Computers & Geosciences 106 (2017) 49–59

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Computationally efficient variable resolution depth estimation

B.R. Calder[a,*], G. Rice[b]

[a] Center for Coastal and Ocean Mapping and NOAA-UNH Joint Hydrographic Center, University of New Hampshire, Durham, NH 03824, USA
[b] NOAA Hydrographic Systems and Technology Branch, NOAA-UNH Joint Hydrographic Center, University of New Hampshire, Durham, NH 03824, USA

ARTICLE INFO

Keywords:
Bathymetric models
Hydrographic survey
Variable resolution surfaces
cube
chrt
Data density estimation
Data-driven estimation

ABSTRACT

A new algorithm for data-adaptive, large-scale, computationally efficient estimation of bathymetry is proposed. The algorithm uses a first pass over the observations to construct a spatially varying estimate of data density, which is then used to predict achievable estimate sample spacing for robust depth estimation across the area of interest. A low-resolution estimate of depth is also constructed during the first pass as a guide for further work. A piecewise-regular grid is then constructed following the sample spacing estimates, and accurate depth is finally estimated using the composite refined grid and an extended and re-implemented version of the CUBE algorithm. Resource-efficient data structures allow for the algorithm to operate over large areas and large datasets without excessive compute resources; modular design allows for more complex spatial representations to be included if required. The proposed system is demonstrated on a pair of hydrographic datasets, illustrating the adaptation of the algorithm to different depth- and sensor-driven data densities. Although the algorithm was designed for bathymetric estimation, it could be readily used on other two dimensional scalar fields where variable data density is a driver.

1. Introduction

In a number of remote sensing modalities the density of observations achievable by any given sensor is a function of the environment. For example, in many sonar and lidar ocean mapping systems, the observations are made at a given angular separation across a swath, or are spaced equidistantly across a swath, so that the average density of observations is inversely proportional to water depth below the sensor. Attempts to increase the data density in deeper water are found wanting in independence of observations and inefficiency of survey effort due to the necessarily reduced swath width.

Consequently, the fidelity with which the true spatial configuration of the measurand can be reconstructed is limited not only by the point spread function of the observing system, but by a system-mediated sample spacing which will inevitably alias at some wavelengths. Attempting to reconstruct the measurand beyond the sample spacing supported by the data, observing that the Nyquist limit (Proakis and Manolakis, 2006) for the measurand is not well defined, and unknown *a priori*, are therefore at best inefficient and at worst misleading for the user (compare Gardner et al. (2014) for a limited approach to this in deep water). Further, an estimation or reconstruction scheme that does not allow the sample spacing to adapt to the data is forced to choose a compromise sample spacing that is guaranteed to be incorrect almost everywhere. Optimistic estimates of achievable sample spacing can be inefficient in both memory and computational load and lead to poor reconstruction stability where the data density falls off; pessimistic estimates can lead to significant aliasing in regions of high data density.

Algorithms that allow for spatial variations in sample spacing, and even for adaptation, are common in finite element modeling and rendering for computer graphics (Borouchaki et al., 1997a, 1997b; Borouchaki and Frey, 1998; Cao et al., 1999), and are a standard method for spatial data structures in the form of the quadtree (Fischer and Bar-Yoseph, 2000; Samet, 2006). These methods, however, generally require that the location of the estimation nodes be stored individually, and often have a significant cost to index each observation being applied to the reconstruction. This works well when limited numbers of observations are being considered (the definition of "limited numbers" naturally changing with advances in processor, memory, and storage systems), but for remote sensing datasets of billions of points the theoretical overhead costs can be significant. To be practical, a reconstruction method has to be computationally efficient on a "per observation" basis.

A practical reconstruction method must also be robust to noise inherent in the observations. In ocean mapping applications, in addition to the measurement noise that affects every observation, it is common to have "blunders" where the instrument mis-detects the ocean depth, for example because the acoustic energy bounced off a target in the water column prior to interacting with the seafloor, or

[*] Corresponding author.
E-mail addresses: brc@ccom.unh.edu (B.R. Calder), glen.rice@noaa.gov (G. Rice).

http://dx.doi.org/10.1016/j.cageo.2017.05.013
Received 1 November 2016; Received in revised form 23 May 2017; Accepted 25 May 2017
Available online 31 May 2017

because sidelobe interference caused the instrument to detect the specular return in an off-nadir beam. It can be difficult or impossible to predict where these events will occur, and therefore any useful reconstruction method can make only limited assumptions about the data, and if adapting to the data, must continue to do so successfully in the face of such blunders.

A method is therefore required, and is here proposed, for a remote sensing reconstruction scheme that can adapt to the achievable sample spacing inherent to the observations gathered. It maintains these variations in sampling spacing within a single coherent data structure, is computationally efficient, and is robust to the effects of observation blunders through the use of a multi-hypothesis tracking sub-algorithm; the primary focus here, however, is on the data adaptation.

Conventional algorithms within the hydrographic (nautical charting) community mostly focus on selecting, from all observations, which ones are believed to be plausible (that is, they form no grid, except perhaps as a means to identify which observations to keep, and which to remove from consideration). Methods has been attempted based on pointwise testing (Lirakis and Bongiovanni, 2000), emulation of human subjective editing (Du et al., 1996), various statistical tests (Debese, 2007; Debese and Michaux, 2002; Debese et al., 2012; Eeg, 1995; Ware et al., 1992), and constructing triangulated irregular networks (TINS) directly from points (Canepa et al., 2003; Arge et al., 2010), with varying computational costs, degrees of robustness, and scalability. No one method has become dominant in practice. Within the geosciences community, where there are fewer concerns about preserving the shallowest point within an area and the consequent liability that might ensue, processing algorithms have been most focussed on (fixed resolution, regular) grid-based techniques, with methods based on (weighted) averages of various kinds (Gourley and Dodd, 2000), geostatistical methods (Cressie and Wikle, 2011), scale-control interpolation (Cleveland and Devlin, 1988; Plant et al., 2002), robust surface fitting (Chen et al., 2016; Nurunnabi et al., 2015), or splines in tension (Smith and Wessel, 1990) (see Hell (2011) for a variable resolution method). Again, no method has become dominant in hydrographic practice, although the method of Smith and Wessel (1990) is probably the most commonly used in conventional bathymetric estimation, at least in deep water. A technique that straddles both camps is the CUBE algorithm (Calder and Mayer, 2003), which is designed to compute the best estimate of depth at any given point in the area of interest, taking into account measurement uncertainty, which is often repeated over a regular grid to reconstruct the measurand. Modifications of the CUBE algorithm for sparse data have also been proposed (Bourgeois et al., 2016; Zambo et al., 2015). The CUBE algorithm has become a widely accepted approach for bathymetric data processing, being incorporated into a large majority of the software packages used for this purpose.

The use of a gridded reconstruction is computationally beneficial, simplifies data structures, and allows for assimilation of information from a collection of observations in order to robustly process raw observations with blunders. The CUBE algorithm is robust to typical blunders, and uses the estimated measurement uncertainty of observations not only to weight the contribution of the observations but also to separate them into internally consistent but externally distinct groups so that blunders do not contaminate the developing estimate of depth. Although the algorithm does not require it, being a pointwise estimator of depth, all known implementations of the algorithm use a regular grid of estimation nodes for efficiency.

Consequently, an extended version of CUBE is here proposed that relieves this constraint, while still remaining computationally efficient. The algorithm, called CUBE with Hierarchical Resolution Techniques (CHRT), extends the CUBE algorithm by initially estimating the data density achieved by the observations, which is translated into an estimate of sample spacing that can be supported by the observations. A low-resolution estimate of depth is simultaneously generated in order to aid decisions about further processing. A set of piecewise-constant sample spacing (PCSS) grids are then constructed over the area of interest, and the observations are processed using a CUBE-derived estimation algorithm to reconstruct depths and uncertainty pairs at each estimation node, along with other metrics to inform the reconstruction. The supporting data structure is designed to scale to essentially arbitrary-sized areas, and maintain the variable resolution grids within a single structure.

The remainder of this manuscript describes the design of the CHRT algorithm in detail. In particular the method for estimating data density and translating it to sample spacing estimates, and the data structure to support essentially arbitrary-sized areas of interest (Section 2). Illustrative examples of the estimator in use with dense hydrographic datasets are given in Section 3. A discussion of the design and conclusions follow.

