and therefore could have significant variance. They are broken down into the following categories:

- Very Small is less than one (1) day of effort,
- Small is between one (1) day and five (5) days of effort,
- Medium is between one (1) week and two (2) weeks of effort, and
- Large is more than two (2) weeks of effort.

Where a more accurate ROM is available it may be provided in brackets.

3.1 Main application or suggestion group

Table 1 is a summary of selected issues being tracked in Jira for future upgrades to STAR with regards to cetacean detection, analysis and estimation. These issues are further explained in the following subsections.

Table 1: Jira issues tracking future enhancements for the cetacean data algorithm development project

Jira ID	Type	Summary	Estimated Effort
AKOT-227	Improvement	Improve cetacean simulator to allow for animal motion	Small (2 days)
AKOT-228	Improvement	Integrating RMC trace detection with STD correlograms	Medium (1 week)
AKOT-244	Improvement	Multipath optimization	TBD
AKOT-245	Improvement	Integrate dive event analysis and STD into SPPACS streams	TBD
AKOT-246	Improvement	Upgrade to use a database for detections	TBD

3.1.1 Improvements to cetacean simulator

The current cetacean simulation software uses a simple approach to model a stationary clicking animal. The simulation software would be much more realistic and allow for better, robust algorithm testing if animal motion was considered. The current software uses STAR map positions for the animal location, so it would be a relatively small upgrade to associate a STAR track with each animal.

The simulation software could also be upgraded to generate simulated acoustic data from the animal. The current impulse click trains and basic environmental uncertainties are a very idealistic view of the situation. A simulator upgrade that generates synthetic acoustic clicks and takes into account directivity and basic propagation would be an incredibly useful tool for algorithm prototyping, but would require significant effort.

3.1.2 Integrate trace detection

Akoostix recently collaborated with researchers at the the Royal Military College (RMC) of Canada to develop trace detection algorithms for images, in this case for multistatic active sonar clutter-mitigation images. Application of a similar algorithm to these data would complete an important step in automatically extracting the number of targets from a STD correlogram. The algorithm is still in the initial stages of prototyping and development, but the related software may be integrated with the density estimation algorithms.

3.1.3 Multipath optimization

The simple approach of filtering multipath arrivals based on signal duration proved useful in removing false traces from the STD correlogram. The current algorithm to calculate duration is implemented in IDL and uses a large amount processing power as it performs convolution. This can be optimized and implemented in a streaming SPPACS application, which also allow it to be used more generally.

3.1.4 Integrate dive event analysis and STD into SPPACS streams

The algorithms prototyped in STAR-IDL under this contract could be ported into streaming SPPACS applications. The algorithms could then be further optimized in C/C++, where they could also be implemented in user-friendly software applications such as ACDC, or implemented on embedded systems, used for automatic detection (e.g. gliders, PARB, etc.).

3.1.5 Upgrade to use a database for detections

Detection data is currently stored in text file logs and different categories of logs (i.e. for different targets) are managed using directory structures. This method for managing the data is inefficient, as it consumes extra disk space, processing power, and analyst time during parsing and sorting operations. It is also limiting because utilities like OPD and ACDC are not aware of the meaning behind custom directory structures and so complex queries and compound sets cannot be used.

This approach may also lead configuration management errors, caused by people editing the detection logs or misplacing output files.

Ideally, detections should be stored in a database (e.g. SQLite, SQL) that would make filtering/querying, relating, and managing detection data more efficient. The data base could be used in programs such as ACDC and OPD to quickly and effectively share detection information and annotations without the manual adjustments currently performed to work around design assumptions that limit utility for the current solution.

4 Summary and conclusions

This project focused on developing algorithms to improve cetacean detection and density estimation. It was successful, delivering significant advances in algorithms to improve the efficiency and effectiveness of analysts. These advances include:

- An algorithm to produce STD correlograms. STD correlograms allow analysts to view the STD of the same cetacean click train between two sensors in a sonogram-style plot. These images separate what appear as complex overlapping click trains in a sonogram into easily-identified traces in the STD correlogram. Each trace corresponds to a different arrival time, which should relate to a single animal but can also indicate multipath. The slope of these traces is indicative of cetacean motion and could be used for tracking if three or more sensors were detecting at once. These images greatly improve an analyst's ability to quickly estimate the number of animals in an area of interest.
- An algorithm to classify direct arrival and multipath detection. This algorithm significantly reduced the number of multipath detections for the test data, thus improving the quality of the associated STD correlograms.
- An algorithm to group detection data into dive events, while filtering spurious false alarms. This approach provides an analyst with pre-identified regions of interest, avoiding analysis of the each detection, which can number in the thousands. It is most effective for data where there are significant gaps between valid detection.

