JOINT LOCALIZATION AND SEPARATION OF SPERM WHALE CLICKS

Paul M. Baggenstoss
Naval Undersea Warfare Center, Newport RI, 02841, p.m. baggenstoss@ieee.org

ABSTRACT

In this paper we consider the joint problems of separating and localizing sperm whale click trains. Click train separation is the single-sensor problem of grouping the clicks from each animal together when the clicks of more than one animal are present at a given sensor. Localization is the problem of localizing the animals based on the measurement of time delays of the same click events at multiple sensors. The two problems are inherently connected. We first consider the two problems independently using novel applications of statistical signal processing methods. For separation, we employ an algorithm inspired by the Viterbi algorithm from dynamic programming. For localization, we employ an algorithm inspired by the expectation-maximization (EM) algorithm. Finally, we use the two algorithms to "assist" each other in a joint localization/separation solution. We demonstrate the algorithm on real data.

SOMMAIRE

En cet article nous considérons les problèmes communs de séparer et de localiser des trains de clic de cachalot. La separation de train de clic est le problème de simple-hydrophone de grouper les clics de chaque animal ensemble quand les clics de plus d'un animal sont présents à une hydrophone donnée. La localisation est le problème de localiser les animaux basés sur la mesure du temps retarde des mêmes événements de clic aux sondes multiples. Le problème deux sont en soi relies. Nous considérons d'abord les deux problèmes employant indépendamment des applications de nouveaux des méthodes statistiques de traitement des signaux. Pour la séparation, nous utilisons un algorithme inspireé par l'algorithme de Viterbi de la programmation dynamique. Pour la localisation, nous utilisons un algorithme inspireé par l'algorithme de E-M. En conclusion, nous employons les deux algorithmes pour nous aider dans une solution du joint localization/séparation. Nous démontrons l'algorithme sur de vraies données.

1 CLICK TRAIN SEPARATION

1.1 Introduction and Problem Definition

When recorded at a single hydrophone, multiple sperm whale vocalizations are difficult to separate into the click-trains of the individual whales. Previous work has utilized various clues including spectral and temporal features and inter-click correlation as well as multi-sensor time-delay [1],[2],[3]. Based on the structure of the problem, involving many interrelated clues spread over time, we believe there is significant room for improvement through the application of dynamic programming. With dynamic programming, a globally best solution can be approximated through efficient time-recursive processing. Let us assume that a series of clicks has been received at a sensor. For the purpose of this discussion we define an arbitrary error metric $E_{i,j}$ as the result of comparing clicks i and j. It is not important for the discussion how the clicks are compared. We can assume that information about the time duration, amplitude, and spectral content plus a measure of inter-click correlation has been used to develop this error metric. The goal is to arrange clicks into groups or *chains*. If the inter-click error is measured only between adjacent clicks within a chain, then the total error is minimized when the clicks are properly grouped. Of course there must be a penalty for creating a new chain. Otherwise, one could place each click in a separate chain to minimize the errors. The total error for a *complete grouping* is the sum of the inter-click errors in each group plus the penalty value. The solution could be found by brute-force evaluation of all possible groupings, but in general this is computationally prohibitive. To obtain an efficient algorithm, we employ dynamic programming.

1.2 Dynamic Programming Algorithm

Dynamic programming is a means of solving problems, whose complexity would grow at an exponential rate with time if solved brute-force, recursively with linearly increasing complexity. For details, we refer the reader to the classic book by Bellman [4]. Although the problem we try to solve does not meet the requirements to be solved by dynamic programming, we use an algorithm inspired by dynamic programming. The goal of the algorithm is to group together clicks at a single sensor that come from a given whale and a given propagation path. Before we describe the algorithm, let us define the following terms. **Chain:** a set of associated sequential clicks. **Over-Complete Grouping:** a set of chains that make up a complete set including all clicks, but with duplicates. **Complete Grouping:** a set of chains that make up a complete set including all clicks with no duplicates. **Best Complete Grouping:** the complete grouping with the minimum error of all complete groupings.

The algorithm operates on the clicks received at a single sensor. Assume the algorithm has processed all clicks up to index $n - 1$ and has an *over-complete grouping* list of candidate chains and chain error values. To terminate the algorithm, it would be necessary to search the list for the *best complete grouping*. If the list of chains is not very large, and since the error metrics have already been computed and the totaled up for each chain, the problem is not computationally prohibitive (since we still have additional clicks to process, we are not going to terminate). We now process the remaining clicks.

