Handling large amounts of multibeam data

S. Broen
Kongsberg Maritime, Hydrographic Department, Subsea, Norway

Abstract

Today's multibeam echo sounders are capable of outputting enormous amounts of data, at an exceedingly higher rate. Following the evolution of this technology, a need has arisen to be able to process and view this data before it is used to create an end-product. The user wants to be able to confirm that the echo sounder is performing as it should and that the data collected is making sense, before it is stored. He / she does not want to wait until the end of the day after the survey has completed to do this, or that day would have been wasted if the data collected does not meet expectations. It must happen in real time so the user can make necessary calibration and / or adjust the echo sounder settings before further data collection, if the user suspects that something is not right. Also, when such large datasets are being collected there is always a certain probability that individual data units are erroneous. Correcting such fluctuations manually can be very time-consuming so there is also a need to perform such corrections automatically.

1. Introduction

Kongsberg Maritime (KM) has developed technology to meet the multibeam echo sounder (MBE) users expectations of real time processing and display of depth points. It is being utilized through KM's software application for MBE control and data logging, named Seafloor Information System (SIS). A core technology in this application is named the Grid-Engine (GE). The GE will accept all depths in real time, in addition to seabed image data which is 3-5 times more, and make a bin model out of them. This bin model is then processed in real time, and a Display Model is created. This Display Model will create and maintain several grids in different resolution and detail. At the highest resolution, this Display Model presents the terrain model at full resolution. The operator can quickly change the map scale and get another view of the whole area in a lower resolution. The GE is primarily used to create terrain models from Multibeam data, but there is no restrictions so it has also been used to create terrain models from laser data. In this project the height of trees was displayed using the GE and the Display Model.

This text will initially give a rough introduction to how we collect data. Next it will explain the path that the data takes through the system, followed by an explanation of how data is processed and displayed. Finally there is a summary and some thoughts of what the future might bring.

2. Depth points

KM MBE systems consist of several core parts. Perhaps the most obvious parts are the transducers. There is a transmitter transducer and a receiver transducer, or an array of such installed underneath a boat vessel. The transmitting transducers convert electrical voltage to sound waves, and the receiver does the opposite. The received sound signal is basically sound echoes from the seafloor. Data are created based on the sound-travel time and received strength. Due to the sound-absorbing properties of the seafloor sediment, one is able to determine the sediment type based on the received signal strength. This is what constitutes the seabed image. The travel time of the sound pulse, combined with the known beam-emitting angle from the transducer, constitutes a basis for the depth points. This basis has to be combined with sensor data that monitor the vessel orientation. Finally an estimate of the water columns sound travel times has to adjust the points. After these steps of processing, we have achieved a best-possible (realtime) estimate of an actual point at the seafloor. Point clouds consist of consecutive swaths of depth points, as illustrated by Figure 1.

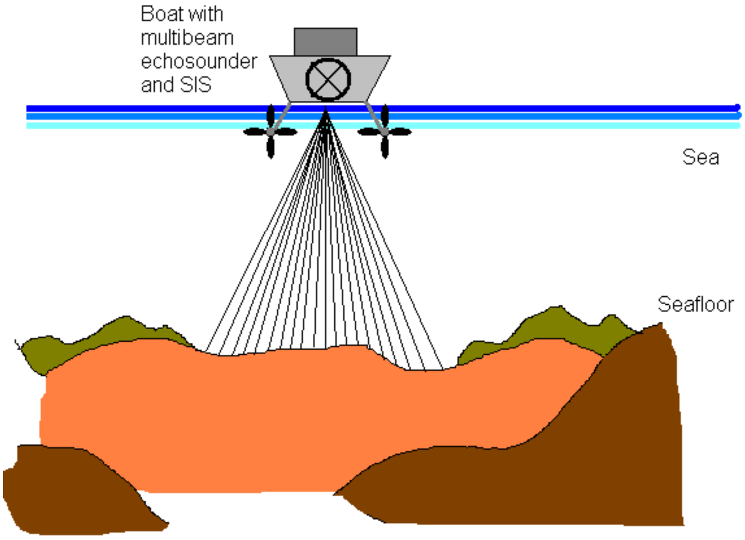

Figure 1. Collection of depth points.