Though significant progress was made during this contract, a list of suggested improvements remains and should continue to be addressed as budget and schedule permit. Additional research should also be performed including:

- The current approach to generating STD correlograms should be compared to the results for coherent STD using time-series data, such as the algorithms available in PAMGUARD. It may be possible to generate similar images for closely-spaced sensors, providing additional flexibility for sensor design.
- Density estimation algorithms should be evaluated using data that includes visual sighting data. Comparison of density estimates using acoustically- and visually-based estimates may provide additional insight, or reveal estimation errors related to either approach.
- More research is required to determine robust and appropriate click train and dive event clustering algorithm parameters for various species.

References

[1] Hood, J. and Glessing, B. (2005), Marine Mammal Detection Final Report (DRDC Atlantic CR 2005-XXX), MacDonald Dettwiler and Associates Ltd., Dartmouth, Nova Scotia.

[2] Theriault, J., Hood, J., and Deonarine, K. (2006), Acoustic Cetacean Detection Capability, In *Proceedings of ECUA Portugal 2006*, Carvoeiro, Portugal.

[3] Theriault, J., Mosher, D., Hood, J., Flogeras, D., and Murphy, T. (2007), Detection of Beaked Whales Using Autonomous Underwater Vehicle (Glider), In *Proceedings of 3rd International Workshop in the Detection and Classification of Marine Mammals Using Passive Acoustics*, Boston, MA, USA

[4] Theriault, J., and Hood, J. (2007), Acoustic Cetacean Detection Capability Performance with the Workshop Dataset, In *Proceedings of 3rd International Workshop in the Detection and Classification of Marine Mammals Using Passive Acoustics*, Boston, MA, USA.

[5] Hood, J. and Bougher, B. (2010), MMOS `07 and Sirena `08 Analysis Report (DRDC Atlantic CR 2010-060), Akoostix Inc., Dartmouth, Nova Scotia.

[6] Hood, J. and Bougher, B. (2010), MMOS `09 Analysis Report (DRDC Atlantic CR 2010-060), Akoostix Inc., Dartmouth, Nova Scotia.

[7] James Theriault et al. (2010), Cetacean Click Detection: Gliders in the Tongue of the Ocean and Alboran Sea (DRDC Atlantic TM 2010-062), in preparation.

[8] Bougher, B., Hood, J., and Theriault, J., Improved Band-limited Marine Mammal Detector using Signal Excess, presented at the *Fifth International Workshop on Detection, Classification, Localization, and Density Estimation of Marine Mammals using Passive Acoustics*, 21-35 Aug 11, Mount Hood, Oregon.

[9] Hood, J. and Macmichael, C. (2005), STAR Whale Localization Aids – Final Report (DRDC Atlantic CR 2005-127), MacDonald Dettwiler and Associates Ltd., Dartmouth, Nova Scotia.

[10] Hood, J., and Bougher, B. (2010), Transient Localization using Sonobuoys (DRDC Atlantic CR 2010-XXX), Akoostix Inc., Dartmouth, Nova Scotia.

[11] Ma, T., Collison, N., Theriault, J., Hood, J., Pecknold, S., and Bougher, B. (2010), Detection of Precise Time Events for Marine Mammal Clicks, presented at 2010 Meeting of the Canadian Acoustical Association, 13-15 October 10, Victoria, British Columbia.Ma, T. et al. (2010), Detection of Precise Time Events for Marine Mammal Clicks, In *Proceedings of the Acoustics Week in Canada*, 30-31, Victoria: Canadian Acoustical Association.

[12] Kandia,V., Stylianou, Y., and Dutoit, T. (2008) Improve the accuracy of TDOA measurement using the Teager-Kaiser Energy operator, In *Proceedings of the International Workshop on New Trends for Environmental Monitoring Using Passive Systems (PASSIVE '08)*, IEEE Xplore, Hyeres, France.

[13] Frantzis, A., et al. (2002), Clicks from Cuvier's beaked whales, *Ziphius cavirostris*, *J. Acoust. Soc. Am.*, 112(1), 34-37.

[14] Hooker, S. and Whitehead, H. (2002), Click Characteristics of Northern Bottlenose Whales (*Hyperoodon Ampullatus*), Marine Mammal Science, 18(1), 69-80.