To add click n, we do the following. First, we assume that detection n may not be a member of any existing chain so we add it as a "chain of one" with error value P and retain all existing chains. Second, we assume detection n may append to an existing chain. We copy all existing chains, thereby doubling the number of chains, and append click n to the end of each copy (subject to limits on click period). Having added click n, we then proceed to click $n + 1$. The list of chains can grow substantially with each added detection, in fact it grows exponentially. Before the number of chains gets too large to manage, it is necessary to pare down the list. The act of paring down the list we call a *collapsing search* because it collapses the list down to the *best complete grouping*.Note that there is no guarantee that the collapsing search performed on clicks up to n won't lose a chain that is part of the *best complete grouping* of clicks up to $n + 1$. To minimize the chance of losing such chains, we do two things. First, the collapsing search is performed at intervals of T click updates. Second, a set of parallel lists are maintained that perform the collapsing search also at intervals of T click updates, but a different time offset (phase). The use of multiple list phases is illustrated in figure 1.

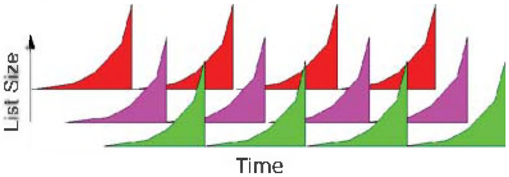

Fig. 1. Illustration of a multiple search phases.

The list size of each list phase grows exponentially until the collapsing search is performed (every T updates).

We performed an experiment to see the reduction in error rate as T increased. We performed a collapsing search at an interval of length T, and kept T phase lists, so one list was performing a search at each update. We used two test problems: (a) a benign problem with a single high-SNR whale with a direct and a reflected propagation path, and (b) a difficult case with up to three whales at low SNR. In each trial, we selected at random 12 consecutive click detections. Then we located the best complete grouping by brute-force search. We then ran the recursive algorithm, with the inter-click error metric described below, with collapsing search every T updates (and kept T phase lists). We considered it an error if the best complete grouping was lost. The error rate was the fraction of the trials in which the best complete grouping was lost. The results are shown in figure 2. In the benign problem the error rate dropped exponentially as T increased until no errors were found above a value of $T = 2$. In the difficult problem, the error rate decreased less rapidly and never reached the point where no errors were found. This is expected because the globally minimum solution is probably not "truth" in any case. In other words, solutions found using the recursive algorithm did not appear any better to the eye as the global best solution.

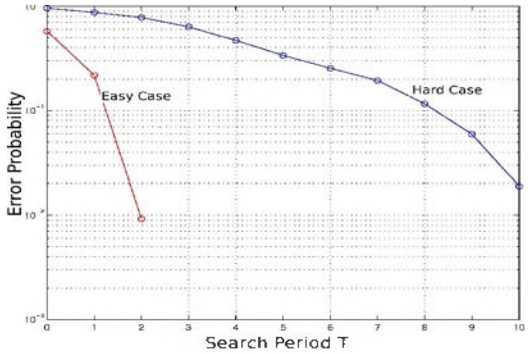

Fig. 2. Error rate as a function of search interval T. The benign case is the lower trace. There were no errors at T = 3 and higher.

1.3 Inter-Click Error Metric

We used a probabilistic inter-click error metric. This was accomplished by extracting 10 features from the click pair to be tested. Features included measures of spectral and temporal closeness, correlation measures, etc. We considered the binary decision: associated (H_1) versus not associated (H_0). For training, feature samples of H_1 were obtained from adjacent clicks from hand validated groupings, while feature samples of H_0 were obtained from random click pairs. A Gaussian mixture probability density function (PDF) estimate [5] was used to estimate the distribution under H_0 and H_1. The probabilistic error metric was obtained from the log likelihood ratio using the log likelihood ratio $E_{i,j} = -\log\{p(x|H_1)/p(x|H_0)\}$. This is the negative of the log-likelihood of the probabilistic test for H_1 vs. H_0. We used a new chain penalty value of -8 in all our experiments.

2 MULTI-SENSOR LOCALIZATION

We now describe a model-based whale localization algorithm. It is distinguished from existing algorithms in its use of the expectation-maximization (EM) algorithm and use of probabilistic "soft" association of click-pairs.

