

University of New Hampshire University of New Hampshire Scholars' Repository

Center for Coastal and Ocean Mapping

Center for Coastal and Ocean Mapping

1-29-2007

CUBE User's Manual

Brian R. Calder

University of New Hampshire, Durham, brian.calder@unh.edu

David E. Wells

University of New Brunswick

Follow this and additional works at: <https://scholars.unh.edu/ccom>

 Part of the [Oceanography and Atmospheric Sciences and Meteorology Commons](#)

Recommended Citation

Calder, Brian R. and Wells, David E., "CUBE User's Manual" (2007). *Center for Coastal and Ocean Mapping*. 1217.
<https://scholars.unh.edu/ccom/1217>

This Report is brought to you for free and open access by the Center for Coastal and Ocean Mapping at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Center for Coastal and Ocean Mapping by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.

CUBE User's Manual

Version 1.13

Brian Calder

Center for Coastal and Ocean Mapping
and NOAA/UNH Joint Hydrographic Center
University of New Hampshire

David Wells

Center for Coastal and Ocean Mapping
and NOAA/UNH Joint Hydrographic Center
University of New Hampshire

Department of Marine Sciences
University of Southern Mississippi

Department of Geodesy and Geomatics Engineering
University of New Brunswick

[This User's Manual is a work in progress. Check with the authors for later versions. In particular Appendix B "Uncertainty Attribution" from B.2.1 onwards is presently under construction. As well, feedback from software suppliers who have implemented CUBE, and may invoke specific features, remains to be incorporated. Feedback from ANY reader is encouraged - how can this be made more useful for you? Let us know.]

30 January 2007

Table of Contents

Table of Contents	2
1 - Introduction.....	3
1.1 What is CUBE?	3
1.2 CUBE Modes of Operation	5
1.3 History of CUBE	6
1.4 Relationship to the Navigation Surface.....	7
1.5 Relationship to the Open Navigation Surface Project	7
2 - CUBE overview	8
2.1 CUBE input	8
2.2 CUBE functions.....	8
2.3 CUBE output.....	8
2.4 What is assimilation?	9
2.5 What is intervention?	11
2.6 What is disambiguation?.....	12
2.7 Caveats	12
3 - Assimilation	14
3.1 Reordering	14
3.2 Scattering	16
3.3 Gathering	17
3.4 Assimilation memory fading.....	18
4 – Tracking and Intervention	19
4.1 Bayes Theorem in Summary	19
4.2 Applying Bayes Theorem to CUBE.....	19
4.3 Estimate Offset Value	21
4.4 Bayes Factor Threshold.....	22
4.5 Run-Length Threshold	22
5 - Disambiguation.....	23
5.1 CARIS's Locale Rule	24
5.2 CARIS's 'Null Hypothesis' Rule	24
5.3 CARIS's Blended Density/Locale Rule	25
6 - Summary	27
7 - References	28
Appendix A: CUBE Axioms.....	29
Appendix B: Uncertainty attribution.....	33
Appendix C: Bayes' Theorem and Dynamic Linear Models.....	48

1 - Introduction

1.1 What is CUBE?

CUBE is an algorithm that implements computer-assisted hydrographic processing (a.k.a., 'algorithmic hydrography'); it is an acronym for 'Combined Uncertainty and Bathymetry Estimator'. The objective is to process bathymetric data from raw measurements resolved in a local-level coordinate system into estimates of water depth in a particular position on the face of the earth. The algorithm is aimed mostly at high resolution Multibeam Echosounder (MBES) data, although other suitably dense bathymetric data could be used.

The primary aim of the algorithm is to use as much information as possible from the data to determine the true depth at any point in the survey area from the noisy estimates of that depth (i.e., the soundings), and to quantify an uncertainty on the depth estimate. CUBE attempts to answer the fundamental question of "What is the depth, and how well do we know it?" rather than the more traditional hydrographic question of "How good is this sounding?" The simultaneous estimation of bathymetry and uncertainty is the 'combined' in CUBE.

CUBE makes a distinction between systematic miscorrection, blunders and uncertainty. (The term "error" is now deprecated since it is overloaded in common usage with unintentional meaning. In particular "error" is correctly the difference between the true value and the estimated value since we never know the true value, we can never really compute the error. Uncertainty is our estimate of this error's magnitude. [10]) Systematic miscorrection is a correction that should have been made and either was not, or was made incorrectly. For example, this could be an incorrect lever-arm offset between MBES and GPS receiver. CUBE assumes that all systematic effects are corrected as well as they can be before data is presented to it, and makes no attempt to correct or compensate for these effects. It will, however, readily highlight such effects.

Blunders are missolved beam solutions generated by the MBES which do not correctly represent the seafloor. This could be due to reflections from mid-water targets, or side-lobe reception in the MBES, etc. CUBE attempts to isolate these from other data by accumulating them into one or more potential depth reconstructions with the hope that when the 'best' reconstruction is selected, these will be avoided. (If not, it is the operator's task to identify the event and correct for it.)

CUBE's primary concern, however, is the residual uncertainty of the soundings. That is, the random variability of the soundings that remains after all appropriate corrections are made. This is a fundamental property of all real data, and cannot be avoided; at best, it can be understood and managed. This axiom is at the heart of CUBE's view of data, and the data processing problem.

The primary drivers behind the development of CUBE were speed of processing, and promotion of objectivity. The first is a consequence of the data rates possible with newer MBES systems. The standard methods of processing this data required that a human operator should individually examine each sounding; a practice brought forward from Vertical Beam Echosounder (VBES) methods. In practice, most modern systems for shallow water examined the data in batches of 60-100 pings at a time, with the user trimming the 'obvious' outlier data points; some systems also provide area-based editing, where soundings are examined in

geographical space either instead of, or subsequent to, ping-by-ping examination. In either case, however, data rates from MBES systems are such that the time taken to process all of the data far exceeds the time taken to capture the data given the (limited) human resources available. CUBE aims to improve on this situation by replacing the majority of this human inspection with automatic algorithmic examination of data based on the estimated uncertainties of the soundings, followed by a human 'quality control' inspection. In this model, the operator's primary task is to examine the results of the algorithm's assessment of the data, and determine whether they correspond to a prudent hydrographer's interpretation of the same data. If so, then the algorithm's results are allowed to stand; if not, the operator adjusts the parameters of the algorithm, corrects problems with the data, overrides the algorithm's selection of data, or otherwise rectifies the problem. However, because inspection of the algorithm's outputs does not require that every sounding be examined, and most data collected is provably good, the result is that significantly less time is required to process the data when using CUBE.

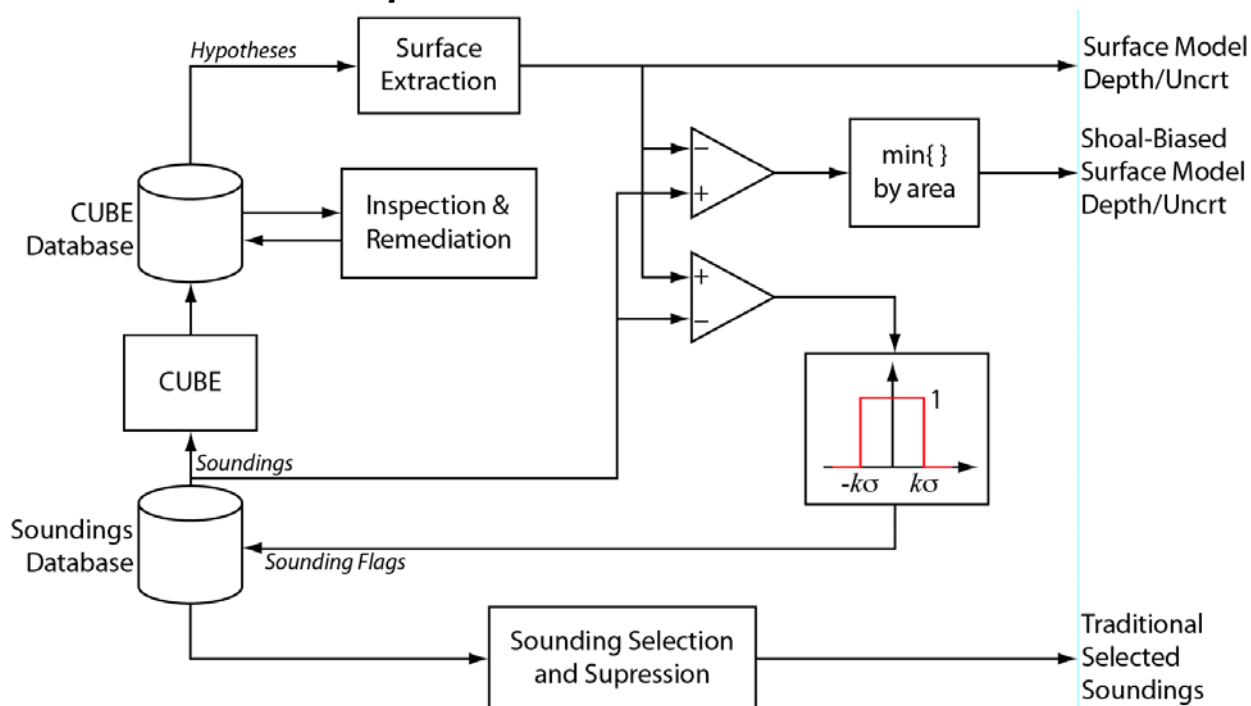
This redefinition of the operator's role also addresses the requirement for objectivity in the processing chain. In the standard model, the operator examines each sounding and subjectively determines whether it should be considered 'acceptable' or 'inappropriate'. There is, in most systems, no log of this, and few defined procedures for what is acceptable for further processing except experience of the operator. Justifying these decisions at some later date would be problematic. CUBE starts with the premise that all soundings are uncertain to some degree, representing an estimate of the depth in the area, and not necessarily 'the' depth. The question therefore is how to accommodate this uncertainty, and how to make use of it in answering the fundamental dual question of depth and uncertainty of that depth. By appropriate use of estimates of the sounding uncertainty, and models of propagation and combination of uncertainty, CUBE can provide estimates of the most likely depth at any point in the survey area (and hence by extension over the whole survey area at an appropriate resolution) which are unaffected by any human bias, and for which there is quantified statistical support. Since the process is algorithmic, it is also guaranteed repeatable with the same inputs. The algorithm also has in-built checks for consistency of the data, and supports multiple potential depth reconstructions at each point. The user is always provided with information on these potential reconstructions, and CUBE's estimate of their relative strengths. The goal is to provide not only objective estimates of depth, but also objective estimates of the quality of these reconstructions so that the operator has tools to decide whether to agree with the algorithm or not.

This last point is very important. CUBE is by definition not a complete solution to the hydrographic data processing problem. It will not – ever – replace completely a human operator. The goal is to provide a power-tool to assist the operator with interpreting the data, not to switch off the operator's hydrographic abilities. There are hydrographic situations for which the algorithm does not have sufficient information to make an adequate decision, and it is the operator's responsibility to ensure that these are recognized and corrected if required. A classical example is where there is scant evidence for a shoal point within dense deeper data, e.g., a single sounding at the bow of a wreck. To the algorithm, this appears as a potential reconstruction, but one of low likelihood since there is only one sounding supporting it; the alternatives at the point are much more statistically likely. The algorithm has no understanding of wrecks, and therefore would probably report the alternative depth reconstruction to the user as its 'best hypothesis' on depth, while retaining the shoaler reconstruction for user inspection. An appropriately wary operator might hypothesize that the sounding came from a jack-staff on the bow, although in fact there is no evidence that this is the case. It could equally well be a false return from a passing fish. However, hydrographic caution would mandate that this shoal point be preserved with appropriate markings for the future. This is possible with most implementations of CUBE, but relies on the operator's ability to make these decisions. To

repeat: CUBE is a power-tool to assist operator decision-making, not a replacement for sound hydrographic practice.

CUBE is, then, fundamentally a point estimator of depth. It takes into account the uncertainties of the component measurements of depth (i.e., the soundings) and attempts to estimate the most likely depth at any particular point in the survey area. It can adapt to the case where there is evidence in the data for multiple different possible reconstructions of depth and can keep the supporting evidence for each potential reconstruction independent of the others. It maintains all potential reconstructions on-line at all times, and is capable of updating them in real time to reflect new evidence as it is collected (i.e., it does not have to have all data available before starting to make decisions). The algorithm also maintains metrics on the potential reconstructions, their relative likelihood and the algorithm's confidence in its choice of 'best' reconstruction. The output from CUBE once all ambiguity in potential reconstructions is resolved is a collection of point estimates of depth and their associated uncertainties spread across the survey area. For efficiency in implementation, this generally takes the form of a regular grid, although this is not mandatory. CUBE re-focuses the operator paradigm from inspection of soundings to aided quality control of algorithm outputs.

1.2 CUBE Modes of Operation



This flow diagram illustrates three modes of using CUBE, two of which are currently in widespread use. The right side of the diagram lists the three outcomes that result.

The intended use of CUBE, the main thrust of this manual, is traced as follows: from the Soundings Database, soundings are processed through CUBE into a CUBE database, which is subjected to inspection and remediation by a human operator. Disambiguation of alternative hypotheses results in the extraction of a depth surface model (defined by CUBE's nodal depth estimates), and associated uncertainty estimates, which form the output intended for CUBE - a

realistic depth surface model attributed with uncertainty. When required (e.g. for nautical charts) this surface can be subjected to the Navigation Surface processing [5], as noted in Section 1.4 below.

However, not all CUBE users have yet been convinced to replace traditional sounding selection by such a surface model. CUBE can still play an important role for these users. The CUBE surface model can be used to assess the appropriateness of soundings in the Soundings Database, and to flag those that do not belong to the hypotheses that were selected by the disambiguation engine, playing an efficient and important role in "data cleaning". The resulting output is shown as the bottom output, whereby soundings that remain unflagged by this CUBE comparison, are subjected to traditional sounding selection and suppression (shoal biasing) algorithms.

The middle option represents an alternative that (to our knowledge) has not yet been widely used. This option uses the minimum (unflagged) soundings in an area to shoal bias the depth surface for that area, adjusting the depth surface uncertainty values appropriately. This results in a shoal biased depth surface, as does the Navigation Surface approach. However in this case, the defocussing and double-buffering processes of the Navigation Surface are replaced by a more direct shoal-biasing by the minimum soundings in an area.

This manual concentrates on the processes involved in obtaining the depth surface model and uncertainty. However, this flow diagram shows that these results can be applied in other ways than been accepted as the final product. That is not our recommendation - it is our belief that the depth surface model does the best job of "telling the truth about the data". However, we accept that it will take time for enough confidence to be built in the major paradigm shift represented by this alternative, for it to be widely accepted.

1.3 History of CUBE

CUBE was developed as a research project within the Center for Coastal and Ocean Mapping and NOAA/UNH Joint Hydrographic Center (CCOM/JHC) at the University of New Hampshire, starting in the fall of 2000. The primary motivation was initially that the author was badly seasick during a rough mapping cruise, and felt that there had to be a better way of processing data than trying to focus on little colored dots while feeling ill. Uncertainty estimation of soundings was added early in 2001 by adaptation of the Hare-Godin-Mayer MBES uncertainty model [1] and multiple hypothesis tracking and model intervention were added late in 2001/early 2002. The algorithm was first presented in a very limited form at the US Hydrographic Conference in 2001 (Norfolk, VA), and in more detail with multiple hypotheses at the Second International Conference on High Resolution Survey in Shallow Water in the fall of 2001 (Portsmouth, NH); results of comparisons against human testing was presented at the Canadian Hydrographic Conference in 2002 (Toronto, ON), and results of comparison of processing efficiency against the then-standard processing chain were presented at the US Hydrographic Conference in 2003 (Biloxi, MS). The algorithm has been published in peer reviewed journals, [2] [3] Testing of the algorithm for robustness, operational performance improvement and validity was carried out during 2002 and 2003, including processing of three full-scale NOAA/NOS surveys in Snow Passage, AK (H10949), Woods Hole, MA (H11077), and Valdez Narrows, AK (H11182). CUBE was packaged into a 'reference' release from the research source code on 2003-02-14, and was offered for royalty-free non-exclusive license to all interested parties through the University of New Hampshire's Office of Intellectual Property Management. The source code distribution is

maintained in a subversion repository at CCOM/JHC so that all implementations have access to, and start from, the latest source code.