[15] Zimmer, W. et al. (2005), Echolocation Clicks of Free-ranging Cuvier's beaked whales (*Ziphius cavirostris*), *J. Acoust. Soc. Am.*, 117(6), 3919-3927.

[16] Hood, J. and Burnett, D. (2008), Compilation of Marine Mammal Passive Transients for Aural Classification (DRDC Atlantic CR 2008-287), Akoostix Inc., Dartmouth, Nova Scotia.

[17] Theriault, J., Hood, J., Bougher, B., and Radcliffe, M., Sorting Out Numbers: How Many Animals are Clicking?, presented at the Fifth International Workshop on Detection, Classification, Localization, and Density Estimation of Marine Mammals using Passive Acoustics, 21-35 Aug 11, Mount Hood, Oregon.

[18] Hood, J., Bougher, B., and McInnis, J. (2011), CFMETR 2011-1 Trial Support (DRDC Atlantic CR 2011-XXX), Akoostix Inc., Dartmouth, Nova Scotia.

[19] Hood, J. (2011), Introduction to the Software Tools for Analysis and Research (STAR): Tools and Processes Version 1.0 (AI MA 2011-007), Akoostix Inc., Dartmouth, Nova Scotia.

[20] Ward, J. et al. (2008), Passive acoustic detection and localization of *mesoplodon densirostris* (Blainville's beaked whale) vocalizations using distributed bottom- mounted hydrophones in conjunction with a digital tag (DTAG) recording, *Canadian Acoustics,* 36(1), 60-66.

[21] Kliewer, J. and Mertins, A. (1998), Audio Subband Coding With Improved Representation of Transient Signal Segments, In *Proceedings of Europ. Signal Processing Conf.*, 1245-1248, Island of Rhodes, Greece: EUSIPCO.

[22] McInnis, J. and Ryan, G. (2011), OPD User Manual Version 1.5 (AI MA 2008-001), Akoostix Inc., Dartmouth, Nova Scotia.

[23] Flogeras, D. (2010), Active LAND Buoy Operator Manual Version 1.0 (AI MA 2010-009), Akoostix Inc., Dartmouth, Nova Scotia.

This page intentionally left blank.

Annex A Configuration management

The final software deliverable for this contract was provided on the STAR release CD - version 6.6.11. The CD was generated and delivered on 30 March 2012. This software release contains the original project deliverables. This section of the document describes the content of that CD.

A.1 STAR branch and release information

Each logical grouping of software modules has been independently versioned on the CD that is also versioned. The current STAR release version is 6.6.11 and contains the following:

- OPD 2.6.1,
- ACDC 2.1.8,
- SPPACS 1.3.0. and
- Analysis Tools 6.13.0 (STAR-IDL).

The 6.6.11 release CD was generated in conjunction with other DISO call-ups. Installation instructions are located in the root directory on the release CD.

A.1.1 STAR software documentation

Some manuals, API documentation, and other design documents are provided with the 6.6.11 software release CD. In a standard STAR distribution they can be found by opening *the /usr/local/atools/star-6.6.11/documentation.html* file in a standard web browser. This page contains links to several sets of documentation including:

- Software revision history,
- IDLDoc for the analysis tools (STAR-IDL) and DOxygen generated documents for OPD, ACDC and SPPACs,
- The STAR user manual,
- STAR quick reference guides,
- STAR-IDL application user manuals,
- STAR application user manuals, and
- Useful third-party Documentation.

A.2 Issue summary

The issue summary in Table 2 shows the current state of known defects for all of the software release candidates listed in A.1 as of 30 March 2012.

The distribution of issues is indicative of the maturity of the software. Though maturing, much of this software is composed of various evolutions of an iterative design, especially command line SPPACS applications and STAR-IDL components. This software would benefit from general design improvements and refactoring. There are no active blocker issues but there are several critical issues. These are obscure or infrequent bugs that were discovered during current work, but budget or schedule has been insufficient to address them yet. Critical issues are issues that still allow the operator to perform their function but could cause erroneous results or loss of data in those instances. These bugs should be fixed in the near future. Only Blocker issues do not have a work-around and need to be addressed before a contact can be completed successfully.

Table 2: Issue Summary (Severity vs. Status) for all software on STAR release 6.6.11

	Opened	**Reopened**	**Resolved**	**Closed**
Blocker	0	0	3	33
Critical	14	1	0	71
Major	115	4	7	215
Minor	37	2	0	25
Trivial	7	0	0	3
Undecided	0	0	0	5

Table 3 summarizes the critical issues that remain open, but only for software relevant to this contract. None of the critical issues had any effect on the success of this contract. Resolution of these issues may increase efficiency during the execution of future call-ups or contracts.