2.1 Introduction

The problem of localizing sperm whales using click-trains received at multiple sensors works primarily by measuring time delays between the click vocalizations received at multiple sensors and comparing with a propagation model. The task is made difficult by the problem of associating clicks, that is, knowing if clicks received at separate sensors are actually from the same click vocalization. This is made even more difficult by the existence of multiple propagation paths and the existence of multiple whales. While existing approaches solve the association problem by correlating click trains [6], [7], [8], [9], we seek to employ the E-M algorithm to sort out the association problem. We explore the advantages and disadvantages of the approach and test it using real data.

2.2 Algorithm Description

The EM algorithm [10] has been successfully applied to many problems where there is inherent association ambiguity. In the application of the EM algorithm to such problems, the problem can be regarded as a mixture density and the algorithm takes a particular form [5]. There is a fixed number N of data samples, and a fixed number M of probabilistic "models".

In our problem, a model is a potential whale location and a data sample is a measured 2-sensor inter-click time delay. Each data sample (inter-click time delay) is regarded as a possible statistical realization of one of the M solutions. Let there be set of NM click association probabilities, denoted by $w_{i,j}$, and representing the probability that data sample i is a realization of model j. The EM algorithm consists of 2 steps:

1) **The "E"-step:** Given the localization solutions, estimate the click pair association probabilities $w_{i,j}$, and solution weights α_j. The solution weights are a measure of validity of each solution.
2) **The "M" step:** Given the click association probabilities, estimate the parameters of each localization solution. Parameters include position and time delay variance.

The "E" and "M" steps are repeated until convergence. Solutions with low α_j are pruned. Notice that rather than associating one click to another, we associate measured click pairs to a source solution hypothesis. We allow all click pairs to exist even if they are false pairings. We add a special "error" localization solution located at the center of the sensor field and with a high location variance (σ_j^2). The algorithm should associate invalid click pairs to the "error" solution. The EM algorithm uses "soft" association - a solution is associated with a click pair with certain probability. It may have nonzero association with all click-pairs. The concept is illustrated in figure 3.

2.3 Algorithm Details

2.3.1 Solution Probabilistic Model

We assume that $\square_i$ is a Gaussian random variable with mean $T(z_j, s_i, r_i)$ and variance σ_j^2. Let $\square_i$ be the time delay measurement for click pair i. Let $L_{i,j}$ be the likelihood function value for click pair i and solution j

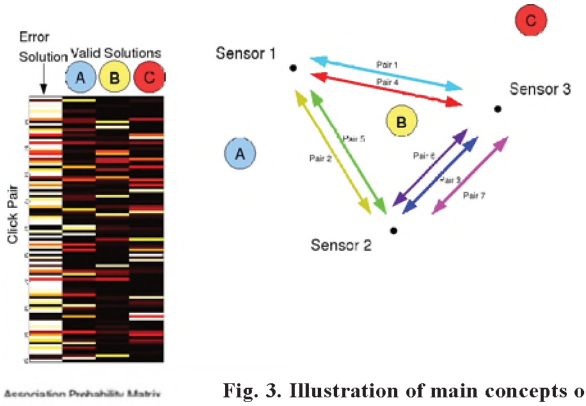

Fig. 3. Illustration of main concepts of EM algorithm.

On the left is the click-pair association probability matrix (CPAPM) represented as an intensity image. The Y-axis is the index of the click pair and the X-axis is index of the solution. On the right is a geographical representation of the sensors and solutions. The three solutions, A, B, and C are represented on the right in geographical position and on the left as columns of the CPAPM. The additional Error solution is shown as the first column. Each horizontal row of the CPAPM is the probability of a given click pair having been generated by a whale at each of the solutions, and sums to 1 over all solutions. Each vertical column can be regarded as a given solution's probability of ownership for the click pair. Those solutions with the largest column sums are associated with more click pairs, and are therefore more important. The EM algorithm alternatively updates solutions based on the weighted click pairs, and then updates the CPAPM.

defined by,

$$L_{i,j} = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left\{- \left[\tau_i - T(z_j, s_i, r_i)\right]^2 / (2\,\sigma_j^2)\right\}$$

where σ_j^2 is the time delay error variance for solution j, z_j is the current position vector for solution j, (s_i, r_i) is the sensor pair from which click pair i has been obtained, and T (z_j, s_i, r_i) is the model time delay.

2.3.2 Initialization

We detect individual clicks at each sensor, then create "click pairs" from every two-sensor pair of clicks (that can reasonably be associated with each other given the dimensions of the search area, the maximum range of reception, and the spacing of the hydrophones). Let there be M initial location solutions obtained by a grid-search using any model-based localization procedure. A large number of initial solutions can be used. The solution variances σ_j^2 can be initialized based on the grid-search quantization size. We include an "error" solution, located at the center with wide variance. Let α_j, $1 \leq j \leq M$ be the solution weights for the M solutions. They can be initialized to 1/M.

