Distinguishing features from outliers in automatic Kriging-based filtering of MBES data: a comparative study.

Peter Bottelier [*], Christian Briese [†], Natasha Hennis [‡], Roderik Lindenbergh [‡], Norbert Pfeifer [§]

fax: +31 15 278 3711, mail: r.c.lindenbergh@lr.tudelft.nl

June 15, 2004

Abstract

Multi beam echo sounding is the state of the art way for surveying sea bottoms. The elevation of the sea floor is obtained strip wise by measuring the time it takes for sound signals, emitted simultaneously in different directions, to travel to the sea bottom and back. In this paper we compare various ways of filtering erroneous soundings from MBES data sets, all based on the interpolation method Kriging. This research was initiated because of the problems that a classic filtering method had with distinguishing blunders from points belonging to features, like pipelines. Fortunately, methods developed for filtering laser altimetry data do not exhibit this unwanted behavior.

1 Introduction.

Multi beam echo sounding (MBES) is the state of the art technique of surveying sea floors. A set of sound signals, called a ping, is emitted, at distinct angles, towards the sea floor. The time it takes for a signal to travel to the sea floor and back to the emitter is used, together with the angle of emittance, to determine the position and depth of the point of reflection on the sea floor. MBES surveys produce large data sets. Typically, several measurements are available for every square meter in coastal waters. In case of offshore engineering, often real-time processing of the MBES data is required, for example to verify if a pipeline construction turned out successfully. The processing of MBES consists basically of two steps: outliers should be removed and the data density should be decreased, while maintaining a realistic model of the sea bottom. For this purpose a first automatic filtering and thinning algorithm was designed, based on Kriging. Unfortunately, this algorithm had an important drawback: not only the blunders, also points representing pipelines were removed by the algorithm.

As the removal of features like pipelines is highly unwanted, methods for improvement were considered. In this paper we discuss these methods and test them on four different data sets of MBES data containing different configurations of pipelines.

The first new method is an extension of the original algorithm. In the original algorithm, a covariance function is computed in one specific direction, the so called ping direction. This 1D covariance function is used by the interpolation method Kriging to predict a depth value that is compared with the measured value. If the difference exceeds a certain test value, the measured

[*] Fugro Intersite BV

[†] Institute of Photogrammetry and Remote Sensing, Vienna University of Technology

[‡] Mathematical Geodesy and Positioning, Delft University of Technology

[§] Photogrammetry and Remote Sensing, Delft University of Technology

depth is considered an outlier. Measurements from pipelines perpendicular to the ping direction are easily considered outliers. We show that by considering a 2D covariance function this problem can be partially solved.

An alternative method, also using the Kriging paradigm, was originally designed for filtering laser altimetry data. This type of data, where the time is measured that an emitted light pulse needs for traveling from the laser sensor, mounted on an aircraft, towards the earth and back, is often used to determine a Digital Elevation Model of the bare earth. But laser points are not only reflected by the bare earth but by trees as well. The 'tree points' are filtered away as follows. By using a covariance function with a smoothing effect, an average elevation is determined of all available laser points. Now this average elevation divides the measurements in two groups: points below the average are probably bare earth points, points above it are probably tree points. A recursive version of this algorithm turns out to be very effective. We apply this method on our pipeline data after determining suitable parameters. In the multi beam setting we want to filter the 'real' outliers/spikes from the sea bottom data including the pipes.

Finally the results of the methods are compared on the data sets, giving very satisfying results in most cases.

2 Multi Beam Echo Sounding.

Echo sounding is based on the principle that water is an excellent medium for the transmission of sound waves and that a sound pulse will reflect from the bottom of a water mass to all directions and thus return to its source as an echo. If a pulse is emitted from the bottom of the ship at an angle ψ with the vertical line through the emitter, the depth d and the position y of the sea floor hit by this pulse are determined from

$$(d, y) = \frac{1}{2} ct(\cos\psi, \sin\psi)$$

where t denotes the time it takes between the initiation of the sound pulse, traveling with velocity c, and reception of the echo, see also Figure 1.