Table 3: Known Critical issues for STAR

Module	Issue ID	Summary	Description
STAR++	STPP-3	Error in STAR IDL hyperbola fixing	Starting from line 1102 of star_cross_data_support in the create_hyperbola_hyperbola_contact function there may be a problem. The variables xbh1, xbh2, etc are not used but instead a set of vectors xbh, ybh, BB, and SS are created. These vectors contain all three buoys instead of just the two from one hyperbola. These are what is used in getbc and then for bayesprobs. The original IDL code from Joe Maksym doesn't make this simplification (idemo.pro about lines 23813 and 25263). This needs to be checked to make sure that this is valid or the posterior probabilities will be incorrect.
OPD	OPDY275	Crash on deleting data	A crash was experienced while deleting old data while processing other data. If processing is completed the data should be deleted as well. See comments for system dump.
OPD	OPDY245	OPD crash if EADAQ changes state during processing but haven't stopped OPD processing	OPD will hang if the user stops processing and EADAQ resets. The display will freeze and the user is unable to stop processing.
OPD	OPDY244	OPD crash while trying to stop processing after EADAQ stops.	OPD will crash if the user tries to process EADAQ when EADAQ is not recording.

OPD	OPDY174	Bad combination of zero padding/overlap in the processing parameters dialog can cause a hang	If a user chooses a 8192 FFT size, 8191 zero pads, and anything but a 0% overlap, the user will end up with (8192-8191)*(1-overlap) which for any overlap but 0% will cause an integer round to 0. Typically users will not configure it this way.
OPD	OPDY160	Crash with extreme processing parameters	A crash occurs using the following processing parameters: File: meg//scratch/OPD_test_data/bof_99.dat Channels: 1-16 Default settings except: 1M pt FFT 99% overlap Look at Thread 9 of first comment for crash report on the Mac. It hangs almost immediately then crashes after a few seconds.
OPD	OPDY77	OPD hangs when validating data sources	OPD tries to verify the data stream by reading as much as required to determine the format and sensors. It stalls the system until it receives this information. An array server can be accepting connections but not sending data. OPD connects to Northern Watch Array Server automatically as soon as a valid IP and port are given. OPD saves the IP and port to save from retyping them all the time. So if the user stops feeding data on one port (but still allow connections) and start it on another, then restart OPD, it tries to connect to the previously saved port and hangs. Workaround: 1. Port being saved is in the registry settings. It can be changed manually before resetting OPD or 2. The user can start the array server on the previously saved port. Permanent Fix (tentative): 1. Prevent OPD from validating data sources automatically by using some kind of connect button. 2. Have a timeout for validating sources since we still have the issue that a socket connection can be made but the data is not flowing.

SPPACS	AKSP91	sp_correlate output time is not correct	The output file time for sp_correlate is being set to 31-DEC-69 23:59:59 when the input time is actually 01-JAN-80 03:00:00 They should be the same.
SPPACS	AKSP72	sp_median_nrmf is referencing a null pointer occasionally in win32	A couple of times now, we have seen sp_median_nrmf cause a crash (in OPD) by dereferencing a null pointer. This seems to only happen on Win32, but may just be being hidden in UNIX environments (windows has a history of being more strict, especially in debug mode). Before the crash (during construction of the module) a bunch of warnings are printed to stderr, "Can't find remove point" A quick look in the code revealed that there is a comment saying "if we got here, there is a bug".
STAR-IDL	AKOT-225	Trial recon crashes when a template cable model is provided	The following error was produced when a template cable model was placed in the cable_model directory and then trial_recon.pro was run. % Variable is undefined: CABLE_NUMBER_LIST. % Execution halted at: STAR_CABLE_MODEL_DAO::IMPORT 201 /opt/tools/analysis_tools/active/src/lib/tactical_database/data_access/sta r_cable_model_dao__define.pro
STAR-IDL	AKOT-163	Failure with ping to source mapping- track feature extraction	The user will get a failure when trying to map a specific ping to a source when trying to run the tracking feature extraction application. The error is similar to the issue affecting the main clutter application. Error message: Could not find active source for blast. Ignoring blast
STAR-IDL	AKOT-153	Animation with selected tracks- bad behaviour	When a user selected tracks in ITAC then started animation some strange tracks were observed. If the user turns off selections, then animation behaves normally again.