2.1 Sensors

Necessary directional properties are being measured by a number of sensors. Examples of such properties are roll, pitch, heading and position of the boat, as illustrated by Figure 2.

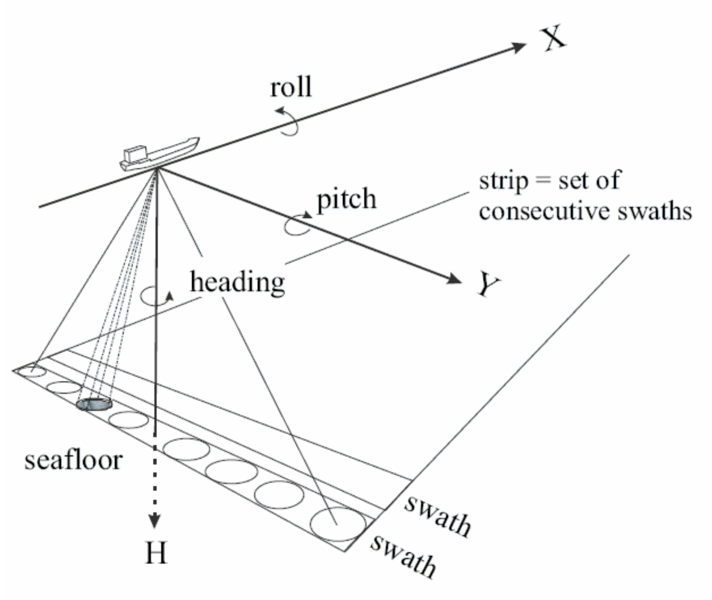

Figure 2. Vessel directional parameters.

Another important parameter is heave, which is also monitored by sensors. The data coming from these sensors are being combined with the depth data from the transducer in a Processing Unit (PU). The PU uses the sensor data to adjust the depth points, so that we can tell where the soundbeam actually hit the seafloor. Because of the often very long travel path of the sound waves, even the smallest change in vessel orientation will result in a corresponding change in depth points at the seafloor that are several multitudes larger. Because of this, it is critical that the sensors are being as accurate as possible. Also, it is necessary with a buffer to temporarily store raw depths because of the travel time of the beams. The received echo has to combine with the corresponding sensor reading at the time of transmittance. The result will be a seafloor point estimate, where we have assumed a consistent soundspeed through out the sea. Of course this is not the case, as we will see in section 2.2.

2.2 Water Sound Speed Profile

Snells law (*see Hovem, 2000; chapter 3*) states that a ray of sound traveling in liquid through the border between two layers of water with different densities and sound speeds, will result in a reflected and a refracted ray. The reflected ray will have an angle between its direction and border, that is the same as the angle between the incoming ray and the border. The refracted ray will have an angle based on the angle between the incoming ray and the border, and the difference in soundspeed and density between the layers. Because of this, a ray of sound moving from the echosounder to the seafloor will not be a straight line, but rather a line that will constantly have its direction altered at the transition between water layers. This can be seen in Figure 3, which is taken from SIS. The rightmost figure shows the zoomed-in area defined in the mid-figure.