The CUBE algorithm has subsequently been licensed by IVS3D (released 2003, integrated into a PFM editor), CARIS (released 2005, integrated into their BASE surface technology), QPS (released 2006 in QCloud, a 3D editor), Kongsberg Simrad (released 2005 in SIS), Reson, Triton Imaging International (released 2005 in ISIS), SAIC, and IFREMER.

1.4 Relationship to the Navigation Surface

CUBE and the Navigation Surface [5] developed in parallel, and are designed to be complementary; they are, however, distinct entities. CUBE's outputs are compatible with the Navigation Surface approach to representation and processing of bathymetric data for hydrography (and other tasks), although CUBE is not required to build surfaces appropriate for these methods.

1.5 Relationship to the Open Navigation Surface Project

The Open Navigation Surface Project [6] was developed to standardize a means to represent bathymetric surfaces appropriate for the Navigation Surface approach to data management. The intent of the project is to define a file format that can contain bathymetric data and its uncertainty in a regular grid structure, any hydrographic modifications required by the operator (e.g., over-rides of depth/uncertainty pairs to account for hydrographically significant objects), and meta-data to describe the object. The project also provides an Open Source library to read and write the file format, known as a Bathymetric Attributed Grid, or BAG, which is supported for both Windows and Unix platforms. The File Specification Document (FSD) [7] for the project defines all required components of the file, and how to interact with the Project. It is available from the Project's website, <http://www.opennavsurf.org>.

2 - CUBE overview

CUBE has been called a data-cleaning package. It has been called a gridding package. These it may be (with caveats), but it really is much more multi-faceted than that. It is a fresh attempt to “tell the truth about the data” that is acquired during a bathymetric survey, particularly a high-density survey (e.g. resulting from the use of a multibeam echosounder, or airborne LIDAR).

The CUBE algorithm is based on seven accepted truths, or axioms. These are listed in Appendix A to this document. They are referred to by number in the following.

2.1 CUBE input

CUBE ingests uncertainty-attributed depth measurements. We will henceforth refer to these as “soundings”.

Uncertainty attribution means that each sounding measurement has estimates attached to it, of its uncertainty in depth and its uncertainty in horizontal position. Hence the input to CUBE is a data set conceptually constructed as $\{x_i, y_i, z_i, \sigma x_i, \sigma y_i, \sigma z_i\}$, $i = 1, \dots, n$, where z is depth, x, y defines the location of the sounding, and the σ values are estimated uncertainties.

2.2 CUBE functions

CUBE transforms measured soundings at randomly spaced locations, to regularly spaced depth estimates. To do this CUBE incorporates three main functions: assimilation, intervention, and disambiguation. Each of these is briefly defined in this chapter, and each has a chapter devoted to when, why and how the user may “tweak” CUBE’s processes of assimilation, intervention, and disambiguation.

Assimilation is a common function of gridding algorithms (although it may be done in many different ways). Model intervention is a standard time series analysis task. Disambiguation is basically an unsupervised classification task. However their combination in the CUBE algorithm represents a major paradigm shift in the processing of bathymetric data.

2.3 CUBE output

CUBE produces depth estimates at predefined nodal grid intersections. For each node, four values are produced: depth, depth uncertainty, number of hypotheses, and “hypothesis strength”. Hence the output from CUBE is a data set conceptually constructed as $\{z_i, \sigma z_i, nh_i, hs_i\}$, $i = 1, n$, where z is depth, σz is depth uncertainty, nh is the number of hypotheses, and hs is the hypothesis strength. In addition, the grid origin, spacing, grid width and height (in nodes) are defined in metadata, from which the real-world coordinates for each node can be computed. Note that $n = \text{number of nodal depths} = (\text{grid width} + 1) * (\text{grid height} + 1)$.

Each of these four sets of values is really a collection of point estimates, but can loosely be considered to be a “surface”. Hence, for manipulation, display and analysis, CUBE output is four surfaces:

- (a) Depth surface – the final depth value at each node, after the competing hypotheses have been disambiguated.
- (b) Uncertainty surface – the final uncertainty associated each nodal depth
- (c) The number of hypotheses at each node
- (d) The uncertainty associated with the selection of the correct hypothesis by the disambiguation engine. This “hypothesis strength”, is a measure of algorithmic certainty - how convinced CUBE is that it's telling you the real story.

2.4 What is assimilation?

Assimilation is the assembling of statistically based nodal depth estimates and depth uncertainties, based on the input sounding measurements and their 3D uncertainty estimates. Three steps involved in this process are:

2.4.1. Establish a network of nodes over the survey area

This is the basic first step in any gridding operation. Node points are selected to

- (a) be dense enough to completely capture all features of interest,
- (b) no denser than needed,
- (c) usually regularly-spaced, and
- (d) at horizontal locations that are “perfect” since they are selected by convention, and do not involve any measurements.

“Gridding” often implies smoothing. This is not the intention in CUBE. The nodes are arranged as a dense grid for depth estimation “at a point”, not to smooth the results, and the nodal separation (spatial resolution) may not be much different than the separation between soundings.

This step is fundamental to implementing Axiom 2 – estimating depths and depth uncertainties, rather than accepting measured soundings as “golden”.

2.4.2. At each node, maintain sequentially-determined estimates of depth, and depth uncertainty

“Sequentially-determined” is the innovative component to this step. The current estimate can be updated by new sounding measurements nearby. This means that, in principle, that CUBE can be used as a real-time estimation tool (built into the MBES data acquisition software), rather than only in post-processing software. However this requires an initial something to be updated (a foundation of “Bayesian estimation”, which implies an updating process).

An early incarnation of CUBE (before Steps 4 and 5 were thought up), obtained this initial something by first ordering the initial set of soundings according to internal consistency, so that the initial something was unlikely to have unfortunate pathology that would distort the Bayesian process. However, the inclusion of alternate hypotheses (Step 4), and later disambiguation (Step 5) allows unfortunate pathology to occur (a starting hypothesis that will be later discarded). The reordering algorithm remains (it can do no harm, and may do some good), as a

rolling (sequential) median window, but could likely be removed, or reduced to three data points. It probably improves the rate of convergence (but not the convergence uncertainty). However, we've drifted too deep into the weeds in this discussion!!

2.4.3. Compare incoming sounding measurements against existing nodal estimates

Each incoming sounding is considered to contain some (imperfect) information about the depth at each of the nodes surrounding it. This information is passed to each of these nodes using a propagation strategy that involves three processes that appropriately dilute the 3D sounding uncertainty based on the sounding / node separation. The nodal horizontal locations are "perfect" (have zero uncertainty) since they are defined by convention, not by measurement. Incoming soundings that are estimated to lie closer to the node should have more influence on the nodal depth than those that lie further away. Incoming soundings that have a smaller estimated vertical uncertainty should also have more influence on the nodal estimation than those with larger vertical uncertainty. Incoming soundings that have a larger estimated horizontal uncertainty should have less influence on the nodal estimation than those with lower uncertainty.

The first process is to propagate (increase) the **vertical** uncertainty of the sounding as a function of some power of the sounding / node distance. The second power (distance squared) is the default power used at present. This process (the power selected) is based on the assumption that the vertical uncertainty being propagated is primarily "random" in nature (rather than systematic).

The second process is to convert the **horizontal** uncertainty of the sounding into a vertical uncertainty at the node, as a power of the sounding / node maximum distance (the nominal sounding boresight distance from the node, increased by the horizontal uncertainty). The first power (linear distance) is the default power used at present. This process (the power selected) involves assumptions about possible seafloor slopes between sounding location and node.

The third process is the statistical **comparison** between the incoming sounding (propagated to the node location), and the already-assimilated information at the node. The simple basis for this comparison is whether the incoming sounding (and its propagated vertical uncertainty) is statistically consistent with at most one of the potential depth reconstructions at the node (and its vertical uncertainty). In other words does the nodal depth lie within the uncertainty region of the sounding, or, more importantly, does the sounding lie within the uncertainty region of the previous nodal depth estimate?

Two important points to bear in mind here:

- (a) a particular incoming sounding will generally contribute to updating the estimates at several surrounding nodes, not just the closest one, and
- (b) the process just described serves to collapse three-dimensional (horizontal and vertical) uncertainties in soundings to one-dimensional uncertainties (vertical only) at nodes.

2.5 What is intervention?

Intervention is a decision to interrupt the straightforward assimilation process when an incoming sounding measurement is not statistically compatible with any of the existing nodal depth estimates. Intervention is based on the practical experience that sometimes soundings are associated with more than one object (the classic being soundings coming from both the seabed and a school of fish floating above it). Intervention results in the creation of an "alternative hypothesis".

CUBE draws a clear distinction between incoming soundings that are statistically consistent with previous nodal depth estimates, and those that are not. CUBE employs the jargon of statistical theory in dealing with cases where the sounding is statistically INconsistent with the previous nodal depth estimate: the incoming sounding is judged to have "failed the null hypothesis that it is consistent" with one or previous nodal depth estimates.

Now comes the really innovative bit – CUBE then creates a new "hypothesis" track, based on the inconsistent incoming sounding.

Why is this so innovative? Two reasons.

- (a) It differs from previous approaches by deferring final judgment on appropriateness of a sounding until all the evidence is in.
- (b) It recognizes the possibility that soundings in the same area may be sensing more than one depth trend (e.g. the top of a school of fish as well as the bathymetry beneath them).

Incoming soundings are only compared against one hypothesis before being considered to be a new track. The hypothesis is chosen on a least error basis, and the assumption is that if the data doesn't match the closest hypothesis, then it won't match any others either. This isn't quite true (if the next closest had a larger uncertainty, it might assimilate the data), but it's usually fairly valid.

Deferring judgment on the true depth is also a requirement for real-time processing. We have to accept that the estimate (and choice) of depth is data dependent, and may vary as further data is added. Therefore, we must keep updating our current state of knowledge based on new evidence from new data. This is deliberately a lot like everyday life decision-making.

Of course, we don't have to wait for **all** of the evidence to be in. We can make a decision (run the disambiguation engine) at any time, as long as we accept that the resulting decisions are malleable (they could get modelled by adding more data).

New hypotheses are not only about catching different real returns, they're also about (or even primarily about) making sure that good data and bad data don't get mixed together. CUBE doesn't really know good from bad in any concrete sense, so it is better to think of CUBE as partitioning data into 'mutually consistent' subsets that are 'inconsistent' with alternative 'mutually consistent' subsets (reserving the right to have an army of one for a really isolated piece of data). This interpretation allows one to think of CUBE as an unsupervised clustering / classification system, if preferred.

2.6 What is disambiguation?

Disambiguation is the process of deciding which of many hypotheses about the depth at a particular node (i.e. the ambiguity in the nodal depth) is the best estimate of the “true” depth. In CUBE an algorithm called the “disambiguation engine” makes this decision.

The “disambiguation engine” drops the second innovative shoe. It is specifically designed to attempt to decide which of several hypotheses that may exist at a node is most likely to be the correct one.

The disambiguation engine will be subjected to (perhaps never ending) refinement over time, as more is learned about the nature of and causes for multiple nodal depth hypotheses. However for now, the reference disambiguation engine has three “metrics” or tests that are used to resolve which hypothesis is most likely the correct one at each node. These metrics are:

- (a) The popularity contest. Which hypothesis is consistent with the greatest number of incoming soundings at that node?
- (b) The local consistency test. Which hypothesis is closest to a prediction based on nearest neighbor nodes that have only one hypothesis?
- (c) The external consistency test. Which hypothesis is closest to an “external” (probably lower resolution) reference surface? This surface could be generated by a median filter through all the soundings being processed, or based on an older chart or survey data.

In practice, our experience is that the most reliable metric is the 'popularity contest', even though it's almost embarrassingly naive. It also runs a great deal faster than the other metrics.

Further metrics are under construction. Brian thinks the eventual best solution will probably involve

- (a) some sort of local coherence,
- (b) a quality measure on suspiciousness of reconstruction, and
- (c) a 'try harder' mode to use more powerful techniques where simple ones vacillate.

2.7 Caveats

CUBE is a very good tool for efficiently producing a product that “tells the truth about the data”. But it remains just a tool. It is not perfect. Measured soundings exhibiting pathological (unreasonable to expect) behavior may elude CUBE's attempts to tell the truth about them. Hence caution should always be exercised.

2.7.1 Re-CUBE as necessary

Read Axiom 4. CUBE is a triage tool. Anybody who claims to have a fully automatic algorithm is lying or hasn't seen enough data. CUBE does its best to get as much right as possible, but always needs a user at the end to resolve problems that the algorithm cannot. CUBE's job is to provide as much guidance as possible on how to do this, and only then to minimize how much is required. This also makes the user's job different: not so much about killing dots, more about quality assurance. Fully processing data is an iterative process, but of course not all of the data needs to be reworked at each pass.

2.7.2. CUBE is just a (better) tool – not the final answer

Read Axiom 7. CUBE inhabits the model world, not the real world. CUBE is not the final stage in the processing pipeline. Users should not take raw CUBE grids and use them for charting. CUBE will generate best estimates of depth, but that's not what a nautical chart shows (and never will be).

For example, we return to the concept of "Golden Soundings". The hydrographer should be allowed to override the algorithm, and retain those (rare) "critical to navigation" sounding values in any given survey. But now we're entering the territory of the "Navigation Surface" (described elsewhere). But, like the CUBE algorithm, it is another aspect of the principle of determining the true surface as well as possible. A more traditional alternative, proposed by Peter Kielland is to subtract the 95% uncertainty values from the untouched CUBE nodal depths, to give a statistically shoal-biased surface for navigation.

3 - Assimilation

Assimilating the data, from input soundings to nodal depth estimates, involves three stages: reordering, scattering, and gathering.

Reordering is used to delay consideration of outliers.

Scattering is a sounding-centric process - each new input sounding presents itself to many surrounding nodal points for consideration in updating each of their nodal estimates.

Gathering is a nodal-centric process - at each node, the capture distance is determined, which is the maximum radius from the node within which input soundings will be accepted for updating the nodal estimate.

There are user-accessible parameters associated with each of these three stages.

3.1 Reordering

Reordering means changing the sequence that input soundings are presented to the nodal estimation procedure. The purpose is to delay consideration of outlier soundings as long as possible. This procedure was essential in the early research (single-hypothesis) version of CUBE, before intervention and disambiguation had been incorporated into the algorithm. With released versions of CUBE it is not essential.

Reordering is implemented using a running median filter. It works like this:

- 1 - Construct a list (queue) containing an odd number of input soundings (say 11 soundings).
- 2 - Reorder the soundings in this list according to their depth value (shallowest at the top, deepest at the bottom).
- 3 - Once the queue is full, export the median sounding to the nodal estimation process (the middle sounding on the list - in our case, 6th from both top and bottom - is the median).
- 4 - Replace the exported median with another input sounding.
- 5 - Return to step 2, unless there are no more input soundings.
- 6 - When there are no more input soundings, export the remaining 10 soundings on the list from the center of the list outwards.

Outliers will stay near the top and bottom of the queue, and will be the last to be processed.

It is important to note that no input data is changed. Every sounding gets presented to the estimation procedure unchanged. Only the order of presentation is affected.

With this reordering, the nodal estimates are protected from outliers, while they “learn” about the true depth. Outliers must compete with well-established single-hypothesis nodal estimates. This minimizes the damage they can do.

Reordering is still important for single-hypothesis estimation algorithms, but not for the released (multiple-hypothesis) version of CUBE, since early outliers will spawn their own hypotheses, and soundings closer to the median depths will generate a hypothesis that is more likely to be selected during disambiguation. However the procedure has been retained in CUBE because it does no harm.

3.1.1 Queue length

One of the user-accessible parameters in CUBE is the reordering median filter queue length (we assumed a queue length with the default value of 11 above).

Queue Length: length of pre-filtered queue that CUBE uses before assimilating data into a node. Value range is odd number {3 to 101}, default = 11

When using the released (multiple-hypothesis) version of CUBE, changing this parameter should have no significant effect.

Lengthening the queue will consume additional memory, perhaps significantly. This may also increase processing time, if virtual memory swaps to disk are increased as a consequence of larger arrays in memory.