STAR-IDL	AKOT-150	Crash/Hang in STAR++	A hang was observed when using STAR++ to investigate AKOT-149 on OSX (tracking3) and a crash on TMAST02 data (Linux and likely slightly different version). User was running the Monte-Carlo simulation for the runs on OSX and had them off once and on another time for the TMAST02 runs (crashed both ways). Generally settings were: 2 detections, 10% threshold 2 Clusters, 3km radius El-El, El-Brg, El-Hy crossings Monte-Carlo simulation with all error types and cluster results Search time adjustment of 0.5 seconds On TMAST 02 user had designated on channels 2,3,4,5 For Tracking3 user set a threshold of 550.
AKOT-107	STAR-IDL	Ownership of overlays and contained data	There is a general problem with ownership of overlays and contained data by the tactical plot. The tactical plot can be closed and reopened several time during an application's lifetime, so if it destroys all data that it contains it will be lost to the application and cannot be used on subsequent instantiation of the tactical plot, or usage by other modules (i.e. tracker). It may be reasonable just to own the overlay itself and not the contained data, but then we need to assign ownership of the contained data to someone. Another option may be to tell the tactical plot when it owns an overlay. Right now image overlays are owned by the tactical database, because it would have been more work to create a data container in the database and then force the creation of an overlay after the fact. This might need to change depending on how this issue is resolved. If two objects are created (image container and image overlay) then we need to be careful how data is passed between them to avoid expensive data copies for big images.

Annex B Software tools

This section provides background information necessary to understand the role that DRDC software played in this contract. This flexible reusable software enabled the project to make better use of the available budget, advancing the research more than would be otherwise possible. Their relationship to the project is described below while a high-level description of the tool itself is provided in the subsections below:

- SPPACS was used to perform batch data processing primarily initial detection processing, along with basic data conversion and utility tasks such as WAV to DAT file format conversion and time stamping.
- STAR-IDL was used to perform all of the advanced processing and prototype development, along with some of the data analysis. Software libraries that perform utility tasks, such as parsing detection logs, were integrated into prototypes, allowing researchers to focus on core scientific concerns and new capability.
- OPD was used for detailed spectral processing and investigation of specific concerns. For example, manual STD analysis was performed using sonograms generated in OPD. OPD was also used to visualize data and establish initial settings for click train and dive event clustering algorithms.
- ACDC was used as a secondary tool to examine detection results, playing only a minor role on this contract.
- The AS was used onboard the Slocum gliders to sample and record acoustic data, and perform live detection processing. These data were analyzed under this contract.
- The DAS was used to replay selected real and synthetic cetacean vocalizations during CFMETR trials. Their integrated GPS logging, precise time synchronization, and ability to quickly adjust replay parameters proved very useful during these experiments.

B.1 Signal Processing Packages (SPPACS)

SPPACS is a group of software programs that are written in the C/C++ programming languages, with each application providing a specific processing or utility function. They are designed to run on Linux and OSX based PCs and typically work with Defence Research Establishment Atlantic (DREA) formatted data files (DAT), though format converters are also contained in the suite. SPPACS has slowly evolved to its present day state.

The SPPACS software suite consists of two types of software. One type is runtime executables. These applications have proven to be very useful in simplifying data management and sonar processing tasks by providing a set of tools from which to build the necessary, often much customized, processing streams. These streams can be run from the command line or assembled into scripts to perform batch-processing tasks allowing large amounts of data to be automatically and incrementally processed.

The second form of the software is a group of library functions that can be used by other programs to efficiently perform standard tasks. These library functions are extensively used by the runtime software, but can also be used for other applications, such as OPD. There are several types of libraries of which three are most commonly used in SPPACS:

- Utility (e.g. math, geo, filesystem, …) libraries that consist of utility routines for performing tasks, such as header manipulation, geospatial data representation, and command line parsing.
- Signal Processing (e.g. splib) libraries that contain modules for low-level signal-processing. A new SPPACS module typically consists of one or more SPLIB modules linked together with an SPPACS user interface.
- Sonar Processing (e.g. sonlib) libraries that contain modules consisting of several SPLIB modules linked internally to create a complex sonar module, such as passive processing.

B.1.1 Background and design Information

More generic and reusable software was created by separating the library code above from SPPACS. These modules are independent of the data header format, time-stamping method, etc., and are suitable for integration in real-time processing systems. The libraries can be built to run on a number of UNIX, OSX or Microsoft Windows platforms and on less common processors such as the ARM core and Texas Instruments (TI) DSP. Once successfully ported, the CMAKE build environment supports subsequent builds with a command line option.