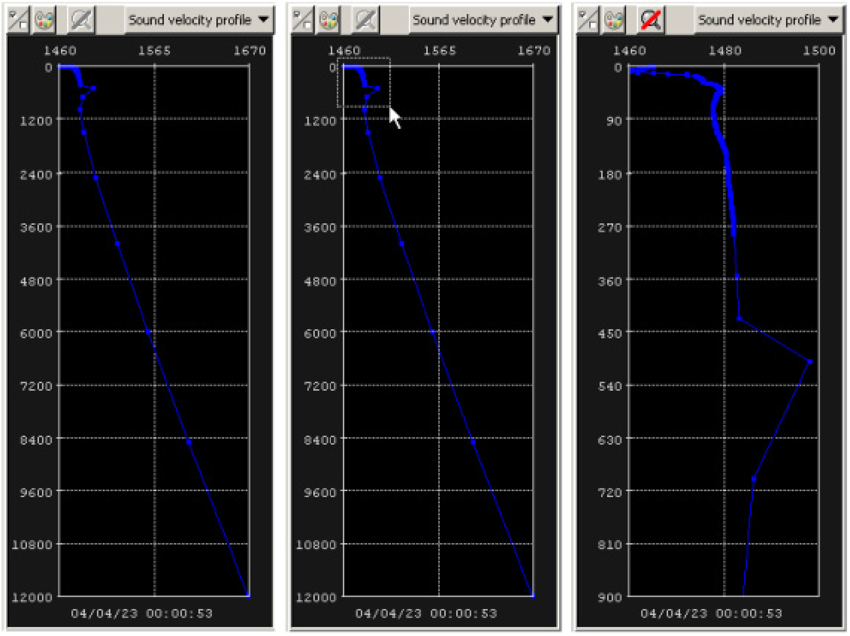

Figure 3. A ray of sound being altered by differences in water layers.

The smaller the angle between the border and the incoming ray, the larger the distance between measured consecutive points at the seafloor. In some cases, the outlying rays might never even reach the seafloor. Because of this property, we will need to measure the sound speeds in the ocean prior to surveying. To measure the sound speeds, it's common to use a soundspeed sensor probe that is sunk down the ocean, and then pulled up again. The result of this measurement will be a sound speed profile (SSP), and this profile will adjust the measured depth points.

2.3 Projection

We need to be able to reference our depths to the real world. To achieve this we need to **project** our depths onto a world reference system. There are various such systems, and they can be set be the SIS-user. When the necessary **parameters** in SIS are set correctly and some necessary pre-processing is done, the depth records may now be displayed and logged. The SIS geographical view may now look like Figure 4.

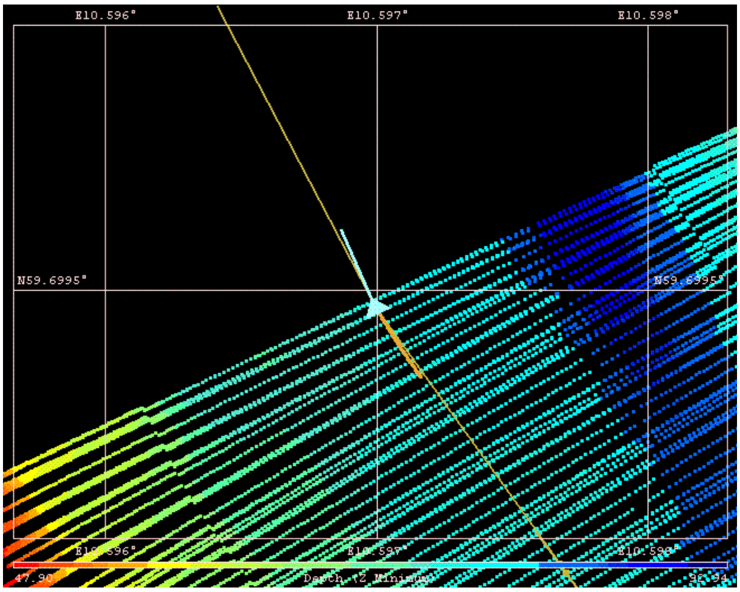

Figure 4. SIS Geographical View.

Note that every logged depth isn't necessarily displayed in the geographical view at once. The world coordinates relative to our selected projection can be seen on the left, right and upper part of the figure.

2.4 Datapath

The resulting data is then broadcasted on an Ethernet at which an operator PC (HWS) is also connected to. This PC has SIS installed on it. It is used to monitor data, log data and to control the echosounders. This datapath is illustrated by Figure 5.