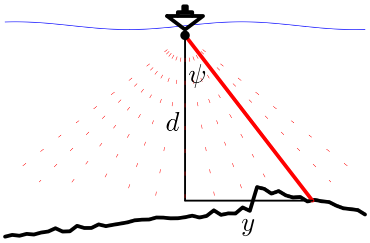

Figure 1: The multi beam geometry

As is illustrated in Figure 1, a swath MBES system, [3], transmits an acoustic pulse that resembles a fan. It is wide in one direction (across track) and narrow in the perpendicular direction (along track). The echo is received by a transducer, which segments it into multiple smaller beams. The widths of these beams are in the order of one to a few degrees, depending on the system. Per pulse transmission (ping) a high number of depths are thus generated. From a single vessel's track a band of depths are obtained, opposed to a single line of depths obtained from the traditional single beam echo sounder. By combining the depth d and the position y with the position of the ship, determined real-time by GPS (Global Positioning System) and INS (Inertial Navigation System), one obtains coordinate system referenced xyz-data of the sea floor.

The total list of MBES error sources is extensive. A major error source is wave-induced movement of the ship that can be divided in pitch, roll, heave and heading errors. However, these errors should be eliminated immediately by the INS motion sensors mounted at the ship. Another important error source is the positioning of the ship, done by GPS. The errors we consider however are not the systematic ones, but the real blunders, caused by reflection of the signal on fish or debris, or by occasional electronic errors.

3 Filter methods.

All methods we consider are based on the geostatistical interpolation method Kriging, [2, 4]. Therefore we first recall its basics. The first two filter methods apply Kriging-based cross-validation. In the first case a 1D covariance function is used, in the second a 2D covariance function. The last, recursive, method uses the covariance function for defining a smoothed approximation of the sea floor.

3.1 Kriging.

Kriging, as an interpolation method, determines weights w_i for the prediction of a depth $\hat{z}_0 = w_1 z_1 + \ldots w_n z_n$ at location p_0, given depth observations $z_1, \ldots, z_n$ at locations $p_1, \ldots, p_n$. First we discuss the covariance function, that encodes the variability of the measurements, then we show how Kriging uses the covariance function to determine the weights in an optimal way.

Correlation: the covariance function.

The covariance function of a set of, say, height measurements describes the correlation, the degree of relation, of the measurements as a function of the distance r between two measurements. In general one expects that two measurements that are close to each other are more correlated than measurements that are on a larger distance. The most simple case, the case that we consider, is when the correlation is assumed to depend on the distance only and not on other parameters like direction.

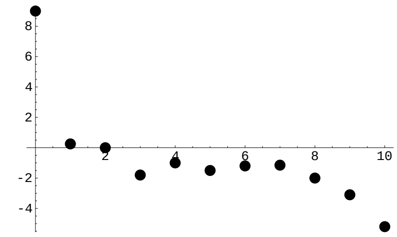

Figure 2: An empirical covariance ‘function’ showing periodic correlation.

The classic way to obtain a covariance function is to analyze the differences $z_i - \overline{z}$ between the height observations and the average of the height observations. More precisely, if the expected height is independent of the position, that is, if the signal is stationary, a first estimation for all expected heights z_i is given by $E\{z_i\} = \overline{z}$. The covariance between two height observations z_i and z_j is now defined as

$$\mathrm{Cov}(z_i, z_j) \;=\; E\{(z_i - E\{z_i\})(z_j - E\{z_j\})\} \;=\; E\{(z_i - \overline{z})(z_j - \overline{z})\}.$$

By computing and grouping the covariances between any two observations, an empirical covariance ‘function’ is determined. An example of a 1D covariance function $C(r)$, where the grouping is only done with respect to the distance r, is shown in Figure 2.

One can also take a covariance function that is suited to perform some special task. For example, a covariance function that drops relatively slow has a smoothing effect on the data interpolation.

Ordinary Kriging.

Suppose that, as above, we are given height measurements $z_1, \ldots, z_n$ and want to predict a height $\hat{z}_0 = w_1 z_1 + w_2 z_2 + \ldots w_n z_n$ at position (x_0, y_0). Assume moreover that we are given a covariance function $\tilde{C}$ that produces a correlation value $\tilde{C}_{ij}$ between two positions (x_i, y_i) and (x_j, y_j). The ordinary Kriging system consists of $n + 1$ equations:

$$\begin{aligned} w_1\tilde{C}_{i1} + w_2\tilde{C}_{i2} + \cdots + w_n\tilde{C}_{in} + \mu &= \tilde{C}_{i0} \qquad \text{for all } i = 1, \ldots, n \\ w_1 + w_2 + \cdots + w_n &= 1. \end{aligned} \tag{1}$$

This implies that the weights can be found by

$$\begin{pmatrix} \tilde{w}_1 \\ \vdots \\ \tilde{w}_n \\ \mu \end{pmatrix} = \begin{pmatrix} \tilde{C}_{11} & \ldots & \tilde{C}_{1n} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \tilde{C}_{n1} & \ldots & \tilde{C}_{nn} & 1 \\ 1 & \ldots & 1 & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \tilde{C}_{10} \\ \vdots \\ \tilde{C}_{n0} \\ 1 \end{pmatrix}. \tag{2}$$

The μ is a so-called Lagrange multiplier and is an extra variable added to make the system solvable.