The C and C++ elements of the libraries are intentionally separated to ensure that the core capability, found primarily in the C modules, can be readily ported to systems that don't support the more complex language features employed in the C++ version of the libraries. For the most part, the C++ layer consists of a wrapper on the C layer that provides a more generic method of instantiating, connecting and running modules. This is provided by inheritance that is, in part, the adoption of a common interface from a base class allowing parts of the system to interact with a module without knowing the details of the module. Connection of SPPACS applications using UNIX pipes provides similar functionality at the application layer.

SPPACS is also supported by a number of libraries, such as the Fastest Fourier Transform in the West (FFTW), helping to ensure that the SPPACS software runs as efficiently as possible, while providing a significant reduction in coding effort. These dependencies, and the associated licenses, are tracked for those projects that require knowledge of intellectual property (IP).

B.2 STAR-IDL

The STAR-IDL[2] tools were developed to support general research and analysis objectives at DRDC Atlantic. The actual software goes hand-in-hand with an analysis process that is intended to help formalize a reliable and consistent research and analysis methodology [19]. The primary objectives of the STAR-IDL tools are:

[2] The STAR-IDL tools were formerly referred to as the Software Tools for Analysis and Research (STAR). The STAR Software Suite has now come to mean the greater tool set, including OPD, ACDC, SPPACS, etc.

- Provide scientific grade analysis tools that allow for efficient, detailed quantitative and qualitative analysis of a data set.
- Provide scientific grade algorithm prototyping and refinement tools that can be used to quickly realize a variety of algorithm options, validate the basis of the algorithm, and determine the best approach to use for system prototypes.
- Support synergy between DRDC groups and the Department of National Defence (DND) by providing a common software base for analysis. This synergy encourages inter-group communication and simplifies user training, analysis process development, documentation and data portability.
- Support cost and analysis efficiency by providing software reuse and common tools and data formats. Examples of efficiency would be using the output of analysis from one group to feed the inputs of another, or using common software components to lower development cost of several custom analysis tools.

Most STAR-IDL components are currently implemented using Interactive Data Language (IDL), though the design is not restricted to IDL. For example, localization algorithms contained in C++ libraries are accessed from IDL.

Applications in the STAR-IDL tools are built using a combination of reusable and custom components that meet the requirements of each application. The layered design and common components allow for rapid and logical development of new capabilities. Though currently focused on two main areas - sonar data processing and analysis, and target localization, tracking, and multi-sensor data fusion - the tools are capable of expanding to meet other analysis and research requirements.

B.3 Omni-Passive Display (OPD)

OPD is a standalone signal processing application designed to run on UNIX, OSX, and Microsoft Windows platforms. It can be used to quickly produce sonogram, energy-time indicator (ETI), amplitude-line indicator (ALI), and time-series output from DREA digital acoustic tape (.DAT/.DAT32) files, wave files, sound card, EADAQ, Rapidly Deployable System (RDS), and Northern Watch. The following functions summarize its capability (detailed information can be found in the OPD User Manual [22]):

- A user can quickly set up the desired signal processing by loading in a preset configuration from storage, or by simply defining the desired frequency and time resolution. A more sophisticated user can define a wide range of parameters, including Fast Fourier Transform (FFT) size, zero padding, overlap, quantization range, decimation, sonogram compression and much more.
- OPD provides an optional beamformer and is capable of processing complex heterodyned time-series data.
- Annotations can be added to the data.
    - The user can assign a category (or classification) to the annotation from a list of presets as well as provide free-form text to associate with the annotation.

  - Previously generated annotations are displayed on screen when processing data associated with the annotation.
  - The annotation format is compatible with STAR-IDL and ACDC.
- Each processing result is stored in memory and can be selected for viewing and analysis. Analysis tools include a crosshair cursor for time-frequency measurements.
- The entire sonogram can be saved to an image file to capture the output for reports, etc.
- A WAV extraction tool allows the operator to define a region within a sonogram and clip the raw data associated with the selected bounds into a wave file.
- Operational measurement tools such as harmonic, banding, periodic event and Doppler cursors can be used to analyze advanced features in data and learn tactical information about potential targets.

B.4 Acoustic Cetacean Detection Capability (ACDC)

The ACDC application was developed to provide an initial cetacean monitoring capability for DRDC with the hopes of growing the application to provide broader, generic support. The vision is to create a component that can be connected to a variety of sonar systems and configured to automatically monitor data streams for cetacean vocalizations. Eventually detections would be vetted by more complex classification software before being presented to an operator for validation and mitigation. An intuitive user-friendly display would allow an operator to operate the system part-time and automatically log detections with annotation showing mitigation action. This log information could also be merged with other streams, such as ping logs, to provide comprehensive evidence gathering to support the crew in the case of an incident.