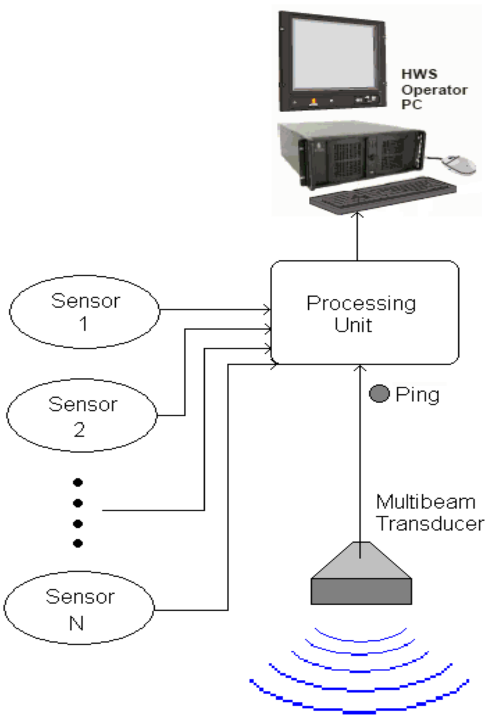

Figure 5. Depth point data path.

After just a few seconds with surveying the depth points start to seriously **stack up**. It becomes evident that there should be some mechanism to arrange all the points into something more eye-pleasing, like a terrain model. This is where the GE comes in.

3. The Grid Engine

The GE is a standalone server-based application. It is written in Java which ensures platform independence and automatic memory allocation/deallocation by the Java Runtime Environment. The bottleneck memory-wise then becomes the actual amount of installed RAM on the HWS. The server approach makes the GE see connected applications (SIS) as client ”websites”, and thus there are in principle no restrictions as to how many websites that can connect to this server. Storage of data also becomes more convenient this way.

3.1 GE datapath

The GE is split into several ***servlets*** which each handle different kinds of requests. SIS has a GE communication interface that responds to user input and transforms them to URL commands. For example, the user might want to zoom into a specific area, specified by a bounding area. Such an area can easily be drawn in the geographical view in SIS by left-clicking and move the mouse the draw a rectangle. An URL command is the sent from the GE-interface in SIS requesting the desired operation along with parameters, and taken care of by the *Survey servlet* in GE. This servlet may in turn take other servlets in use to perform its task. I will not go into each servlet in this talk, as it will be too time consuming. Figure 6 explains the basic flow of data inside a HWS. Figure 7 shows the basic flow of data of the GE.

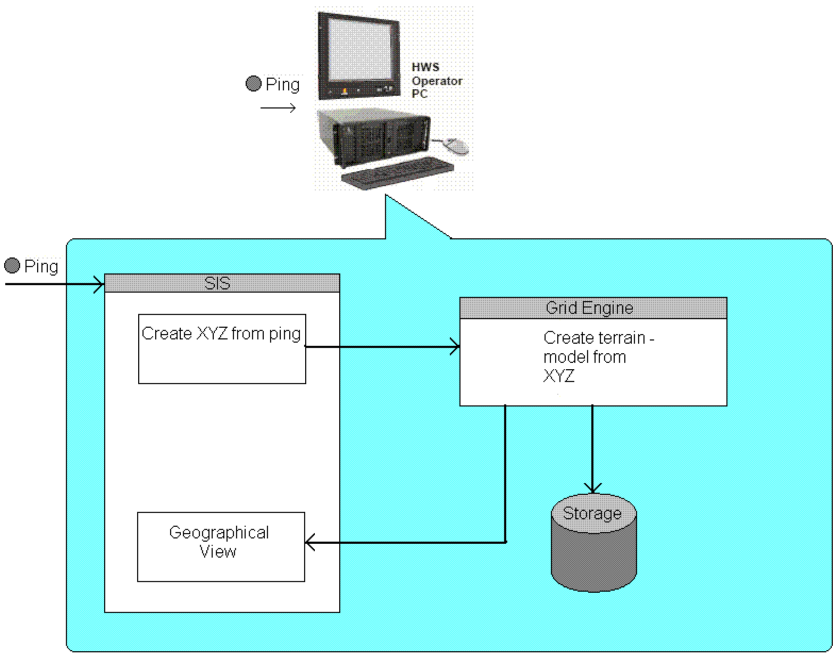

Figure 6. HWS datapath.

As can be seen in 7, data is being processed when it enters the system. A process grid is then created and stored to disk, then display grids are created and stored to disk. Connected clients may request to view the stored files. The following sections will explain each individual step of Figure 7.