3.1.2 Quotient limit

In the research (single-hypothesis) version of CUBE, it was important to have an outlier identification test, to eliminate outliers from contaminating the nodal estimate. This test is performed before soundings are exported from the queue. It uses the mean value of all soundings in the queue, and the discrepancies of each member of the queue from this mean. The test is to compare ratio of (a) the square of the potential outlier's discrepancy from the mean to (b) the sum of squares of discrepancies from the mean, for all other soundings in the queue, except the potential outlier. An outlier is declared and eliminated from further consideration when this ratio exceeds a user-accessible CUBE parameter:

Quotient Limit: outlier quotient upper allowable limit. Value range {0.1 to 255.0}, default = 255.0

With the released multiple-hypothesis version of CUBE this parameter is no longer necessary, and could, if set inappropriately low, eliminate valid soundings from consideration. It is now a dangerous parameter, rather than a useful one.

Hence it should always either be set to its maximum value of 255, or else removed as a user-accessible parameter.

3.2 Scattering

Scattering involves the calculation of the depth uncertainty to be associated with each input sounding, once it has been transferred from its own location to a nodal location. This will be a function of the depth and horizontal uncertainty attributes associated with the sounding measurement itself, as well as the distance from the nodal point. Actually the distance involved should include the possibility that the sounding is at the far edge of some confidence region. A simple propagation model that incorporates all these ideas is

$$\sigma_p^2 = \sigma_v^2 \left(1 + \left[\frac{dist + hes * \sigma_h}{node_spacing} \right]^{de} \right)$$

where

- σ_p is the propagated uncertainty (standard deviation), in depth only, after translation of a sounding to a nodal point,
- σ_v and σ_h are the vertical and horizontal uncertainty attributes associated with the input sounding (at its original location),
- dist* is the distance from sounding location to node,
- node-spacing* is the distance between nodes.

The remaining two parameters are user-accessible:

hes = Horiz_Error_Scalar: the value used to scale the horizontal error of each sounding when used in the radius of influence computation. Value range {0.0 and 10.00}, default = 2.95 (99% CI)

de = Distance Exponent: used to control exponential growth of a sounding's vertical uncertainty as a function of distance from the node. Value range {1.0 to 10.0}, default = 2.0

This equation was selected to have this form so that:

- soundings with higher σ_v will have less weight
- soundings farther away will have less weight
- soundings with higher σ_h cause the uncertainty to scale faster, and hence have less weight.

Altering these parameters from their default values may be appropriate under one of three conditions, each of which represent failure of one of the axioms in Appendix A:

3.2.1 Inadequate sounding density

CUBE assumes a flat seabed in the scattering process - the sounding depth is left unchanged as it is translated to nodal positions. This is equivalent to assuming that the soundings are dense enough to provide multiple samples of the same feature, or same depth area, and that the node spacing is dense enough to capture the shapes of features of interest.

Changing one or both of the two user-accessible parameters can partially accommodate / remediate inadequate sounding density. For example, when sounding along a sharp seabed scarp, there will be distinct "true" depths at adjacent nodes (on either side of the scarp). The

default value for *hes* (horizontal error scalar) of 2.95 means that the propagated uncertainty σ_p will include significant scaling of σ_h meaning that all soundings will be given significant scaled vertical uncertainty; at the edges of objects, the data will appear “compatible” from both sides of the edge and will form one hypothesis rather than two (albeit with significantly higher standard deviations).

Reducing the value for *hes* (horizontal error scalar) will reduce the propagated uncertainty, making it more likely that soundings coming from opposite sides of the edge will create two different hypotheses.

3.2.2 Soundings contain systematic effects

CUBE also assumes that systematic effects have been cleaned / remediated from the data before it is CUBEed. In practice, this may not always be the case. Changing one or both of these two user-accessible parameters may also partially accommodate / remediate the existence of common systematic effects, such as the “smiles” and “frowns” that are characteristic of inadequate sound speed sampling.

The default value of *hes* may be overly pessimistic, in which case it unduly increases the propagated uncertainty of all soundings. Reducing the value of *hes* increases the influence of distance as compared to the influence of horizontal uncertainty, in diluting the certainty of the sounding. However, more importantly, it will reduce the overall value for propagated uncertainty, making hypothesis splitting more likely.

In one particular example [9] reducing the *hes* (horizontal error scalar) value from its default of 2.95 to 0.5 resulted in systematic outer-beam refraction errors being split into alternative hypotheses, which was not the case with the default value.

3.2.3 Soundings are correlated

Correlation among sounding measurements means that there is a common dependence upon various sources of these uncertainties. To some extent, sounding uncertainties are always correlated. The sounding uncertainty attribution process assumes correlation is zero. This can lead to lower uncertainty values than in reality. We may be able to partially compensate for this by changing the *hes* and *de* parameter values.

3.3 Gathering

Gathering involves the selection of which soundings will actually be assimilated at each node. The selection criteria are based upon distance from the node. The “capture distance” is the outer radius from the node within which data will be accepted. There are two user-accessible parameters that control the capture distance. The first is:

Capture_Distance_Scale: scale on predicted or estimated depth for how far out to accept data. Value is a percentage of depth used to limit the radius of influence a sounding may have on the grid. Value range {1.00 to 100.00} default = 5.0

This parameter determines the capture distance as a function of depth, with a default value being 5% of depth. For example, at 10 m depth, only soundings located within 0.5 m of the

node, or at 100 m, only soundings within 5 m of the node, will be considered in the estimation of the nodal depth and depth uncertainty estimates.

When data is sparse, the default value of 5% of depth may result in capture areas containing no soundings, leading to holes in the CUBE grid. In this case the `Capture_Distance_Scale` parameter could be increased, to fill in these holes. Of course widening the capture area may also result in some smoothing, as soundings from different seabed features and depths are combined. [CUBE's goal is NOT to do interpolation in the grid, which is properly a downstream process, after the best estimate has been determined.]

For shallow depths, the `Capture_Distance_Scale` parameter may result in too few soundings being considered. Therefore a second parameter is needed to set a floor below which the capture distance cannot fall.

Capture_Distance_Minimum: minimum value (in metres) for how far out to accept data. This value is used in conjunction with the `Capture_Distance_Scale` to limit the radius of influence of a sounding. Value range {0.0 and 100.00}, default = 0.5

In the case of very close (e.g. sub-metre) nodal spacing (and equivalently dense soundings), it may be appropriate to decrease the `Capture_Distance_Minimum` to half the grid spacing.

The maximum value of the two capture distance parameters is used to set the actual capture distance. Hence:

Capture distance = MAX (`Capture_Distance_Minimum`, depth * `Capture_Distance_Scale`)

3.4 Assimilation memory fading

In a sequential estimation process like CUBE, it is possible to design the process to “forget” or “discount” the influence of previous input contributions, as they “age” or fade into the past. This may be appropriate when the parameter being estimated is itself evolving with time (e.g. weather parameters). This is NOT appropriate with the release version of CUBE. The user-accessible parameter that controls this fading is:

Discount Factor: for evolution noise variance. Value range {0.8 to 1.0}, default = 1.0

This parameter should always be set to its default value of 1.0, or else removed as a user-accessible parameter.

4 – Tracking and Intervention

4.1 Bayes Theorem in Summary

Bayesian inference is the generic name given to the use of Bayes Theorem to discover something new about data, or to make some decisions about data. The key feature of the Bayesian approach to probability is that it allows the user to say something about the probability of hypotheses independent of the observed data via the prior distribution. In this way, it allows the analysis to include side information to constrain problems, a very powerful idea. This is also controversial, because it allows for the probability assessed for the same event by different people to vary, possibly dramatically, depending on the assumptions with which each person computes the value.

Most data processing methods rely on a model of the data being processed in order to allow predictions to be made; by comparing the model's predictions against the data being observed, information can be developed about the data's properties. Bayesian theory can be applied to this problem, typically in the form of a Dynamic Linear Model (DLM) which (in this instance) allows us to make a prediction about the new observations based only on our current state of knowledge about the data. The Bayesian processing of a DLM is optimal in the mean square sense under certain (fairly general) assumptions, and intuitively balances the estimates from previous data against new observations according to the relative uncertainties of the two.

Any model based analysis of data is only as good as the model. It is therefore important to make sure that the model still represents the data before using it to do the processing. Since the data can change dynamically, this process has to be done in real-time as the observations are being made. One scheme for doing this is to compute Bayes factors – the ratio of the probability of the observations under two different modeling assumptions – and check when the evidence starts to favor the alternative to the current model. A variant of hypothesis testing, this scheme allows the processing algorithm to determine when the input data is sufficiently different from the current model to indicate that some intervention is required. A number of different problem specific intervention schemes are possible. In CUBE, the simplest is used: each observation (including the first) that is different from all extant depth hypotheses at an estimation node starts a new hypothesis. All depth hypotheses are treated equally by the algorithm until a decision about which is most likely is required, since there is no way to tell which is going to contain the depth estimate that matches the true depth at the estimation node.

CUBE is a practical implementation of a DLM with Bayesian sequential testing model monitoring and 'fresh start' Multiple Hypothesis tracking. Due to its history as a research tool, many of the parameters used in the assimilation scheme are exposed to the user level and can be tuned for particular datasets. This does not imply, however, that this is necessarily a good idea – unless the user properly understands the implications of the tuning.

4.2 Applying Bayes Theorem to CUBE

During the process of assimilation of data, CUBE's estimator will occasionally receive data, which are inconsistent with anything that has been presented previously. This data could occur because of either a systematic miscorrection or a blunder, but in either case it is essential to detect the condition and correct for it since otherwise the inconsistent data will be mixed, leading to corruption of two or more hypotheses.

Conceptually, the process is one of comparison against the estimator's current state of knowledge about the potential depths. Initially, each node contains no information about

potential depths. When the first sounding works through the input queue structure (c.f. section 3.1), scatter, and gather computations (c.f. section 3.2 & 3.3) to be presented to the node, it is adopted as the first hypothesis about depth. The second sounding to be presented is compared with this hypothesis; if they agree on the depth within their uncertainty limits (as defined in more detail following), then the second sounding is integrated with the hypothesis; if not, then it forms the basis of a second hypothesis. Subsequent soundings are treated as the second, except that they are compared to all hypotheses extant at the node to determine the closest match, and are compared with the closest one for potential assimilation or intervention. This process is a variant of model monitoring with real-time intervention [8]

Given

$\hat{x}_{n|n-1}$ the current estimate of depth
 $\sigma_{n|n-1}^2$ the uncertainty associated with $\hat{x}_{n|n-1}$
 Y_n the sounding being tested for assimilation
 v_n the observation variance associated with Y_n

then

e_n the normalized difference between the observation and the current estimate of depth is given by

$$e_n = (Y_n - \hat{x}_{n|n-1}) / \sqrt{\sigma_{n|n-1}^2 + v_n}$$

The normalization is being done here with respect to the one-step prediction uncertainty (i.e., $Q_n = \sigma_{n|n-1}^2 + v_n$) and we assume as usual that Gaussian distributions are sufficient to describe the second order statistics of these data.

The fundamental assumption in CUBE is that we are estimating at a point position, and therefore that all appropriate soundings should agree on the depth. This assumption (null hypothesis, with probability P_0), applied specifically, means that the sounding being tested Y_n will be (statistically) consistent with the current estimate of depth $\hat{x}_{n|n-1}$.

The alternative to this null hypothesis (alternative hypothesis, with probability P_1) is that the sounding Y_n is separated from $\hat{x}_{n|n-1}$ by a (statistically) significant vertical step, the Estimate Offset Value h (the value of h is a CUBE parameter, described in §4.1).

The process of monitoring whether the null or alternative hypotheses are more likely is most simply described in terms of Bayesian sequential testing. In Bayesian inference new evidence (in our case the sounding Y_n) is used to update or newly infer the probability that a hypothesis might be true. When we have both a null and alternative hypothesis, a ratio of probabilities, called the Bayes factor (B), is used.

$$B = \frac{\Pr(E | H_0)}{\Pr(E | \text{not } H_0)}$$

where

E is the new evidence

$Pr(E/H_0)$ is the probability that the null hypothesis is true
 $Pr(E/not H_0)$ is the probability that the alternate hypothesis is true.
 B is the Bayes factor

When the new evidence to be considered is presented as a sequence (in our case new soundings Y_n $n = 1, 2, \dots$) we have a corresponding sequence of Bayes factors B_n

$$B_n = \frac{P_0(Y_n | D_{n-1})}{P_1(Y_n | D_{n-1})} = \exp\left\{\frac{1}{2}(h^2 - 2h|e_n|)\right\}$$

The parameters of h and the critical Bayes factor ratio threshold are used in CUBE to determine when an incoming sounding is inconsistent with the current hypothesis, and are explained in detail below.

This method allows a single outlier sounding to be selected, but would fail with a suitably slowly changing sequence of outliers, which can sometimes occur. In order to check for this, the algorithm also computes cumulative Bayes factors,

$$C_{n+1} = B_n \min\{1, C_n\}$$

and the number of samples for which the sequence has had Bayes factors partially supporting the alternative model (i.e., with $B_n < 1$),

$$L_{n+1} = \begin{cases} L_n + 1 & B_n < 1 \\ 1 & B_n \geq 1 \end{cases}$$

If the cumulative Bayes factor falls below the critical ratio, it is evidence that the incoming soundings are drifting in depth, and a new hypothesis should be started; if the sequence length exceeds a threshold, it is evidence that the alternative model has been favored marginally for some time, and hence that the data are drifting and deserve an alternative hypothesis. This run-length test threshold is the third intervention parameter described below.

4.3 Estimate Offset Value

The Estimate Offset Value (a.k.a., `monitor_offset`) describes the size of the vertical step which is considered significant. Since normalized differences are used, this is computed in terms of the standard Normal distribution. Evidence starts to turn in favor of the alternative hypothesis when $\log B_n = 0$, and therefore when

$$\begin{aligned} (h^2 - 2h|e_n|) &= h(h - 2|e_n|) = 0 \\ h &= 2|e_n|, h > 0 \end{aligned}$$

Hence, to consider a variation outside of the 95% CI significant, we would choose a critical point of $|e_n| = 1.96$, and therefore $h = 2|e_n| = 3.92$.

This value can be chosen in the range [0.1, 10.0], with default value 4.0. Setting this value smaller means that smaller deviations in depth from the current tracked hypothesis level will be considered significant, and setting the value higher will allow more variability in depth before the algorithm considers an alternative hypothesis as being required. As with other variables in this group, changing this parameter will primarily cause more hypotheses to be generated. Setting this value too small, however, will result in data that it is appropriate to assimilate being split into multiple hypotheses, with consequent weakening of the robustness of the algorithm. For most users, the default value should be appropriate; expert users might decrease this slightly to encourage hypothesis splitting when outliers hover just above or below the true depth, or where the estimated uncertainty for the soundings is believed to be too high and cannot be modified by any other means.

4.4 Bayes Factor Threshold

The Bayes factor measures the odds of the current model (H_0) against the alternative (H_1); conventionally, significant odds in favor of H_1 are indicated if and only if $\log B_n \leq -2$, or $B_n \leq 0.135$, and this threshold is the 'Bayes Factor Threshold' (a.k.a., `monitor_tau`) used in CUBE. The nominal value of 0.135 should be sufficient for most uses, although it might be revised upwards if the user wanted to construct hypotheses for smaller magnitude outliers. An upper limit of 1.0 is mandated by theory (otherwise the evidence is favoring the null hypothesis), although this would be inadvisable since any small difference between soundings would be considered evidence for a new hypothesis. Generating more hypotheses is not in itself dangerous, but it is inefficient since the operator has to inspect them all and determine whether the algorithm picked the correct one for reconstruction. In general, more hypotheses also makes it more likely that the algorithm will choose the incorrect one since each hypothesis will have less evidence associated with it.

4.5 Run-Length Threshold

The run-length threshold (a.k.a. `monitor_runlength`) value is the number of sequential samples over which the evidence is stronger for the alternative than the null hypothesis before the algorithm considers a new hypothesis is required for subsequent data. The new hypothesis is constructed from the data point being tested for assimilation. The default value is 5 samples on a range of [1,10]. This choice is essentially arbitrary, and there is a trade-off between choosing a small value where many hypotheses are generated, but each avoids any damage from outlier sequences, and a large value where fewer hypotheses will be created (by this means) but each one may be corrupted by the soundings that are incorporated before the algorithm detects the trend. For most users, the default value will most likely be sufficient; expert users might reduce this value to encourage sharper hypothesis generation with sparse data.