The software is contained in two separable components; display processing and control (ACDC); and signal processing (sp_transient_processing). This was intentional and allows signal processing to take place off-line or in a remote system such as the Slocum Glider, though it can also be run as part of ACDC. The heart of the detection processing is the Sentinel sonar library (SONLIB) module that can be tuned for transient detection. Processing results are stored in up to five formats; American Standard Code for Information Interchange (ASCII) log files, WAV files, and DREA DAT formatted power files containing black and white GRAM images, power files containing raw spectral data, energy time indictor (ETI) files containing band vs. time data, and amplitude line integration (ALI) files containing the averaged spectra data. The detection results are dynamically read into the ACDC application for operator analysis and verification. Dynamic reading allows the processing and analysis to run simultaneously, providing automatic updates as detections are made. ACDC will function on any data set once provided with a directory in which to find the required detection results.

B.5 Acoustic Subsystem (AS)

The AS is a general-purpose embedded acoustic recording and detection system composed of an integrated set of reusable software modules. The AS operates in one of three modes:

- Transient detection and recording mode – In this mode onboard detection processing is performed, the entire sample period is recorded, and individual captures (WAV files) are created for each detected transient along with an ASCII detection log. When operating at 40 kHz bandwidth the maximum sampling duration, on the current hardware, is 5 minutes with a duty cycle of ~50%.
- Target (vessel) detection and recording mode – In this mode onboard detection processing is performed, the entire sample period is recorded, and an ASCII detection log is created. In this mode, the bandwidth is often limited allowing the AS to process at real-time.
- Ambient recording mode – In this mode the AS records acoustic data and operates real time, so recording duration is not limited on the current hardware, provided enough flash memory is available.

The system is intended for soft real-time operation and is normally installed on a low-power, fixed-point, general-purpose processor and paired with other technology that acts as the vehicle (e.g. Slocum Glider, Passive Acoustic Reusable Buoy (PARB), Stealth Buoy).

When used for marine mammal detection, it is most commonly paired with the Slocum Glider, where the current acoustic sensor bandwidth (40 kHz) supports detection of a broad range of species. The AS is designed to work with ACDC, where ACDC provides post-processing – if required – and data visualization.

The AS is composed of a number of technology layers. Its capability will be described at each layer, as capability varies significantly. At the topmost layer, the AS is:

- Single channel.
- Data is sampled at 16-bit resolution at rates up to 100 kHz.
- Acoustic bandwidth is variable from 2 – 40 kHz.
- Acoustic preamplifier gain is variable from 0 – 35 dB in 5 dB steps. The A/D also has adjustable input voltage ranges that are ± 1, 2, 5 and 10 Volts.
- Data is recorded on a standard Compact Flash card and recordings can be taken up to the limit of the card capacity.
- The AS provides a serial interface for a host controller interface with a basic command set to allow an external system to control it. The extensible command set includes control of the sample period, time setting, query of detection status, and power off. Where an external interface is not available the same interface is controlled by an AS host controller via a socket and onboard software. The AS host controller provides additional control and functionality over a serial user interface (terminal interface).
- The AS provides a serial user interface (terminal interface with text menus) for direct user access to manage modes, logs, data, etc.

Underlying the AS is modular technology with greater potential. It allows for processing of any number of data channels at any sample rate, using floating-point numbers. This includes both detection processing and recording. This software is written so that any data format can be supported via an appropriate module at the front of the processing stream. WAV, WAV64, and other formats supported by *libsndfile* are currently supported along with a number of DRDC proprietary formats, one PC-104 A/D, and various soundcards that are supported by standard API on MS Windows, OSX, and Linux. New data sources are regularly added.

B.6 DARB and DAS

The dynamic active reusable buoy (DARB) and the closely related dynamic active simulator (DAS) are valuable tools for experimentation. They are used to transmit pre-defined signals into the water. The DARB includes:

- A flotation system (buoy) which contains the electronics,
- On-board computer used to control the buoy and generate signals,
- Amplifier used to transmit signals,
- Batteries for power,
- GPS for location and time-synchronization, and
- Freewave modem for control and position updates.

The on-board software uses a simple terminal-driven menu interface to control the buoy. All commands are logged along with other important information (e.g. NMEA track). The NMEA track is also broadcast so that the buoy's position is known, when not in the menu.