2.3.3 Solution weights and Click Pair Association Probability Matrix (CPAPM) Update (E-step)

The CPAPM is estimated as follows. The un-normalized CPAPM is computed as $\tilde{w}_{i,j} = \alpha_j\, L_{i,j}$ where i indexes the click pairs and j indexes the solutions. Normalization is then performed so that for each i, $\sum_j w_{i,j} = 1$.

Solution weights, α_j, are obtained by summing the CPAPM along each column (fixed j) to determine the effective number of click pairs associated with solution j, then normalizing so the solution weights sum to 1.

2.3.4 Solution Estimation Procedure (M-step)

The parameters of each solution include the time delay error variance σ_j^2 and the position vector z_j. These parameters are estimated by weighted maximum likelihood (ML) by maximizing

$$Q(z_j, \sigma_j^2) = \sum_{i=1}^{N} w_{i,j} \log L_{i,j}(z_j, \sigma_j^2)$$

over the parameters z_j, σ_j^2 . Space does not permit a detailed description of the maximization procedure, however, ML is a well-studied method [11].

2.3.5 Solution Error Ellipse

A well-known property of ML is that the statistical random error of the parameter estimates approximates Cramer-Rao lower bound matrix [11]. This is useful for drawing solution error bound ellipses on the geographical solution plot. Let C be the 4-by-4 error covariance matrix for the 4-dimensional parameter set $\theta = \{z_1, z_2, z_3, \sigma_j^2\}$ for a general location solution. The Cramer-Rao lower bound [11] matrix is given by $C = I^{-1}$ where

$$I_{p,q} = -E\{\partial^2 Q(\theta) / \partial p \partial q\}$$

where p and q represent components of θ. The solution error ellipses are contours of constant value of the inner product $\theta' I \theta$.

2.3.6 Recursion

The algorithm repeats the E-step and M-step until convergence. Solutions with very low solution weights are removed.

2.4 Algorithm Modifications

The algorithm as described above is a special case of the E-M algorithm for mixture densities. Some modifications may be necessary, however, which deviate from the E-M algorithm. In difficult problems, the majority of click pairs may be "invalid", that is they are time delays measured between two clicks that are not both from the same acoustic click event. Normally, these should get assigned to the "error" solution. Nevertheless, because they often have time delays matching a given model solution by chance, their existence causes significant problems in the convergence of the algorithm. The algorithm can be modified to effectively deal with the problem. First, by using a power in the exponent of (1) higher than 2, the distribution is no longer Gaussian, but the effect of outlier time delays is minimized. Second, we use *click-based* solution weights. In the unmodified algorithm, the solution weights are *click-pair based.* They are proportional to the effective number of click-pairs assigned to each solution since they are derived from the CPAPM. To remove the influence of invalid click pairs, we create *click-based* solution weights, based on a click detection association probability matrix (CDAPM). Let $\beta_{k,j}$ be the probability that detection k is assigned to solution j. We estimate $\beta_{k,j}$ by approximating the number of click pairs assigned to solution j containing click k. More precisely, $\beta_{k,j} = \sum_{i \in I_k} L_{i,j} \alpha_j$, where I_k is the set of click pairs containing click k. We then normalize $\beta_{k,j}$ in the same way as $w_{i,j}$. We estimate the solution weights α_j from $\beta_{k,j}$ instead of $w_{i,j}$.

3 COMBINING SINGLE AND MULTI-SENSOR ALGORITHMS

3.1 Assisting single-sensor separation with multi-sensor information

One way to assist the click-train separation is to develop a measure of the probability that two clicks, received at the same sensor, are from the same animal and same propagation path. This requires "support" from two additional clicks. Let clicks A_1 and B_1 be received at one sensor. If A_1 and B_1 are indeed from the same animal, then it is possible that these same two clicks have been received at another sensor. Denoting these two clicks as A_2 and B_2, we would find that the time delays (A_1 to A_2) and (B_1 to B_2) should match to a high degree of accuracy. With this in mind, we create the match measure

$$r_{i,j} = \sum_{m,n} e^{-(\tau_{i,m} - \tau_{j,n})^2 / (2\sigma_\tau^2)}, \qquad (3)$$

where $\tau_{i,m}$ is the time delay between click detections i and m and σ_τ^2 is a time delay error variance parameter. Indexes i, j represent A_1 and B_1. Indexes m, n represent all potential pairs A_2 and B_2. The search is limited to likely candidates for A_2 and B_2 based on time delay limits. An example of matrix $r_{i,j}$ is shown in figure 4. The information is utilized in click train separation by adding $-\log\ r_{i,j}$ to the error metric $E_{i,,j}$.