2. Algorithm design

2.1. Variable resolution with piecewise-constant sample spacing

To establish the core structure, consider a coarse uniform grid over the area of interest with sample spacing S-m (in both axes), such that $S > s_{\max}$, the coarsest refined sample spacing expected in the final reconstruction. Around each grid point establish a super-cell, covering area $\mathrm{A}(\mathbf{p}) \subset \mathbb{R}^2$ for sample reference $\mathbf{p} = [u, v]' \in \mathbb{Z}^2$, so that $\bigcup_{\mathbf{p}} \mathrm{A}(\mathbf{p}) = \mathbb{R}^2$. Assume that from a first pass over the data it is possible to compute for each super-cell the sample spacing achievable by the observations, $s(\mathbf{p})$. Then, at each sample point generate a regular grid $\mathrm{G}(\mathbf{p})$ with sample spacing $s(\mathbf{p})$-m, with

$$n(\mathbf{p}) = \lfloor S/s(\mathbf{p}) \rfloor + 1 \tag{1}$$

nodes in each axis, centred at

$$\mathbf{x}(\mathbf{p}) = (\mathbf{p} + 1/2)S \tag{2}$$

so that the area covered by the grid, $\mathrm{A}^{\mathrm{G}}(\mathbf{p}) \subset \mathrm{A}(\mathbf{p})$, Fig. 1, since $(n(\mathbf{p}) - 1)s(\mathbf{p}) < S$ by construction. The edge buffer is $b(\mathbf{p}) = S - (n(\mathbf{p}) - 1)s(\mathbf{p}) < s(\mathbf{p})$ so the spacing between edge nodes in adjacent super-cells maintains at least the estimated sample spacing. The ultimate estimation of depth is conducted on the PCSS grid $\mathrm{G} = \bigcup_{\mathbf{p}} \mathrm{G}(\mathbf{p})$. Fig. 2 shows a synthetic example of this refinement on an area where there is sufficiently rapid change in supportable sample spacing to show the effects.

The simple PCSS grid is memory and computationally efficient, but is limited in that the sample spacing can only be modified at S-m intervals. Since each $\mathrm{G}(\mathbf{p})$ is independent, however, it would be possible to replace those where further flexibility were required with a more complex data structure (e.g., a quadtree) without loss of efficiency elsewhere. This topic is considered further in Section 4.

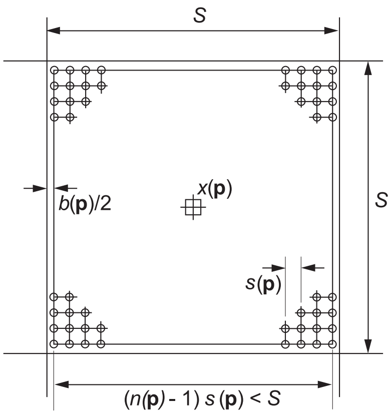

Fig. 1. Geometry of super-cell refinement. The refined grid is a proper subset of the area supporting the super-cell, with edge area that is not covered being less than one (refined) sample spacing.

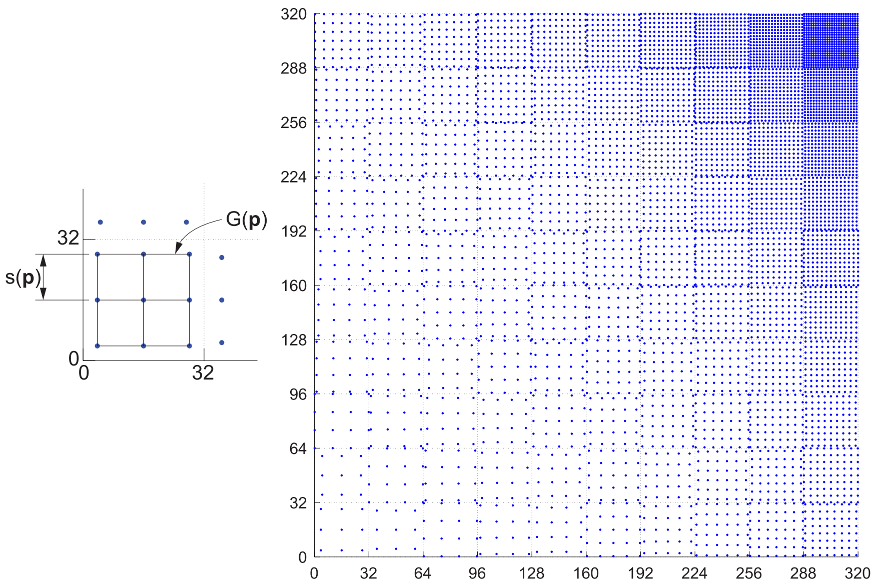

Fig. 2. Example of G(**p**) and $s(\mathbf{p})$, and simulated refinement node placement, where dotted lines indicate super-cell boundaries. The density of nodes always at least matches the sample spacing required, although it can exceed this on super-cell edges. Axes here are arbitrary projected units.

2.2. Resolution estimation

For most estimators, information is gleaned from observations in the vicinity of the point of estimation, with the estimation improving with the number of observations assimilated. Consider, then, observations in the closed disc $B_c(\mathbf{x}) \in \mathbb{R}^2$, radius c-m about location $\mathbf{x}$ used to estimate the depth at $\mathbf{x}$, where n_{req} observations are required on average to form a sufficiently precise estimate (n_{req} is a user parameter, see Section 4). Allowing that a proportion, ϵ, of observations will be blunders and not used for estimation, for data density $\rho(\mathbf{x})\text{m}^{-2}$, it is clear that $\pi c^2 \rho(\mathbf{x}) \ge n_{\text{req}}(1+\epsilon)$ is required for stable estimation. Define $\gamma \triangleq s(\mathbf{p})/c$, so that this constraint becomes

$$s(\mathbf{p}) \ge \gamma\sqrt{\frac{n_{\text{req}}(1+\epsilon)}{\pi\rho(\mathbf{p})}}. \tag{3}$$

In order to ensure representability of objects of linear scale L-m, it is also required that

$$s(\mathbf{p}) \le \kappa L,\ 0 < \kappa \le \frac{1}{2} \tag{4}$$

from the Nyquist theorem. In order to avoid missing observations, the ratio γ must be chosen to ensure that the whole area of interest is considered; for efficiency, choose

$$c = \min\left\{\delta\colon \mathrm{A}^{\mathrm{G}}(\mathbf{p}) \subseteq \bigcup_{\mathbf{n}\in \mathrm{G}(\mathbf{p})} B_{\delta}(\mathbf{x}(\mathbf{n}))\right\}, \tag{5}$$

which is equivalent to setting $\gamma = \sqrt{2}$. The resolution is then bracketed by the pair of inequalities

$$\kappa L \ge s(\mathbf{p}) \ge \sqrt{\frac{2n_{\text{req}}(1+\epsilon)}{\pi\rho(\mathbf{p})}},\ 0 < \kappa < \frac{1}{2}. \tag{6}$$

Given suitable user-selected values for n_{req} and ϵ, the only unknown is the data density, which must be estimated directly from the observations during an initial pass over the data. This is a variant of spatial point density estimation (Cressie and Wikle, 2011), where each observation location $\mathbf{x}_i \in \mathbb{R}^2$, $1 \le i \le N$ represents area $\mathrm{a}_i \subset \mathbb{R}^2$ in the plane; the effective area occupied within a super-cell is therefore

$$\mathrm{a}(\mathbf{p}) \triangleq \mathrm{A}(\mathbf{p}) \cap \bigcup_{\{i:\mathbf{x}_i\in \mathrm{A}(\mathbf{p})\}} \mathrm{a}_i \tag{7}$$

and hence the observation density can be estimated as $\hat{\rho}(\mathbf{p}) = N(\mathbf{p})/\|\mathrm{a}(\mathbf{p})\|$ where $N(\mathbf{p}) = \sum_i [\mathbf{x}_i \in \mathrm{A}(\mathbf{p})]$ is the number of observations that fall in the super-cell.

Computing (7) is expensive. For random point data, an approximation is to digitise a representation of a_i, say $k_i(\mathbf{x}(\mathbf{q})) = [\mathbf{x}(\mathbf{q}) \in \mathrm{a}_i]$, $\mathbf{q} = [a, b]' \in \mathbb{Z}^2$ at some suitable sample spacing S/U, $U > 1$ so that $\mathbf{x}(\mathbf{q}) = (\mathbf{q} + 1/2)(S/U)$ and compute

$$\mathrm{a}(\mathbf{p}) \approx \sum_{\{\mathbf{q}:\mathbf{x}(\mathbf{q})\in\mathrm{A}(\mathbf{p})\}} \max\left\{1, \sum_{i=1}^{N} k_i(\mathbf{x}(\mathbf{q}))\right\}, \tag{8}$$

choosing U sufficiently large to provide a reasonable approximation to the true areas.

