Article

Multi-Algorithm Swath Consistency Detection for Multibeam Echosounder Data

By Brian Calder, Center for Coastal and Ocean Mapping & NOAA-UNH Joint Hydrographic Center, University of New Hampshire, Durham, NH 03824, USA.
E-Mail: brc@ccom.unh.edu

Abstracts

It is unrealistic to expect that any single algorithm for pre-filtering Multibeam Echosounder data will be able to detect all of the "noise" in such data all of the time. This paper therefore presents a scheme for fusing the results of many pre-filtering sub-algorithms in order to form one, significantly more robust, meta-algorithm. This principle is illustrated on the problem of consistency detection in regions of sloping bathymetry. We show that the meta-algorithm is more robust, adapts dynamically to sub-algorithm performance, and is consistent with operator assessment of the data. The meta-algorithm is called the Multi-Algorithm Swath Consistency Detector.

Résumé

Il n'est pas réaliste de s'attendre à ce que chaque algorithme pour le pré-filtrage des données des échosondeurs multifaisceaux puisse détecter tous les «bruits» de ces données, à chaque moment. Cet article présente donc un projet de fusion des résultats de nombreux sous-algorithmes de pré-filtrage afin de constituer un seul méta-algorithme, bien plus robuste. Ce principe est illustré avec le problème de la détection de cohérence dans les régions à bathymétrie en pente. Nous montrons que le méta-algorithme est plus robuste, que sa dynamique s'adapte au fonctionnement du sous-algorithme et qu'il est cohérent avec l'évaluation des données effectuée par l'opérateur. Le méta-algorithme est appelé «détecteur multi-algorithme de cohérence dans une bande».

Resumen

No es realista esperar que cualquier simple algoritmo, para filtrar previamente los datos procedentes de un Sondador Acústico Multihaz, podrá detectar todo el "ruido" contenido en dichos datos todo el tiempo. Este documento presenta pues un esquema que utilizará los resultados de muchos sub-algoritmos de pre-filtrado para formar uno solo, significativamente más robusto, un meta-algoritmo. Este principio queda demostrado en el problema de la detección de una coherencia en regiones de batimetría de retroceso. Mostramos que el meta-algoritmo es más robusto, se adapta de forma dinámica al funcionamiento de un sub-algoritmo y es coherente con la evaluación de los datos por el operador. El meta-algoritmo es denominado Detector de Coherencia de Multi-algoritmos por Bandas.

1 Introduction

There have been many approaches to automatic detection of outliers in bathymetric or hydrographic data [1-13], most of which are based on some idea of consistency of the data. The methods proposed all attempt to identify "outliers" in data by some robustness method, and fall into essentially two camps: detection based on spatial context in a ground-fixed local coordinate system, or detection based on beam-to-beam comparisons of data in the body-frame of the MBES itself. The algorithms then "flag" the data considered to be outliers so that they are not considered further (CUBE [12,14] is an exception to this rule.) However, while each of these algorithms has some features to recommend it, all of them attempt to solve all of the problems, all of the time. While a noble goal, it is unrealistic to expect one algorithm to solve everything consistently. As an alternative, we consider here the benefits of using multiple simple sub-algorithms that each identify a part of the problem, with the aim of utilising all of these in concert to attempt to solve the whole problem robustly. That is, can we use ideas of multi-sensor fusion [15, 16] to harness simple sub-algorithms into a more robust meta-algorithm?

In principle, a data fusion scheme has much to recommend it. (We consider here a "data-level" fusion in which data derived measures are expressed as probabilities and are combined, or fused, before any decision is made about them.) Each sub-algorithm only has to solve a part of the problem, and therefore can be made more targeted as to its opinion of the data (that is, its probabilistic description of the problem) so long as it expresses neutrality in an appropriate manner when the feature being sought does not occur. So long as there is a sufficient diversity of opinion among the sub-algorithms, the failure of one will not generally outweigh the opinions of the others, resulting in a significantly more robust meta-algorithm overall. In a compute-resource limited environment – e.g., in real-time applications – the sub-algorithms could be applied sequentially rather than in parallel until such time as either a sufficiently solid opinion emerges, or the meta-algorithm runs out of time and has to make a decision anyway. The only significant difficulty is finding a means to express the results of the sub-algorithms in an appropriate manner, and manage the fusion of results. These are the issues motivating the current work.

To focus the discussion, we consider here as an example the problem of Multibeam Echosounder (MBES) bottom detection in areas of heavily asymmetric bathymetry (Figure 1). In this case, the heavy slope of the seabed engenders at least three mechanisms that cause bottom detections to fail. Firstly, the grazing angle of the beams can be very low, resulting in lower signal-to-noise ratio (SNR). Secondly, the outer beams may observe only watercolumn before the combination of spreading and absorption loss result in insufficient energy return to make an adequate detection. In this case, the most energy generally comes from the primary sidelobe of each beam observing the orthogonal return of the outgoing energy from the nearest point of the seabed, resulting in clusters of "noise" points at about the

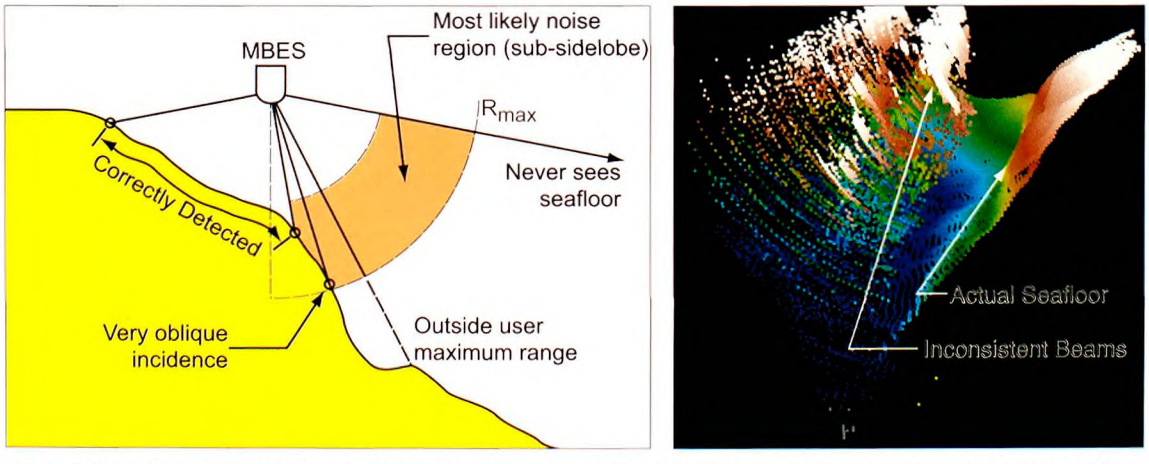

Figure 1: Schematic and example data for the downhill problem. The difficulties are due to very oblique incidence on heavy slopes, and user-specified maximum ranges. Note the spatial coherency of returns indicated in the example data (vertical scale is 1x here). Data from Valdez, AK, courtesy of the NOAA Ship Rainier.

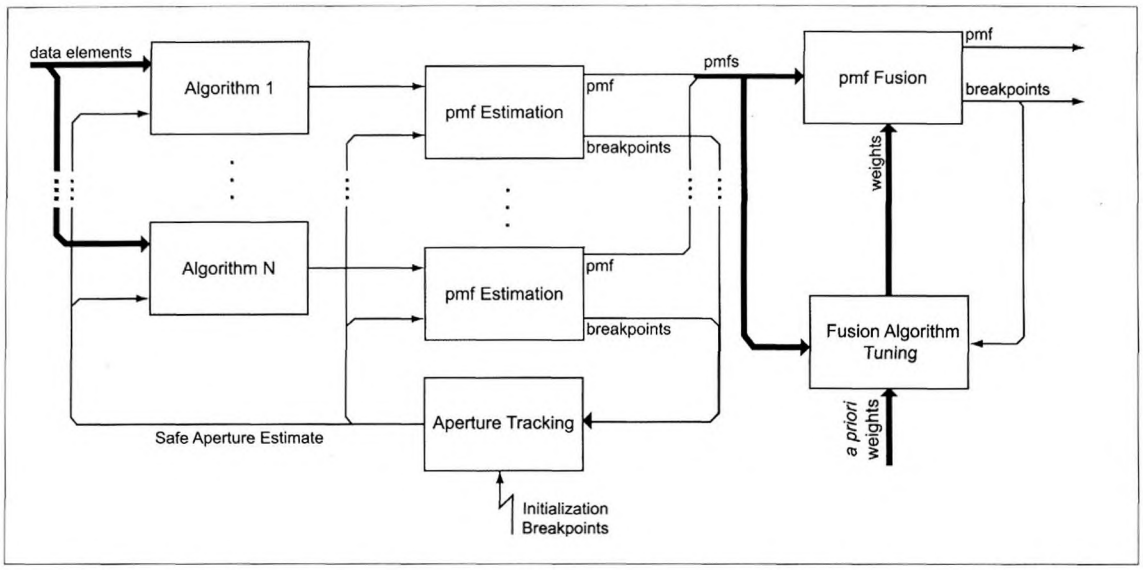

Figure 2: Data flow-path for the MASC'D meta-algorithm. Bathymetric data enters at the left, and is processed in parallel by the basic sub-algorithms which each predict a probability mass function (pmf) for breakpoint location at each edge of the swath. Fusion of the pmfs results in an overall conclusion of most likely breakpoint. Feedback of fusion result and breakpoints allows the meta-algorithm to tune the fusion weights and safe aperture tracker, respectively.

same range from the MBES. Thirdly, in order to artificially increase the ping rate of the MBES so that the highest density data is collected in the hydrographically more significant shoal areas, it is common in this situation to limit the MBES maximum range. While this improves the along-track resolution and hence rate of advance for the same coverage requirement [17], it also means that the primary return in many beams will not be observed before the limit, leading to similar misdetection problems as outlined previously.