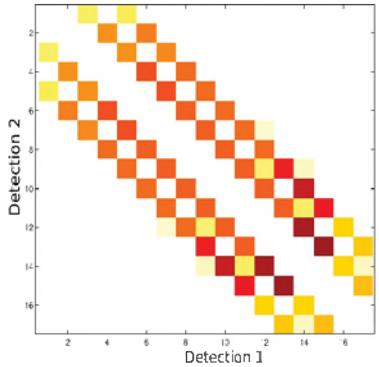

Fig. 4. Example of single-sensor inter-click error metric ri j based on multi-sensor information. Darker shades indicate good match. Even numbered clicks associate with even clicks and odd with odd. This is a situation where there is one animal and each direct-path click detection is followed by a reverberation detection. Use of $r_{i,j}$ prevents association of reverberation and direct-path clicks.

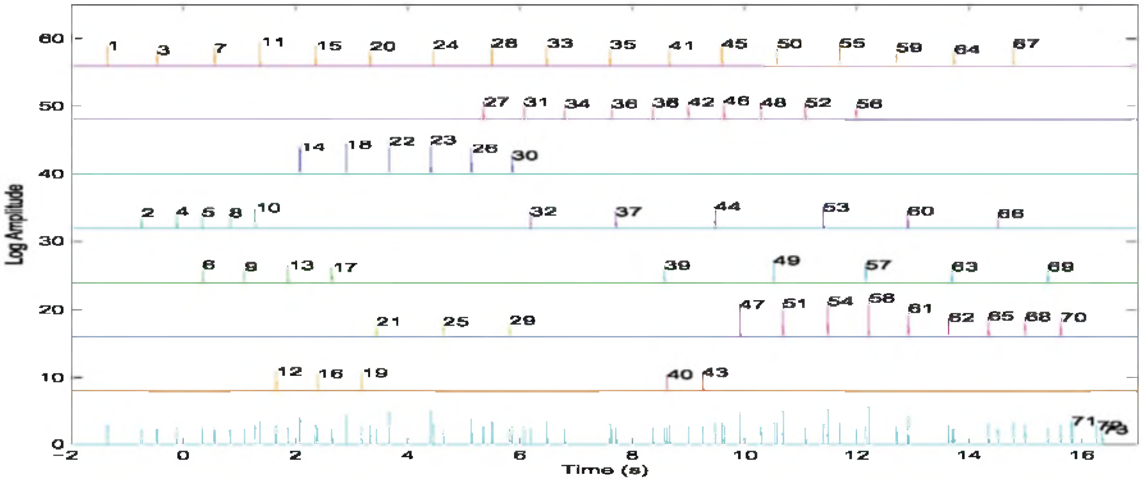

Fig. 5. Example of the sperm whale click train separation. Shown is the first 18 seconds of data from sensor 1 of 6. The lowest trace shows all clicks (log amplitude vs time). Separated click trains are displayed with vertical and/or horizontal separation. The presence of two nearby clicks of different amplitude gives the appearance of "crooked" clicks.

3.2 Assisting localization with single-sensor information

Going back to the example above, if click pair (A_1,A_2) is a valid pairing, then it stands to reason, there exists another pair (B_1,B_2) such that (a) A_1 and B_1 belong to the same click train, (b) A_2 and B_2 belong to the same click train, (c) pairs (A_1,A_2) and (B_1,B_2) have the same time delay. With this in mind, we create the click pair quality measure

$$q_i = \sum_{j \in J_i} e^{-(\tau_i - \tau_j)^2/(2\sigma_\tau^2)}, \qquad (4)$$

where J_i is the set of click pairs that are from the same single-sensor click train as pair i. Click pairs with low values of q_i can be eliminated.