The ordinary Kriging system is obtained within a *random function model.* This means that with every position a random variable is associated. In the case of ordinary Kriging it is assumed that the expected height is independent of the location and that the mean of the heights is unknown. Ordinary Kriging aims at optimizing two parameters and this optimization results in Equations (1) and (2).

First of all the expected error $r_0 = \hat{z}_0 - z_0$ in the height prediction should be *unbiased.* It can be shown that this condition $E\{r_0\} = 0$ leads to the equation $w_1 + \cdots + w_n = 1$. The other aim is to minimize the error variance $\mathrm{Var}\{r_0\}$. Looking for the *best* solution for the weights under this condition gives the other Ordinary Kriging equations. Moreover, one obtains a formula for the error variance $\tilde{\sigma}^2_{\text{prediction}} = \mathrm{Var}\{r_0\}$:

$$\tilde{\sigma}^2_{\text{prediction}} = \tilde{\sigma}^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \tilde{C}_{ij} - 2\sum_{i=1}^{n} w_i \tilde{C}_{i0} = -\mu + \sum_{i=1}^{n} w_i \tilde{C}_{i0}$$

As the predicted height $\hat{z}_0 = w_1 z_1 + w_2 z_2 + \ldots w_n z_n$ forms a *linear* combination of the measured heights, it is by now clear why Ordinary Kriging is often called BLUP, Best Linear Unbiased Prediction.

3.2 1D cross validation.

The first method we discuss was developed by Peter Bottelier as part of a data thinning algorithm, [1]. Given one ping of MBES data, outliers are eliminated in two steps. In a first step, all blunders, defined as soundings above a certain minimal depth and below some maximal depth, are eliminated. These two threshold values are based on a priori depth information on the surveyed area.

The second step is a cross validation step: a depth value $\hat{z}_{(x,y)}$ is predicted for every sounding location (x, y). This prediction $\hat{z}$ is compared to the actual measurement $z_{(x,y)}$. If the difference between the predicted and measured depth, relative to the standard deviation, is tested too big, the sounding is rejected.

Predicting the depth value, 1D case.

The prediction of the depth values is done ping-wise by means of Kriging using a covariance function based on the soundings in the ping. For this purpose first an experimental, discrete covariance function $C(w_k)$ is determined for every ping. A bin width $w = \sum_{i=1}^{n-1} \|p_i - p_{i+1}\|/(n-1)$ is defined as the average horizontal separation distance between consecutive soundings in the ping. The bin W_k consist of all pairs of soundings $\{p_i, p_j\}$ s.t.

$$(k - \frac{1}{2}) \cdot w < \|p_i - p_j\| \leq (k + \frac{1}{2}) \cdot w \qquad \text{for} \quad k = 1, \ldots, K$$

The experimental covariance function is given by

$$\begin{aligned} C(0) &= \frac{1}{n} \sum_{i=1}^{n} (z_i - \overline{z})^2 \\ C(w_k) &= \frac{1}{2 \cdot |W_K|} \cdot \sum_{\{i,j\} \in W_k} \frac{(z_i - \overline{z})(z_j - \overline{z})}{(z_i - \overline{z})^2 + (z_j - \overline{z})^2} \cdot C(0), \qquad \text{for} \quad k = 1, \ldots, K \end{aligned}$$

To decrease the irregular tendency, this empirical covariance function is smoothed with a moving average of five points. From this smoothed function the distance d of first zero crossing, and the correlation length ξ, that is, the distance at which the covariance value is dropped, for the first time, to half of the value at distance zero, is determined. The curvature κ at the origin is defined as $\kappa = (\log 0.3149)/\log(\xi/d)$. These parameters are used to fix the following analytical covariance function

$$c_A(s) = c_A(0)(1 - (\frac{s}{d})^{\kappa})e^{-(\frac{s}{d})^{\kappa}} \quad \text{where} \quad c_A(0) = C(0) \quad \text{and} \quad s, d > 0.$$

Please check that indeed $c_A(\xi) \approx \frac{1}{2} c_A(0)$. Using this covariance function, a predicted value $\hat{z}_{(x,y)}$ is determined for each measurement position (x, y) in the ping. Here, only the two neighboring soundings on the left and the two neighboring soundings on the right of the sounding to cross validate are used in the interpolation. Note that from the Kriging we obtain $\sigma_{\text{prediction}}$ as well.