5 - Disambiguation

Disambiguation is the process by which CUBE attempts to choose one of the potential depth reconstructions (in CUBE-speak, 'hypotheses') at a node to represent its best estimate of the depth there. The notion of 'best' is of course variable: what is 'best' in one case may not be 'best' in another. In order to allow the user to make decisions on this, the CUBE algorithm is built to store and maintain all hypotheses, so that the rules by which the 'best' is chosen can be changed by the user at run-time. This means that the user can mix and match rules for hypothesis selection, per node if necessary. In general, it is better practice to use one method over a given area since otherwise it is very difficult to keep track of which rule was used where, but the extra flexibility is available if required.

How to choose a hypothesis as 'best' can depend on the characteristics of the hypothesis, its metrics, and its relationship to the other nodes in the area. Of all of the components of CUBE, this is the one that can be different in separate implementations of the algorithm. Commercial vendors are encouraged to add new rules to the algorithm if they think they have a better method for selecting the 'correct' hypothesis more often. As supplied in source-code form, CUBE has four rules that are normally selected by changing the 'extractor' parameter in the configuration file:

1. 'prior'. This uses the logic that the hypothesis that has assimilated the most soundings is most likely, and selects accordingly. This works well in most data, and is very fast in operation, but can fail with burst-mode blunders. This is also known as 'density' mode in some implementations, and 'Num. Samples' in others.
2. 'likelihood'. This attempts to get around the problem of burst-mode blunders by looking out from the node of interest for another node that has only one hypothesis. Theorizing that this single hypothesis is likely the true depth, the node of interest selects the hypothesis closest in depth to this 'guide' value. The minimum and maximum search radii can be set to the user's understanding of the effects of noise in the dataset. This method is more powerful than `prior`, but is slower due to the local searching required. This is known as 'locale' mode in some implementations, and 'neighborhood' mode in others.
3. 'posterior'. This is a combination of `prior` and `likelihood` that attempts to blend the best of both methods. This is generally shown as the concatenation of the two component rules in implementations.
4. 'predsurf'. This attempts to avoid problems with local tests by appealing to a reference surface (e.g., from previous surveys or low-resolution version of the data constructed by some other means), and selecting the hypothesis closest in depth to this reference value. This is intermediate in speed between `prior` and `likelihood` methods, but suffers from the problem that obtaining sufficiently good predictions of depth to make it worthwhile can be quite difficult. This is known as 'Initialization Surface' in some implementations.

Some commercial implementations use these names; some use synonyms. It is not required that all of these methods are used in all implementations, and some may have more methods. The rules described below are known variants of, or differences from, the canonical methods.

5.1 CARIS's Locale Rule

The CARIS implementation of CUBE has a difference in its Locale rule which makes it more like the `predsurf` canonical method. Here, the algorithm does not use the depth of a neighboring node's single hypothesis to provide a guide as to which hypothesis to pick, but instead computes a slightly robustified mean value of the 'best' hypotheses' depths in the neighbor nodes, and picks the hypothesis closest in depth to this value. That is, at each node of interest:

1. Establish a radius of influence around the node, as controlled by the `locale_radius` parameter. This value is in nodes, rather than meters. Only nodes within this radius are used to determine the robustified mean depth.
2. For each node in the influence radius except the node of interest, determine if the hypothesis strength is sufficiently low (i.e., that CUBE's assessment is that the reconstruction is sufficiently robust to be reliable); if so, the selected 'best' hypothesis at that node is used, if not the node is skipped in the remaining computation. This threshold is set by the `locale_strength_max` parameter, and has the same scale as the hypothesis strength parameter in CUBE.
3. For all remaining nodes in the influence radius, compute a trimmed mean of the reconstructed depths. The trimmed mean is computed by removing the shoalest and deepest soundings, and then computing the arithmetic mean of the remainder.
4. At the node of interest, select the hypothesis with depth closest to this robustified mean value as the 'best' reconstruction.

Some operator experimentation with both of the control parameters here will probably be required to find an optimal setup for this rule.

5.2 CARIS's 'Null Hypothesis' Rule

CUBE is designed to examine all soundings and report what it finds. If there are only 'blunder' soundings, then CUBE will report the blunder as the best available data in the region. This often happens at the edge of a survey area where there is very shallow water since MBES systems have difficulties with multipath returns in these areas. The consequence is a scattering of nodes with one or more hypotheses, occasionally on 'dry land' due to the blunders, and frequently at odds with any reasonable depth reconstruction in the area. The operator has to examine all of these, and eliminate them, before the outputs are suitable for use, which can be time consuming.

CARIS's 'Null Hypothesis' rule is an attempt to reduce this problem. (Note that the use of 'null hypothesis' is not correct in the statistical sense and should not be confused with its use in the section on intervention; here it means whether the node has an acceptable reconstruction or is marked as NULL, or not for use.) This is not another disambiguation rule in the sense of those above, but an extra test that is used to try to eliminate these isolated 'noise' reconstructions.

The algorithm can be turned on or off with the `enable_null_hypothesis` parameter, and is by default off. As the algorithm operates, it tests each node against its neighbors in order to determine whether it is a reliable reconstruction or not. The algorithm proceeds as follows:

1. For each of the 8-connected neighbors of the node of interest, count the number that contain at least one hypothesis. If this is greater than the `null_hypothesis_min_neighbours` parameter, then the node can be further tested; if it is smaller, then the node is marked NULL, and no reconstruction is made during the disambiguation process.
2. For survivor neighbors, the algorithm computes the standard deviation of the 'best' reconstructed hypotheses' depths, and then again leaving out the node of interest. The ratio of these standard deviations is then compared to the `null_hypothesis_ratio` parameter; if the ratio is higher than the parameter, the node is marked as NULL and is not reconstructed.

In general, the nodes that are being targeted by this method occur as 'salt and pepper' events, and have few or no connected neighbors. Therefore, setting the `null_hypothesis_min_neighbours` parameter too high will result in even fully connected neighbors being marked as NULL, or may increase the size of any missing nodes in the output, since any node adjacent to a node with no hypotheses – and hence no reconstruction – will have fewer valid neighbors.

The logic behind the standard deviation test is that a 'blunder' hypothesis is likely to be formed from a depth estimate that is very different from the true depth in the area. Therefore, the standard deviation estimated over all nodes will be significantly higher than that computed without the node of interest. Outlier hypotheses will therefore have very high ratio values, and will be marked as NULL. This is a form of leave-one-out testing like the quotient limit in the assimilation component of CUBE (see section 3.1.2). It is possible to form sample statistic distributions for this ratio and conduct hypothesis tests for whether the difference in standard deviations is significant or not, but this is not implemented. Some operator experience (and testing) would generally be used to select a 'good' threshold value that eliminates the 'noise' nodes without significantly effecting the remaining reconstructions.

5.3 CARIS's Blended Density/Locale Rule

The CARIS implementation of CUBE has a 'try harder' mode, where a simple rule is used to start with, and then a more robust rule is used if the result of the first test is suspect. The implementation in HIPS 6.0 uses the 'density' mode (a.k.a. `prior` mode) as the first test since it is fast, and their modified 'locale' mode (see above) as the follow up test. The decision about when to try the second test is made based on the hypothesis strength of the first attempt at reconstruction, in effect testing CUBE's belief in the robustness of the reconstruction. If the value is higher than the `density_strength_cutoff` parameter, the second test is conducted to determine the final hypothesis selection. This parameter has the same range as the hypothesis strength ratio itself, and some operator experimentation may be required to find an optimal value for the particular dataset. Setting the ratio threshold to 5.0 results in no second tests; setting it to 0.0 results in second tests for all nodes with any doubt about the

reconstruction. There are time implications in doing too many second tests, and robustness implications in doing too few. A compromise value is most likely to give the best blended results, but the magnitude of this value may vary from dataset to dataset depending on the number of blunders expected, density of blunders, and other parameters that affect the number of hypotheses that are constructed.

6 - Summary

This CUBE User's Guide is an attempt to explain, in sufficient detail to permit informed use of CUBE, how CUBE works, the role of the user-accessible CUBE parameters, and the conditions under which it may be appropriate to change the value of one or more of these user-accessible parameters away from their default values.

We have re-iterated several times through the document that CUBE must be treated as a new, powerful bathymetric data processing "power tool", but only as a tool. It does not replace the judgment and experience of the bathymetric specialist. CUBE results should always be evaluated for reasonableness against expected results. CUBE results *assist* in bathymetric decision-making, but do not replace sound bathymetric practice and experience.

That said, CUBE does provide several advantages over previous approaches to bathymetric processing. It is objective. It provides repeatable results. It is efficient. It makes very effective use of uncertainty attributes that are more commonly becoming attached to sounding measurements. It provides an equally spaced gridded surface product. Within the limitation of the axioms listed in Appendix A, CUBE will "tell the truth about the data" better than traditional approaches.

CUBE is being widely accepted and implemented in bathymetric processing software. It represents a dramatic paradigm shift from traditional processing methods. This shift is best captured by changing the question to be asked from "How good is this sounding?" to "What is the best estimate of depth (and depth uncertainty) at this location?"

CUBE performs three functions: assimilation, intervention and disambiguation.

Assimilation is the assembling of statistically based nodal depth estimates and depth uncertainties, based on input sounding measurements and their 3D uncertainty estimates. Three stages in CUBE assimilation are reordering, scattering and gathering. Reordering delays the consideration of outliers. Scattering is the presentation by each sounding measurement of itself as a candidate for updating depth estimates at several surrounding nodes. Gathering is the determination of which surrounding soundings will be accepted for updating the nodal depth estimate.

Intervention is the decision to interrupt the assimilation process when an incoming sounding measurement is not statistically compatible with the existing nodal estimate. Intervention results in the creation of an alternative hypothesis concerning the "true" depth - a separate tracking of the nodal depth and depth uncertainty. Intervention allows CUBE to respond appropriately to several failure modes: soundings from multiple real objects (seabed and school of fish; wharf pilings and seabed beneath wharf), burst-mode blunders (echosounder bottom detection failures), and systematic uncertainties in the soundings.

Disambiguation is the selection of which of several alternative hypotheses is most likely to represent the "true" depth. Several metrics may be used. However at present the popularity contest, and the locale metrics are most commonly used.

7 - References

- [1] Hare, R., A. Godin and L. A. Mayer. *Accuracy Estimation of Canadian Swath (Multi-beam) and Sweep (Multi-transducer) Sounding Systems*, Tech. Rep., Canadian Hydrographic Service, Ottawa, ON, 1995.
- [2] Calder, B. R. and L. A. Mayer. *Automatic Processing of High-Rate, High-Density Multibeam Echosounder Data*, *Geochem. Geophys. And Geosyst. (G-Cubed)*, DID 10.1029/2202GC000486, 4(6), 2003.
- [3] Calder, B. R. *Automatic Statistical Processing of Multibeam Echosounder Data*, *Int. Hydro. Review*, 4(1), 2003.
- [4] Calder, B. R. and S. M. Smith. *A Time Comparison of Computer-Assisted and Manual Bathymetric Data Processing*, *Int. Hydro. Review*, 5(1), 2004.
- [5] Smith, S. M. *The Navigation Surface: A Multipurpose Bathymetric Database*, Masters Thesis, Center for Coastal and Ocean Mapping & Joint Hydrographic Center, University of New Hampshire, 2003.
- [6] Calder, B.R., J. S. Byrne, B. Lamey, R. T. Brennan, J. D. Case, D. Fabre, B. Gallagher, R. W. Ladner, F. Moggert, M. Paton. *The Open Navigation Surface Project*. *Int. Hydro. Review*, 6(2), 2005.
- [7] Moggert, F., and the Open Navigation Surface Working Group. Description of the Bathymetric Attributed Grid Object (BAG) Version 1.0.0, online: http://www.opennavsurf.org/papers/ons_fsd.pdf, 2006.
- [8] West, M. and J. Harrison. *Bayesian Forecasting and Dynamic Models*, 2 ed., Springer-Verlag, New York, 1997.
- [9] Smith, S. M. *CUBE workflow on TJ (and implementation at NOAA's Office of Coast Survey)* Presentation at CARIS 2005, Halifax, NS, October 2005.
- [10] International Standards Organization (1993) *Guide to the expression of uncertainty in measurement (GUM)*. Prepared by ISO Technical Advisory Group 4 (members being nominated by 7 organizations: BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, OIML). American version available as American National Standard for Expressing Uncertainty *U.S. Guide to the Expression of Uncertainty in Measurement*, ANSI/NCSL Z540-2-1997, National Conference of Standards Laboratories, Boulder.
- [11] Bayes T. (1764) *Essay Towards Solving a Problem in the Doctrine of Chances*. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London*, 53:370–418 (original), and Bayes, T. (1763/1958) *Studies in the History of Probability and Statistics: IX. Thomas Bayes's Essay Towards Solving a Problem in the Doctrine of Chances*, *Biometrika* 45:296–315. (Bayes's essay in modernized notation)

Appendix A: CUBE Axioms

These are the accepted truths upon which CUBE is built. When any of these axioms are violated, CUBE may tell us something important about the data, but CUBE will not necessarily be able to perform its primary function - to provide estimates of depth and depth uncertainty that are objective, reliable representations of the “truth”.

A.1 Most data, most of the time, are mostly okay.

This axiom has two practical consequences. Before sounding data are presented to CUBE, some preprocessing is required so that:

Blunders (outliers, inappropriate soundings) exist, but not commonly.

Systematic uncertainties have been remediated (to the extent possible).

A.2 A sounding (the noun, not the verb) is not a depth

This axiom is new, when stated so starkly. It highlights the fallacy of the “golden sounding”, based on the belief that soundings are “real”.

A favorite topic kicked around by hydrographers, usually when surrounding a pitcher of beer is: “What does an echo-sounder measure, anyway?” There are 200 kHz “depths”; 38 kHz “depths”, the “depth of refusal” for a lead-line weight (which in turn depends upon its shape and weight), a different depth of refusal for the sounding pole, and in clear shallow waters the “LIDAR depth”. More specifically for MBES, there are the “amplitude-detect depth” and the “differential phase-detect depth”. In general all of these “soundings” (results of attempted depth measurement) may yield different values. Are any equal to the “true” depth?

However, this axiom is more profound than these concerns. Hydrographers were softened up to accept this axiom when IHO Special Publication 44 (4th edition) drew a distinction between measured soundings (read measurements), and “bathymetric models” (read depths). This distinction fundamentally changes hydrography – see Axiom 3.

A.3 The appropriate question to ask is “What is the depth (and its uncertainty) here?” NOT “How good is this sounding?”

Maybe this is a just a corollary of Axiom 2. But I think it has enough substance to qualify as an independent axiom.

This axiom has fundamental and far-reaching consequences, mainly within the souls of traditional hydrographers. It means the intent of data cleaning is the estimation of derived parameters (depths at nodes), rather than assessment / flagging of measured soundings. This is a major paradigm shift for those whose careers have been to select appropriate subsets from available soundings. It will take some getting used to.

However, one argument advanced to support this approach is that what we call “soundings” these days are themselves estimated derived parameters, based on a multitude of measurements (two-way travel time, beam direction, roll, pitch, heave, dynamic draft, etc.), and

processes (ray tracing using an imperfectly known sound speed profile, etc.). So why not go that one tiny additional step, and derive best estimates for depths at nodes, letting the “measured soundings” drop through the same one-way trap door that the two-way travel times, roll measurements, etc. dropped through when we derived the sounding values in the first place?

This debate will rage on.

A.4 Data cleaning, in whatever form, is basically a Triage Operation.

The goal of Algorithmic Hydrography is to extend mankind's grasp: to provide power tools that reduce mankind's workload; wherever possible to replace subjectivity with objectivity. Well, maybe not the last – perhaps the selection of appropriate algorithms remains somewhat subjective. Even the selection of these Seven Axioms could be viewed as somewhat subjective. At least an (appropriate, non-self-learning) algorithm will yield repeatable results every time it is run, which is more than can be said about human hydrographic data cleaners.

Quibbles aside, Algorithmic Hydrography tools always divide data into

- (a) clearly bad,
- (b) clearly good, and
- (c) questionable, requiring further attention.

These tools should be judged on how much of that “further attention” requires (expensive, subjective) human intervention. Here CUBE (potentially) shines. The CUBE *Disambiguation Engine* (more on this below) is designed to provide as much “further attention” as the ingenuity of the “CUBE community” can devise over the next several years (or at least as long as they are still having fun with it).