4 ALGORITHM SUMMARY

```
1) Make click detections on each sensor.

2) Develop a list of potential inter-sensor click pairings
   with time delay $\tau_i$ subject to constraints on time delay.

3) For each sensor:
     a) For every pair of clicks (i, j).
          i)  calculate single-sensor inter-click error metric
              $E_{i,j}$ (section 1.3).
          ii) calculate inter-click match metric based on
              multi-sensor time delays $(r_{i,j}$ in equation 3).
              Add $-\log r_{i,j}$ [illegible] $E_{i,j}$.
     b) Run the click separation algorithm described in
        section 1.
4) From the single-sensor separation results, develop the
   click pair metric $q_i$ in equation (4).

5) Run the localization algorithm described in section 2.3:
     a) Use a grid-search to find a set of potential whale
        locations $z_j$. Initialize the solution variances
        $\sigma_j^2$ to reflect the time-delay variance
        corresponding to the grid-search quantization.
        Initialize the solution validity measures $\alpha_j$ to
        a constant $\alpha_j = 1/M$ where M is the number of
        solutions.
     b) Compute CDAPM $\beta_{k,j}$ and validity
        probabilities $\alpha_j$ (section 2.4). Eliminate locations
        with low $\alpha_j$.
     c) For each j, maximize (2) over $\sigma_j^2$ and $z_j$
        and compute the Cramer-Rao lower bound
        covariance for $z_j$ from which the error ellipses
        can be drawn. Go to step 5-b, repeat.
```

5 EXPERIMENTAL RESULTS

We utilized sperm whale data from bottom mounted sensors from the Monaco 2005 workshop. The data consisted of two sets which can be described as "easy" and "difficult". Data set 2 (easy), was a 25-minute run where a single whale was present with high SNR. This data set was used to produce figure 4 and the lower trace in figure 2. Data set 1 (difficult) was a 20 minute run with two and possibly three whales. This data set was used to produce figures 5, through 9 and the upper trace in figure 2.

5.1 Click Separation Results

Using "difficult" data set, we obtained single sensor click-train separation results using the described recursive algorithm assisted by multi-sensor information using (3). Results from the first 18 seconds of sensor 1 are typical and are shown in figure 5. A total of 70 click events were grouped into 11 chains ranging from 2 to 17 clicks in length. From the first 18 seconds of data from 6 sensors, the single-sensor groupings were used to calculate q_i (equation 4). A total of 9564 multi-sensor click pairs and associated time-delays were created. By setting a lower bound of $q_i = 0.8$, it was possible to eliminate 8643 of the 9564 click pairs. An initial set of 287 initial solutions were found by looking for local minima in the $x - y - z$ grid-search. Using the "easy" data set, the algorithm provided two very strong and tight solutions, the direct path.

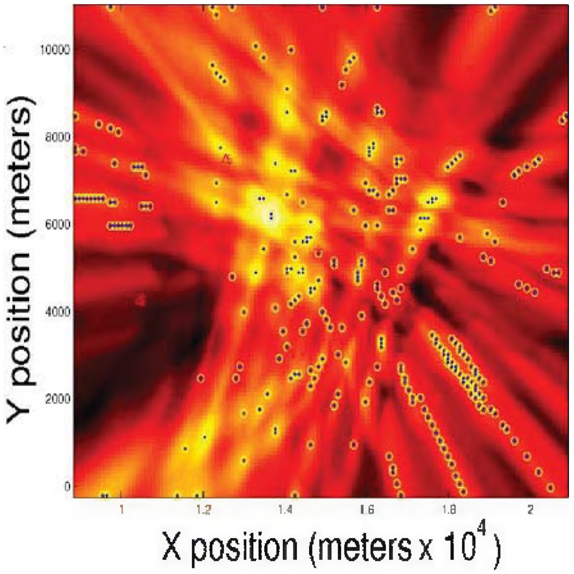

Fig. 6. Initial set of 287 solutions on a geographical plot. Initial solutions were local maxima of the function 1 searched over z1, z2, z3 in a grid. Image intensity was maximized over depth (z3).

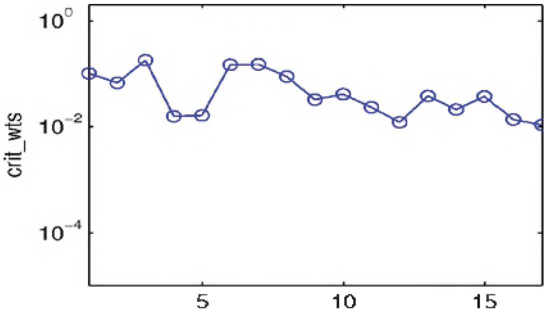

Fig. 7. Solution weights after 100 iterations. Solutions 3, 6, and 7 have appreciable weight.