The system operates in the following acoustic modes:

- Standby – no acoustic transmissions are sent.
- Synchronous – starts transmitting pulses at a specified offset from the minute and repeats at a specified interval (based on GPS time).
- Continuous – transmits a signal continuously.
- SubGPS – the buoy transmits NMEA strings using an underwater modem with a custom modulation scheme developed by Akoostix personnel for DRDC Atlantic. The scheme allows the transmission rate to be scaled, based on required range from 1 bps to 100 bps.

Signals can be WAV formatted files, or wave trains that are a serious of WAV formatted files with user-specific gaps between the files. Wave trains are defined to avoid the requirement to create large files containing mostly zeroes to fill in gaps.

The system also comes with a user manual [23] that can be used for more detailed information and instructions.

List of symbols/abbreviations/acronyms/initialisms

A/D	Analogue to Digital
ACDC	Acoustic Cetacean Detection Capability
ALI	Amplitude Line Indicator
API	Application Programming Interface
ARM	An incorporated technology company
AS	Acoustic Subsystem
ASCII	American Standard Code for Information Interchange
AUTEC	Atlantic Undersea Test and Evaluation Center
AUX	Auxiliary
CD	Compact Disk
CFAV	Canadian Forces Auxiliary Vessel
CFMETR	Canadian Forces Maritime and Experimental Test Ranges
CM	Configuration Management
CMAKE	A cross platform build tool
CR	Contract Report
CSA	Contract Scientific Authority
DARB	Dynamic Active Reusable Buoy
DAS	Dynamic Active Suitcase
DAT	DREA Data Format
DAT32	32-bit version of DAT
dB	decibel
DFO	Department of Fisheries and Oceans
DIFAR	Directional Frequency and Ranging
DISO	Departmental Individual Standing Offer
DND	Department of National Defence
DOxygen	An inline documentation generator
DRDC	Defence Research and Development Canada
DREA	Defence Research Establishment Atlantic
DSP	Digital Signal Processing/Processor
DTAG	Digital Tag

EADAQ	Environmental Acoustic Data Acquisition
ECUA	European Conference on Underwater Acoustics
El-Brg	Ellipse/Bearing Crossing (EB)
El-El	Ellipse/Ellipse Crossing
El-Hy	Ellipse/Hyperbol Crossing
ETI	Energy Time Indicator
FFT	Fast Fourier Transform
FFTW	Fastest Fourier Transform in the West
FM	Frequency Modulation
GPS	Global Positioning System
GRAM	Sonogram
HF	Human Factors
ICI	Inter-Click Interval
ID	Identification
IDL	Interactive Data Language
IDLDoc	IDL Documentation (similar to Doxygen)
IEEE	Institute of Electrical and Electronics Engineers
IP	Intellectual Property
ITAC	Integrated Tracking and Aural Classifier
kHz	Kilohertz
KM	Kliewer-Mertins (algorithm)
LAND	Lagrangian Ambient Noise Drifter
m	Metre(s)
MMOS	Marine Mammals and Ocean Science
MPA	Marine Protected Area
MS	Microsoft
NMEA	National Marine Electronics Association
NO	Number
NS	Nova Scotia
NUM	Number
NW	North-West
OPD	Omni-Passive Display

OSX	Operating System X (Ten) - Apple OS
PA	Project Authority
PAMGUARD	Passive Acoustic Monitoring Guard
PARB	Passive Acoustic Reusable Buoy
PC	Personal Computer
PC-104	An embedded PC standard
PWGSC	Public Works and Government Services Canada
R&D	Research and Development
RDS	Rapidly Deployable Systems
RMC	Royal Military College
ROC	Receiver Operating Characteristic
ROM	Rough Order-of-Magnitude
SNR	Signal to Noise Ratio
SONLIB	Sonar Library
SOR	Statement of Requirement
SPLIB	Signal Processing Libraries
SPPACS	Signal Processing Packages
SQL	Structured Query Language
SQLite	A lightweight SQL compliant database
STAR	Software Tools for Analysis and Research
STAR-IDL	IDL specific applications of STAR
SVN	Subversion
TBD	To Be Determined
TDOA	Time-Difference of Arrival
TI	Texas Instruments
TM	Technical Memorandum
TMAST	TTCP Multi-scale Active Sonar Technology
TOC	Table of Contents
UNIX	A computer operating system
USA	United States of America
WAV	Wave file format (i.e. .wav)
WAV64	64-bit Wave format