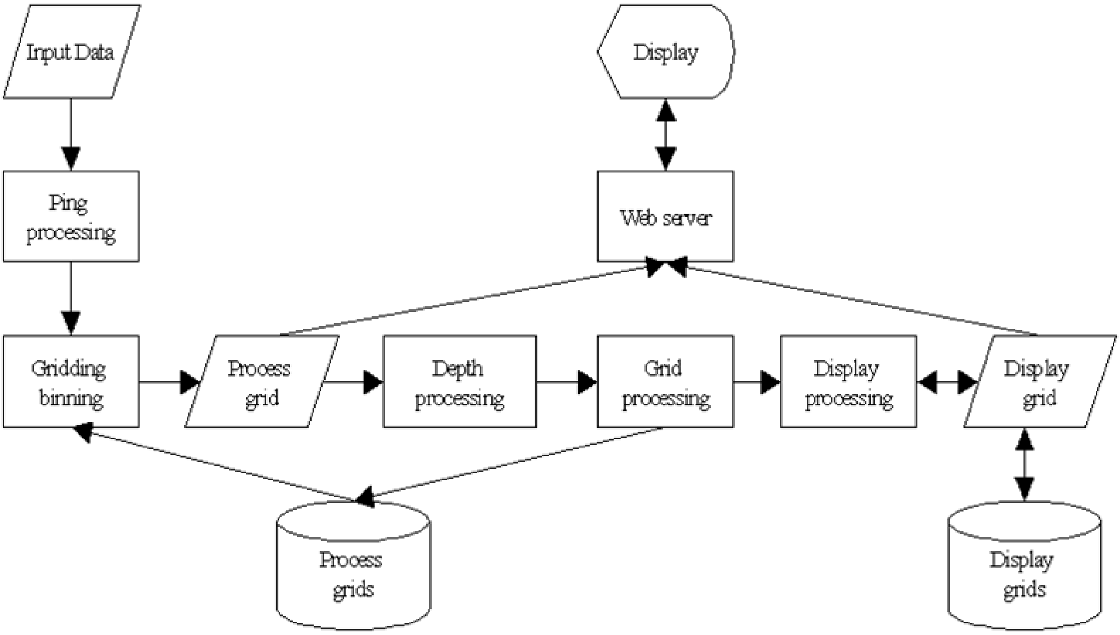

Figure 7. GE datapath.

3.2 Realtime ping processing

Realtime ping processing is performed right after the data enters the system (Røang et al., 2002), and before gridding and binning of data. The purpose of the ping processing is to flag points that probably are erroneous according to some relatively simple rules.

The rules can be divided into two main types; the first kind only looks at a single ping at a time, while the second kind needs a neighborhood around each beam in every ping. The rules are applied in sequence, and the single ping rules are always applied before the neighborhood rules.

The extraction of neighborhoods requires a buffer of pings, which results in a small delay. The size of the delay depends on how many pings needs to be stored in the buffer, which in turn is depth dependent.

There are two single-ping rules. First, the time rule, which flags beams whose two-way time range is longer than the time difference between two consecutive pings. This rule is rarely used. Second, the overhang rule, which flags beams that create an overhang situation where the loca-