For data that is collected in swathes, however, a more efficient approximation is to determine the effective area covered by the extents of each swathe, and connect these together into an outline polygon, Fig. 3, before rasterising the polygon at an appropriate resolution to provide an estimate of the area covered by the observations. The spatial extents of the swathe can be determined directly from the positions of the observations in the edges of the swathe; this is subject to blunders in positioning or solutions for the observation's position with respect to the sensor, but these have a minimal effect on area estimation and almost always occur on the edge of the observed area where they have lesser significance. In order to ensure that the along-track extent of the

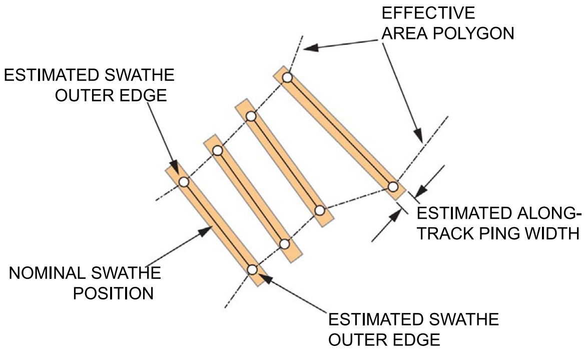

Fig. 3. Geometry of the swathe's effective-area polygon computation. Each swathe is one set of observations derived from a single acoustic transmission, or "ping"; due to transmission geometries the observations form an approximate line on the seafloor. Rasterising thick lines (of the estimated along-track ensonified area) in addition to the polygon ensures a minimum area coverage estimate even in the case of a static observing platform.

observations are considered (e.g., for a stationary observer), an approximate estimate of the mean depth $\mathbb{E}[z]$ of the observations in each ping can be determined, after which given the along-track beamwidth, β, the estimated along-track ping width is simply $w = 2\mathbb{E}[z]\tan(\beta/2)$, which can be rasterised as a line in addition to the overall polygon. In the proposed implementation, the mean depth estimate is constructed over the observations about the middle of the ping using a trimmed mean estimate (i.e., first computing a weighted arithmetic mean and standard deviation of all observations using the estimated vertical uncertainty of the observations as weights, and then recompute ignoring all observations more than three standard deviations from the initial mean).

Hardware rasterisation can be achieved using a suitable graphical processing unit (GPU) library, such as OpenGL (Kessenich et al., 2016), rendering to an offscreen frame buffer, rather than directly to the screen buffer. Such implementations are often limited as to the size of the frame buffers that can be used, which limits the over-sampling rate U that can be achieved. In order to minimise the quantisation noise associated with the estimation of area by this method, graphical anti-aliasing techniques can be used, which allows for partial sub-cells to be maintained. The GPU hardware can also be used to implement an approximation to (7) by rendering each swathe into a "saturation overflow" stencil buffer, and then reading the frame buffer back through a boxcar convolution kernel (Proakis and Manolakis, 2006) to compute the final summation of all of the anti-aliased partial and full sub-cells within the domain of a super-cell and thereby generate the area estimate.

Typical values of n_{req} range over [3,10] (Rice and Calder, 2009) with a common choice being five observations (National Ocean Service, 2016). The choice of "noise factor" ϵ depends to some extent on the degree of blunders in the observations; typical ranges during testing were $[0.01, 0.10]$. Selection of these parameters is considered further in Section 4.

2.3. Estimation node placement

Estimation nodes must be placed within the super-cell so as to maintain the edge guard region $b(\mathbf{p})$ shown in Fig. 1 by manipulating the sample spacing. This ensures nodes are not placed exactly on the boundary of the super-cell, which can be inefficient, and cause difficulties for visualisation code. In order to avoid these problems, define a nominal relative boundary gap of

$$B = k\frac{s_{\min}}{2S} \tag{9}$$

for some arbitrary constant $k \in \mathbb{R},\ 0 < k < 1$. For spacing $s(\mathbf{p})$ the number of estimation nodes is $N_e = \lfloor S/s(\mathbf{p}) \rfloor + 1$; if $(N_e - 1)s(\mathbf{p}) > (1 - 2B)S$, the edge guard region would be violated, therefore remap the spacing to

$$\begin{aligned} s'(\mathbf{p}) &= \frac{(1 - 2B)S}{N_e - 1} \\ &= \frac{S - ks_{\min}}{\lfloor S/s(\mathbf{p}) \rfloor} \end{aligned} \tag{10}$$

which results in a remapped width for $\mathrm{G}(\mathbf{p})$ of

$$\begin{aligned} W' &= (N_e - 1)s'(\mathbf{p}) \\ &= \left\lfloor \frac{S\lfloor S/s(\mathbf{p}) \rfloor}{S - ks_{\min}} \right\rfloor \left(\frac{S\lfloor S/s(\mathbf{p}) \rfloor}{S - ks_{\min}} \right)^{-1} S. \end{aligned} \tag{11}$$

This can unfortunately result in $W' = S$ by an exact cancellation if

$$f = \frac{S\lfloor S/s(\mathbf{p}) \rfloor}{S - ks_{\min}} \in \mathbb{Z} \tag{12}$$

so that $\lfloor f \rfloor = f$. This condition pertains if $S/(S - ks_{\min}) \in \mathbb{Q}$ since $\lfloor S/s(\mathbf{p}) \rfloor \in \mathbb{Z}$ by definition. However, since $k \in \mathbb{R}$ is arbitrary (with preference for $k \lesssim 1$), $1 < S/(S - ks_{\min}) < \infty$ so there is a choice $(S, s_{\min}, k)$ that avoids cancellation, and can be determined *a priori*. Examination of the structure of (10) (S. MacGillivray, personal communication) can demonstrate that any choice of k satisfies the conditions required with reasonable assumptions on the properties of the inputs; in practice, corner cases make these assumptions hard to ensure, and a one-time computation of k to avoid (12) is more reliable.

2.4. Core estimator

Given the PCSS grid $\mathrm{G} = \bigcup_{\mathbf{p}} \mathrm{G}(\mathbf{p})$, any estimation algorithm can be applied to the component nodes, with index costs per observation (i.e., the cost to determine which node is closest to an observation) being limited to three multiplications and two additions per observation (i.e., $O(n)$). Although any algorithm could be used, the proposed method uses the core estimator from the CUBE algorithm (Calder and Mayer, 2003), with some simplifications due to the direct link between data density and sample spacing induced by CHRT.

CUBE operated by a scatter-gather process: each observation was offered to a subset of the estimation nodes within its local vicinity (scatter), and at each node a decision was made as to which offered observations to accept for estimation (gather). The scatter and gather radii were user parameters. In CHRT, however, automatically computing the sample spacing and selecting $\gamma = \sqrt{2}$ in (3) means that all observations are required only to be propagated to their nearest neighbour estimation nodes, and each node automatically accepts all observations that are propagated to it. Similarly, determining which observations need to be propagated from one super-cell to its neighbours can be greatly simplified.

CUBE also assumed that each observation contained both a depth estimate and its associated 3D uncertainty; depth estimates constructed also included an uncertainty estimate. The algorithm used an uncertainty propagation equation to reflect the idea that the uncertainty of an observation that is used at some distance from its nominal location should be higher (i.e., the information content is diluted to some extent). Experience with the original equation has demonstrated that the uncertainty often increases too rapidly, and CHRT uses an equation based on the mean distance of propagation, given the horizontal uncertainty of the observation, rather than CUBE's worst-case approach. For more details, and examples, the reader is directed to Calder and Elmore (2017).

As with CUBE, as each observation is propagated to an estimation node, the CHRT algorithm assimilates it against the appropriate depth estimate using a simple dynamic linear model (West and Harrison, 1997), which could also be considered a simple Kalman filter, or a sequential Bayesian estimator. Prior to assimilation, each observation is tested against the null hypothesis that the current estimate and the new observation are consistent in depth within their respective uncertainties, with the alternative hypothesis being a step change in depth. If the null hypothesis is rejected, a separate tracking structure is initialised from the new observation (called a "hypothesis" in CUBE for obvious reasons); otherwise, the new observation is assimilated into the current estimate. When more than one hypothesis is being tracked, new observations are sequentially tested against the set of hypotheses until one matches or all fail and a new hypothesis is spawned. The algorithm therefore sequentially constructs a set of hypotheses as to the true depth at the node location as observations are made available, but defers a decision as to which is most likely to be the true depth until the user requests a reconstruction, at which point one hypothesis is reported as most likely, along with metrics to quantify the algorithm's reliability in this choice. A number of methods are available to make this choice, from selecting the hypothesis with most observations to selecting the hypothesis that is closest in depth to that declared at the nearest node with a unique reconstruction (i.e., with only one hypothesis). Many variations are obviously possible. For more details, the interested reader is directed to Calder and Mayer (2003).

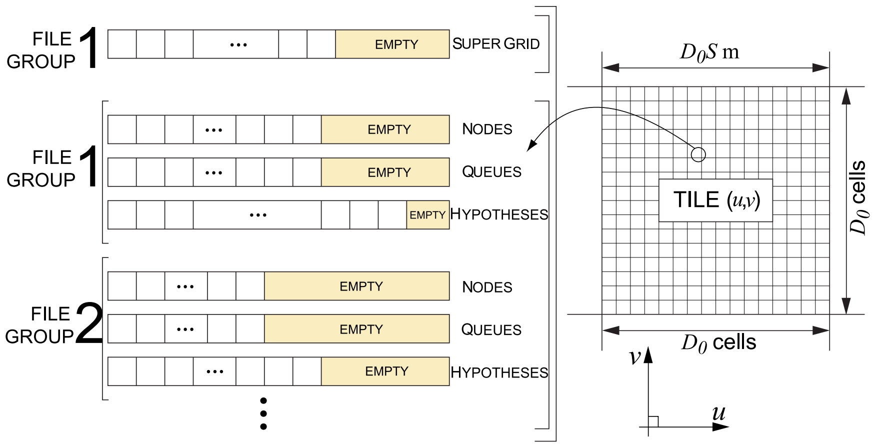

Fig. 4. Structure of memory-mapped components of an individual tile. Each tile maintains a single super-cell file, and any number of file groups for CUBE depth estimation purposes.