CUBE allows adaptive sampling. When CUBE and human intervention in the office fails to resolve alternative hypotheses with appropriate certainty (this could happen in near real time), collect more (or different) data to support or amend the disambiguation.

CUBE can be used for survey optimization. Collect your MB data with minimal or no overlap, run it through CUBE. If the uncertainty surface is within specifications and no unresolved blunders exist, move on to the next area. If problems are apparent, collect additional data until specifications are met and all blunders and alternative hypotheses have been resolved.

A.5 Unlike man, not all soundings are created equal (hence, every sounding solution must include estimates of its uncertainty).

This axiom is a familiar mantra in geomatics. It is essential to any form of uncertainty propagation, including within CUBE.

Promisingly, it is a key feature of IHO S-44, 4th edition, and of the ISO 191xx family of standards, and consequently is starting to be taken seriously by the hydrographic community (and the wider geomatics community).

CUBE advances the importance of this axiom in a major way, by demonstrating what can be achieved when it is satisfied. The question remains about where these uncertainty estimates should come from. There are four general answers.

The fourth best approach is to use a non-discriminatory regional model, such as that represented by “source diagrams” or the ZOC approach. This would not work with CUBE.

The third best approach is to use a predictive model for uncertainty, such as those developed by Rob Hare and his colleagues. This is the approach upon which CUBE is based.

The second best approach is to allow the data itself tell us about its uncertainty, using redundancy tests like cross-line analysis (e.g. comparing nadir beams against outer beams to calibrate the beam-angle-dependent component of the predictive uncertainty model). For this approach to be useful, cross-lines and their processing have to be very well controlled, and in any case this provides only a “reality check” of existing predictive models. Perhaps future tools growing from CUBE will make use of this approach.

The best approach is all of the above. Use the best predictive models, but calibrate these with initial patch test results, and ongoing cross-line redundancy analyses. This is what the kriging software HydroStat does in later incarnations – leave some sample points out of the weighting calculation, to be used to calibrate the model.

A.6 MBES data density and spatial resolution are sufficient to capture all bathymetric features of interest.

This axiom is the hydrographic equivalent of “you can’t make a silk purse out of a sow’s ear”. It recognizes that there will always be features (small sand ripples, scour, pebbles or boulders) that are so small that the MBES data will not be capable of detecting them.

A.7 Models represent truth

Models are what we can derive from measurements, and estimation processes driven by those measurements. Even the most detailed and sophisticated model is a simplified description of the complexity that is Mother Nature.

CUBE and other estimation models are based on mathematical processes. The real world in which we live, touch, see, smell, hear, and taste, but will forever imperfectly understand, is much more subtle and cunning than anything based purely on reason.

This is clearly distinct from the abstract world of mathematics, which is built on logical rules and processes (such as these axioms and process steps), is a human creation, and well within human understanding (at least those of us that are mathematically aware).

The symbiotic relationship between the real world and the model world (of mathematics) works something like this:

(a) In the real world, we ask a question, about something in the real world. Hydrographically, this question could be “Will I run aground if I head in this direction?”

(b) We build a mathematical model that simplifies the aspects of the real world that are relevant to the question. This is the transition {real-to-model}. Hydrographically, this would be a bathymetric model based on an echo sounding survey.

(c) Inside the model world we pursue logical well-understood processes that lead to some conclusions that might be relevant to our original question. Hydrographically, this is where CUBE lives.

(d) This fourth step is the tricky one, and the subject of this axiom. It is the reverse transition {model-to-real}. HOW relevant are the model conclusions to the real world? This is often labeled "interpreting your results". What real-world conclusions can we draw from the model-world conclusions? There are no easy answers here. It all depends upon how "realistic" (close to real-world complexity) the model is. We should never take the validity of model results for granted. Hydrographically, this means, "the prudent navigator never depends upon a single source of information." Only semi-facetiously, perhaps the prudent hydrographer should never rely on the same sensor to resolve alternative hypotheses!

So the bottom line of this axiom is that the model world bottom line does not necessarily represent the real-world bottom line. Take all model world results with a grain of salt. Test them against real-world experience. Sometimes human experience and gut feelings are more reliable than model results.

Appendix B: Uncertainty attribution

CUBE assumes, and requires, that each sounding under consideration has an associated uncertainty. That is, some measure of the expected variability that we should see in the data measurements that we are using. Since the data we are concerned with are soundings in three dimensions, the uncertainty is also three dimensional and we should consider a complete three dimensional ellipsoid of uncertainty around each point. In general, however, it is difficult to work with full three dimensional uncertainties, and we specify the uncertainty as a horizontal component and a vertical component. It is a mistake to assume, however, that there is no correlation between horizontal and vertical uncertainty because of this: in any location where there is a slope of any kind, any horizontal uncertainty engenders a vertical uncertainty. We might assume that the ellipsoid's axes differ from the local-level by at worst a rotation about the vertical axis, but we cannot assume that the axes are decorrelated because of this.

CUBE itself does not compute uncertainties, relying on some external agency to provide the attribution. (There is sometimes some confusion on this, since the distribution of CUBE source code contains a module that does uncertainty attribution; it is not part of the core CUBE algorithm, however – merely an adjust to the code until the (long anticipated) day when all data will be recorded with uncertainty as a matter of course.) However, to properly understand how CUBE behaves, it is important to understand its underlying model of uncertainty and how such uncertainties are estimated. This appendix describes some of the background theory on uncertainty estimation, and common models and methods for applying this in practice.

B.1 The Meaning of an Uncertainty

B.1.1 Origins of Uncertainty

The sounding uncertainty arises from a number of potential sources, both in the measurement process and in the modeling process. The former is simpler to understand. In essence, it is simply acknowledging that no measurement system is perfect. No matter how hard we try, every physical measurement is affected by noise of some kind, with the result that we are always slightly unsure of the true value of the physical property being measured. For example, what we measure with an acoustic system is time – generally two-way time of flight from transmission of the acoustic energy to the seafloor and back to the transducer. However, we measure time with reference to a clock of some kind, possibly just an interval timer that is reset to zero when the acoustic energy is released, and this oscillator is subject to variations in the rate at which it counts out seconds due to temperature variations, fluctuations of the voltage supply to the circuitry, and aging among other problems. Because of this, the relationship between an apparent second as measured by the oscillator and a true SI second is complex and variable from event to event. There is an error in the time indicated by the oscillator, and consequently anything that we compute from the two-way travel time so derived is also affected by this variability. In particular, when we compute the nominal depth by scaling the time by the speed of sound, the depth so estimated would vary from ping to ping even if the sounder was absolutely stationary with respect to the local-level coordinate system, and no other physical changes took place in the system. We can make the same assertion about any other measurement that we take – the sensors in a CTD system, for example, are affected by similar problems, and the temperature, pressure and conductivity are therefore subject to small variabilities that we cannot predict in advance, cannot measure directly, and which change from measurement to measurement. So when we finally combine the conductivity, temperature, and

pressure measurements to form an estimate of sound speed, it inherits the base uncertainties; when combined with the two-way travel time to estimate depth, we're combining two measurements each with their own uncertainty. It is remarkable that the depths that are estimated are anything close to the true value.

The second problem to examine is that of model uncertainty. In order to make predictions about something that we don't measure, a mathematical model is required relating that thing to measurements that we can make. Often, the models are not especially complex, although apparent simplicity is sometimes misleading. For example, for a vertical beam echo-sounder, we could estimate the depth as $ct/2$ for speed of sound c (ms^{-1}) and two-way travel time t (s). Buried under this example, however, are a number of assumptions. We are assuming that either the speed of sound is a constant in all of the water through which the sound travels (which is unlikely in the wild), or that we have somehow computed a nominal speed of sound which compensates for the changes in sound speed with depth and/or environmental conditions. For the latter case, we obviously need another model to do the computation; typically, the geometric mean sound speed is used. To compute this model, however, we need to have measurements of the speed of sound at all depths between the echo-sounder and the seafloor, which requires that we use a CTD probe before we take the acoustic measurement. The CTD probe does not measure the speed of sound, however: it measures the conductivity, temperature and pressure of the local water and computes the speed of sound, another model. In addition, the CTD probe cannot measure speed of sound everywhere and at all times; when we correct for speed of sound while making depth measurements, we are assuming that differences of sound speed in time and space are small, so that a measurement taken elsewhere can still be used for the current acoustic event. (Otherwise we would spend all of the time making measurements with the CTD probe and never use the echo-sounder!)

Clearly, even the simplest measurement model can result in a complex array of sub-models, each of which has its own assumptions about particular aspects of the natural world. These assumptions are generally made because if we did not, it would be very difficult, or expensive, to do the computations required in order to transform our measurements into the information that we require. Each model's assumptions, however, abstract the final result a little more from the real world about which the measurements are meant to be informative. Consequently, every time we apply a model to the estimation process, we induce a little error between what the model says the real world looks like, and what the real world actually does. In essence, we are trading computational tractability for physical veracity. The prime question is whether doing so significantly or adversely affects the final predictions that we make.

B.1.2 The Difference Between Error and Uncertainty

In the uncertainty modeling literature, it is common to see the terms 'error' and 'uncertainty' used, rather loosely, as synonyms. Common terms like 'Total Propagated Error' reinforce this trend. There is, however, a very important technical distinction between error and uncertainty.

Each measurement that we make attempts to describe the true value or state of some physical entity. The error in the measurement is the difference between the true value of the measurand and the measurement that we make. The true value of the measurand is, however, unknown and unknowable: if we knew what the true value was, we would not need to make the measurement. Consequently, we cannot ever compute the actual error in the measurement that we make. At best, all we can do is to compute an estimate of the likely magnitude of this error, given what we know about the measurement system and procedures.

This summary description is the uncertainty of the measurement: a statistical description of the likely variation of any measurement (rather than the particular measurement) about the true value of the measurand, with respect to some probability distribution. It is therefore incorrect to describe this as an error, or to provide an uncertainty unless we also say what the assumed distribution is, and how the summary was derived.

B.1.3 Modeling of Uncertainties

In order to make progress with turning measurements into information, then, it is evident that we need to be able to say something useful about the uncertainty in order to take it into account in subsequent processing and analysis. The primary goal here is to provide some means by which any individual sounding can have, attributed to it, a horizontal and vertical uncertainty, irrespective of the features of the sounding such as position with respect to the local-level, depth range, sounding mode, etc. In general, this means that we have to have a predictive model of some kind, since it is unlikely that we can gather enough data to drive a fully empirical model (this is taken up in more detail in B.2-B.3)

Since we cannot say anything directly about the error of the particular measurement, it should be understood that we are attempting to summarize the likely variability of any measurement, from all potential sources. The natural choice for this is to develop a statistical description of the likely distribution of the true error for any measurement, and therefore we focus on estimating the parameters of a suitable probability distribution to describe the likely variability of the measurements. The actual distribution of each measurement may be evident from the mechanism by which it is made; backscatter from an active sonar, for example, is typically modeled by either a Rayleigh, Rice, Log-Normal or K distribution. Working with these distributions can be problematic, however, and frequently we make the simplifying modeling assumption that the distribution is Gaussian, and therefore can be summarized completely by its mean value (usually zero) and variance (or equivalently, standard deviation). Typically, either the standard deviation or a scaled version of it is used as the summary of the variability of the measurement.

In practice, the use of Gaussian distributions is so common it is frequently the case that an uncertainty is given in the form '0.23 m (95%)'. This should be taken to mean that the probability of the measurement being within 0.23m of the true value is 95%, given that the distribution is Gaussian. Knowing percentage points of the Gaussian distribution (or looking up a suitable reference) shows that this occurs at 1.96 standard deviations from the mean, and therefore that the standard deviation in this case is 0.117m. This sort of description is acceptable as long as there is a generic statement somewhere in the supporting document to the effect that these assumptions should be made unless otherwise stated.

Two immediate complications ensue with this simple scheme. First, not all uncertainties are caused by random variabilities, and second, not all known modeling uncertainties can be well calibrated.

The former complication is well understood in the mensuration community, where it is often the case that a particular measurement may have some term that deviates from the true value, but does not change from measurement to measurement. Therefore, as measurements are averaged to reduce the uncertainty, this bias term, or systematic uncertainty, remains. A simple hydrographic example is the case where one water-level correction is used for multiple soundings. The water-level correction is computed from other data, and therefore has an associated uncertainty (both measurement and modeling). However, because the correction is

the same for any soundings averaged together, the averaging process only reduces the relative uncertainty of the depth being corrected, and not the uncertainty of the depth with respect to datum.

In practice, it is rare that this effect is significant when considering the uncertainty of individual soundings, as is the case here. There are seldom directly repeated measurements to average, and therefore the distinction of a systematic uncertainty is lost. That is, for one measurement, the systematic component is a realization of the underlying systematic distribution, and therefore essentially equivalent to a random uncertainty for that specific measurement. In addition to this effect, it is very difficult to determine the likely value of systematic uncertainties in general, since the true value of the measurand is not known. Although it is possible to do this by field calibration with portable standards (a special example is a bar-check on VBES systems, although this is by no means a straightforward process for MBES systems), circumstances when this is practical are rare. Consequently, it is unusual to see systematic uncertainties carried through in the theoretical models used for uncertainty measurements on MBES systems, and impossible to estimate them in data-driven (reductive) models.

The latter complication deserves more attention, since it can be much more significant for practical purposes in modeling. As mentioned previously, it is often the case that when a model is built for some physical system, some simplification is required to make the model tractable. The same situation applies in some measurement systems, where the value of a measurand cannot be easily estimated. For example, consider the dynamic draught of a small launch. In theory, the draught of the launch changes as a function of time and speed through water: as fuel burns, the launch becomes lighter and rises; as the speed through the water increases, the launch might dig down by the stern, or start to hydroplane. In either case, the depth of the transducer with respect to the mean water surface changes, and therefore a correction is required to the depths. Although settlement and squat trials are generally conducted either with a shore station or a kinematic GPS solution to resolve the latter, estimating how much the launch rises due to fuel burn is very difficult. If the launch is also refueled from a mother-ship after each day's survey, then it is unlikely that the tanks will ever be run completely dry, and therefore it would be very unlikely that the full range of potential draught effects would be observed. Under these circumstances, what is the appropriate model for loading effects?

It could be argued that this is a static offset, and should be corrected on all depths before use; not correcting for it induces a systematic residual uncertainty. Assuming, however, that the static draught is measured when appropriate, it is fair to say that the amount of fuel burned off during any one day's activities is unlikely to change the draught of the vessel significantly, and that determining the rate of burn in order to make the correction would be very difficult to implement reliably. Therefore, as a simplifying assumption, it might be justified to ignore the correction completely, and accommodate this in the uncertainty model by adding a small component to cover the expected variation in loading that is being ignored. The assumption here is that for any one sounding, its position in the loading cycle is essentially random. Therefore, the correction not being made can be well approximated by a random variable drawn from a suitable distribution about the mean load draught, and therefore can be well modeled by a random uncertainty added to the vertical corrections budget like the other random components. The magnitude of the component being added would have to be determined, either from previous experience of fuel burn rates, or by conducting an experiment to show the mean burn over a full tank of fuel, or over a 'typical' survey day. Whichever method is chosen, however, the result is a system with very good predictive power, but much simpler implementation than a one where the loading had to be computed for each sounding.

This modeling decision is trading fidelity for complexity in implementation, which is a common occurrence. This is a process that has to be done with some care; the special circumstances that allow the simplification to be made are not guaranteed to be universally valid, and implementations should take care to understand the assumptions implicit in each model in order to avoid mistakes. Used properly, however, this is a very pragmatic solution to a problem that might be difficult or impossible to resolve in any other way.

The goal of any model is to provide insight into the phenomenon being modeled, rather than to be an absolutely perfect model of the physical nature of the thing itself. This is especially true in uncertainty modeling, where our goal is to understand the likely variability of our measurements and derived quantities. It is important to remember, however, that the goal is insight – if the model has to be abstracted from reality a little in order to gain that insight, we should not be overly concerned. As a rule of thumb, you might care very much about being 10% out in the value of an estimated quantity, but nobody is going to be concerned about whether you are 9% out, or 11% out: the same relative variability in the uncertainty is typically insignificant. Hence, the goal of most uncertainty models is to achieve a ‘first order’ accuracy estimate of the uncertainty, in the sense that the estimate has to be roughly of the right magnitude, rather than attempting to tie down every loose end. When the magnitude of the loose ends can be on the order of 0.1% of the total uncertainty budget, a very strong case would have to be made for chasing down a better estimate of the component in question. Most models, then, attempt to focus on the big-picture components of the measurement uncertainties being used, and neglect smaller details which are generally insignificant. Understanding when such decisions are appropriate, however, is essential in making principled use of this type of model in analysis of data.