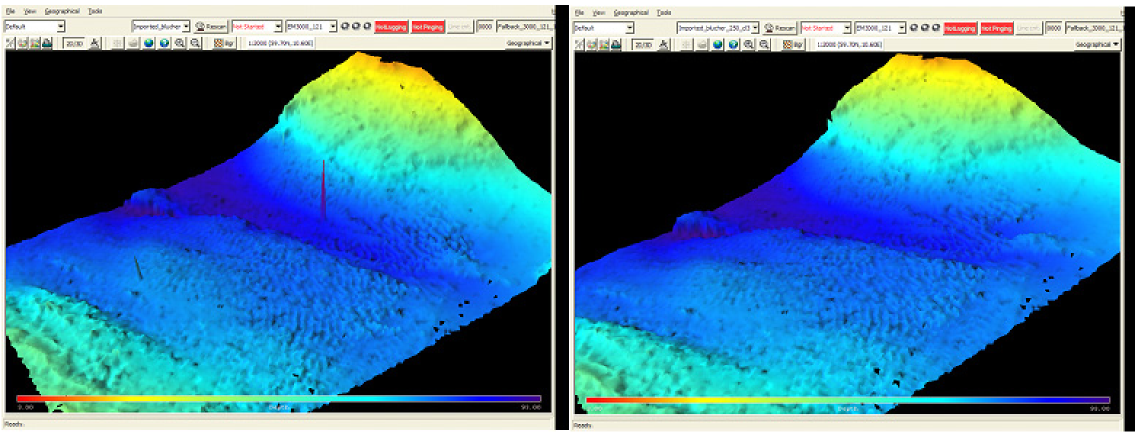

Figure 8. Removing spikes with ping processing.

tion of the depth of that beam is closer to the vertical projected depth from the sounder to the seafloor, than the depth of the neighboring beam that should have been closer.

There are two neighborhood rules. First, the median rule, which compares the absolute value of the depth difference between the center point and each point in the neighborhood. The center point is flagged if its depth differs too much from the neighbors. Second, the quantile rule, which is more adaptive than the median rule, in that it takes into account the local variability of the topography and the noise. Figure 8 demonstrates the result of a ping processing.

3.3 Realtime depth processing

Data are now being put into a Process Grid, where they can be processed realtime (Debese, 1998). A Process Grid contains a square number of *cells*. The number- and size of the cells (in m^2) may be set by the user. In deeper waters, larger cells are more appropriate in order to get required amount of points inside each cell. Points are sorted into the cells by their xy-value. When we later come back to the same area, data from the previous lines will be read back and reprocessed together with data from the next line. This is called Area Based Data Cleaning (ABDC). But before we can do this we have to split and merge all cells in the grid to ***processing units.*** Based on the number of points and the depth variation, we try to merge or split with neighboring cell(s). Like this, steeper seafloor will require smaller units in the xy plane. Units can also be empty. After an iteration of this sorting, the grid may look for instance like Figure 9.

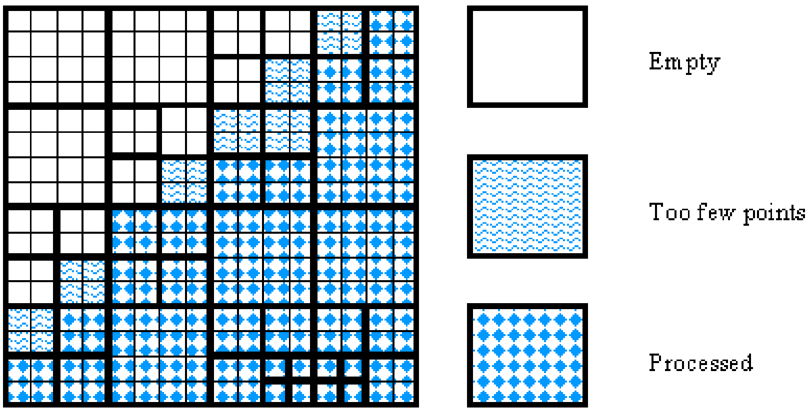

Figure 9. Arrangement of process units.

After the processing grid has been arranged into ***processing units***, the list of units is traversed and a surface estimate is calculated for each of them. The traditional approach is to simply use the mean depth value, meaning the surface approach will always be flat. Instead of doing this, the GE uses the iterated reweighted least squares (***IRLS) algorithm*** to calculate a ***first, second or third degree polynome surface*** that model the terrain much better (Debese and Bisquay, 1999). See the Figure 10.

The IRLS will take the set of points, and start off by calculating a ”least squares” surface. Now each point is given a ”weight” based on their distance to this surface. For each iteration, the surface is recalculated based on the point weights. Point weights will pull the surface closer, ”heavier” points have more influence than ”lighter” points. Then each point is given a new weight based on their new distance to the surface. This goes on until either the surface converges, or the number of iterations has passed a threshold. Note that outlying points (4 red dots) in Figure 10 have been flagged out based on their weight.