These problems manifest themselves as a "breakpoint" in each swath of soundings (i.e., a point in the swath at which the behaviour of the data breaks from one regime to another), with consistent data on one side (typically towards nadir) and radically inconsistent data on the other. (Each swath most often has a breakpoint on one side of nadir, but can have a breakpoint on each side in certain conditions.) The role of the sidelobes in formation of the problem makes the "noise" data spatially coherent and hence difficult to process by typical methods both algorithmic and manual, leading to great expenditure in time and effort. The objective here, therefore, is to develop multiple sub-algorithms to detect this breakpoint, and then use the fusion engine to bring these to a consistent, robust solution. The intent is to allow this to be used as a pre-filter for further processing, and therefore the remediation method is to simply apply conventional "do not use" flags against the data.

The remainder of this paper describes this Multi-Algorithm Swath Consistency Detector (MASC'D) meta-algorithm, with particular emphasis on the fusion system. The meta-algorithm is illustrated on data collected in steep bathymetry and it is shown to subjectively match what a human operator might do in the same circumstances. We then conclude with some perspectives on utilisation of the meta-algorithm, and extensions to other scenarios.

2 Theory

The MASC'D meta-algorithm consists essentially of two parts: probability mass function (pmf) generation based on the data, and fusion of the pmfs. In the general scheme, the pmf estimates summarise some features of the data of interest, e.g., consistency of data points, likelihood of object presence, probability of correct bottom detection, etc., and the fusion scheme ties the estimates together, taking into account the relative current strengths of the sub-algorithms. In the example here, the feature being summarised is the location between the portion of the swath with consistent data (generally about

the nadir) and that with inconsistent data, potentially on either side of nadir within one swath. The overall flow of data within the meta-algorithm is shown in Figure 2.

2.1 Safe Aperture Tracker

A number of the sub-algorithms used need to estimate some basic properties of the data (e.g., slope, noise variance). Given that the data has not been "cleaned" or otherwise processed, it is important that, in addition to suitably robust techniques, the sub-algorithms are given an idea of where in the swath is likely to be unaffected by the noise problems. This is called a "safe aperture", and is represented as a start and stop beam number, $\check{b}(p)$ and $\hat{b}(p)$ respectively, for each ping (with $0 \le \check{b}(p) < \hat{b}(p) < N$ for N beams, numbering from port to starboard). In order to provide a scheme that adapts to protect the aperture's integrity as much as possible, the region between the breakpoints is segmented as shown in Figure 3. The tracking algorithm is structured so that the segmentation maintains a safety margin around the safe aperture as long as possible, then reduces the safety margin before finally starting to collapse the estimation aperture. The tracker uses the information provided by the fusion system to determine the likely location of the safe aperture on the next ping, which is then fed back to the sub-algorithms. Details of the tracking algorithm can be found in Appendix A.

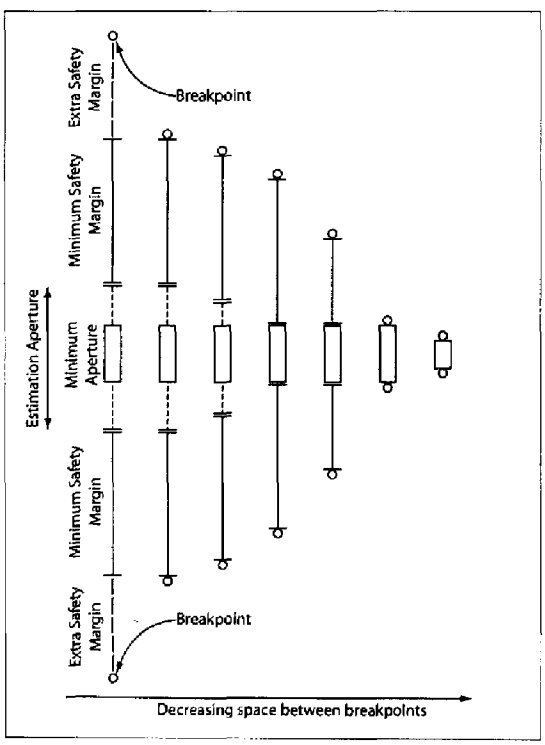

Figure 3: Progression of safe aperture tracker as the space between breakpoints decreases. The tracking algorithm attempts to preserve the safety margin for as long as possible before starting to encroach on the user-specified minimum estimation aperture.

2.2 Sub-Algorithms

Each of the sub-algorithms is designed to construct, from the raw input data and the current estimate of the safe aperture, an estimate for each beam of the likelihood of that beam being the breakpoint between consistent and inconsistent data. That is, they compute the probability $P_L(p,b|\alpha)$ and $P_R(p,b|\alpha)$ for the port (left) and starboard (right) side, respectively, interpreted as the probability of beam b at ping p being the breakpoint, given the sub-algorithm α. These are defined over beams $0 \le b < \check{b}(p)$ and $\hat{b}(p) \le b < N-1$, respectively, to avoid any problems of identification.

The structure of the meta-algorithm is designed to allow a variable number of sub-algorithms to be added or removed from the decision-making process, potentially on a dynamic basis. The sub-algorithms are designed to identify different features of the data so that they are as near as possible conditionally independent given the data, and therefore have independent information in their probability mass function (pmf) estimates to contribute to the fusion scheme. In the example here, four sub-algorithms were used:

- Grazing Angle. This sub-algorithm computes the slope of the swath using a suitably robust filtering method and combines this with the normal minimal grazing angle allowed for the MBES when the seafloor is flat in order to determine a slope-corrected minimal grazing angle. The probability distribution is computed based on how close the beams are to this angle. This sub-algorithm is simple and efficient, and although it generally underestimates the extent of the problem it is very good at indicating the likely location of the problem area as a vaguely informative adjunct to the other methods.
- Variance Excess. This sub-algorithm computes an *a priori* estimate of the expected variance in the depth solution for each beam, using the method of Hare *et al.* [18] as implemented for the CUBE

algorithm [12]. The sub-algorithm then computes a sample estimate of the variance observed in the beam over a small window of pings and uses the former as a threshold on the latter to develop a sequence of binary indicator variables indicating whether the sample estimate is above or below the theoretical prediction that should be valid if the data is behaving as we would normally expect. A probability distribution for the likely breakpoint location is developed from these by treating them as Bernoulli trials with fixed probability of success, so that the sum to the left and the sum to the right of each potential breakpoint should both be binomially distributed. The product of these two probabilities, suitably normalised, provides a distribution with higher mass where a breakpoint is likely.

- Range Limit. This sub-algorithm starts from the observation that the inconsistent data is more frequently clipped at the maximum range gate specified by the user than it is in consistent data. By applying a threshold to the data by range, therefore, the sub-algorithm can judge closeness to maximum range as a percentage of that maximum range, and thereby develop a binary indicator variable for likely inconsistency. These are processed as for the Variance Excess method to develop the required probability estimate for each beam.
- Turning Angles. This sub-algorithm starts from the observation that consistent data changes depth relatively slowly, even on slopes. The offset vector from one beam depth solution in the body-frame of the MBES to the next beam in sequence should therefore be similar to that of the next offset vector in sequence. Consequently, if we consider the turning angle between these two vectors, consistent data should have a very peaked angular distribution while inconsistent data should have a more closely uniform distribution. This distinction is a simple hypothesis test in the theory of circular statistics [19] and therefore directly forms a sequence of binary indicator variables for processing as before.

None of the sub-algorithms, which are described in full in Appendix B, are expected to be perfect in the sense that they determine the "correct" breakpoints for all pings and all datasets. All that is required is that they get the computation right most of the time and fail in different ways when they do fail. The fusion of the disparate results then uses the consistent probability mass of the correct sub-algorithms to outweigh those that are inconsistent, providing a more robust final solution.

2.3 Algorithm Fusion

2.3.1 Fusion Structure

Let the *a priori* probabilities of sub-algorithm reliability be $P_{\alpha,L}(p)$ and $P_{\alpha,R}(p)$ for port and starboard sides, respectively. Then, the fusion of pmf estimates resolves to computation of the total probability:

$$\tilde{P}_L(p,b)=\sum_{\alpha\in A}P_L(p,b|\alpha)P_{\alpha,L}(p)$$

$$P_L(p,b)=\tilde{P}_L(p,b)\Big/\sum_b\tilde{P}_L(p,b) \quad 1)$$

$$\tilde{P}_R(p,b)=\sum_{\alpha\in A}P_R(p,b|\alpha)P_{\alpha,R}(p)$$

$$P_R(p,b)=\tilde{P}_R(p,b)\Big/\sum_b\tilde{P}_R(p,b) \quad 2)$$

where A is the set of all sub-algorithms. The fused breakpoints for the ping are then computed as:

$$b_F^L(p)=\arg\max_{0\le b<\delta(p)}P_L(p,b) \quad 3)$$

$$b_F^R(p)=\arg\max_{\delta(p)\le b<N-1}P_R(p,b) \quad 4)$$

and are then used to feed the "safe" aperture tracker, (11)-(13), and to flag the data outside the breakpoints as "not for use".