5.2 Localization Results

The first 18 seconds of the "difficult" data set provides a good illustration of algorithm behavior. The data of figure 5 was one of six sensor inputs provided to the localization algorithm. The algorithm was able to positively identify the locations of two whales in the sensor range. After 100 iterations, the solution weights were as shown in figure 7. Solutions 3, 6, and 7 have appreciable weight. Solutions 6 and 7 were clearly whales and solution 3 was a potential whale, although the solution was ambiguous. The solution error ellipses are shown in figure 8. Error ellipses were obtained from the CR bound analysis of equation (2). The error ellipse for solution 3 resembles a line, an indication that click pairs from only two sensors are available, or the positioning is unfavorable for exact localization. Ellipses for the valid whale solutions 6 and 7 are very tight. The clicks associated with each solution can be determined using the CDAPM. In Figure 9 we show all clicks, those associated with whale 1, and those associated with whale 2. The clicks have been time-aligned in accordance with the time delay from each sensor to each solution's position.

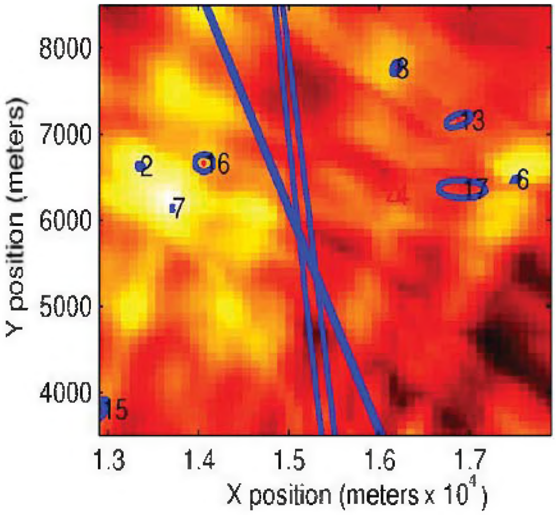

Fig. 8. Solution error ellipses. The diagonal line is actually an ellipse for solution 3 indicating wide error in one direction probably due to having information from just 2 sensors.

5.3 General comments and future work

The click separation algorithm worked perfectly in all 18-second segments of the "easy" data set, separating the direct from the multi-path chains. In the "difficult" data set, it appeared to work very well although it is difficult to validate the results without positive localization solutions for all clicks. By comparing localization and separation results from the first 18-seconds (i.e including figures 5 and 9) we were able to find no clear cases where clicks were improperly grouped together, but many cases where click trains were separated into smaller pieces. Much of this behavior can be controlled by the "new chain penalty" P.

The localization algorithm performed very well provided good initialization solutions are provided. A fine grid search was necessary causing the algorithm to be very slow, requiring up to five minutes to process the solution for 18 seconds of input data. However, as the number of solutions is reduced to a handful, a single E-M iteration requires only a fraction of a second. In the "difficult" data set, good whale locations were obtained in most of the 18-second intervals, and whales could be tracked throughout most of the data set, but it was not uncommon that invalid solutions dominated. It was clear that additional work was needed to eliminate these invalid solutions, possibly by using information from the separation algorithm.