2.5. Data structure

The CHRT data structure is in essence a collection of regular grids. For efficiency, however, careful arrangement of the memory structures is required. There are three primary design requirements: efficient core data structures that can be expanded readily during observation assimilation, the ability to do lazy allocation of resources and minimise resource usage, and the ability to operate over large physical areas and datasets without manual intervention of the user. The first two requirements addresses computational performance; the last address user experience and ease of use.

A spatially-tiled data structure is used to allow for demand paging. Within each tile of $D_0 \times D_0$ super-cells, the data structures required include a base object to hold statistics on sample spacing, and a set of structures for CUBE. Each estimation node consists of a core structure, a short queue of working data, and zero or more hypotheses on the true depth, augmentable as new observations are assimilated. For efficient addition of hypotheses and data transfer, these structures are held as separate files that can be memory-mapped in file groups, Fig. 4. For simplicity, the hypotheses associated with a given estimation node are required to appear in the same file group, and a single integer-indexed singly-linked list allows for random allocation of hypotheses. A 32-bit index is used, with the CUBE core structure holding the root index, or a null index if there are no hypotheses; theoretically, $2^{32} - 1$ hypotheses can be indexed in a file group, although other constraints typically preclude this.

Establishing the maximum area that can be accessed requires a determination of the tile sizes allowed. Nodes and queues are always paired so only a single index is required to find refinements; the nodes and queues for a refinement are stored sequentially within their respective files, which constrains how many refinements can be placed in a file group. Multiple file groups are therefore required per tile. To do this, a 32-bit index is used, split between a b-bit file group reference and a $(32 - b)$-bit intra-group index. A primary design choice is therefore the selection of b.

Consider a file group with $2^{32-b} - 1$ nodes (reserving one value to represent an invalid index), each of m_n B. In the worst case, for example if a swathe covered the corner of four separate tiles, with load-before-delete semantics, plus a spare, up to nine file groups might need to be mapped to avoid paging. A maximum of $M = 9m_n(2^{32-b} - 1)$B is therefore required, and for a 32-bit address space, $M < 2^{32}$, giving

$$b > 32 - \log_2\left(\frac{2^{32}}{9m_n} + 1\right) \tag{13}$$

or, since $2^{32}/(9m_n) \gg 1$,

$$b > \log_2(9m_n) \approx 3.17 + \log_2 m_n \tag{14}$$

For the development implementation, $m_n = 212$ B, so $b > 10.90$ bits, or $b \geq 11$ bits.

Establishing an upper limit for b requires a detailed analysis of the node count in each tile, considering the $D_0{}^2$ nodes required for the low-resolution first-pass depth, the worst-case refinement node count of $\lceil S/s_{\min}\rceil^2$, and that $M(D_0, S, s_{\min}) = \max\{D_0^2, \lceil S/s_{\min}\rceil^2\}$ nodes might be unused at the end of the file group due to local addressing constraints. It can be shown, however, that the constraints can be satisfied (Calder and Rice, 2011) if

$$D_0 \leq \sqrt{\frac{(2^b - 1)(2^{32-b} - 1)}{2^b + \lceil S/s_{\min}\rceil^2}}, \tag{15}$$

and

$$b \leq 32 - \log_2(\lceil S/s_{\min}\rceil^2 + 1) \tag{16}$$

which leads to a range of allowable tile sizes given b and the refinement ratio $r = S/s_{\min}$, Fig. 5.

Combined, these limits provide for tile sizing, which can be physically extensive. For $b = 11$ bits, $D_0 \in [319, 1324]$ and for

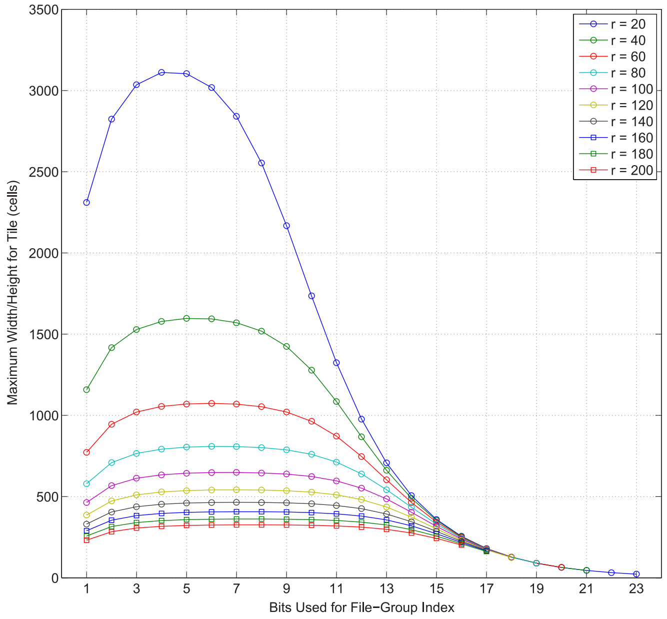

Fig. 5. Maximum allowed tile sizes for different choices of file group index bits b, and refinement ratio $r = S/s_{\min} \in [20, 200]$, assuming no other constraints. Increasing the refinement ratio generally reduces the maximum tile size allowable, although other limits might be stronger in practice.

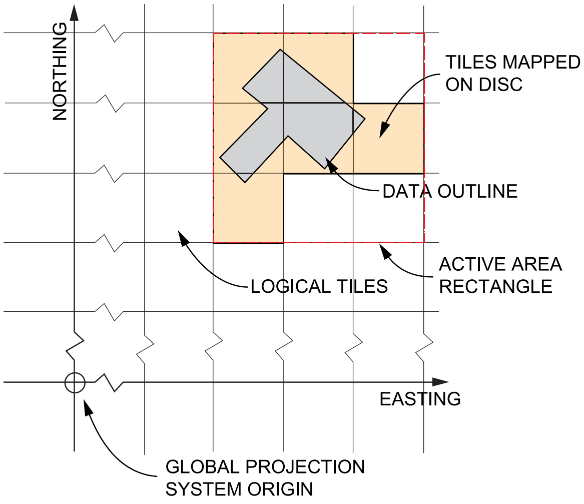

Fig. 6. Illustration of global grid structure, which covers the projected coordinate system with fixed-size data storage tiles that are only allocated (orange shading) in the active area (red outline), i.e., the minimum bounding box of tiles that cover the area in which observations occur (grey). Orange shaded cells are allocated on disc, and mapped to memory when required; cells in the active area, but not containing data, use no resources. The algorithm automatically maintains the list of active tiles and the current bounding box based on the data observed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

$(S, s_{\min}) = (20, 0.1)$ m, $D_0 = 319$, implying ≈6.4 km tiles; for $(S, s_{\min}) = (30, 0.25)$ m, $D_0 \approx 510$ implies ≈15.3 km tiles. Since the tiles are indexed by 32-bit integers, the overall size of the addressable space is greater than the circumference of Earth.

Hence, there is no practical limit on the size of survey area that can be addressed, as required.

Establishing an addressing scheme that can circle Earth supports the design goals of user experience and lazy allocation. Since the whole projected plane is addressable, the user does not have to specify a bounding box *a priori*: the tiles, and their refinements, can be fixed to the origin of the coordinate system, Fig. 6. It is then simple to turn on only required tiles as new data is observed. Within each tile, a compressed bit-mask is used to identify active super-cells, and resources are only allocated where required, providing for lazy allocation of resources with low overhead.

3. Estimation examples

3.1. Woods Hole, MA

Woods Hole, MA was surveyed in 2001 by the NOAA Ship *Whiting* (Barnum, 2001), eventually becoming registered as survey H11077. The primary instrument used was a Reson 8101 multibeam echosounder, with ApplAnix POS/MV 320 motion sensor, and differential-aided GNSS using Trimble receivers; sidescan and singlebeam echosounder data were also collected, but are not used here. For processing through CHRT, the data were converted using CARIS HIPS data processing software (version 9.0) for manipulation, and then exported in projected coordinates (UTM zone 19N on NAD83 ellipsoid). A total of 37.2×10^6 observations were used, creating 5.36×10^6 populated nodes after refinement.

The survey is in relatively shallow water with vertical range of 2–30 m, limiting changes required in sample spacing, and allowing for a relatively small super-cell width of $S = 8$ m. The CHRT first-pass output products are given in Fig. 7. The estimated data density is relatively constant, although the edges of the survey show lower sample spacing estimates as might be expected; some overlay of multibeam echosounder data density differences are also observable, highlighting the survey line pattern used. A mean sample count per hypothesis (n_{req}) of 11 observations and a noise compensation (ϵ) of 0.10 were used to compute the sample spacing estimates.

A section of the variable resolution estimate of depth, in the main channel heading north to Woods Hole, is shown in Fig. 8, and is shown with an overlay of the estimate node positions in Fig. 9. As expected, the algorithm has adapted to the variability in data density within the region, increasing sample spacing from approximately 0.6 m to 1.3 m over a depth range of 8.5–15.5 m; these conservative sample spacing estimates reflect the choice of n_{req} and ϵ, which were chosen to ensure that 95% of the estimation points had at least five observations, as required by current NOAA survey specifications (National Ocean Service, 2016).