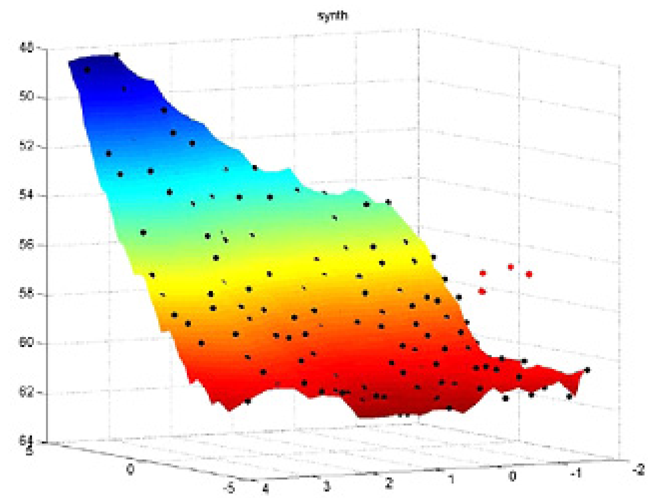

Figure 10. Terrain model estimated for one unit.

For each iterations, points are given new weights based on their residue according to a Tukey function, see Figure 11. This function has been subject to a lot of research, and the approach Kongsberg Maritime has chosen is based on Debese (1998).

Points that are (most likely) erroneous will be given little or no weight, and after the iterations they can be flagged out based on , for example, a threshold weight value. The valid points are sorted and some of the values from the minimum, maximum and median points are transferred to the ***display grid***. All “point set variables” (i.e. all the “z” values) are sorted and transferred separately to the display grid.

The display grid contains “point set” data and cell data. Point set data are minimum, maximum and median point data for each point set variable (“z “variable). Cell data are scalar data valid for the entire cell such as average backscatter and maximum depth difference. Data is stored in different “layers” (files on disk) where each layer represents a level of detail (LOD). SIS chooses which LOD to display based on the current span in latitude / longitude in the geographical view (zooming). The cell data for the lowest level are calculated in the processing grid using all valid points in a cell. When data are processed from one LOD to a higher LOD, data from one to four Display Cells are used to compute the values for the Display Cell above, see

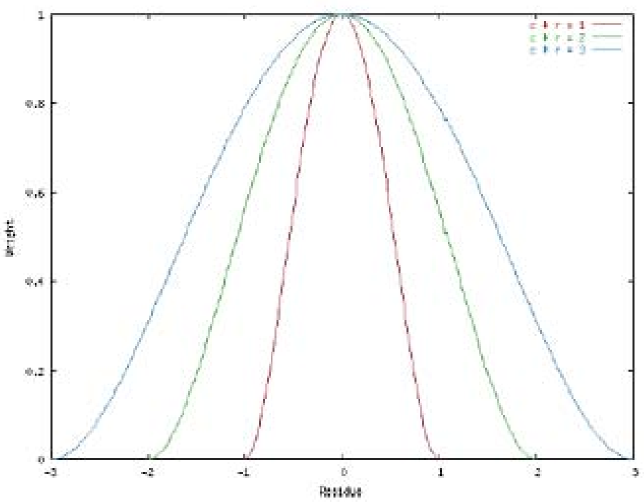

Figure 11. Tukey weight functions of residue and weight.

Figure 12. The minimum and maximum point set data is computed by taking the point set with the minimum and maximum value from the level below. The new median depth is computed by taking the median point set from the level below which is closest to the center.