2.3.2 Computation of Algorithm Probabilities

The sub-algorithm probabilities, $P_{\alpha,L}(p)$ and $P_{\alpha,R}(p)$, have to be adaptively computed since the performance of each sub-algorithm is expected to change over time and space depending on the structure of the noise observed. The probability of each sub-algorithm giving a correct decision is a combination of an overall prior probability, $P_\alpha(0)$ assessed *ad hoc* by the operator, and a likelihood, $\lambda_\alpha^s(p)$ that depends on the data being processed. (Here, $s\in\{L,R\}$ is the side being processed.) With these definitions,

$$P_{\alpha,s}(p)=P_\alpha(0)\lambda_\alpha^s(p) \quad 5)$$

A feedback solution is utilized to carry out the adaptation, based on the concept of balanced penalties and bonuses, Figure 4. Formally, the likelihood of each sub-algorithm (i.e., the likelihood that the sub-algorithm is performing correctly and assessing the breakpoint position reliably) is assessed at the end of each ping cycle, and is updated as:

$$\lambda_\alpha^s(0)=1 \quad 6)$$

$$\lambda_\alpha^s(p+1)=\min_{\{0,1\}}\max\left[\lambda_\alpha^s(p)-\pi_\alpha^s(p;\cdot)+\beta_\alpha^s(p;\cdot)\right] \quad 7)$$

where the penalty function is given by:

$$\pi_\alpha^s\left(p;\lambda_\alpha^s(p),P_s(p,b|\alpha),b_F^s(p),w_{\max}\right)=\lambda_\alpha^s(p)w_{\max}\left(\max_b P_s(p,b|\alpha)-P_s\left(p,b_F^s(p)|\alpha\right)\right) \quad 8)$$

and the bonus function by:

$$\beta_\alpha^s\left(p;\lambda_\alpha^s(p),c_\tau\right)=c_\tau\left(1-\lambda_\alpha^s(p)\right) \quad 9)$$

A hard-limiter,

$$\min_{\{a,b\}}\max f(x)=\min\left\{\max\left\{f(x),a\right\},b\right\}=\max\left\{\min\left\{f(x),b\right\},a\right\}$$

is used to ensure that the results remain valid probabilities.

The penalty function $\pi_\alpha^s(p;\cdot)$ is designed to penalize sub-algorithms that have significant mass far from the final fusion point $b_F^s(p)$, while only mildly penalising sub-algorithms that have very diffuse probability functions, Figure 5. This allows the use of "vague but informative" methods, such as the Grazing Angle sub-algorithm, which always give an estimate that is mostly correct, but are never very certain about what the true answer is. An immediate corollary is that this method heavily penalizes sub-algorithms that are over-confident in the precision of their predictions. Given the properties of the $P_s(p,b|\alpha)$, it is immediately evident that $w_{\max}$ is a scalar on how much of a modification to attempt at any stage (i.e., how bad is "bad"). The bonus function, $\beta_\alpha^s(p;\cdot)$, is designed to drift the likelihood back to its nominal steady state of $\lambda_\alpha^s(p)=1$ in the absence of further penalties. If no other penalties are applied, the recursion of (7) can be solved to show that choosing

$$c_\tau=1-\exp\left\{(\tau_R+1)^{-1}\ln(\varepsilon/w_{\max})\right\}\ (\varepsilon/w_{\max}<1) \quad 10)$$

allows the bonus to recover the likelihood from $1-w_{\max}$ to $1-\varepsilon$ (for some suitably small ε such that $\varepsilon/w_{\max}<1$ so that a valid solution exists) in τ_R pings. Choosing τ_R is significantly simpler than trying to choose an appropriate c_τ directly.

2.3.3 Choice of Algorithm Prior Probabilities

The choice of sub-algorithm prior probabilities is essentially subjective. In the experiments here, they were selected based on experience with the reliability of each sub-algorithm, but were set in very approximate fashion so that the Grazing and Turning Angle sub-algorithms had $P_\alpha(0)=1.0$ and the Variance Excess and Range Limit sub-algorithms had $P_\alpha(0)=0.5$. These reflect the (operator) assessment that the former pair are generally reliable everywhere, while the latter pair are occasionally significantly in error about their breakpoint assessments. It is in theory possible to apply another adaptive system to tune these at a much slower rate – for example at the scale of a survey line, or over a survey area – but this has not been pursued further in this case. Although subjective, it is not expected that these assessments should have to change from location to location. The prior probability is, by definition, an assessment of the reliability of the sub-algorithm before any data is observed and is more a description of the nature of the sub-algorithm than its behaviour on particular data.

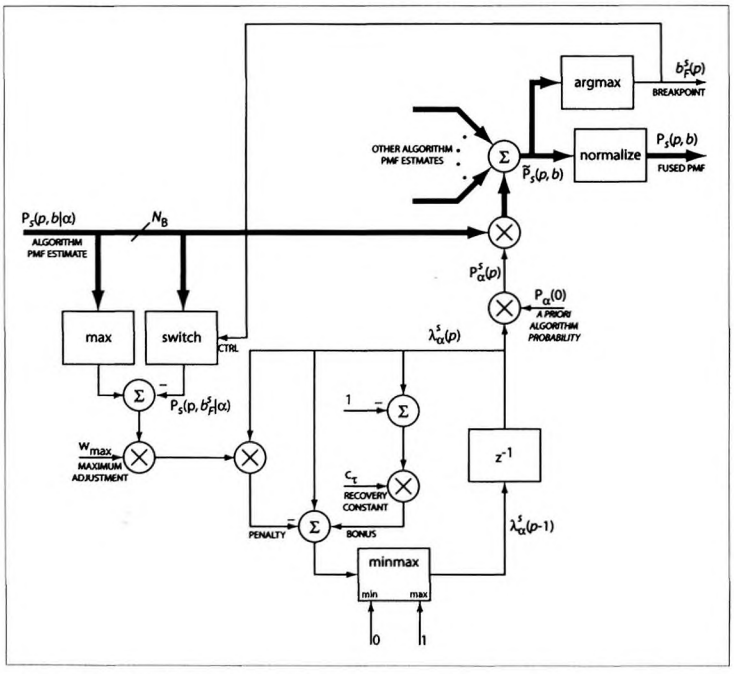

Figure 4: Fusion feedback system for per sub-algorithm likelihood probabilities (i.e., the probability that the sub-algorithm is operating correctly at the current step). The system of penalties and bonuses adjust the estimated probability of correct operation based on the past performance of the sub-algorithm on the data.

3 Examples

The example here uses data from the south-west Pacific around the

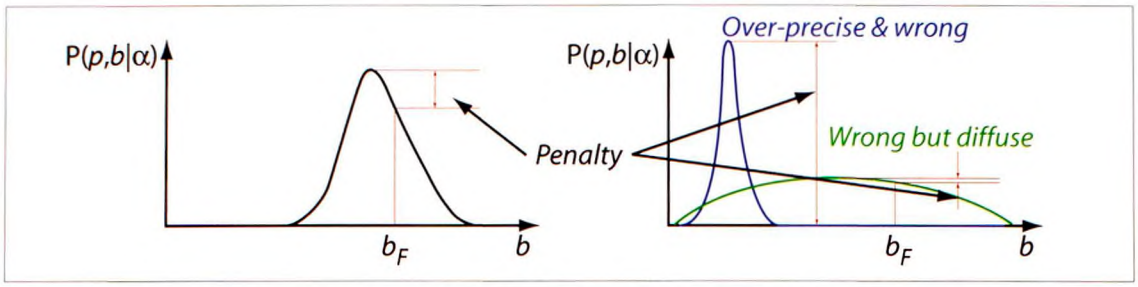

Figure 5: Structure of the fusion penalty function. The function is designed to penalise sub-algorithms that have significant mass far from the fused breakpoint solution.

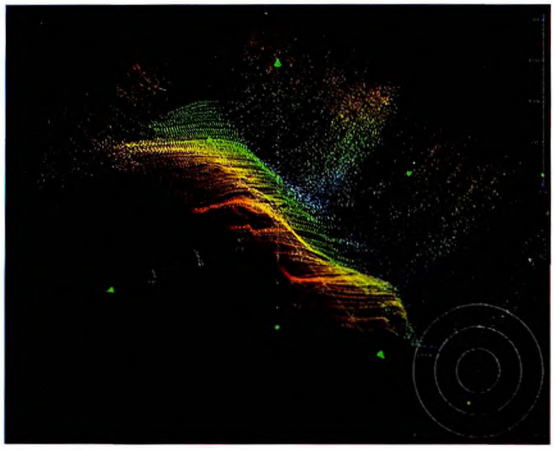

Figure 6: Section of example data at a depth of 60-100 m. Data is colored by depth and clearly shows the heavy "noise" induced by a maximum range $\hat{r}(p) = 300\,\mathrm{m}$. Starboard is to the right of the image.

island of Rota (14° 09'N, 145° 12'E, part of the Mariana Islands group in Western Micronesia). The data was gathered with a Reson 8101 MBES in approximately 60-100m depths, and clearly show evidence of swath breakpoints on both sides of the swath, Figure 6. The problems in this example occur mostly on the starboard side of the swath, but occasionally break-through on the port side as shown here. The estimated slope of the surface is shown in Figure 7, indexed in pings from start of line. The line is approximately 9 min., or 2.2 km long, with a ping rate of ~ 3Hz. In all cases, the code was run with aperture tracker safety margin of $\tau_S = 10\,\mathrm{beams}$, and aperture recovery time of $n_{\max} = 20\,\mathrm{pings}$, a minimum aperture of $\pm 5°$ and a nominal aperture of $\pm 30°$.

3.1 Component Algorithms

The example file was processed separately with each sub-algorithm (for illustration), and then with the fusion scheme in place. (Note that algorithm parameters are explained fully in the detailed descriptions of the sub-algorithms in Appendix B.) For the Grazing Angle sub-algorithm, an estimation window width of W_G = 10 pings was used, with maximum and minimum acoustic opening angles of $\check{\theta} = -75°$ and $\hat{\theta} = 75°$. The breakpoint locations predicted, $(\check{b}_G(p), \hat{b}_G(p))$, are shown in Figure 8, and the corresponding flagged data is shown in Figure 9. The breakpoints picked by the sub-algorithm are generally good, although some examination of the filter-

Figure 7: Estimated slope of the example line. "Slope" is not well defined in variable cross-track environments, but only a rough estimate is required here.

ing results shows that the sub-algorithm does underestimate the extent of the problem, failing to flag many of the points that are affected by the noise problem. This sub-algorithm does, however, provide basic information on the likely location of the breakpoints, and very rarely has any difficulties with outliers.