The survey area in H11077 is relatively benign except for the many small objects on the seafloor. It is interesting to note, however, that the sample spacing selected is not driven solely by depth: the northwest corner of the area shown in Fig. 8 is deeper than the southwest, but still has smaller sample spacing estimates due to the higher density of data points in the area.

3.2. Ernest Sound, AK

Ernest Sound, AK was surveyed by the NOAA Ship *Fairweather* in 2009 (Baird, 2009), eventually becoming registered as survey H11825. The primary instruments used were Reson 8101, 8111, and 8125 multibeam echosounders (selected according to depth range), with ApplAnix POS/MV 320 motion sensor and differential-aided GNSS using Trimble receivers. A sub-set of the overall survey was extracted as the survey progressed, and was handled as before. A total of 8.57×10^6 observations were used, creating 1.87×10^6 populated nodes after refinement.

As with many Alaskan locations, the area covers depths from shoreline to approximately 220 m, leading to dramatic changes in sample spacing that can be accommodated by the multibeam echosounders in use, which also differ significantly in the number of beams they can produce, and the swathe angle over which they can operate. This mandates a larger super-cell width of $S = 32$ m to ensure stable estimation of data density and sample spacing, Fig. 10: at smaller super-cell widths, the spacing between beams of the multibeam echosounder at depth are greater than the super-cell size, leading to poor estimates. Algorithm control parameters were otherwise maintained as before.

The significantly higher dynamic depth range is reflected in the data density estimates, which vary over more than five orders of magnitude. The predicted sample spacing mostly corresponds to depth, although variations due to instrument choice, repeated survey lines (at centre), and lack of density on the outer swathe edges (west edge, southeast corner) are also observed. The greater dynamic range also requires the sample spacing to vary significantly faster, making this a good test for the assumption that only PCSS grid refinements are required. Fig. 11 shows a sub-set of the variable resolution depth estimates in the vicinity of the southern-most island in the northeast corner of the area, with the over-laid estimation node positions in Fig. 12. The algorithm clearly adapts to the data density variation without any evident "tearing" that might indicate a discontinuous estimate.

4. Discussion

The CHRT algorithm clearly adapts to the available data density, and appears to be sufficiently adaptable to accommodate typical datasets. The current model of a PCSS grid is computationally efficient, although it is conceivable that there may be topographies for which it is inadequate. It would in theory be possible to replace any component regular grid with a more complex data structure where this occurs, however, and only pay the runtime cost for this where absolutely required. In addition to standard spatial indexing structures such as a quadtree, it would be possible to adopt more complex schemes such as a curvilinear

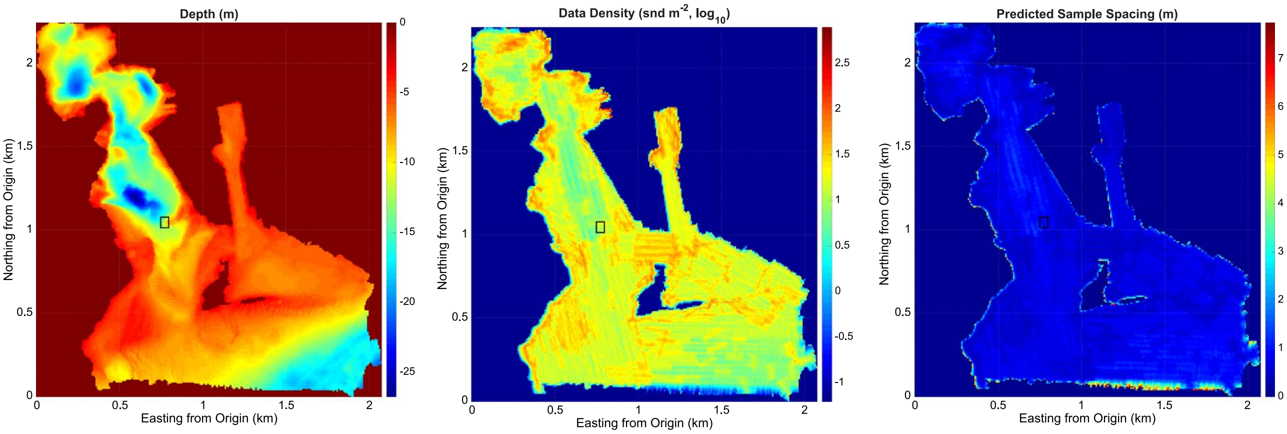
Depth (m)
Data Density (snd m^{-2}, log$_{10}$)
Predicted Sample Spacing (m)
Northing from Origin (km)
Easting from Origin (km)
0
0.5
1
1.5
2
-5
-10
-15
-20
-25
2.5
-0.5
-1
1
2
3
4
5
6
7

Fig. 7. First-pass estimate of (left to right) depth, data density, and predicted sample spacing for Woods Hole, MA, NOAA survey H11077 of 2001. Note logarithmic scale on data density; black rectangle marks location of Figs. 8 and 9; details of super-cell geometry, etc. are given in the text.

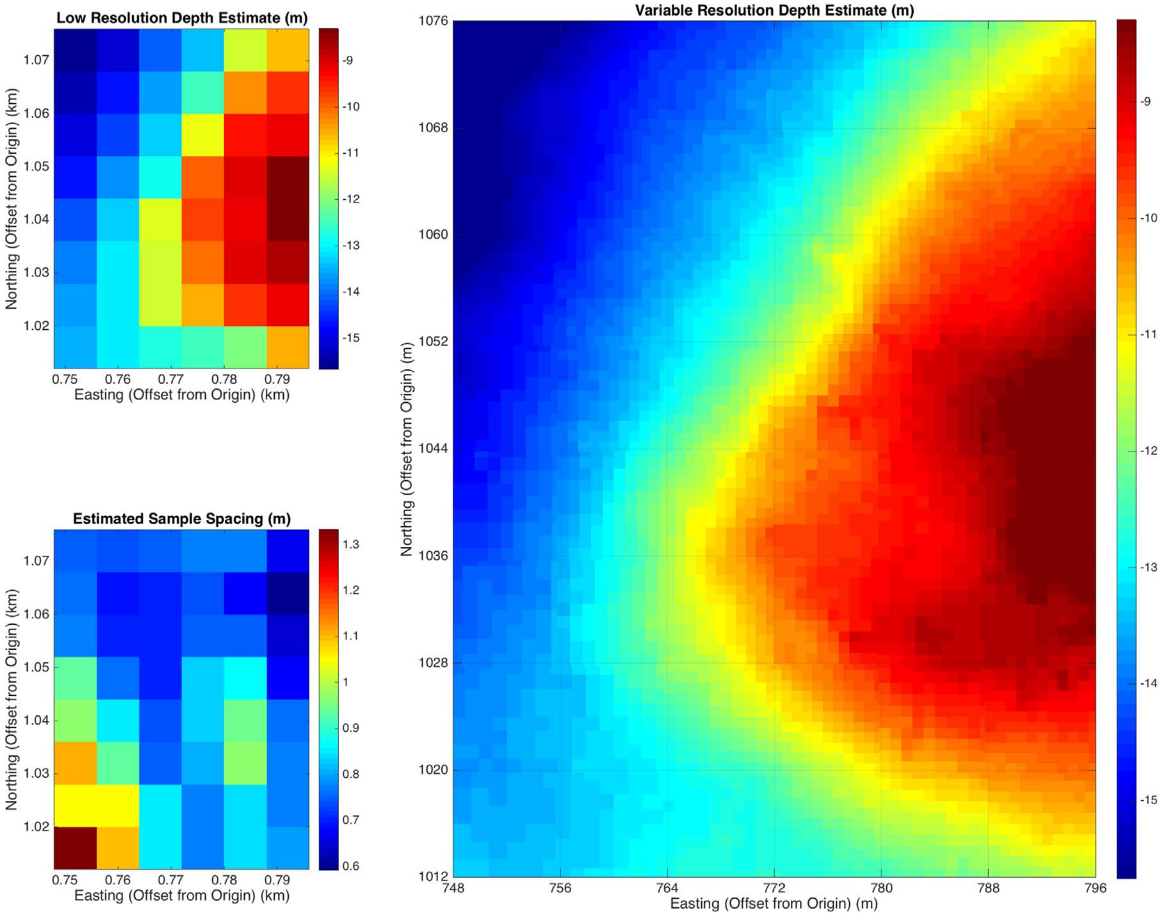

Fig. 8. Variable resolution depth estimate subset (right) with fixed (low) resolution first-pass depth and sample spacing estimates (left). The variation in data density over even relatively small depth ranges results in a significant difference in stable sample spacing.

coordinate frame with suitable basis to provide continuously variable sample spacing, or even an adaptive node placement that was density-seeking. The only question is the trade-off between fidelity of representation and computational complexity.