This is not to say that there is no relationship between the uncertainty model and reality; if the uncertainty model is not close to what is observed then it is of very limited use. What we are looking for is a reasonable first order accurate model that explains the most significant components of what we observe in nature. In practice, this is enforced by the CUBE algorithm itself in the sense that the algorithm will exhibit very different behaviors if the uncertainty model predicts either too high or too low. If the uncertainty is too low, for example, significant structures will not generate alternate hypotheses, and the effect will be clearly evident; if the uncertainty is too high, many more hypotheses than are appropriate will be generated, splitting the surface. Either condition is readily detected as a deficiency in the uncertainty model during inspection of the data. This is colloquially known as the ‘Goldilocks Theory’ of uncertainty modeling.

B.1.4 Types of Uncertainty Model

For processing data, for example with CUBE, the requirement is for a predictive model. That is, a model that takes the information about each sounding – two-way travel time or range, angle, MBES type, raw measurement uncertainties, etc. – and computes the likely uncertainty in horizontal and vertical planes for it. These sorts of models are covered in Section B.2.

Traditionally, there has been some reluctance to use these for serious work, however, with the feeling being that they were too complex, or required too many free variables to be set. The conventional method in hydrography has generally been to examine the soundings after they have been collected, attempting to deduce some summary of their behavior by reducing them to some summary statistics. Methods such as cross-line analysis are common, and are covered in Section B.3.

In reality, however, both methods are required. Fundamentally, reductive models only report on repeatability of the measurements that are being generated. That is, they measure the relative uncertainty of the soundings against each other, but say nothing about the absolute uncertainty of the sounding with respect to some absolute datum. One could argue that, repeated indefinitely, such methods would eventually approach the true uncertainty of the soundings from below, but as a practical method this does not have much to recommend it. Few surveys have indefinite time to devote to such practices.

On the other hand, it is immediately evident that a predictive model's predictions are only as good as the calibration parameters used to describe the fundamental measurement uncertainties associated with the model. If the user mis-calibrates the driving uncertainties, the predictions could be higher or lower than the true uncertainty with respect to absolute datum, and/or higher or lower than the observed repeatability that would result from a reductive model. Given that hydrographers are by nature conservative and cautious, it is more likely that the uncertainty from a predictive model will over-estimate the true uncertainty and converge to it from above.

Finally, the primary aim of the methods described here are to attribute new soundings with uncertainties. Unfortunately, however, most data being used in hydrographic products was collected prior to a full and nuanced understanding of the reporting requirements for uncertainty attribution and consequently has no uncertainty attribution, and frequently does not contain the information required to compute an uncertainty attribution. The task of uncertainty attribution for sparse data is still a notoriously unsolved problem, but some models have been developed for dealing with data that might be described as 'semi-dense' in the sense that it was collected in a systematic manner for hydrographic purposes, or may be derived from MBES soundings summarized at the smoothsheet/fairsheet scale. For completeness, some details of these types of models are outlined, along with their problems, in Section B.4.

B.2 Predictive Forward Modeling

Predictive forward modeling describes the process where a mathematical construct is used to compute uncertainties for a specific sounding based on the measurements and meta-data associated with the sounding. That is, based solely on the current measurement, a probable uncertainty is derived, potentially using some calibration constants for the fundamental measurements that are derived from previous experience. The characteristic of this type of model is that it is driven only by calibration constants and therefore can make predictions for any sounding rather than having to interpolate or extrapolate from other data observations.

B.2.1 Hare-Godin-Mayer MBES Model

The Hare-Godin-Mayer model (HGM) was first developed in 1995 for the Canadian Hydrographic Service [x], and was subsequently published in 19xx [y]; subsequent revisions have been made to the basic model, with the most recent being defined in a technical report for the US Naval Oceanographic Office in 20xx [z]. The model is split into two parts: one which deals with uncertainties that are specific to the sonar sensor (the device model) and one that deals with integration of all of the components of the survey system (the system model); the device model is based primarily on work published by Simrad [Pohner, 1993; Hammerstad, 1995] for their EM100 and EM1000 series MBES. The original report has a section on MBES systems and another on sweep systems (i.e., multiple VBES transducers arranged across with width of a vessel), although only the MBES portion has seen much practical application. It is, by a significant margin however, the most popular model for MBES system uncertainty in current use.

The model is a formal Propagation of Uncertainty (PoU) development based on the fundamental equation for MBES systems,

$$d = r \cos P \cos(\theta + R)$$

where d is the depth, r is measured range, P is instantaneous pitch, θ is the angle of arrival, and R is the instantaneous roll. The method of PoU is a formal description of what occurs to the uncertainty when two or more measurements that are uncertain are combined to form another value; essentially, the second-order statistics of the distributions used are propagated through the equation describing the combination of terms and the resultant uncertainty is computed.

Formally, if $y = f(x_1, \dots, x_n)$, each of the variables has variance σ_i^2 and pairs have covariance $\text{cov}(x_i, x_j)$, then the variance of the result is:

$$\sigma_y^2 = \sum_i \sum_j \left(\frac{\partial y}{\partial x_i} \right) \left(\frac{\partial y}{\partial x_j} \right) \sigma_{ij}$$

where the covariance matrix term σ_{ij} is:

$$\sigma_{ij} = \begin{cases} \sigma_i^2 & i = j \\ \text{cov}(x_i, x_j) & i \neq j \end{cases}$$

For simplicity, the HGM model assumes that correlations can be safely ignored, and therefore that the resultant uncertainty is given simply as:

$$\sigma_y^2 = \sum_i \left(\frac{\partial y}{\partial x_i} \right)^2 \sigma_i^2$$

so that the majority of the complexity in the model is in deriving the partial differential terms of the resultant with respect to the input variables and dealing with the consequent non-linear equations.

This complexity turns out to be non-trivial: breaking the primary equation down into its major components results in a sequence of over a hundred component equations that have to be combined to form the final result. This is more readily seen in a flow-diagram such as that of Figure B.1, which illustrates the input constants (in red circles) and fundamental measurements (boxes with no predecessors) as they are combined into the measurement of vertical uncertainty at the top of the tree. A similar diagram applies to the horizontal uncertainty, of course.

Due to the detailed nature of the model, there are many parameters that are required to drive the computation. Most, however, represent the uncertainties of fundamental measurements, which make them simpler to estimate. This applies to measurement uncertainties for the GPS device, the Inertial Motion Sensor unit (IMU) instantaneous angular measurements of roll, pitch and yaw, and of heave, and uncertainty of offset measurements for positioning of the components of the survey system with respect to the body frame of the survey platform. Typically, these can be estimated either from manufacturer's specifications, or from normal survey methodologies.

More complex cases are those uncertainties for which suitably strong physical models do not yet exist, or which are very specific to a particular region. For example, as in the discussion above, estimating the loading effect uncertainty can be troublesome for typical small survey launches, as is predicting the uncertainty in the squat correction. One particularly egregious case is that of water-level corrections, particularly the modeling uncertainty of predicting the tide at the sensor based on measurements elsewhere (e.g., zoning or co-tidal models); another is the spatio-temporal variability of sound speed profiles, about which almost nothing is known in the context of hydrographically significant areas and which in any case would change considerably from area to area.

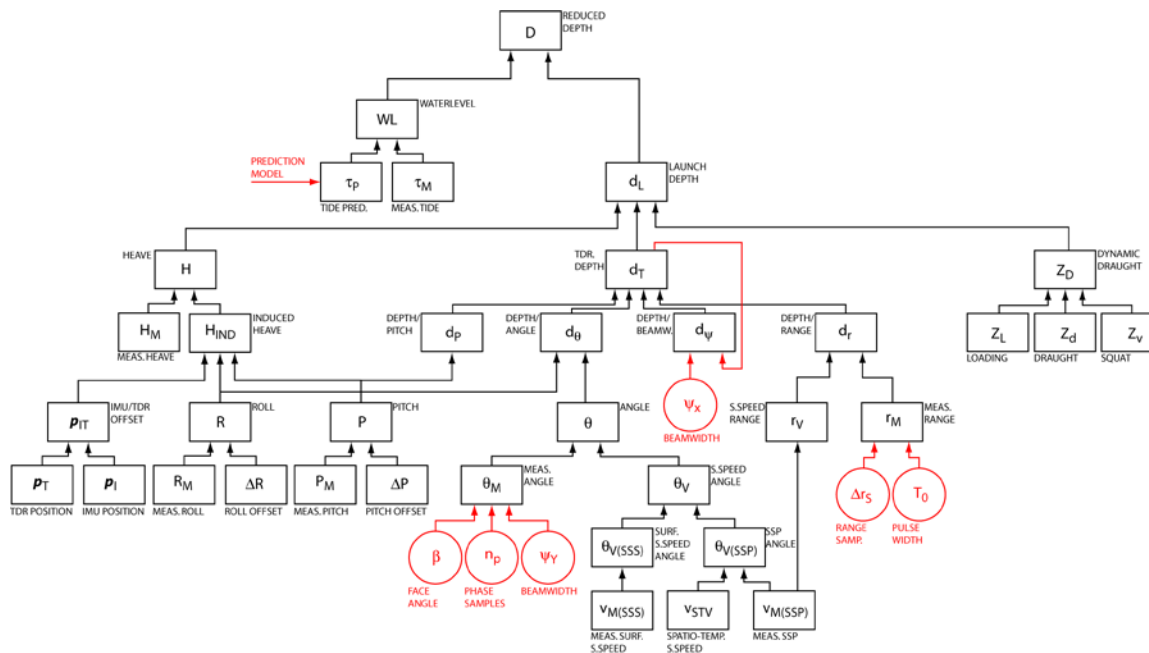


Figure B.1: Data flow diagram for vertical uncertainty component of the Hare-Godin-Mayer MBES uncertainty model. Fundamental constants are given in red circles; boxes with no predecessors represent the uncertainties of fundamental measurements; other boxes are derived uncertainties as intermediate stages of the computation. Data flows from bottom to top in the diagram.

In some cases, these can be resolved by careful testing, or modeling assumptions as described previously for the loading effect. For example, the uncertainty of the vessel alignment angles computed from the patch test could be estimated by doing multiple patch tests within a short time (i.e., so that any true angular variation is unlikely) and then computing the spread of estimates; or, a method that explicitly includes uncertainties could be used [Bjorke & Nilsen, 2005]. (It is generally worthwhile to ensure that multiple people compute estimates from each patch test data set, since there is significant variability between analysts when using manual estimation methods.) In other cases, however, this might not be possible, in which case the only fall-back position is to estimate the uncertainty directly by physical intuition and prior experience. While this might seem arbitrary, there is support for this in the literature [GUM], with the constraint that a selection of expert opinions should be solicited in order to reach an appropriate consensus opinion.

The HGM model is the most commonly used uncertainty model at the present time. It is known to provide a reasonable first order approximation to observed MBES uncertainties. However, it

has a number of limitations, and has received some criticism for them. First, the model completely ignores all correlations between the components of the model. Second, the model approximates the solution of the forward geodetic problem (i.e., to compute the position of a point given a base latitude and longitude and an offset in local level coordinates). Third, the device-specific part of the model is only strictly applicable to one specific MBES system. Fourth, the model applies a particularly simple form of speed of sound correction for refraction. Fifth and finally, the model has a number of free parameters that can significantly change the predicted uncertainty, but in common usage there is no calibration of the model from observed data.

The question of correlations is simply a function of mathematical complexity. Every correlation cross-term makes the model significantly more difficult to compute, and keeping appropriate accounts for which terms are correlated (and determining how) is prohibitively expensive in a model of this complexity. In the end, it is simply a question of abstraction of physical reality in order to support insight: if we insist on having the full complexity of the correlations, the model becomes intractable. We accept the simplification to first order in order to be able to make any predictions at all.

Approximation of the forward geodetic problem is likewise a simplification to make the computations tractable; the direct solution is iterative and difficult to model in closed form. In practice, the model is used most often in shallow water environments where the swath width is sufficiently small that the approximation is very good; in deeper water environments where the swath width is higher, the positioning uncertainty is also generally significantly worse because of the effects of motion sensor uncertainty and refraction and therefore the little extra modeling uncertainty added by the approximation is very slight. Again, we trade uncertainty for tractability.

The HGM device model inherits directly from the work at Simrad in the early 1990s. Not surprisingly, the model is most appropriate for Simrad systems of that period, specifically the EM100 and EM1000. The model is parameterized by such features of the sonar as beamwidth and pulselength, however, and with suitable modifications to these can be used for sonar systems of similar structure. Luckily, most sonars prior to the current generation (e.g., EM3002, 8125) are very similar in structure to the prototypes and therefore the model is a reasonable approximation. (For flat-faced sonars, suitable modifications to the across-track beamwidth as a function of angle area required.) The current generation systems are generally focused on receive (and possibly transmit), and this will result in a significant difference to the signal-to-noise ratio for the received beams. Whether this is practically important to effective bottom picking uncertainty is currently unknown, and there are no published models specifically for these systems. Criticism of the model on this count is probably justified, but it is equally misplaced. Specification of the uncertainty of a sonar device is properly the responsibility of the sonar's manufacturer. No other science would buy an instrument without either a calibration certificate or a calibration model supplied by the manufacturer. There is no reason that hydrographers should either. It is therefore incumbent on all users of MBES systems to request, and then require, that the suppliers of their equipment adequately document their component of the uncertainty model in a manner that is suitable for ingestion into models such as the HGM. It is not required that the manufacturers complete the whole uncertainty model; indeed it is unlikely that they will be able to do so reliably in real-time, and the effort would be wasted. But they must report the range and angle uncertainty inherent in the fundamental measurements that they record, preferably as explicit values for each sounding; as an alternative, a published model that can be implemented by others using the available recorded metadata would be sufficient. Some progress towards this has been made (see Section B.2.4),

but there remains much to be done, and progress is unlikely until users actively drive the debate.

The model's approximation for sound speed refraction effects is another mathematical simplification: in effect, the assumption is that there is at worst a step change in sound speed at the midpoint of the watercolumn. A variational argument then supplies the uncertainty associated with refraction. The handling of this argument has been criticized by a number of observers, including Hare and Eeg, and was one of the significant changes in the 'second edition' (NAVOCEANO) version of the model. This version is still not universally accepted, and NAVOOCEANO themselves reportedly use a table-based uncertainty model that matches the look-up table used to make refraction corrections to the data itself. The criticism here is probably valid. Unfortunately, however, a fuller model of the uncertainty associated with refraction quickly becomes very complex as the horizontal stratification of the medium is taken into account iteratively. In addition, there is little literature on the effects of sound speed variability in a spatio-temporal sense on high-resolution bathymetry, and the model would remain at best an approximation even with a vastly more nuanced and complex model for local refraction assuming perfect knowledge of the sound speed profile. The importance of this approximation depends strongly on the depth regime in which it is applied. In shallow water, the refraction effect is expected to be small, and the approximation is likely to have little relevant effect. In deeper water, this is less likely to be true. This area of the model is most overdue for further research and development.

The issue of variable determination has been outlined somewhat above. The majority of the uncertainties are associated with fundamental measurement uncertainties, and therefore can either be estimated from the manufacturer's specifications, or are readily determined by normal survey methodologies (e.g., the three-dimensional offsets between components of the system). Those that cannot be determined in this way are typically those where the model is less certain anyway (such as the spatio-temporal aspect of sound speed uncertainty), and the estimate of a competent expert is probably as good as any other method of construction. The criticism on this aspect of the model, however, is primarily that there is no attempt to use the repeated observations from an actual survey to attempt to determine the parameters. At one end of the spectrum, the implication has been that the sample uncertainties of the soundings (see Section B.3) should be used directly in the model, rather than having something predictive; others would just use the data to check the calibration of the predictions being made. The best methodology is probably somewhere between these two extremes. A system that only used experimental data would be very limited in its ability to make predictions by extrapolation outside of the observed range of the parameter; a predictive model is positively required, rather than simply desired. However, it is possible that for some sonar systems, an empirical device model might be more appropriate (e.g., Eeg [IHR]), and could be integrated into the larger HGM model with a little extra work; this is particularly the case with phase measurement bathymetric systems such as the GeoSwath, C3D, Klein 5410 and Submetrix ISIS systems. Calibration of the model against observed data is a complex problem, since repeated measurements only measure repeatability, not true uncertainty. There are ways around this, however, and the matter is treated more fully in Section B.3, and especially Section B.3.4. In summary, it appears likely that robust field calibration systems will allow for this eventually using a carefully chosen set of repetition lines along with appropriate measurement protocols.