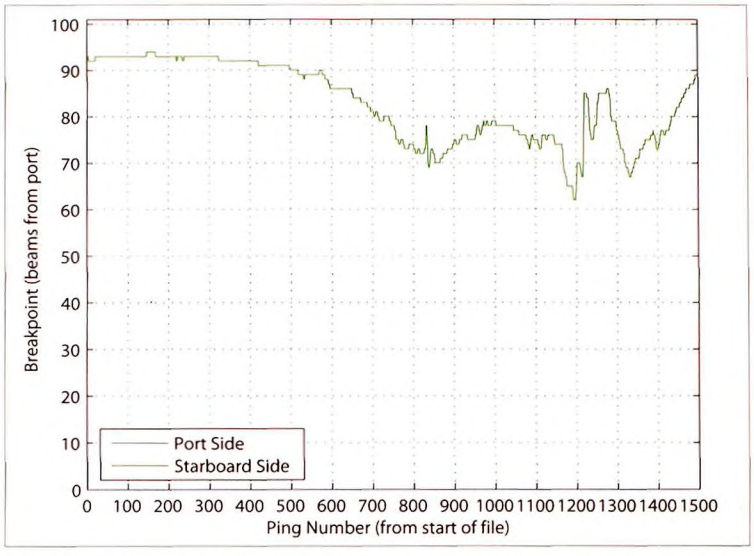

Figure 8: Estimated breakpoints using the Grazing Angle sub-algorithm. Note that the port side breakpoint is fixed at zero since there is only one estimate of slope, and it is always to starboard in this example.

The breakpoints predicted from the Variance Excess sub-algorithm, $(\hat{b}_V(p), \hat{b}_V(p))$, are shown in Figure 10. (The sub-algorithm used an estimation window width of W_V = 10 pings, and hypothesis testing significance α = 0.05 for the χ^2 testing.) The breakpoints here show significantly finer detail than those of Figure 8, primarily because of the greater subtlety in the analysis of the data. Both sides of the swath are processed, clearly showing the "fail safe" nature of the sub-algorithm where no problem exists. The sub-algorithm has predicted more of the swath to be affected by the problem than the previous sub-algorithm, which are more in keeping with the expected values observed from the data. In areas of slight slope, Figure 11, the sub-algorithm corresponds well with what a user might do, although it can be seen to slightly underestimate the extent of the problem on the starboard side of the swath. This in itself is not a very significant difficulty, since this is intended to be a pre-filter. However, the sub-algorithm is also observed to over-estimate the problem when there is very significant slope or very rapid changes in bottom configuration, Figure 12, essentially because the model of expected variance does not currently take these effects into account. This is a more significant problem since this data, pre-flagged, will never be seen by any subsequent algorithm. This example emphasizes the importance of having multiple sub-algorithms available to reach consensus in the fusion stage of processing.

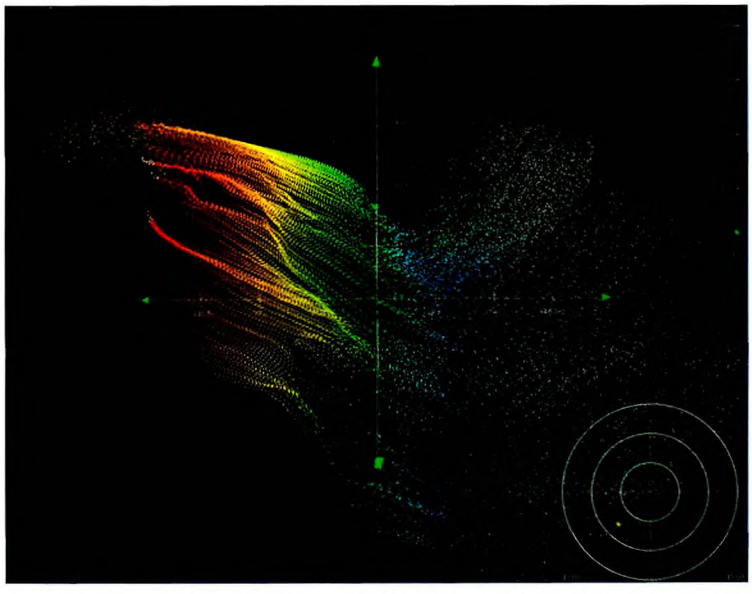

Figure 9: Example of flagging corresponding to breakpoint of Figure 8, constructed by the Grazing Angle sub-algorithm. The starboard side of the swath is to the right; flagged soundings are shown in gray.

The breakpoints computed from the Range Limit and the Turning Angle sub-algorithms show complementary properties to that of the Variance Excess sub-algorithm, with the Turning Angle sub-algorithm appearing more stable with respect to variability in slope. (The Range Limit sub-algorithm was run with estimation window width W_R = 3 pings, range scaling constant k = 0.20 and detection threshold of τ = 0.80. Significance for the Bernoulli trial testing

was α = 0.05; the Turning Angle sub-algorithm was run with estimation window width W_T = 5 beams, and height H_T = 5 pings with hypothesis testing significance α = 0.05. Choosing a larger area of operation in this case is important to ensure that the large-sample assumption of (32) is valid.)

3.2 Fusion Algorithm

The fusion scheme was configured with $w_{\max} = 0.5$, $\tau_R = 20$ pings and *a priori* sub-algorithm probabilities $P_0(G)=1.0$, $P_0(T)=1.0$, $P_0(V)=0.5$ and $P_0(R)=0.5$. The resulting likelihood estimates are shown in Figure 13, which shows different behaviors for the four sub-algorithms depending on their characteristics. The Grazing Angle sub-algorithm fares reasonably well when the data has relatively low slope at the start of the line, but is more heavily penalized as the slope and roughness increase. This is primarily due to the preponderance of mass at the breakpoint in $P_R(p,b|G)$ which is a weakness as the breakpoint moves further towards the nadir. The Turning Angle sub-algorithm, on the other hand, is seen to be penalized only occasionally to any significant extent, and then recovers back to nominal performance in almost τ_R pings. This is partly because the sub-algorithm is preferred in the *a priori* probability assessment, but primarily because it is generally reliable while the Range Limit and Variance Excess sub-algorithms show occasional runs of significant over-estimation of breakpoint. This effect is clearly evident in their likelihood estimates, which are frequently penalized, but do show a complementary nature to that of the Turning Angle sub-algorithm, as the fusion algorithm shifts probability mass between the components.

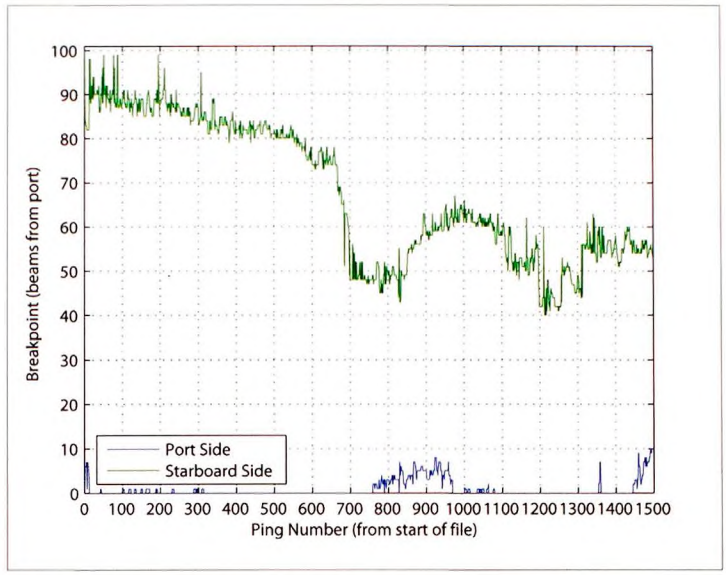

Figure 10: Estimated breakpoints using the Variance Excess sub-algorithm. The estimates are typically good, although some noise spikes are still observed (e.g., around ping 200) and the sub-algorithm occasionally over-estimates the extent of the breakpoints.

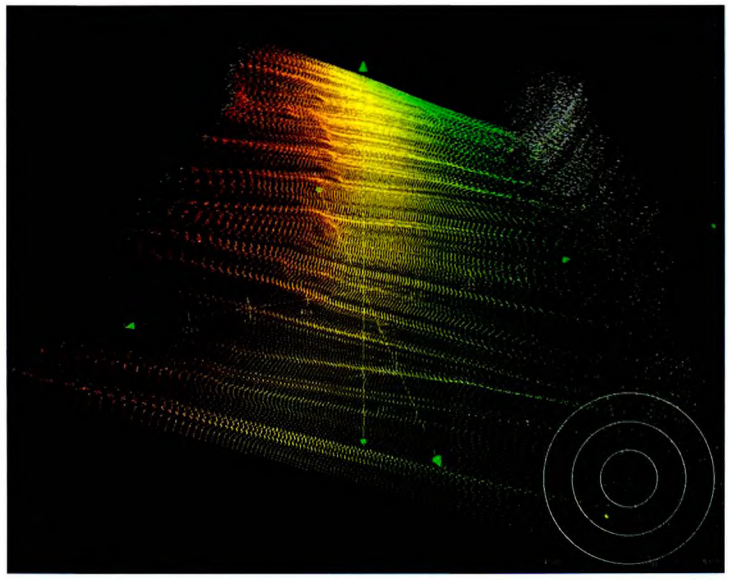

Figure 11: Example of flagging corresponding to breakpoints of Figure 10. The performance is reasonable in relatively shallow areas, with only a few "noise" points allowed through. Slight under-estimation of breakpoint, like this, is not a very significant problem since this is only intended as a pre-processing step for more powerful algorithms.

An example of the fused breakpoint tracks is given in Figure 14 for the steep and rough section towards the end of the example data; Figure 15 shows typical pmf estimates in the same region. The robustness of the fusion scheme with respect to the outliers from the Variance Excess sub-algorithm is evident, primarily because of the success of the Turning An-

gle sub-algorithm in combination with the supporting evidence from the Range Limit sub-algorithm. In the lower-slope conditions at the start of the example data, Figure 16, similar noise tolerance is observed, although another problem appears, visible around pings 70-75. Here, the integer nature of the breakpoint fusion (caused by being forced to choose one particular beam as the breakpoint) means that where there are two sub-algorithms of almost equal information, the meta-algorithm may vacillate between two closely spaced beams leading to some "chatter." In general, however, the difference of one beam in the breakpoint is not of significance.