Data density has been used here as a scheme for predicting the achievable sample spacing (i.e., through (6)). For most sensors, however, it might not be the case that increasing data density continuously improves the achievable resolution. A multibeam echosounder, for example, has a limited aperture per beam, and finite pulselength, with consequent minimum resolvable object size. If the system is stationary, the ground-relative data density will increase without limit, but objects below the resolvable limit will never be resolvable although increasing sample spacing will be possible. Estimating the achievable resolution, given the sensor, might be more physically appropriate, but data density appears to be a reasonable, and pragmatic, proxy. The specifics of the conversion of data density to sample spacing are in any case essentially ad hoc, and the key components of the proposed algorithm do not depend on the details: only that a domain-suitable mapping exists. In theory, it might even be possible to estimate locally, given the noisy, non-uniformly spaced, raw observations, a spatial frequency spectrum (Gardner et al., 2014) and then modify it by an effective point spread function for the sensor. It

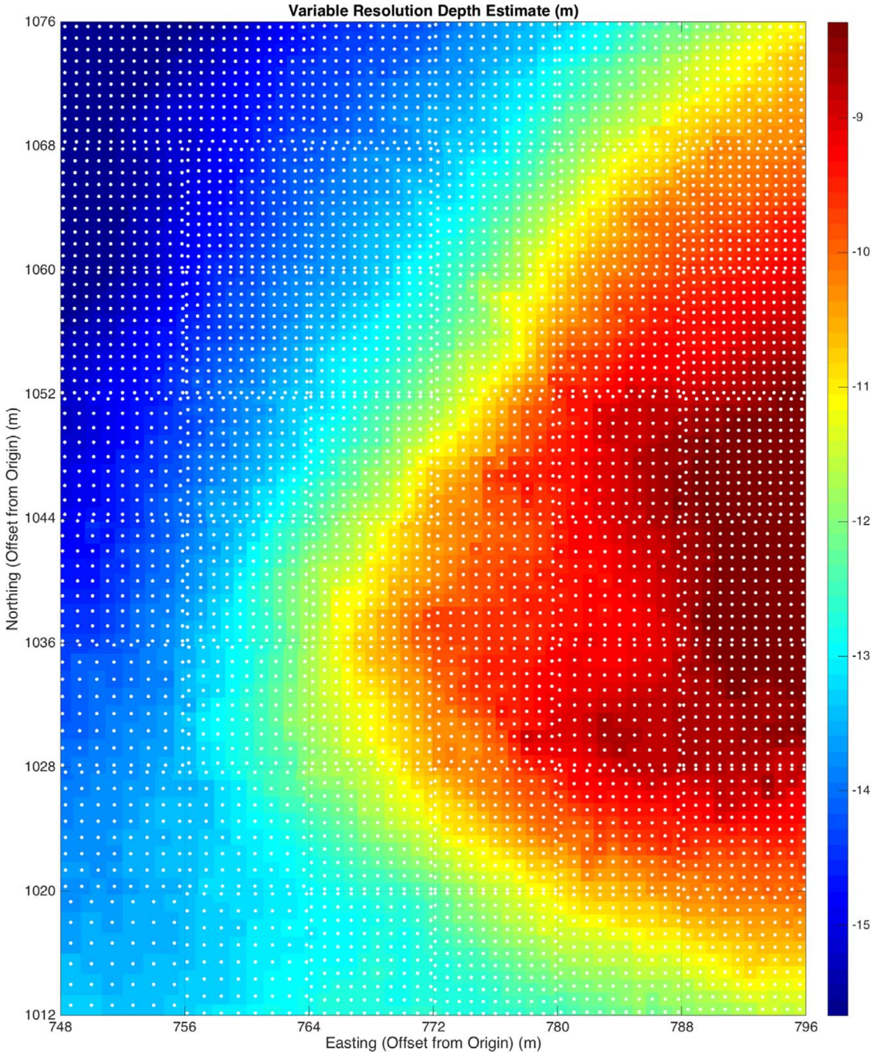

Fig. 9. Variable resolution depth estimates of Fig. 8 with overlaid positions for the estimation nodes derived from first-pass sample spacing estimates.

seems likely, however, that the computational cost of such an approach would heavily outweigh any benefits beyond those derived from a strictly empirical estimate of data density.

Choice of super-cell size is a trade-off between reliability of density estimation and sample spacing flexibility: the larger the super-cell, the more stable the data density estimate, but the less often the sample spacing can be adjusted. Larger super-cells may therefore induce structure in the estimates, since varying background data density will be averaged over the super-cell, leading to over- and under-estimation of the sample spacing. In effect, the data changes more rapidly than the algorithm can adapt. The super-cell must however be at least as large as the maximum expected sample spacing, often a predictable function of the field being measured and the sensor.

Choice of required number of samples is often set by survey specification, but choice of the “noise accommodation” factor, ϵ, has to be based on the data itself. As shown in Fig. 13, increasing ϵ tunes the sample spacing upwards to allow for more observations at each node, and therefore more stable estimation. A suitable value of ϵ is unlikely to be possible to determine *a priori* since the level of noise in the data is not known. Monitoring the performance of a system over time might, however, provide some guidance.

The proposed system assumes implicitly that all data is available *a priori*. The core estimator, however, is not so restricted, and can continuously assimilate data and provide incremental depth reconstructions. The only limitation in extending the system to continuous estimation is the density estimation, which would require adjustment of the refined grids where new data were added. There are no theoretical restrictions on this, although it would require the previous data were available for reprocessing after the refinement is adjusted, since this resets the estimation nodes. A suitable spatial indexing scheme for data would obviously assist in efficiency.

Visibility of objects is a common metric in assessing a survey, which is directly addressed by estimated sample spacing in the proposed system. The sample spacing estimate therefore allows surveyors to demonstrate both that they met any requirement provided, and the converse: that the sample spacing requirement (and hence object visibility) cannot be met with the equipment being used. Such a straightforward metric would avoid unnecessary effort where the equipment available cannot meet the requirements.

To some extent, the core estimator used reflects the nature of the bathymetric estimation problem. The interpretation of multiple competing hypotheses on the true depth being tracked at a node, for example, relies on the principle that there is only one true depth at a point location. Applying the algorithm described here to other scalar fields might necessitate different interpretations, or perhaps a different core estimator. Since the choice of estimator is essentially independent of the supporting data adaptivity, this would affect only the implementation, not the fundamental concepts presented.

Although presented here for bathymetric estimation, the techniques described could be extended to other two-dimensional estimation problems so long as an equivalent to (6) can be formed. In some cases, the same formulation could be used with different estimation techni-

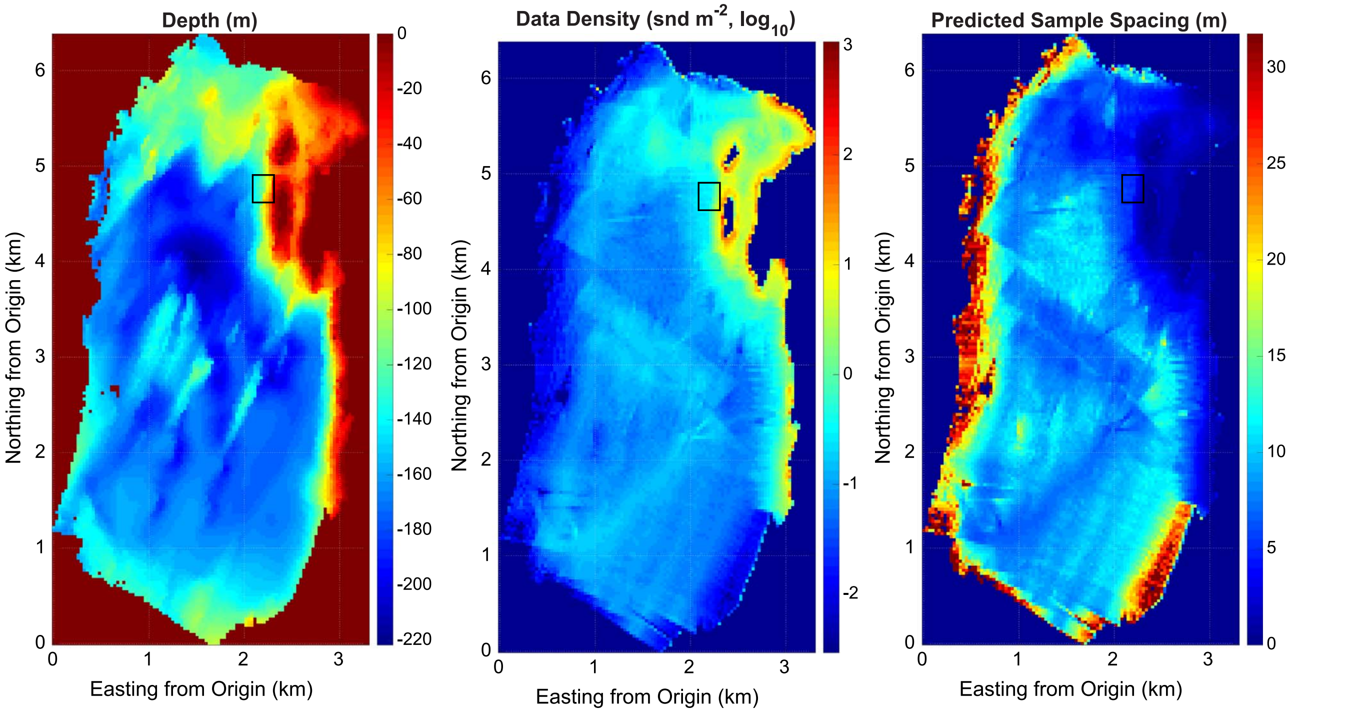

Fig. 10. First-pass estimate of (left to right) depth, data density, and predicted sample spacing for Ernest Sound, AK in the vicinity of Union Point, part of NOAA survey H11825 of 2009. Note logarithmic scale on data density; black rectangle marks location of Figs. 11 and 12.