The HGM model is, therefore, somewhat like Democracy. It may not be perfect, but it is the least bad available system that is readily available for predicting uncertainty of MBES soundings. The future of uncertainty models probably lies in extensions and modifications of

this model to overcome the limitations outlined above rather than wholesale replacement of the model.

B.2.2 Lurton MBES Device Model

DEFINITION OF MODEL (JOE, IHR); ONLY GOOD FOR DEVICE SPECIFIC PART; BETTER MODEL FOR LOW SNR CASES, ABOUT THE SAME FOR HIGH SNR; MODELS FOR DIFFERENT SCENARIOS, INCLUDING PMBS; GOOD SOURCE FOR FUTURE DEVELOPMENT OF WORK.

B.2.3 Other Models

SIMRAD EM1002 UNCERTAINTY MODELS; SUBMETRIX MODEL (?MAY BE CALLED SOMETHING DIFFERENT NOW); GEOACOUSTICS MODEL; MEET.

B.2.4 Sources of Predictive Models

WHERE TO GET IMPLEMENTATIONS AND INFORMATION; CARIS WEBSITE FOR ESTIMATES OF UNCERTAINTY; REQUEST FOR MANUFACTURERS TO MAKE UNCERTAINTY ESTIMATES AVAILABLE AND/OR WRITE ESTIMATES FOR DEVICE PORTION INTO FILES; CARIS INITIATIVE FOR DLL-BASED UNCERTAINTY MODELS.

B.3 Reductive Backward Modeling

The primary general data alternative to a forward predictive model is to attempt to extract some estimate of the uncertainty of the measurements from the measurements themselves. In the general mensuration community, this is by far the most common approach and is preferred over all alternatives; in the ISO scheme for uncertainty, this approach is known as Type A Uncertainty. In principle, at least, the logic of this approach is simple: if we have multiple measurements of the same phenomenon that are statistically independent (or as close to being independent as we can manage), then in addition to computing a better estimate of the measurement than any of the individual measurements, we can also use the repeated measurements to estimate the likely spread of the measurements about the predicted mean value. If we assume (or know) that the measurements are unbiased (i.e., their sample mean is a consistent estimator of the true value of the measurand), then this estimated spread is a valid estimate of the stochastic component of the measurements.

This method has much to recommend it if all of the assumptions can be met: it is simple in concept, always matches what the data is currently doing, and generally requires little beyond standard statistical theory to implement. In addition, it technically does not require any assumption about the underlying distribution of the data, although of course such assumptions are often required to make any use of the estimates subsequently. The main difficulty with this scheme is in the caveat of the assumptions, however: in the sort of data that concern us here, it is frequently very difficult to justify all of the assumptions required to make this method valid.

To appreciate this effect, consider the requirement that the measurand is a constant. On a survey launch on the water, there is no real constancy of any conditions that are significant for surveying. That is, all of the variables that we measure to make corrections to the data change continually: if they did not, we would not be measuring them at all. Consequently, it is difficult to argue that all of the measurements are of the same phenomenon, and therefore that the estimator is consistent. Similarly, consider the requirement for most simple estimators (e.g., arithmetic mean) that the underlying distribution of the samples being considered is a constant. (Quite often, Gaussian distribution is also required to ensure the asymptotic performance of the

estimator, but even if we relax that restriction, the following argument still applies.) A common approach to the comparison of data is to gather an orthogonal pair of lines, and then compare the inner beam depths from one line to the outer beam depths from the other (see section B.3.2). Whether we agree with the forward predictive models completely or not, it can be shown with simple physical arguments that their prediction that the variance of the measurements increases with off-nadir angle is correct. Consequently, we can easily see that the data being compared in the orthogonal lines will have (possibly significantly) different variances, and therefore are not from the same distribution, even if they are from the same distribution family. Obviously, although the estimator will make some prediction in this case, it will have conflated several problems into one number, which requires careful consideration and even more careful explanation and exploitation.

The most significant problem with these methods, however, is not especially technical. Consider the case where the orthogonal swaths for comparison are recorded sequentially in the same area. Since the two lines are recorded close together in time, they will likely share the same water level correctors. Therefore, any uncertainty in the water level correctors will not be expressed in the orthogonal lines, and therefore cannot be observed in the “uncertainty” estimated by these methods. Similar arguments can be made for SSPs, draught effects, etc. In practice, these estimates are always by definition lower than the true uncertainty of the measurements, because they are really measuring the repeatability of the measurement, rather than the true uncertainty.

Due to these limitations, it is not always straightforward to construct reliable estimates of the true uncertainty of the measurements from these sorts of methods. Since they provide another (lower) bound on the likely uncertainty, and because they are readily computable, however, they are quite common in practice. Cross-line analysis, particularly, has been a mainstay of traditional hydrography since the days of leadlines, and is still a deliverable requirement in some survey specifications. With appropriate cautions, therefore, the following sections consider the current methods in use for this style of uncertainty analysis of hydrographic data.

B.3.1 Grid Point Analysis

Consider the whole survey area overlaid with a regular grid, with equal spacing in both cases; there is no required relationship between features of the bathymetry and the grid. Since the grid is a complete cover of the survey area, every sounding measurement can be assigned to one and only one cell in the set. After all of the data has been added to the grid, therefore, each cell (or bin) will contain a collection of co-located data points from one or more passes with the measurement system(s). It's not unreasonable, then, to expect that the collection in each cell should tell us something about the expected uncertainty of the measurements.

If we assume, for example, that the depth is a constant in the cell, we can simply compute sample statistics such as mean and standard deviation, noting that the uncertainty of the measurements should be about the same, or at least related to the sample statistics in some fashion. Alternately, we could use the data points in each grid cell to compute a surface fit of some kind, and then compute the difference of the soundings from the surface as an approximation of the uncertainty of the measurements. Since we do not know *a priori* whether the data is consistent we either have to make such estimates robust (i.e., not sensitive to data outliers that we'd normally flag or otherwise remove from consideration), or we pre-process the data to remove such problems before attempting the uncertainty estimation.

As simple as these schemes are, however, they have a number of limitations that constrain their applicability. In the simplest case, a major limitation is the assumption that the depth is constant in a cell. Imagine the case, for example, where there is a simple unidimensional slope across the cell. The constant depth assumption is equivalent to a horizontal plane across the cell, and when computing standard deviations, the deviations are with respect to this plane.

Consequently, there is a deterministic offset applied to the data as they are being averaged into the variance estimate, and a bias in the estimation of the uncertainty. Clearly, the estimate is only partially linked to the uncertainty of the measurements. A similar argument applies if a surface is fitted to the data before the differences are computed. The problem of the implicit horizontal plane is removed, but the shape of the surface assumed will obviously cause some effect in the estimation. Although the fit to the data will likely be better, there will still be some bias in the uncertainty estimates. Estimates of surfaces from point data are also more sensitive to the measurement noise in the data since they attempt to estimate many more parameters from the observations. Increasing the number of parameters requires larger amounts of data, and therefore a larger area of estimation; larger areas of estimation require more complex surfaces to represent them adequately, and therefore more parameters. Breaking this loop requires some level of compromise.

In both cases, the difficulty is in determining the appropriate resolution at which to process the estimates of uncertainty. Too low a resolution results in bias of the estimates because of the bathymetric effect; too high a resolution results in limited data density and reduced accuracy of estimates. Appropriate choice of resolution is driven primarily by data density; the cells should be sufficiently small so that the approximation of constant depth is appropriate, but sufficiently large to ensure that there are enough soundings present for reliable estimation of sample statistics. (How many soundings are required depends strongly on the statistical properties of the estimator in use.) Obviously, then, the details of the type of sounding system in use and the depth range are critical to understanding the appropriate resolution to use. This, however, ignores the normal requirements for spatial sampling according to Nyquist's Theorem (i.e., that the samples should have at worst half the wavelength, or twice the spatial frequency, of the underlying surface being sampled). To a certain extent this is unavoidable, although questions of spatial aliasing are very important in this problem too. Small objects within a slightly larger cell, or an object bisected by a cell would result in significant asymmetrical bias in any computed uncertainty, or any attempt to fit an algebraic surface to the contents of the cell. The underlying assumption, as with CUBE itself, is that the data density is sufficient to represent any object of interest.

A larger problem, however, is that the basic implementation of these ideas would result in a scheme that conflated the uncertainties of all of the beams in the sounder's output. That is, the methods assume that some 'uniform' uncertainty is appropriate for all of the soundings being generated, which is clearly not the case. The application of such results is, therefore, somewhat limited.

These reservations do not invalidate these methods entirely. It is possible, for example, to stratify the cell size by depth range or some measure of the rugosity of the local surface in order to avoid resolution issues, or to bin the differences between the sounding and the surface in any one cell according to beam number or acoustic launch angle and then average over many cells to estimate a beam-specific uncertainty for the sounding system. (In practice, both of these would be required for a sufficiently robust estimate of uncertainty for further use.) Building a system of estimation that encompasses these constraints makes for complexity that is difficult to automate, however. A practical system might consider the use of a series of carefully chosen estimation locations in order to ensure that the requirements were satisfied. This essentially

requires that the bathymetry of the test location be a priori known, and therefore resolves essentially to that of a Reference Surface (Section 3.3) with the complexity and cost that this involves.

Practically, then, grid point analysis might provide a suitable rough-estimate of the uncertainty associated with the sounding measurements, although it is unlikely, without significant interaction with the user and simplifying assumptions, to provide an estimate that could be used to calibrate a forward predictive model.

B.3.2 Cross-Line Analysis

THEORY OF CROSS-LINES: USE NADIR TO MATCH ORTHOGONAL SWATH AND EXAMING PAIRED DIFFERENCES OR SURFACE/SOUNDING DIFFERENCES; WELLS & SEBASTIAN (?OTHER WAY ROUND); NAVO IMPLEMENTATION; IVS IMPLEMENTATION; CARIS IMPLEMENTATION; LIMITATION IF LINES ARE CHOSEN AT RANDOM; COST OF BUILDING LINES IF ENOUGH STATISTICS ARE TO BE ESTIMATED; FORWARD REFERENCE TO WELL-CHOSEN CROSS-LINES AS UPT.

B.3.3 Reference Surfaces

EXTENSION OF CROSS-LINE ANALYSIS; BUILD A DENSE SURFACE THAT ISN'T GOING TO CHANGE, THEN COMPARE SWATHES AGAINST IT; GOOD FOR RETESTING IN A FIXED SPOT; EXPENSIVE TO BUILD AND MAINTAIN AGAINST ENVIRONMENTAL VARIABILITY; LIMITATIONS IN BUILD IF CORRECTORS ARE NOT GOOD.

B.3.4 Constrained Repetition and Uncertainty Patch Tests

EXAMPLE OF TIDE MODEL AND CHOICE OF CROSS-LINES TO MAXIMIZE STOCHASTIC OBSERVABILITY; REPETITION TRIALS FOR DEVICE SPECIFIC PORTIONS; PROTOCOL-BASED MEASUREMENTS FOR OTHER TESTS; SYNTHESIZED UNCERTAINTY PATCH TEST PROTOCOL DIAGRAM; ESTIMATE OF TIME DURATION; USE FOR CALIBRATION OF PREDICTIVE MODELS.

B.4 Historical and Sub-Dense Data

SUB-DENSE DATA IS IMPORTANT BECAUSE THERE'S LOTS OF IT AROUND AND IT'S STILL THE BASIS FOR COMPILATION; NEED TO ESTIMATE UNCERTAINTIES TO MAKE IT COMPATIBLE WITH OTHER DATA; IDEA HERE IS TO OUTLINE METHODS THAT CAN BE USED FOR THIS.

B.4.1 Metadata-derived Uncertainty

SOURCES OF METADATA; USE OF THE HYDROGRAPHIC REPORT OF SURVEY/DR; UNCERTAINTY ESTIMATION FROM REPORT [JAKOBSSON ET AL., IHR 2005]; UNCERTAINTY ESTIMATION FROM HYDRO MANUALS OF THE PERIOD [CALDER, JOE, 2006]; POSITIONING SYSTEM UNCERTAINTIES; ESTIMATION FROM 'OLD HYDROGRAPHERS'; CAUTIONS ON USE OF UNRELIABLE METADATA.

B.4.2 Comparison with Contemporary MBES

SCHEME FOR USING PARTIAL NEW MBES TO CALIBRATE OLD DATA [CALDER, JOE 2006]; IMPORTANCE OF VARIABLE DEPTH RANGES; HOW MUCH DATA DO YOU NEED?; FIXING A MODEL TO THE DATA [S.44, ED.4]; IMPORTANCE OF SHALLOW-WATER CALIBRATION TO TIE DOWN MODELS; SLOPE EFFECTS WITH WIDE-BEAM VBES;

CAUTION ON THE USE OF 'HYDROGRAPHICALLY SELECTED' SOUNDINGS WITH SHOAL BIASES; CAUTION ON THE USE OF EARLY ECHOSOUNDERS WITH LOW DEPTH RESOLUTION; CAUTION ON THE USE OF DATA FROM NON-DIGITAL SOURCES WITH TRANSCRIPTION ERRORS.

B.4.3 Suck It And See

WHAT DO YOU DO IF NOTHING ELSE WORKS?; ESTIMATION FROM CLASSES OF SURVEY [JAKOBSSON, CALDER & MAYER, 2003]; PROBLEMS WITH VERY HIGH HORIZONTAL UNCERTAINTY, AND LINE/FEATURE CROSS-OVER; WHY DATA AT THIS SCALE DOESN'T CUBE; LIMITATIONS TO CONCLUSIONS DRAWN FROM VERY HIGHLY UNCERTAIN DATA.

B.4.4 Geological Diversity and The Thing Not Seen

THIS SECTION IS PRIMARILY WARNING, RATHER THAN SUGGESTION; YOU CAN'T ASSUME SOUNDING UNCERTAINTY TELLS YOU ABOUT SURFACE UNCERTAINTY; WHY A SURVEY CAN MEET SPECIFICATION AND NOT TELL YOU THE DEPTH; PROBLEMS IN APPLICATION OF SUB-DENSE OR SPARSE DATA TO SURFACE CONSTRUCTION; LIMITS OF SPATIAL RESOLUTION; HYDROGRAPHIC SPATIAL ALIASING; PROBABLE LIMITS ON WHAT YOU CAN SAY FROM SUB-DENSE DATA; IMPLICATIONS FOR REPORTING UNCERTAINTY ON SUB-DENSE DATA TO THE USER; IMPLICATIONS FOR COMBINATION OF SUB-DENSE DATA WITH DENSE DATA.

Appendix C: Bayes' Theorem and Dynamic Linear Models

C.1 Bayes' Theorem

Bayes' Theorem is a mathematical expression, propounded posthumously by Rev. T. Bayes (published 1763 [11]), describing a means to structure problems involving probability statements about hypothesized events based on observed evidence. That is, say that we believe that a number of potential hypotheses describe some physical phenomenon, and we want to try to determine which is most likely to be true, given the observed evidence. To quantify the question, we need to estimate the conditional probability, $P(\cdot)$, that the hypothesis, H , is true given the evidence, E , or $P(H|E)$; Bayes pointed out that it is a fairly straightforward implication of conditional probability that we can express this probability as:

$$P(H | E)P(E) = P(E | H)P(H)$$

$$P(H | E) = \frac{P(E | H)P(H)}{\sum_H P(E | H)P(H)}$$

where the denominator says that the total evidence, $P(E)$, can be computed from the sum of the partial evidence about each hypothesis, since the hypotheses form a mutually exclusive set.