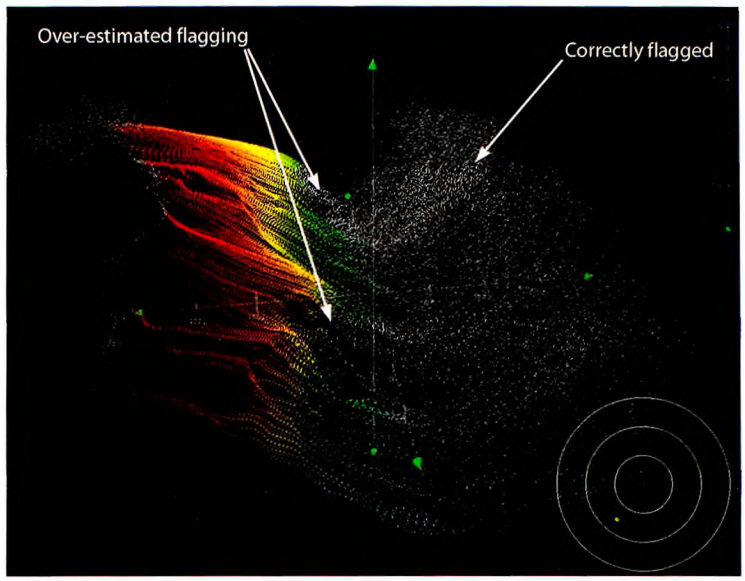

Figure 12: Example of flagging by Variance Excess sub-algorithm in steep slopes and rapidly changing bathymetry.

4 Discussion

The meta-algorithm presented here clearly detects the inconsistencies in the example data, and in general does so at the positions that would be indicated by a human operator (although such indications are, of course, very subjective). The complexity of the sub-algorithms are not particularly high, although there are a number of parameters to determine which can have significant effect on the performance of the meta-algorithm. Methods to reduce or "harden" the parameter choice (i.e., make them estimatible physical parameters rather than arbitrary constants) are currently under investigation. The Grazing Angle, Variance Excess and Turning Angle sub-algorithms have well established theoretical support; the Range Limit sub-algorithm is more *ad hoc* (although *ad hoc* in the strict sense of the term). This makes the parameter choice for the Range Limit sub-algorithm problematic, although in the experiments reported here only the maximum range scale was found to be particularly sensitive. Too low a value results in insufficiently robust detection of the breakpoint; too high a value results in over-estimation of the inconsistent region's extent and subsequent over-flagging. It may be possible to improve on this situation by estimating the variance of the ranges with respect to the maximum range using the grazing angle sub-algorithm's estimate of the breakpoint as something suitably "safe".

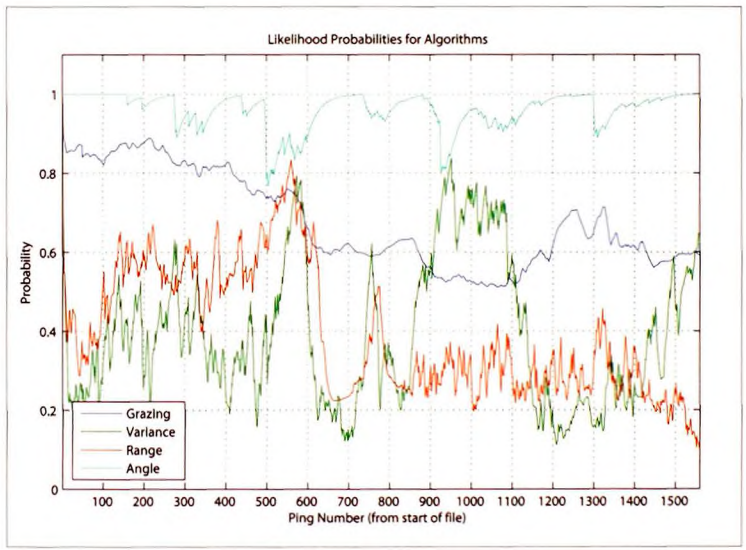

Figure 13: Sub-Algorithm likelihood estimates during processing. The Turning Angle sub-algorithm makes occasional mistakes, but recovers over time; the Range Limit and Variance Excess sub-algorithms are frequently penalized but still contain some information, while the Grazing Angle sub-algorithm survives by being suitable vague in its predictions.

In this experiment, only four sub-algorithms have

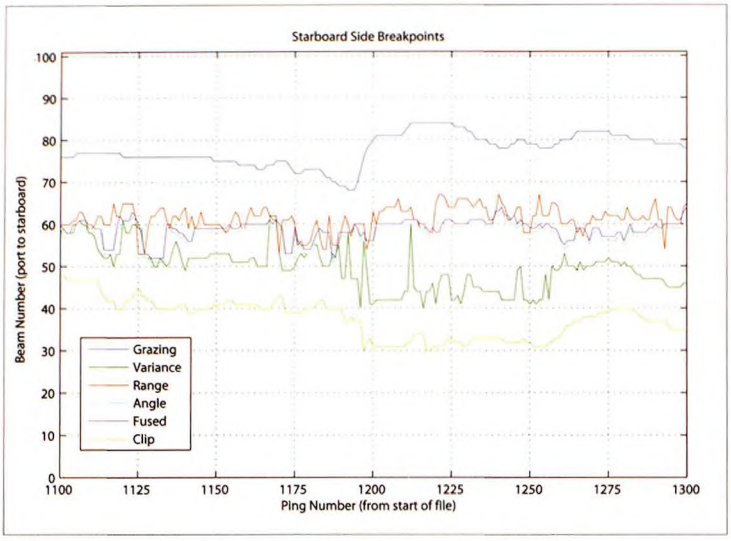

Figure 14: Example of fusion track in steep and rough environment towards the end of the example line. The fusion system observes that the Variance Excess sub-algorithm has inappropriately overestimated the starboard breakpoint relative to the other evidence, and reduces the sub-algorithm's likelihood accordingly (c.f. Figure 13). (Here, "Clip" refers to the safety aperture, as given by (11).)

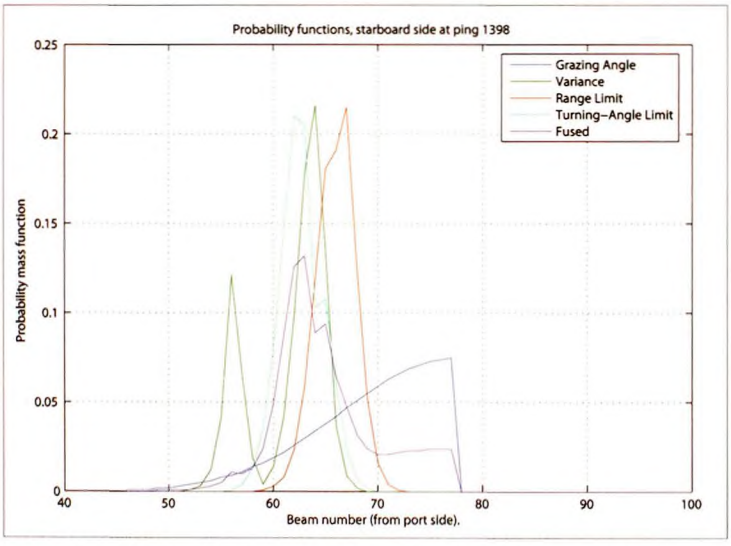

Figure 15: Example of fusion system pmf estimates under normal circumstances. The fusion algorithm favors the Turning Angle and Grazing Angle sub-algorithms (c.f. the likelihoods in Figure 13), resulting in a multimodal pmf which still has its primary mode in the appropriate location.

been investigated. There is an outstanding question of whether four is enough, or if the performance would continue to improve with more opinions added to the mix. In theory, so long as the new sub-algorithms being added provide independent information on the problem at hand, the fused solution should be better. However, since all of the sub-algorithms work from the same original data, there is probably a limit to the number of independent opinions that can be developed. While it could be envisioned that another sub-algorithm might be added to look at, e.g., evidence for acoustic interference from another active source, it is more likely that significant advantage would be derived from extending the scheme to look at other aspects of the overall data quality assessment problem as outlined below. In application of the same basic fusion system to other areas, however, the balance of sub-algorithms might be different.

We have considered here only concurrent fusion of the sub-algorithms (i.e., all of the sub-algorithms are run (theoretically) in parallel, and all of the information is fused in one step) since this is an off-line application; the meta-algorithm is, however, sufficiently fast as to require no special computational techniques to keep up with data collection rates in this case. In more compute resource limited environments such as the real-time processor for an on-line MBES capture system, computational load may become a significant issue. (Irrespective of the speed of the processing system, there will always be trade-offs between the various tasks required; in a restricted environment such as an AUV, these may be significant.) In this case, it would be straightforward to reconfigure the meta-algorithm to compute the sub-algorithms sequentially starting with $\mathbf{argmax}_{\alpha} P_{\alpha,s}(p-1)$ and then in descending order of probability, until sufficient evidence was amassed to make a reliable decision. This would allow the meta-algorithm, on average, to respond more quickly and with less computational load, although it would require some restructuring of the fusion feedback measurement since the pmfs

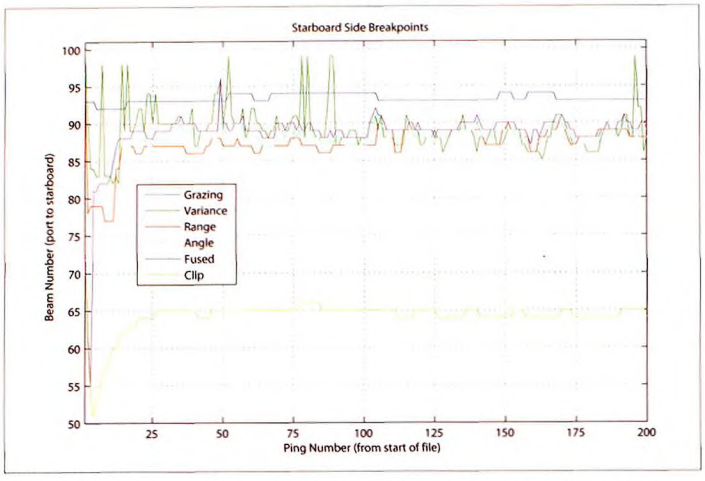

Figure 16: Example of fusion track in smooth environment towards the start of the example line. The fusion algorithm generally ignores spike effects (e.g., from the Variance Excess sub-algorithm around samples 80-85) but the integer nature of the decision point can result in some breakpoint jitter (e.g., around samples 70-75) in otherwise smooth data.

for all sub-algorithms would not all be available for penalty construction.