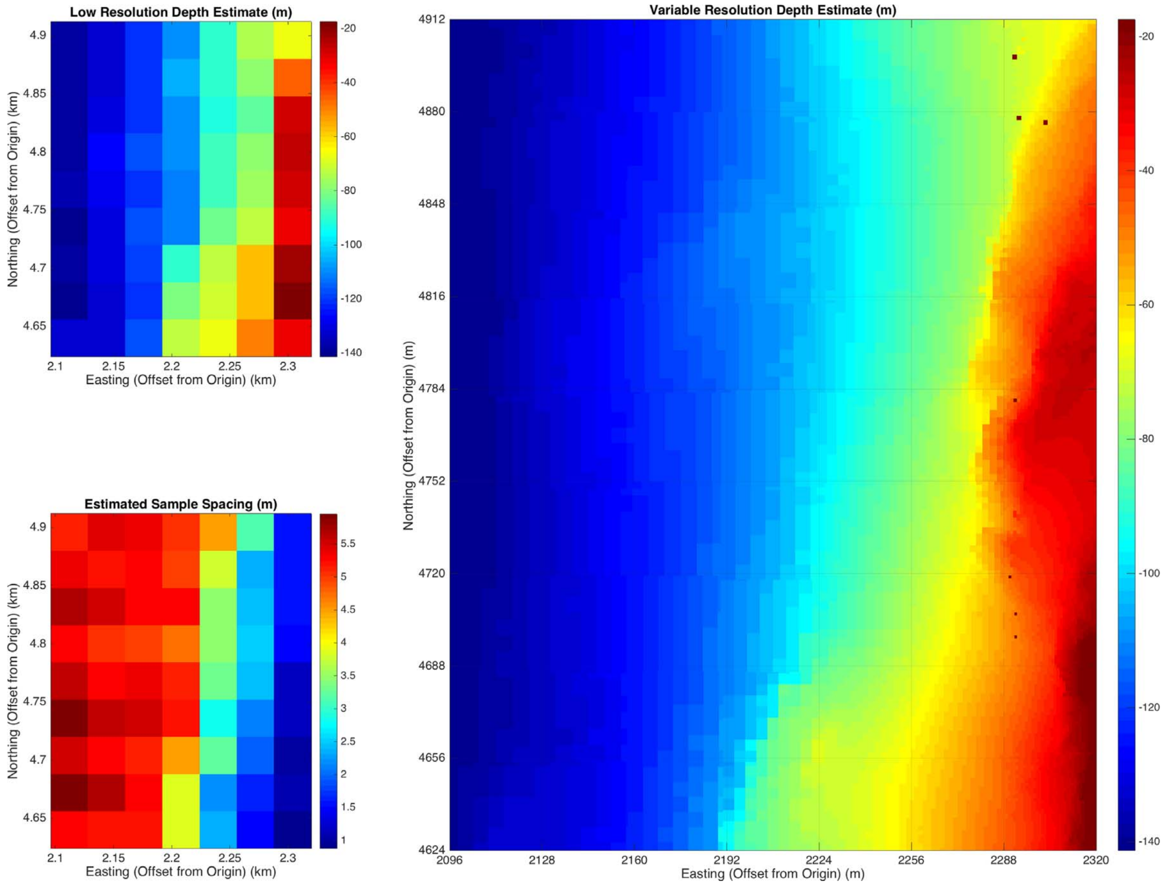
Low Resolution Depth Estimate (m)
Northing (Offset from Origin) (km)
Easting (Offset from Origin) (km)
4.9
4.85
4.8
4.75
4.7
4.65
2.1
2.15
2.2
2.25
2.3
-20
-40
-60
-80
-100
-120
-140
Estimated Sample Spacing (m)
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
Variable Resolution Depth Estimate (m)
Northing (Offset from Origin) (m)
Easting (Offset from Origin) (m)
4912
4880
4848
4816
4784
4752
4720
4688
4656
4624
2096
2128
2160
2192
2224
2256
2288
2320
-20
-40
-60
-80
-100
-120
-140

Fig. 11. Variable resolution depth estimate subset (right) and fixed (low) resolution first-pass depth and sample spacing estimates (left). The rapid drop-off in depth in this area results in a change of sample spacing estimate from approximately 1 to 6 m within 225 m horizontal distance.

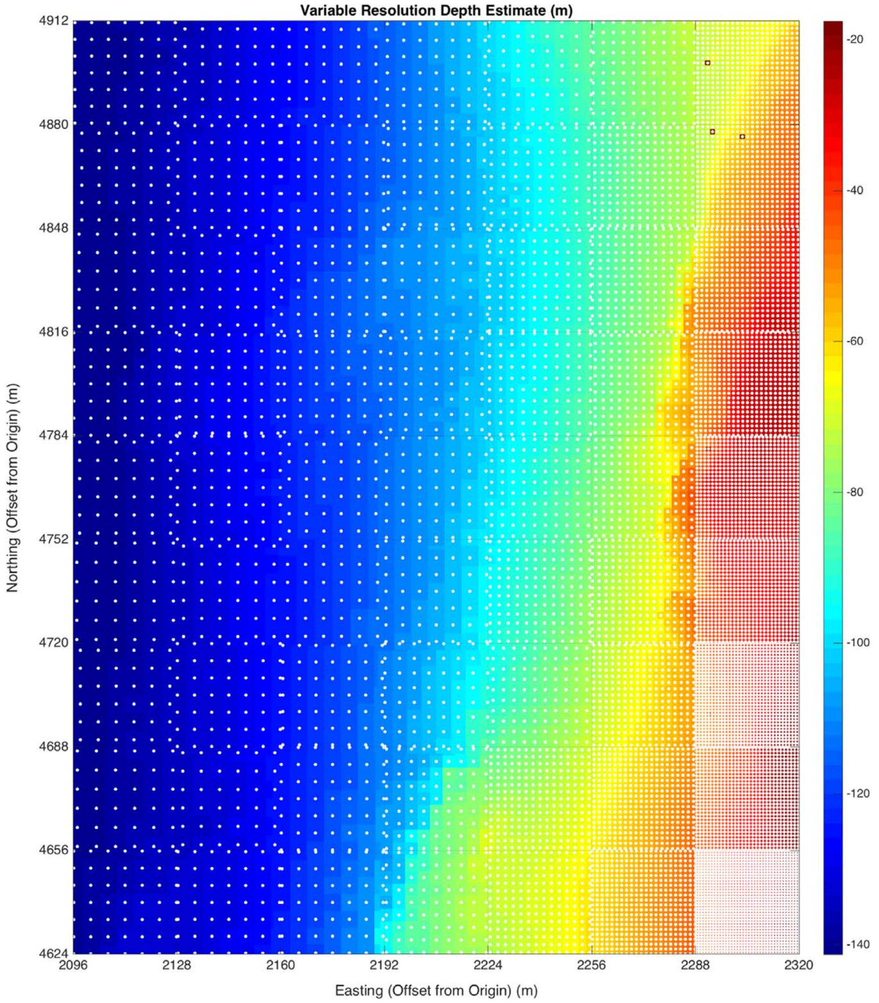

Fig. 12. Variable resolution depth estimates of Fig. 11 with overlaid positions for the estimation nodes derived from first-pass sample spacing estimates.

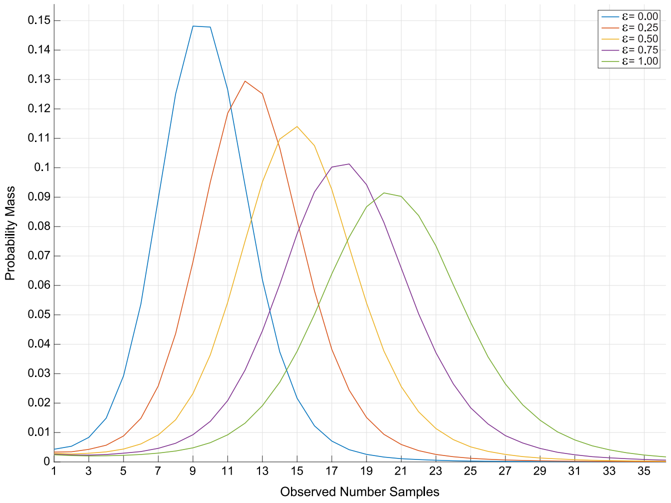

Fig. 13. Probability mass estimates for number of samples in the primary depth hypothesis for varying noise accommodation parameter, ϵ. The predicted sample spacing is proportional to ϵ, so increasing the parameter allows for more observations at each node.

ques for data density; in others, the critical factor might be field complexity rather than data density. In datasets where the observations are not as structured (e.g., where the method of Fig. 3 cannot be used), the estimation of area covered by the observations would be more difficult, but could be approached by assigning each observation a region of support, and computing their spatial union per super-cell. Although the CHRT algorithm is not intended for interpolation per se, such an approach might also allow for the intriguing possibility of using the technique as a means to estimate a plausible interpolation density, given the observations, for a more general sparse interpolation problem. Finally, a different core estimator might be required depending on the properties of the dataset or application area, as outlined above. None of these modifications fundamentally change the core idea of a data-adaptive PCSS grid, with the attendant computational efficiency gains.