As straightforward as this is, the implications are significant: the equation says that the probability of some hypothesis, given the evidence collected, $P(H|E)$ – something about which we are very interested, but about which it is difficult to gather any sensible conclusions – can be explained in terms of the probability that the evidence observed would occur given that the hypothesis is in fact true, $P(E|H)$, and the probability that the hypothesis itself is likely to be true, $P(H)$ – both things about which we can readily gather data and make judgments. That is, it allows us to make predictions about the probability of an interesting but intractable problem via two probabilities that are much easier to estimate.

For example, say we are interested in the probability that a medical patient has measles, given that they are observed to have spots. In this case, the evidence is the observation of spots, and the hypothesis is that the patient has measles (as opposed to some other disease that also has spots – say, being a teenager). We are interested in the probability $P(M|S)$ for “measles, given spots.” On its own, this is fairly hard to estimate – we don't know a lot about the probability that the patient may have measles. Using Bayes' Theorem, however, we can say $P(M|S) = P(S|M)P(M)/P(S) \propto P(S|M)P(M)$ if we're only interested in looking at the relative probabilities of measles as opposed to other diseases. The remainder of the computation falls out easily: the probability of patients having spots, given that they have measles, is very high; say $P(S|M) = 0.999$. The probability of measles occurring in the general population, $P(M)$, can be readily established from epidemiological analysis, since measles is a notifiable disease. In the reporting period for 2001-2003, the Centers for Disease Control (CDC) reported there were 216 cases in the US, population approximately 300 million; a naïve estimate is therefore $P(M) \sim 7.2 \times 10^{-7}$. We can therefore say that the relative probability of the patient having measles is the product $P(S|M)P(M) = 0.999 \times 7.2 \times 10^{-7} = 7.19 \times 10^{-7}$. That is, not very likely at all. This might seem counter-intuitive at first, but the significant difference here is in the value associated with $P(M)$ – our assessment of how likely it is that the hypothesis can occur, before we even start looking for data. This is a subtle, but very important part of Bayesian analysis, and easily the most contentious distinction in all of probability theory: the idea of a personal estimate of probability.

Technically, this is known as a “prior probability” or “prior distribution.” The concept is that we have some generic understanding of the probability with which events should occur – either through previous experience (like the measles example), or through some belief of our own – a personal opinion. Bayes Theory then combines this prior information – our background assumptions – with the evidence from data, known as the likelihood function, $P(E|H)$, in order to update our state of knowledge about the hypothesis probability after we've examined the data, $P(H|E)$, known as the posterior probability.

What makes this idea controversial is that the judgment of a priori probability can change from person to person, and therefore the probability that one person might assess for the posterior probability can be radically different from that another would assess: it all depends on the assumptions – priors – with which you start. Many people find this offensive, and believe that probability should be a uniform construct between all observers, rather than being an observation relative to the observer. Others point out that all human reasoning is based on background assumptions and side information which colors our judgment, and that therefore a scheme for modeling reasoning processes should do the same. The Bayesian description at least gives us something useful that the alternatives do not: it allows us to take advantage of any domain specific knowledge that we have in order to constrain the problem we're trying to solve. All we have to do is to encode them in a form that we can build a prior distribution around. This is such a significant advantage that Bayesian analysis is one of the best defined, most studied and most heavily used forms of probability theory, and is used extensively in practice.

C.2 Dynamic Linear Models and Optimal Updating

Bayes theory is a very general probability tool, and is not limited to just the case of categorical hypotheses and evidence. (This is one very useful case, however – most classification schemes for remote sensing, texture analysis and seafloor characterization use this formulation in some form.) It can also be written for general data, where we use some information on the previous state of the problem of interest to help us make decisions about the future. For example, if we have some data now, say d_n , and we observe some more, d_{n+1} , we might be interested in understanding the probability structure of the problem after d_{n+1} is observed, given all data to that point.

In order to make predictions, however, we have to have a model that relates the current time to the past (or the present to the future – it's just a matter of your point of view). Of the many potential methods, one of the simplest is the Dynamic Linear Model, or DLM. The general form is defined as a pair of coupled equations. The first relates the present to the past:

$$x_n = f(x_{n-1}) + w_n$$

where x_n is the current state of the system – e.g., the depth at an estimate node in CUBE – and w_n is some noise that might corrupt how the system evolves over time and/or space, and the second:

$$y_n = g(x_n) + v_n$$

relates the current state to the observed data, y_n – e.g., the sounding captured by a MBES system, along with some measurement noise v_n – e.g., the random noise of a MBES about the true depth.

The simplest case of these – and the model used in the CUBE code – is where the evolution of the system is constant, and we observe the state variable x_n directly:

$$x_n = x_{n-1} + w_n$$

$$y_n = x_n + v_n$$

These equations say that our estimate – e.g., the depth at a node – should be a constant, because x_n and x_{n-1} differ at worst by some random noise (in CUBE, this is set to zero by default to enforce constancy) and that we can estimate that constant directly since y_n and x_n differ by a random component at worst.

What's almost more important is that these equations also state that all we need to predict what happens next – x_n – is knowledge of what's happening now – x_{n-1} . This is referred to as the "Markov Property," and in practice it means that all we need to keep on-line at any one time is our current best estimate of the variable – e.g., the depth – and that once we've used any observation by updating our state of knowledge to include its effect, we can discard the observation. CUBE's notion of a modifiable "current best depth estimate", its attitude to soundings compared to depths, and its real-time behavior all stem from these observations.

If we're going to absorb the information from the observations only once, however, then we need to make sure that we're getting everything of value out of the soundings at each stage. By design, the DLM can be directly modeled as a Bayesian process, and you can show that under certain conditions, the Bayesian update from x_{n-1} to x_n , given the observation y_n , is optimal in the sense that it minimizes the probability of prediction error in the mean square sense. In practice, this means that all of the relevant information about each observation – e.g., a sounding – is taken into account during the update.

The general form of the update equations is fairly complex to use and derive; in the simple case here, however, many simplifications are possible, and the update reduces to computing the difference between our predictions of what should come next (i.e., the same as what we have now) and what was observed:

$$\epsilon_n = y_n - x_{n-1}$$

and then applying a weighted version of this to the prediction:

$$x_n = x_{n-1} + K_n \epsilon_n = x_{n-1} + K_n (y_n - x_{n-1})$$

The scaling weight factor K_n (sometimes also called the Kalman gain after Rudolf Kalman – the method can also be written as a Kalman filter) is computed based on the uncertainties associated with the current state of knowledge, σ_n^2 , and that predicted for the new observation, q_n , as:

$$K_n = \frac{\sigma_n^2}{q_n + \sigma_n^2}$$

Intuitively, this gives a scheme that makes sense: if we have very poor observed data, then $q_n \gg \sigma_n^2$, $K_n \rightarrow 0$, and the next state essentially stays the same – the poor data is basically ignored. On the other hand, if the observed data is significantly better than our current state of knowledge, then $q_n \ll \sigma_n^2$, $K_n \rightarrow 1$, and the next state essentially takes on the value of the observation, y_n . If the uncertainties are more evenly matched, then the next state is a composite blend of the two values, optimally balanced according to their relative uncertainties.

C.3 Model Monitoring, Hypothesis Testing and Intervention

The optimal updating rule, however, relies on one key assumption: the model predicts accurately what we observe, in the sense that the observations we make agree with the model's predictions within the uncertainties that we assess for measurements, etc. If this isn't the case, we get symptoms of "modeling failure" which can be somewhat unpredictable. In the case of CUBE, for example, a typical scenario is that an outlier point will be introduced to the estimate node, which we assume initially is tracking the true depth. If the model doesn't know better, then it will happily compute a large value for $\epsilon_n = y_n - x_{n-1}$ and, almost irrespective of the Kalman gain, pull the depth estimate away from the true depth and towards the outlier. If we're lucky, the deviation will be small, and the tracker will recover readily as more inlier data is observed. If we're not, we end up with an estimate that's neither here nor there – stuck somewhere between the two extremes. Obviously, then, we need to make sure that the observations still match the model – or conversely that the model still adequately matches the observations – before we attempt to assimilate the information inherent in each observation as outlined before. This process is known as "model monitoring and intervention."

The question that we need to ask ourselves at each stage, immediately prior to adding in some new data, is whether the prediction of the current model is better than the prediction of some suitable alternative that might explain the observation. In essence, we have a form of hypothesis testing that takes us back to the fundamental form of Bayes Theorem, looking at the relative probabilities of competing hypotheses that can both explain the data observed. We can formulate the question mathematically by observing that the status quo – technically the null hypothesis or H_0 – is that the current model satisfactorily explains the next observation in sequence, while the possible change – technically the alternate hypothesis or H_1 – is that the next observation has suffered a step change in depth with respect to the current model. This choice of simple alternate hypothesis is motivated by the typical problem outlined above, where the outlier data will generally be at a very different depth from the true depth in the area. The question then becomes: how do we judge the likelihood of the two hypotheses, and therefore choose at each stage which is more likely?

Classical hypothesis testing would treat this problem by finding some summary statistic for which we can predict the behavior if we assume that the null hypothesis is correct; in general, this means that we should be able to predict the probability distribution that the summary statistic is going to have. For each case to be tested, we then compute the summary statistic and compare it against the predicted statistical distribution; if the predictions do not match at some level of probability, we consider the null hypothesis disproved, and conclude that the alternative hypothesis is more likely given the evidence from the data.

Bayesian hypothesis testing works in a similar way, although the details are slightly different. In the Bayesian scheme, we have a direct way to measure the probability of various hypotheses given the data that we have observed already, and we can compute directly the probability of the next observation given what we've already observed under the null and alternate hypothesis. The ratio of these, called the "Bayes Factor" gives us a measure of how likely either hypothesis is, and therefore the means to determine whether the null hypothesis is disproved – in our context, whether there is evidence that the next sounding is not statistically consistent with the current track, and needs to be treated specially and not assimilated in the normal manner.

For the case of a vertical step change, the Bayes factor can be computed directly and greatly simplified. Assume that the distributions are Gaussian:

$$N(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right\}$$

(with mean μ and variance σ^2) and that to make things simpler, we pre-normalize the difference between the next observation and the current prediction by dividing by the predicted uncertainty for the next observation:

$$e_n = (y_n - x_{n-1}) / \sqrt{\sigma_n^2 + v_n}$$

(i.e., the current estimate of uncertainty for the true depth estimate plus the observation measurement uncertainty) so that in what follows, the distributions have zero mean and unit variance. Then, the Bayes factor is just the ratio of the probabilities for the next normalized observation, or:

$$\begin{aligned} B_n &= \frac{P_0(e_n | D_{n-1})}{P_1(e_n | D_{n-1})} = \frac{(2\pi)^{-\frac{1}{2}} \exp\left\{-\frac{|e_n|^2}{2}\right\}}{(2\pi)^{-\frac{1}{2}} \exp\left\{-\frac{(|e_n| - h)^2}{2}\right\}} = \exp\left\{-\frac{|e_n|^2}{2} + \frac{(|e_n| - h)^2}{2}\right\} \\ &= \exp\left\{-\frac{|e_n|^2}{2} + \frac{(|e_n|^2 - 2h|e_n| + h^2)}{2}\right\} = \exp\left\{\frac{h^2 - 2h|e_n|}{2}\right\} \end{aligned}$$

where D_{n-1} is the data observed previous to the next observation becoming available, and is a summary of the current state of the algorithm, and the absolute value of the difference is taken so that the test is symmetrical with respect to the direction of the outlier difference (i.e., shaller or deeper than the current estimate).

The interpretation of the Bayes factor is relatively straightforward. If the probability for the null hypothesis is higher than that of the alternative, the quotient will be greater than unity; if the probability for the alternative hypothesis is higher, then the quotient will be less than unity (but greater than zero, since the probabilities are positive numbers). Hence, to check whether the alternative hypothesis is supported – in our context, to test whether the next observation is significantly enough different from the current true depth estimate to warrant special processing (i.e., intervention) – all we need to do is compute the Bayes factor and check the magnitude: if it is significantly smaller than unity, then the evidence is against the null hypothesis and we need to take some action; if it is greater than unity, then we need do nothing.

What constitutes 'significantly smaller than unity', like most hypothesis testing, is a matter of opinion. One common technique is to compute the natural logarithm of the Bayes factor (which removes the requirement to compute the exponential function since exponentiation and natural logarithms are inverse operations), and then set a threshold on the result. Evidence starts to be in favor of the alternate hypothesis when $\ln B_n < 0$, and in most texts, the recommendation is to take action when $\ln B_n < -2$ (i.e., when $B_n < 0.135$). The CUBE implementation follows the same rules, as detailed below.

As outlined, the model monitoring method will detect outliers on a one-time basis: that is, a single outlier will be detected if it is sufficiently egregious. It will not, however, detect problems when the effect is gradual – for example, if the estimates of depth change by small incremental amounts that will eventually amount to a difference sufficiently significant that if it happened all

at once, an intervention would be triggered. To get round this problem, most model monitoring schemes also check for sequential Bayes factors, computing the constrained product:

$$C_{n+1} = B_n \min\{1, C_n\}$$

and checking it against the same thresholds as before, while at the same time computing the number of observations for which the evidence has been marginally against the null hypothesis:

$$L_{n+1} = \begin{cases} L_n + 1 & B_n < 1 \\ 1 & B_n \geq 1 \end{cases}$$

as a final check: if the sequence of evidence marginally against the null hypothesis has gone on for too long, then an intervention is also called for. As above, what constitutes “too long” is really problem specific, and is controlled as a user-controlled variable in the CUBE implementation.

The model monitoring scheme outlined here (and described in much more detail in [2]) so far only indicates how to identify when an intervention is required, not what to do about it. What intervention is required depends on the particular problem. In some schemes, the intervention involves just resetting the estimator to the new input value, discarding the previous history; other schemes reprocess some previous data, or adjust other properties of the estimator. In the CUBE implementation, we have taken the simpler and more conservative tack of assuming that if there is indication that an intervention is required, then there has to be evidence for a different depth reconstruction in the same area (i.e., at the estimation node). That being the case, it's important that we simultaneously make sure that we do not mix this data in with the previous estimate (which would corrupt both estimates), and that we preserve this new evidence: initially, we don't know whether it's the new data, or the previous data that is the true depth of the seafloor at the estimation node and therefore we have to treat them equally until the evidence builds up in favor of one or the other.

In the CUBE algorithm, each sounding provides some evidence about the true depth in the area immediately about the estimation node's location, which allows us to build up a hypothesis about the probable true depth that we should reconstruct at the estimation node when the user requests the “current best depth estimate” from the algorithm. The fundamental estimator described in section 4.2 can be used to implement an optimal track for any hypothesis, and therefore we can complete the intervention scheme by noting that the observation (i.e., a sounding) that caused the intervention to be triggered is evidence that there is another hypothesis about the probable true depth at the estimation node that we should consider. Consequently, when an intervention is triggered, the logical course is to respond by taking the observation that caused the intervention and using it as the initiator for a new tracker hypothesis, of equal status to that which we had prior to the intervention. Applying this logic results in each estimation node developing a monotonically increasing number of hypotheses, each of which will assimilate a different number of soundings – some more, some less – and which represents a potential true depth reconstruction at the node's location. This process is the basis of CUBE's Multiple Hypothesis Tracking scheme. (As an aside, this makes the initialization of the node symmetrical: we start with no hypotheses since we've seen no observations; when the first observation arrives it automatically causes an intervention since there was nothing before, and is used to start the first hypothesis track.)

This intervention scheme is a fairly natural extension of the basic estimator, since the assimilation, model monitoring and intervention strategy at each hypothesis is basically the same. We do have to extend the scheme to ensure that each observation is only applied to one of the potential hypotheses however; otherwise we could potentially start a new hypothesis from

each of the old ones for one new observation. There are a number of schemes by which this could be done. We could, for example, test the next observation against all of the current hypotheses using the monitoring scheme outlined above, and only start a new hypothesis if it failed to assimilate against all extant hypotheses. Or, we could start a new hypothesis for each extant hypothesis that triggers an intervention, and then merge them later. In the interests of simplicity, however, the scheme implemented is to compute the difference between the new observation and the current true depth estimate from each hypothesis, normalized by the uncertainty predicted for the hypothesis as above, and then choose the hypothesis which minimizes this difference (in effect the one step forecast error). The new observation is then tested against just this hypothesis using the model monitoring method outlined above. If an intervention is triggered, a new hypothesis is started from the observation; if not, the observation is assimilated into the selected extant hypothesis.