The configuration here is a specific example of a more general problem with inconsistency of swath data, particularly where the data is being monitored in real-time as it is being captured. Many observed problems with field data can be traced to an under-appreciation of the level of data quality that can theoretically be achieved with the systems being used. If software was available that could predict the performance of a sonar system given the real-time environmental conditions and then quantitatively rate the current data according to this prediction, it would provide valuable real-time feedback for operators as to the likely quality of the data being collected. One potential method for implementing such a system would be to continually monitor a number of different conditions using schemes such as the sub-algorithms used here, which are specifically designed to look for particular problems. Appropriate fusion of the individual schemes should lead to an overall quality indicator per beam, which would then be readily summarized for the user. By extension, schemes which included other sources of information (such as multiple passes with the same sonar) might be possible, and would add significantly to the variety of data quality problems that could be addressed. Although presently only a sketch, a scheme of this kind might then form the basis of a realtime integrated quality control mechanism that had the potential to assist in diagnosis of problems, as well as providing indication to the operator in a timely manner.

5 Conclusions

Expecting one algorithm to detect all issues with all datasets is unrealistic; it is much more likely that an effective solution can be built if several sub-algorithms can be used in concert, each one solving a part of the problem well. We have shown here that fusion of multiple sub-algorithms is feasible in the context of swath-based MBES data processing, and that the fusion algorithm can be made to adapt automatically to the current performance of the sub-algorithms being fused.

The proposed meta-algorithm is inherently modular so that sub-algorithms can be added or removed so long as they generate results that are in a form compatible with the fusion engine. In addition, the fusion scheme is sufficiently general so that although in this case the target is breakpoint detection, it would be possible to adapt the meta-algorithm to more general consistency or data quality assurance tasks; in this case an adjustment to the bonus/penalty scheme would be required for meta-algorithm adaptation.

On the example data, it is clear that the simple sub-algorithms proposed can be used to detect the breakpoints, and that they complement each other in their failure modes, a factor that highlights the benefits of the fusion scheme. The flags applied automatically to the data are analogous to those (subjective) decisions that a human operator would make. Future directions include more sub-algorithms, more robust data fusion and better methods for adaptive update of the sub-algorithm prior probability functions.

Acknowledgements

The support of NOAA grant NA05NOS4001153 and ONR grant N00014-00-1-0092 are gratefully acknowledged. The data used in the examples was supplied by the Captain and Crew of the NOAA Ship RAINIER, the Captain and Crew of the NOAA Ship

FAIRWEATHER and Scott Fergusson, NOAA Pacific Islands Fisheries Science Center.

References

[1] M. Gourley and D. Dodd. HIPS: Hydrographic information processing system. Technical Report CARIS White Paper 21, Universal Systems Ltd., Fredericton, NB, Canada, 2000.

[2] D. W. Caress and D. N. Chayes. New software for processing sidescan data from sidescan-capable multibeam sonars. *Proc. IEEE, 2:997--1000*, 1995.

[3] D. W. Caress and D. N. Chayes. Improved processing of Hydrosweep DS multibeam data on the R/V Maurice Ewing. *Mar. Geophys. Res.*, 18:631–650, 1996.

[4] J. E. Hughes Clarke, L. A. Mayer, and D. E. Wells. Shallow water imaging multibeam sonars: A new tool for investigating seafloor processes in the coastal zone and on the continental shelf. *Mar. Geophys. Res.*, 18:607–629, 1996.

[5] M.Gourley and K.DesRoches. Clever uses of tiling in {CARIS/HIPS}. In *Proc. 2nd Int. Conf. on Shallow Water Survey*, Portsmouth, NH, 2001. Center for Coastal and Ocean Mapping & NOAA-UNH Joint Hydro. Center.

[6] C. Ware, L. Slipp, K. W. Wong, B. Nickerson, D. E. Wells, Y. C. Lee, D. Dodd, and G. Costello. A system for cleaning high volume bathymetry. *Int. Hydro. Review*, 69:77–94, 1992.

[7] H. P. Varma, M. Boudreau, M. McConnel, M. O'Brien, and A. Picott. Probability of detecting errors in dense digital bathymetric data sets by using 3D graphics combined with statistical techniques. *Lighthouse*, 40:31–36, 1989.

[8] J. Eeg. On the identification of spikes in soundings. *Int. Hydro. Review*, 72:33–41, 1995.

[9] Z. Du, D. E. Wells, and L. A. Mayer. An approach to automatic detection of outliers in multibeam echosounder data. *Hydro. J.*, 79:19–25, 1996.

[10] N. Debese. Use of a robust estimator for automatic detection of isolated errors appearing in bathymetry data. *Int. Hydro. Review*, 2:32–44, 2001.

[11] N. Debese and P. Michaux. Détection automatique d'erreurs ponctuelles présentes dans les données bathymétriques multifaisceaux petits fonds. Proc. *Canadian Hydro. Conf.*, Toronto, Canada, 2002. Can.Hydro. Soc.

[12] B. R. Calder and L. A. Mayer. Automatic processing of high-rate, high-density multibeam echosounder data. *Geochem., Geophys. and Geosystems (G3), DID 10.1029/2002GC000486*, 4(6), 2003.

[13] G. Canepa, O. Bergem, and N. G. Pace. A new algorithm for automatic processing of bathymetric data. *IEEE J. Ocean. Eng.*, 28(1):62–77, 2003.

[14] B. R. Calder. Automatic statistical processing of multibeam echosounder data. *Int. Hydro. Review*, 4(1):53–68, 2003.

[15] I. Bloch. Information combination operators for data fusion: A comparative review with classification. *IEEE Trans. Sys., Man, and Cyber. (A)*, 26(1):52–67, 1996.

[16] D. L Hall and J. Llinas. An introduction to multisensor data fusion. Proc. *IEEE*, 85(1):6–23, 1997.

[17] NOAA Office of Coast Survey. *NOS Hydrographic Surveys Specifications and Deliverables*. NOAA National Ocean Service, Silver Spring, MD, 2003.

[18] R. Hare, A. Godin, and L. A. Mayer. Accuracy estimation of Canadian swath (multibeam) and sweep (multitransducer) sounding systems. Technical report, Canadian Hydrographic Service, 1995.

[19] K. V. Mardia. *Statistics of Directional Data*. Academic Press, 1972.

Biography

Dr. Brian Calder is a Research Assistant Professor at the Center for Coastal and Ocean Mapping and NOAA/UNH Joint Hydrographic Center at the University of New Hampshire. He graduated MEng and PhD in Electrical and Electronic Engineering from Heriot-Watt University in Edinburgh, Scotland in 1994 and 1997 respectively, but was subsequently seduced into sonar signal processing for reasons that are now obscure. His research interests have previously covered speech recognition, texture analysis, object detection in sidescan sonar, high-resolution sub-bottom profiling, simulation of forward-looking passive infrared images, acoustic modeling and pebble counting. Currently, they revolve around the application of statistical models to the problem of hydrographic data processing. He is a member of the IEE, IEEE, AGU and Hydrographic Society of America, and an Associate Editor for the IEEE Journal of Oceanic Engineering.

Appendix A. Implementation of the Safe Aperture Tracker

In order to provide some resilience against spikes in the breakpoints the safe aperture tracker is implemented in a feedback loop, modeled after a leaky peak-detector circuit. Let the breakpoint determined after data fusion be $b_F^L(p)$ or $b_F^R(p)$ (port and starboard side, respectively). If this value moves towards nadir from either side, the tracker reacts quickly; after the breakpoint returns towards the outside of the swath, the tracker slowly returns to normal at a rate determined by the user. After some manipulation, this idea resolves to iteration of the equations:

$$11)\quad \hat{b}(p)=\begin{cases} b_F^R(p) & b_F^R(p)<\hat{b}(p-1) \\ \hat{b}(p-1)+c_d\left(b_F^R(p)-\hat{b}(p-1)\right) & b_F^R(p)\geq\hat{b}(p-1) \end{cases}$$

$$12)\quad \check{b}(p)=\begin{cases} b_F^L(p) & b_F^L(p)>\check{b}(p-1) \\ \check{b}(p-1)+c_d\left(b_F^L(p)-\check{b}(p-1)\right) & b_F^L(p)\leq\check{b}(p-1) \end{cases}$$

$$13)\quad c_d=\exp\left\{n_{\max}^{-1}\ln(0.5/\tau_S)\right\}$$

where τ_S is a "safety buffer" specified by the user, intended to be the minimum safety margin between the start of the aperture and the breakpoint determined, and c_d is the time constant for a decay to the nominal aperture in $n_{\max}$ pings.

The safe aperture tracker is executed after the fusion decision for each ping, and provides guidance for the processing of the following ping. An example of the tracker's behavior in synthetic data (generated as a power-spectral Fractal with magnitude of the form $|X(\omega)|\propto 1/|\omega|^{\beta}$ with β = 1.1 and $\varphi(\omega)\sim U[-\pi,\pi]$) is shown in Figure 17, where c_d = 0.74; the fast attack (respectively, "smoothed") behavior of the tracker when the breakpoints decrease (respectively, increase) the aperture is clearly evident. Note that a practical issue is to ensure that the predicted apertures do not cross if $\hat{b}(p)-\check{b}(p)<2$ by adjusting the values so that $\hat{b}(p)=\check{b}(p)+2$ iff $b_F^L(p)-\check{b}(p)<b_F^R(p)-\hat{b}(p)$, and $\check{b}(p)=\hat{b}(p)-2$ otherwise.