5. Conclusions

Based on the concept of a data-adaptive PCSS grid, the CHRT algorithm adapts, simplifies, and extends the CUBE estimator, leading to a method for data-adaptive, large-scale, variable resolution scalar field estimation with resource-efficient implementation. The algorithm estimates data density directly from the observations, and uses this to adapt the scalar field estimation to spatially varying sample spacings; a second pass over the observations completes the estimation.

The algorithm implementation is modular, so that it could be readily adapted to different methods for sample spacing estimation from the data density (or other measures), be efficiently extended to different base representations (other than regular grids) if required by the data, and even support other base estimators than CUBE.

Although primarily designed for bathymetric estimation, the algorithm is sufficiently generic that it could be used for reconstruction of other two-dimensional fields where variable data density naturally implies variable resolution of reconstruction.

Acknowledgements

The support of NOAA Grants NA10NOS4000073 and NA15NOS4000200 (brc) and NOAA's Coast Survey Development Lab (gr) is gratefully acknowledged. The authors would also like to acknowledge the contributions of James Hiebert to the locally-regular, but globally-variable grid representations in what has become CHRT, and Stuart MacGillivray at CARIS, who contributed to the node placement logic.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.cageo.2017.05.013.

References

Arge, L., Larsen, K.G., Mølhave, T., van Walderveen, F., 2010. Cleaning massive sonar point clouds. In: el Abbadi, A., Agrawal, D., Mokbel, M., Zhang, P. (Eds.), Proceedings of the ACM Int. Conf. on Advances in GIS. Assoc. Comp. Machinery, Assoc. Comp. Machinery. San Jose, CA, USA, November, pp. 152–161.

Baird, D.D., 2009. Descriptive Report: Hydrographic Survey H11825 (Ernest Sound, AK). Tech. Rep. National Oceanic and Atmospheric Administation, 1315 East West Highway, Silver Spring, MD.

Barnum, S.R., 2001. Descriptive Report: Hydrographic Survey H11077 (Woods Hole, MA). Tech. Rep. National Oceanic and Atmospheric Administation, 1315 East West Highway, Silver Spring, MD.

Borouchaki, H., Frey, P.J., 1998. Adaptive triangular-quadrilateral mesh generation. Int. J. Numer. Methods Eng. 41 (5), 915–934.

Borouchaki, H., George, P.L., Hecht, F., Laug, P., Saltel, E., 1997a. Delaunay mesh generation governed by metric specifications. Part I. Algorithms. Finite Elem. Anal. Des. 25, 61–83.

Borouchaki, H., George, P.L., Mohammadi, B., 1997b. Delaunay mesh generation governed by metric specifications. Part II. Applications. Finite Elem. Anal. Des. 25, 85–109.

Bourgeois, B.S., Elmore, P.A., Avera, W.E., Zambo, S.J., 2016. Achieving comparable uncertainty estimates with Kalman filters or linear smoothers for bathymetry data. Geochem. Geophys. Geosyst. G3, 17. http://dx.doi.org/10.1002/2015GC006239.

Calder, B.R., Mayer, L.A., 2003. Automatic processing of high-rate, high-density multibeam echosounder data. Geochem. Geophys. Geosyst. 4, 6. http://dx.doi.org/10.1029/2002GC000486.

Calder, B.R., Rice, G., 2011. Design and implementation of an extensible variable resolution bathymetric estimator. In: Proceedings of the U.S. Hydro. Conf. The Hydrographic Society of America, Tampa, FL, pp. 1–15.

Calder, B.R., Elmore, P.A., 2017. Development of an uncertainty propagation equation for scalar fields. Mar. Geod., (Submitted for publication).

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

Cao, W., Huang, W., Russell, R.D., 1999. A study of monitor functions for two-dimensional adaptive mesh generation. SCIAM J. Sci. Comput. 20 (6), 1978–1994.

Chen, C.F., Liu, F.Y., Li, Y.Y., Yan, C.Q., Liu, G.L., 2016. A robust interpolation method for constructing digital elevation models from remote sensing data. Geomorphology 268, 275–287.

Cleveland, W.S., Devlin, S.J., 1988. Locally weighted regression: an approach to regression analysis by local fitting. J. Am. Stat. Soc. 83 (403), 596–610.

Cressie, N., Wikle, C.K., 2011. Statistics for spatio-temporal data (Wiley 604 Series in Probability and Statistics). Wiley, Hoboken, NJ.

Debese, N., Michaux, P., 2002. Détection automatique d'erreurs ponctuelles présentes dans les données bathymétriques multifaisceaux petits fonds. In: Proceedings of the Canadian Hydro. Conf. Can. Hydro. Soc. Toronto, Canada, pp. 1–10.

Debese, N., 2007. Multibeam echosounder data cleaning through an adaptive surface-based approach. In: Proceedings of the US Hydro. Conf. The Hydrographic Society of America, Norfolk, VA, May, pp. 1–18.

Debese, N., Moitié, R., Seube, N., 2012. Multibeam echosounder data cleaning through a hierarchic adapative and robust local surfacing. Comput. Geosci. 46, 330–339.

Du, Z., Wells, D.E., Mayer, L.A., 1996. An approach to automatic detection of outliers in multibeam echosounder data. Hydrog. J. 79, 19–25.

Eeg, J., 1995. On the identification of spikes in soundings. Int. Hydrog. Rev. 72, 33–41.

Fischer, A., Bar-Yoseph, P.Z., 2000. Adaptive mesh generation based on multi-resolution quadtree representation. Int. J. Numer. Methods Eng. 48, 1571–1582.

Gardner, J.V., Armstrong, A.A., Calder, B.R., Beaudoin, J., 2014. So, how deep is the Mariana Trench? Mar. Geod. 37 (1), 1–13.

Gourley, M., Dodd, D., 2000. HIPS: Hydrographic Information Processing System. Tech. Rep. CARIS White Paper 21, Universal Systems Ltd., Fredercton, NB, Canada.

Hell, B., 2011. Mapping Bathymetry: From Measurements to Applications (Ph.D. thesis). Stockholm University, Dept. of Geological Sciences.

Kessenich, J., Sellers, G., Shreiner, D., 2016. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V 9th ed.. Addison-Wesley Professional.

Lirakis, C.B., Bongiovanni, K.P., 2000. Automated multibeam data cleaning and target detection. In: Proceedings of the IEEE Oceans, vol. 1. Marine Technology Society/IEEE Ocean Engineering Society, MTS/IEEE, Providence, RI, USA, September, pp. 719–723.

National Ocean Service, 2016. NOS Hydrographic Surveys Specifications and Deliverables. Tech. Rep., NOAA National Ocean Service, 1315 East West Highway, Silver Spring, MD.

Nurunnabi, A., West, G., Belton, D., 2015. Outlier detection and robust normal-curvature estimation in mobile laser scanning 3d point cloud data. Pattern Recognit. 48 (4), 1400–1415.

Plant, N.G., Holland, K.T., Puleo, J.A., 2002. Analysis of the scale of errors in nearshore bathymetric data. Mar. Geol. 191, 71–86.

Proakis, J.G., Manolakis, D.K., 2006. Digital Signal Processing 4th ed.. Prentice Hall, New Jersey, USA.

Rice, G., Calder, B., 2009. A quantitative approach to the resolution of bathymetric representation. In: Proceedings of the US Hydro. Conf. The Hydrographic Society of America, Norfolk, VA, May, pp. 1–6.

Samet, H., 2006. Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann, San Fransisco, CA.

Smith, W.H.F., Wessel, P., 1990. Gridding with continuous curvature splines in tension. Geophysics 55 (3), 293–305.

Ware, C., Slipp, L., Wong, K.W., Nickerson, B., Wells, D.E., Lee, Y.C., Dodd, D., Costello, G., 1992. A system for cleaning high volume bathymetry. Int. Hydrog. Rev. 69, 77–94.

West, M., Harrison, J., 1997. Bayesian Forecasting and Dynamic Models 2nd ed.. Springer-Verlag, New York.

Zambo, S.J., Elmore, P.A., Perkins, A.L., Bourgeois, B.S., March 2015. Uncertainty estimation for sparse data gridding algorithms. In: Proceedings of the U.S. Hydro. Conf. The Hydrographic Society of America, National Harbor, MD, pp. 1–6.