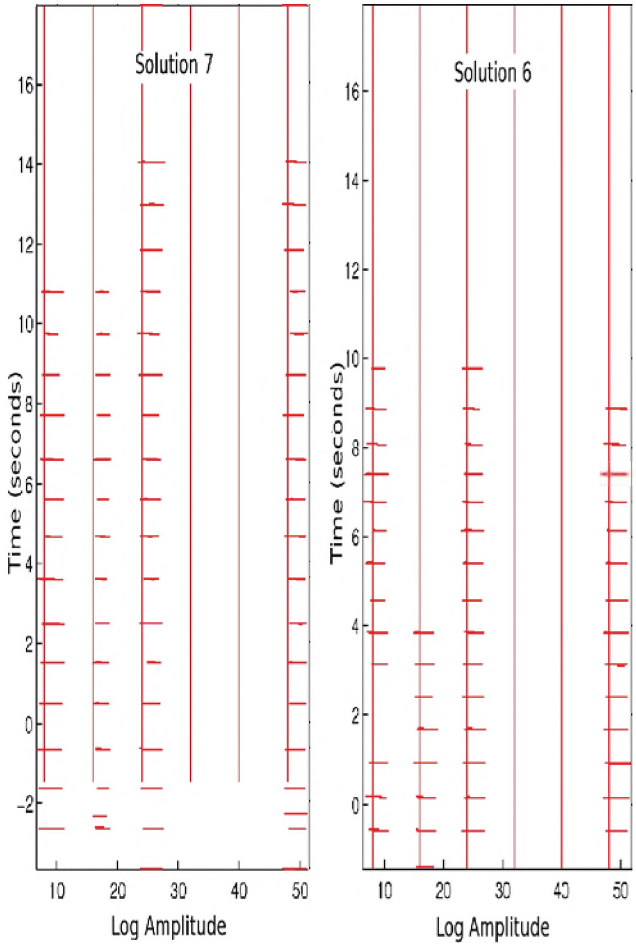

Fig. 9. Example of the sperm whale localization. Log-amplitude on the X-axis and time on the Y-axis. For clarity, the clicks from various sensors are separated horizontally by artificially adding a different bias to the X-axis value of each sensor. Left panel: the clicks with high CDAPM value for solution 7 are shown. Sensors time axes have been time aligned with according to the respective solution to align the clicks. Right panel: clicks with high CDAPM value for solution 6 are shown. These two solutions represent less than half of the available clicks. The rest were associated with the er^{r}or solution or a variety of invalid solutions, some of which appear in figure 8.

6 CONCLUSIONS

We have provided a snapshot of our research effort directed toward joint localization and separation of sperm whale click trains. We have presented two novel algorithms, one for separation, and one for localization, and demonstrated them on real data. The click-train separation algorithm uses an approach inspired by dynamic programming and efficiently seeks the globally best click grouping. The localization algorithm uses the EM algorithm to do “soft” association of click pairs to solutions. The algorithms are loosely tied together by (a) utilizing multi-sensor time delay information to assist the separation algorithm and (b) eliminating false click pairings in the localization algorithm by the use of information from the click separation algorithm. Because the localization algorithm uses the EM algorithm to make “soft” click and click-pair assignments, it may prove superior to existing approaches in many situations, such as when a large number of solution ambiguities exist.

REFERENCES

[1] M. van der Schaar, E. Delory, A. Catal, and M. Andr, “Neural network-based sperm whale click classification,” *J. Mar. Biol. Ass. U.K,* vol. 87, pp. 35–38, 2007.

[2] T. Ura, R. Bahl, M. Sakata, J. Kojima, T. Fukuchi, J. Ura, K. Mori, T. Nakatani, Y. Nose, and H. Sugimatsu, “Development of AUV-based system for acoustic tracking of diving sperm whales,” in *Proceedings of OCEANS 2004 (Kobe,Japan),* pp. 2302–2307, November 2004.

[3] R. Bahl and T. Ura, “Automatic real-time segregation and classification of multiple vocalizing sperm whales,” *Seisan-Kenkyu Bimonthly Journal of IIS, University of Tokyo,* vol. 55, pp. 61–64, May 2003.

[4] R. E. Bellman, *Adaptive Control Processes.* Priceton, New Jersey, USA: Princeton Univ. Press, 1961.

[5] R. A. Redner and H. F. Walker, “Mixture densities, maximum likelihood, and the EM algorithm,” *SIAM Review,* vol. 26, April 1984.

[6] R. Morrissey, J. Ward, N. DiMarzio, S. Jarvis, and D. Moretti, “Passive acoustic detection and localization of sperm whales *(Physeter Macrocephalus)* in the Tongue of the Ocean,” *Applied Acoustics,* vol. 67, pp. 1091–1105, Nov 2006.

[7] S. Jarvis and D. Moretti, “Passive detection and localization of transient signals from marine mammals using widely spaced bottom mounted hydrophones in open ocean environments,” in *International Workshop on the Application of Passive acoustics in Fisheries,* (NUWC), 2002.

[8] P. White, T. Leighton, D. Finfer, C. Powles, and O. Baumann, “Localisation of sperm whales using bottom-mounted sensors,” *Applied Acoustics,* vol. 67, pp. 1074–1090, 2006.

[9] C. O. Tiemann and M. B. Porter, “Automated model-based localization of sperm whale clicks,” in *OCEANS 2003. Proceedings,* vol. 2, pp. 821– 827, Sept 2004.

[10] R. Hogg, J. McKean, and A. Craig, *Introduction to Mathematical Statistics.* Upper Saddle River, NJ: Pearson Prentice Hall, 2005.

[11] S. Kay, *Modern Spectral Estimation: Theory and Applications.* Prentice Hall, 1988.