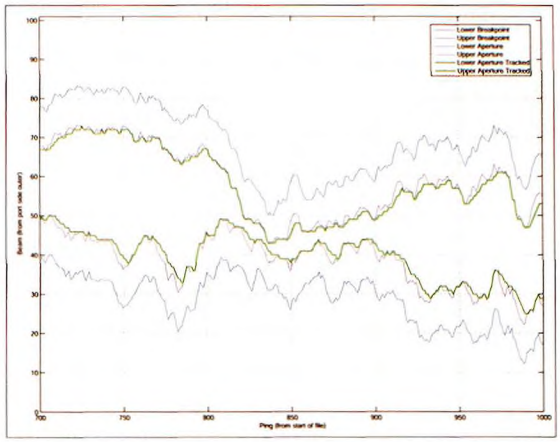

Figure 17: Example of the safe aperture tracker operating in synthetic data. Here, n_{max} = 20 and τ_S = 10 so that c_d = 0.74. The "Lower Aperture" corresponds to $\check{b}(p)$, and "Upper Aperture" to $\hat{b}(p)$.

Appendix B. Implementation of Sub-Algorithms

B.1 Grazing Angle Sub-Algorithm

The simplest explanation for problems observed is that the grazing angles at which the MBES observes the seafloor are too low. At very low grazing angles, little acoustic energy is backscattered and it becomes difficult to make a reliable detection. Normally this is limited because of the maximum opening angle of the MBES, which is in turn limited by refraction correction and physical aperture construction limitations. However on a significant slope this condition can occur somewhere in the outer beams of the swath. (Indeed, for a steep enough slope, it could occur anywhere in the swath.) The Grazing Angle sub-algorithm considers the breakpoint to be at the point where the slope adjusted grazing angle reaches the maximum value allowed by the manufacturer on flat seafloors (Figure 18).

The meta-algorithm determines an estimate of the current slope, and hence the direction of "downhill", by computing across-track beam-to-beam gradients over a small window of W_G pings,

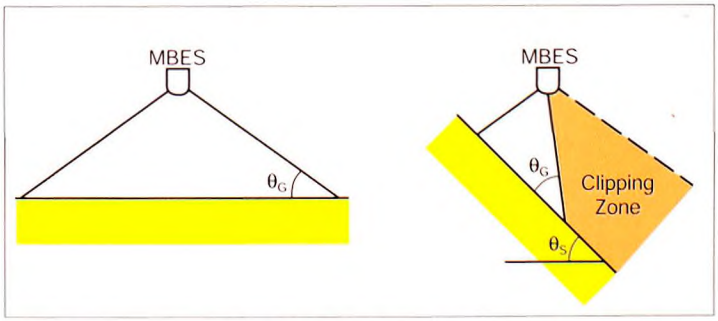

Figure 18: Breakpoint determination based on a slope-corrected minimum grazing angle argument. Beams that have grazing angle lower than that normally allowed when the seafloor is flat are filtered.

$$14)\quad g_0(p,b)=\frac{\Delta z(p,b)}{\sqrt{\Delta x^2(p,b)+\Delta y^2(p,b)}}\qquad (\check{b}(p)\le b\le \hat{b}(p))$$

$$15)\quad g_1(p)=\underset{\substack{0\le\delta p<W_G\\ \check{b}(p-\delta p)\le b\le \hat{b}(p-\delta p)}}{\text{med}}\{g_0(p-\delta p,b)\}$$

$$16)\quad g_2(p)=\underset{0\le\delta p<W_G}{med}\{g_1(p-\delta p)\}$$

where $\Delta x(p,b)=x(p,b+1)-x(p,b)$ for ping p and beam b with corresponding definitions for the other variables, and $\text{med}_x\{\cdots\}$ is a simple median filter. The first median filter provides a rough estimate of gradient from the noisy first differences $g_0(p,b)$; the second median filter provides an improved estimate through re-filtering. It would be possible to improve the estimate robustness by filtering over a wider window. However, the dual method used here re-uses previous filtering and is therefore more efficient.

To compute the maximum aperture allowed, the gradient is converted to slope $s(p)=\arctan(g_2(p))$, and the minimum and maximum angles allowed are computed:

$$\check{\theta}_G(p)=\max\{\check{\theta},-\min\{-\check{\theta}_p-s(p),-\check{\theta}_p\}\}\qquad 17)$$

$$\hat{\theta}_G(p)=\min\{\hat{\theta},\min\{\hat{\theta}_s+s(p),\hat{\theta}_s\}\}\qquad 18)$$

with corresponding beam limits
$\check{b}_G(p)=N/2+\check{\theta}_G(p)/\Delta\theta$ and
$\hat{b}_G(p)=N/2+\hat{\theta}_G(p)/\Delta\theta$, where $\check{\theta}$ and $\hat{\theta}$ are minimum and maximum opening angles for the MBES, $\check{\theta}_p$ and $\hat{\theta}_s$ are the corresponding angles adjusted for any static roll applied to the transducer, $\Delta\theta$ is the beam spacing, and N is the number of beams.

Finally, to make an output compatible with the other sub-algorithms, we must assess the probability associated with the breakpoint detections $\check{b}_G(p),\hat{b}_G(p)$ for all beams in the ping. Based on the observation that this simple scheme generally underestimates the extent of the problem, we model the pdf as a half-Gaussian:

$$\mathrm{N}_h(x;\mu,\sigma^2)=\sqrt{\frac{2}{\pi\sigma^2}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}H(x-\mu)$$

where $H(x)$ is Heaviside's step function. Setting the variance so that the distribution falls in the gap between the breakpoint detected and the safe aperture, the port and starboard distributions are $P_L(p,b|G)$ and $P_R(p,b|G)$, respectively:

$$\tilde{P}_L(p,b|G)=\mathrm{N}_h\left(b;\check{b}_G(p),|\check{b}_G(p)-\check{b}(p)|^2/9\right)$$
$$P_L(p,b|G)=\tilde{P}_L(p,b|G)\Big/\sum_b\tilde{P}_L(p,b|G)\qquad 19)$$

$$\tilde{P}_R(p,b|G)=\mathrm{N}_h\left(\hat{b}_G(p)-b;0,|\hat{b}_G(p)-\hat{b}(p)|^2/9\right)$$
$$P_R(p,b|G)=\tilde{P}_R(p,b|G)\Big/\sum_b\tilde{P}_R(p,b|G)\qquad 20)$$

where $P_L(p,b|G)$ is defined over $\theta\le b<\check{b}(p)$ and $P_R(p,b|G)$ is defined over $\hat{b}(p)\le b<N-1$, and are both zero elsewhere.

B.2 Variance Excess Sub-Algorithm

The second sub-algorithm examines the ping-to-ping variance of the data, which is then compared against an estimate of their theoretical variance computed from an MBES uncertainty model [18] as implemented for the CUBE MBES data processing algorithm [12, 14]. Let $\mathbf{r}(p,b)$ be the vector from MBES to sounding, $\mathbf{r}(p,b)=(x(p,b),y(p,b),z(p,b))^T$, and compute first differences $\Delta\mathbf{r}(p,b)=\mathbf{r}(p,b+1)-\mathbf{r}(p,b)$, $\theta\le b<N-1$. Then, dropping the ping and beam indicators for brevity, the uncertainty associated with the magnitude of $\Delta\mathbf{r}$, $\Delta r=|\Delta\mathbf{r}|$ is:

$$\sigma^2_{\Delta r}=2\Delta r^{-2}\left\{\sigma^2_{xy}\left((\Delta\mathbf{r})^2_x+(\Delta\mathbf{r})^2_y\right)+\sigma^2_z(\Delta\mathbf{r})^2_z\right\}\qquad 21)$$

where σ^2_{xy} and σ^2_z are estimated uncertainties for the beam solution in the horizontal and vertical planes, respectively.

A robust estimate of the expected variance per beam is computed by median filtering:

$$\sigma^2(p,b)=\underset{0\le\delta p<W_V}{\text{med}}\{\sigma^2_{\Delta r}(p-\delta p,b)\}\qquad 22)$$

and then sample estimates of the observed mean, $m(p,b)$, and variance, $s^2(p,b)$, are computed directly. The distinction between consistent and inconsistent beams is typically clearly defined by this method, Figure 19, and comparison of sample and theoretical estimates suffices to determine the breakpoint location. Normal statistical techniques indicate that the test statistic $t(p,b)=(W_V-1)s^2(p,b)/\sigma^2(p,b)$ is distributed as a $\chi^2_{W_V-1}$ random variable given the null hypothesis $H_0: s^2=\sigma^2$, and hence an indicator function (i.e., indicating that the beam is in the consistent region when $i_V(p,b)=1$ and in the inconsistent region when $i_V(p,b)=0$) can be computed for each beam as a result of the test:

$$i_V(p,b)=\begin{cases}1 & t(p,b)<\chi^2_{W_V-1}(\alpha)\\ 0 & t(p,b)\ge\chi^2_{W_V-1}(\alpha)\end{cases}\qquad 23)$$

for suitable significance α.