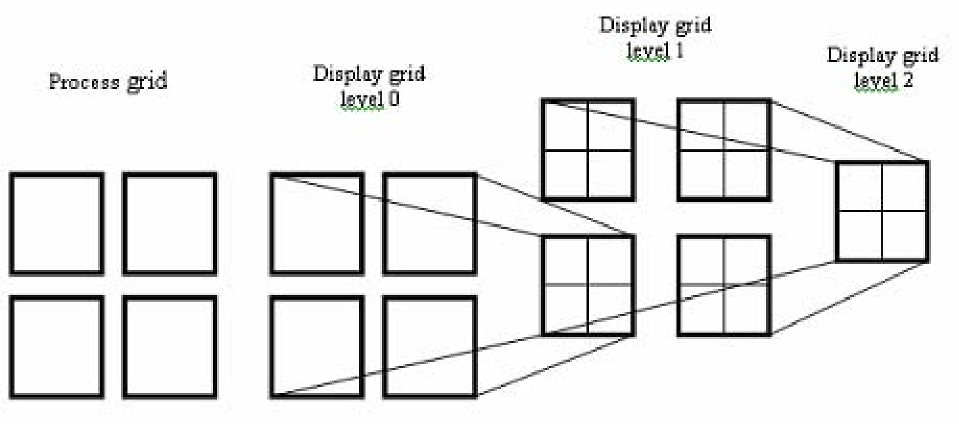

Figure 12. Process grids and display grids.

For the different point set data, separate min, max and median values are calculated for cells at each LOD. Cell data is either calculated as average, area average, minimum or maximum values of the values from the level below. (When area average is computed, the value is weighted with the number of process cells for that cell.).

The number of level of detail is dependant of the number of cells in the processing grid. The more number of cells in each direction the more level of detail. A processing grid with 64x64 cells will have 6 levels of detail, i.e. log2(64).

All the grids at all level of details are stored to disk and updated realtime. By request from SIS, the GE will read the files, and return them to SIS. This means that the user can access the grids in real time, and SIS may switch between the levels of detail seamlessly, based on how far in or out the user is zooming into the seafloor. Figure 13 shows an example of how the geographical view in SIS can display data at a given LOD.

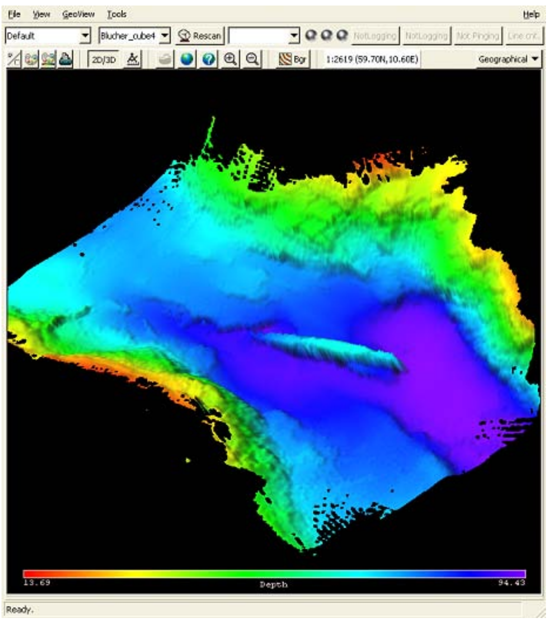

Figure 13. A terrain model made from depth points.

4. Summary and future developments

The KM MBEs produce massive collections of depths. These collections constitute our Point Clouds. The Point Clouds are sent to the GE, which process all data, and creates a terrain model at multiple resolution levels. The GE is under constant development, and has seen numerous upgrades since its release. The next version will contain functionality that removes the requirement that the SIS user manually has to set GE parameters (number of cells, cell size). The *dynamic* grid engine will figure this out on its own, and it will be featured in SIS in the near future. There are plans to further improve it in the future. For example, there are plans to make more effective memory allocations based on predicted vessel heading. So the GE will try to ”guess” which grids it must load into memory.

References

Debese N. (1998). Application d'un estimateur robuste à la détection des erreurs ponctuelles dans les données multifaisceaux: l'estimateur de Tukey. Rapport d'étude du SHOM, 002/98. Brest: SHOM.

Debese, N. and Bisquay, H. (1999). Automatic Detection of Erroneous Soundings in Multibeam Data through a Robust Estimator. Proceedings of the HYDRO ’99 conference, Plymouth, 8 pp.

Hovem, J.M. (2000). Marin Akustikk. Lecture syllabus, Institutt for teleteknikk/akustikk, NTNU, Trondheim, Norway.

Røang K., Eliassen I.K. and Heggelund Y. (2002). Real Time Data Cleaning of Multibeam Echosounder Data. Christian Michelsen Research AS, Bergen, Norway.