Finally, we compute the probability that each beam is the true breakpoint by observing that the binary indicator should be distributed as a Bernoulli trial with probability $1-\alpha$ on the safe (interior) side of the breakpoint, and probability α on the exterior, and so the sum of the indicator towards interior and exterior of the swath from the breakpoint are distributed as a Binomial random variable $\mathrm{Bn}(x;N,p)$:

$$\mathrm{Bn}(x;N,p)=\binom{N}{x}p^{x}(1-p)^{N-x}$$

Therefore, the port and starboard probability mass functions are $P_L(p,b|V)$ and $P_R(p,b|V)$, respectively:

$$\tilde{P}_L(p,b|V)=\mathrm{Bn}\left(\left[\sum_{j=b}^{\check{b}(p)} i_V(p,j)\right];\check{b}(p)-b,1-\alpha\right)$$

$$\mathrm{Bn}\left(\left[\sum_{j=0}^{b} i_V(p,j)\right];b+1,\alpha\right)$$

$$P_L(p,b|V)=\tilde{P}_L(p,b|V)\Big/\sum_b \tilde{P}_L(p,b|V) \qquad (24)$$

$$\tilde{P}_R(p,b|V)=\mathrm{Bn}\left(\left[\sum_{j=\hat{b}(p)}^{b} i_V(p,j)\right];b-\hat{b}(p)+1,1-\alpha\right)$$

$$\mathrm{Bn}\left(\left[\sum_{j=b+1}^{N-2} i_V(p,j)\right];N-2-b,\alpha\right)$$

$$P_R(p,b|V)=\tilde{P}_R(p,b|V)\Big/\sum_b \tilde{P}_R(p,b|V) \qquad (25)$$

where $P_L(p,b|V)$ is defined over $0\le b<\check{b}(p)$ and $P_R(p,b|V)$ is defined over $\hat{b}(p)\le b<N-1$, and both are zero elsewhere (see Figure 20).

B.3 Range Limit Sub-Algorithm

The Range Limit sub-algorithm uses the auxiliary information from the sonar meta-data about the user's selected maximum range ($\hat{r}(p)$), which is recorded on a ping-by-ping basis. A characteristic of the "noise" observed is that it frequently reaches the maximum range set; "real" data seldom does. Therefore, if the excess between the maximum range and the range per beam is computed, a comparison against the maximum range clearly shows the breakpoint (Figure 21).

To express "close" to the maximum range, a pseudo-normal metric is computed from the maximum range:

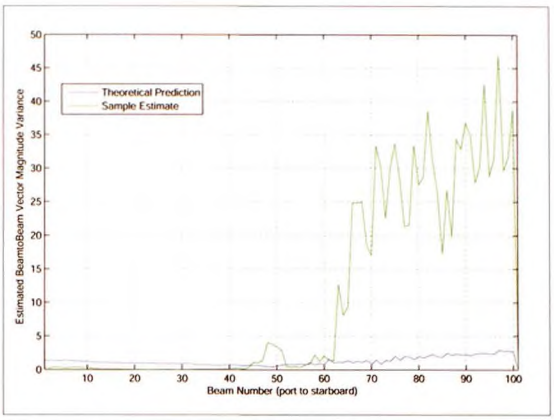

Figure 19: Estimated theoretical and observed beam-to-beam magnitude variance in real data. Inconsistent beam solution in the "noise" area engender significantly higher variance, which is readily detected by comparison with theory.

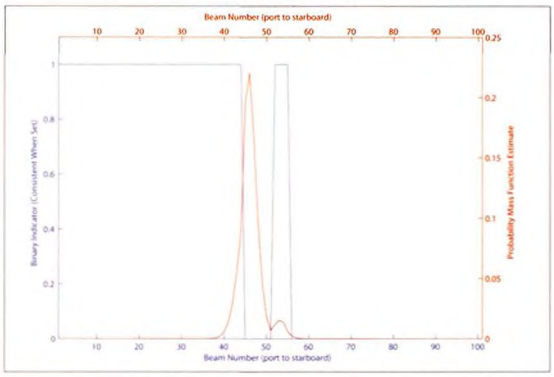

Figure 20: Example of the starboard breakpoint pmf estimation algorithm, (23), (25). The blue line (left scale) is the binary indicator variable in this case (set if the data appears consistent), the red line (right scale) is the probability estimated of the breakpoint being at any specific beam. The port breakpoint estimation is simply the mirror image of this process. Note that this data corresponds to that in Figure 19.

$$n(p,b)=\exp\left\{-\frac{(\hat{r}(p)-r(p,b))^2}{2\sigma_k^2(\hat{r}(p))}\right\} \qquad (26)$$

$$\sigma_k^2(x)=(kx)^2 \qquad (27)$$

where $r(p,b)=\|\mathbf{r}(p,b)\|$ and k is a constant designed to express the degree of "closeness" as a function of the maximum range. The use of a scaled version of the maximum range is suggested by properties of the data in our example datasets, but may not be everywhere optimal. The *ad hoc* nature of this choice is a potential difficulty with application and universality of this sub-algorithm, a topic which is addressed in section 4.

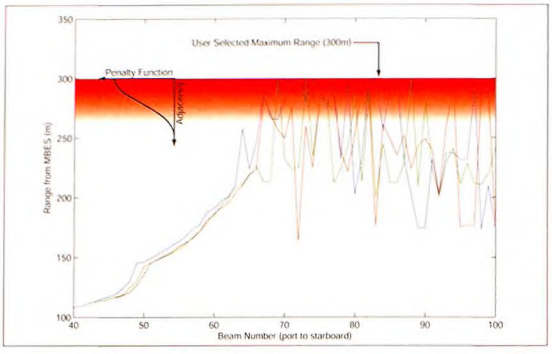

Figure 21: Comparison of per-beam ranges against maximum range selected by user for three typical data swathes. In the inconsistent region, the beam solutions are close to the maximum range more frequently, allowing for a consistent penalty function to be levied based on adjacency to the user-selected maximum range.

To form a compatible binary indicator sequence in the form of (23), a simple threshold of the pseudo-normal metric is used:

$$\hat{n}(p,b) = \max_{\substack{0\le\delta p<W_R \\ -\lfloor W_R/2\rfloor\le\delta b\le\lfloor W_R/2\rfloor}} \{n(p-\delta p, b-\delta b)\} \qquad 28)$$

$$i_R(p,b) = H(\hat{n}(p,b)-\tau) \qquad 29)$$

where $\tau \in [0,1)$ is a suitable threshold value, and the maximization of $n(p,b)$ over a small window serves to compensate for the randomness of ranges after the breakpoint is reached. The remainder of the sub-algorithm follows the form of (24)-(25).

B.4 Turning Angle Sub-Algorithm

The final sub-algorithm uses the angular dependence of the beam-to-beam vectors constructed above. Observing that the angle between subsequent inter-beam vectors is small in the consistent region and therefore will have a relatively peaked distribution, but is more closely uniform distributed in angle in the inconsistent region, Figure 22, testing for distribution of the turning angles for one beam vector across a small window of pings readily distinguishes the two regions.

Let $\mathbf{u} = \Delta\mathbf{r}(p,b)$ and $\mathbf{v} = \Delta\mathbf{r}(p,b+1)$ for simplicity in nomenclature. Then, the turning angle between subsequent inter-beam vectors in the same ping is:

$$\theta(p,b) = -\mathrm{sgn}(v_x u_z - u_x v_z) \cdot \mathrm{acos}\left([u_x v_x + u_z v_z] \Big/ \left[\sqrt{u_x^2 + u_z^2}\sqrt{v_x^2 + v_z^2} \right] \right) \qquad 30)$$

where $\theta(p,b) > 0$ implies clockwise rotation about the ship's direction of motion. Using the theory of circular statistics, Mardia [19] shows that the summary statistic:

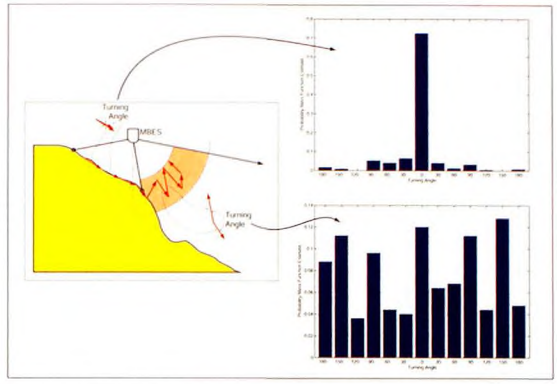

Figure 22: Example distributions of beam-to-beam vector turning angles in consistent and inconsistent regions of the swath. Testing the distribution against the null hypothesis of uniformity allows for separation of the two regions.

$$\overline{C}(p,b) = \frac{1}{K}\sum_{\delta b=-W_T}^{W_T}\sum_{\delta p=-H_T}^{H_T} \cos(\theta(p+\delta p, b+\delta b)) \qquad 31)$$

with $K = (2W_T+1)(2H_T+1)$ is sufficient to test the hypothesis:

$$H_0: \{\theta(p,b)\} \sim \mathrm{U}[0,2\pi) = \mathrm{M}(\theta;0,0)$$
$$H_1: \{\theta(p,b)\} \sim \mathrm{M}(\theta;0,\kappa),\ \kappa \neq 0$$

(i.e., that the distribution of angles is uniform against the alternative of a peaked distribution as shown in Figure 22), where $\mathrm{M}(\theta;0,\kappa)$ is the Von Mises distribution:

$$\mathrm{M}(\theta;\mu,\kappa) = \frac{1}{2\pi I_0(\kappa)}\exp\{\kappa\cos(\theta-\mu)\}$$

and $I_0(\kappa)$ is a modified Bessel function of the first kind and order zero. The critical values of $\overline{C}(p,b)$ are difficult to compute exactly, but may be approximated for large samples as $\overline{C}(p,b) \sim \mathrm{N}(\overline{C};0,(2n)^{-1/2})$ for n samples, and therefore, the test resolves to $\overline{C}(p,b) > \overline{C}_{\mathrm{crit}} = z_c(\alpha)/\sqrt[4]{(2(2W_T+1)(2H_T+1))}$ where $z_c(\alpha)$ is the corresponding unit Normal critical value of α probability of Type I error. Rejection of H_0 implies that the distribution of turning angles is sufficiently far from uniform, and hence that the data is most likely consistent. Therefore, a binary variable is readily computed as:

$$_T(p,b) = \begin{cases} 1 & \overline{C}(p,b) \ge \overline{C}_{\mathrm{crit}} \\ 0 & \overline{C}(p,b) < \overline{C}_{\mathrm{crit}} \end{cases} \qquad 32)$$

and the remainder of the sub-algorithm computes the pmf of the breakpoint location following the form of (24)-(25).