Testing the prediction.

The last step is to compare the predicted value $\hat{z}_{(x,y)}$ to the measured value $z_{(x,y)}$. This is not done directly, but, again, the variability of the measurements is taken into account. One component used is the point noise σ_{noise}, estimated by $\sigma^2_{\text{noise}} = 0.81(C(0) - C(w_1))$. The other component is the prediction error $\sigma_{\text{prediction}}$. This leads to the following test:

$$\frac{|z_{(x,y)} - \hat{z}_{(x,y)}|}{\sqrt{\sigma^2_{\text{noise}} + \sigma^2_{\text{prediction}}}} \quad > \quad C_1$$

If the test value C_1 is exceeded, the measurement is considered an outlier and removed. Here it is assumed that the depth data are normally distributed. The test value C_1 is based on a 5% confidence level, yielding a critical level of $C_1 = 1.96$. For normally distributed data this confidence level implies that 5% of the measurements is expected to be tested as outlier. Of course this test is very similar to the so-called w-test, a test developed in Geodesy to detect a single outlier in a series of measurements, [8].

Computational efforts.

We show that the ping wise determination of an experimental covariance function is the most time consuming part of this method. Suppose we have a data set of n points divided over $\sqrt{n}$ pings and $\sqrt{n}$ beams. For determining a ping-wise covariance function $O(\sqrt{n} \cdot \sqrt{n})$ operations are needed, as all pairs of soundings in the ping are considered. This operation is repeated for every ping, so the total computational complexity for this step is $O(n\sqrt{n})$. Once the covariance is determined, the cross validation is performed in constant time per sounding, as only four neighbors are involved, taking a total of $O(n)$ time. This is also the time needed for the threshold step in the beginning.

3.3 2D cross validation.

The main difference with the 1D cross validation method is that in the 2D method not only soundings in the current ping are used for determining a covariance function and for the actual cross validation. Instead of one ping, a data set of at least three pings and three beams is considered. Again, first the gross blunders are eliminated by thresholding.

Selecting Neighboring points.

The experimental covariance function is determined in basically the same way, only the bin width w is now defined as the maximum of the average separation distances of the pings and the beams. All data in the data set, consisting of different beams and pings, are used for determining the experimental covariance function. The same analytic covariance function model is fitted on the experimental covariance data as above.

Since the purpose of this algorithm is to cross validate in two directions, mandatory neighbors in both different pings and different beams are included in the Kriging prediction. Including the mandatory neighbors, a fixed number of e.g. eight closest neighbors is used in the cross validation.

Computational complexity.

For consistency assume again that we have a data set of n points divided over $\sqrt{n}$ pings and $\sqrt{n}$ beams. If all soundings are included in determining the experimental covariance function $O(n^2)$ operations are needed, as now all pairs in the complete data set are evaluated. The efficiency of the cross validation depends completely on the selection of the neighbors in the Kriging prediction. If the soundings are sorted on forehand, and for example only the neighbors in the current ping and current beam are used, cross validation can once again be performed in constant time per sounding. If in the worst case a fixed number of closest neighbors is found via a brute force approach, one single cross validation takes $O(n)$ time which means that, as for determining the covariance function, $O(n^2)$ operations are needed for cross-validation of all soundings.

3.4 Robust Interpolation

Like the cross validation techniques described in the previous sections, the robust interpolation is a method to filter outliers from a point set describing a surface, see also [6, 7]. Its name is derived from two techniques, i) interpolation, where a continuous surface is interpolated from a given set of points, and ii) the 'robust' approach, developed in adjustment theory [5], which is an iterative technique to remove erroneous observations from over-determined systems of equations by giving suspicious observations less and less influence during the iterations. As interpolation method Kriging is used, and its covariance function will be described next, followed by a description of the iteration scheme and the robust weighting technique.

The covariance model.

The covariance function for the area under study follows the Gaussian model, [4], with the same correlation length ξ, see Subsection 3.2, everywhere. The nugget component of the covariance function is set for each point independently.

$$c_A(s) = \begin{cases} C_0 e^{-(\frac{s \ln 2}{\xi})^2} & : \quad s > 0 \\ V_{p_i} = \frac{\sigma^2}{p_i} + C_0 & : \quad s = 0 \end{cases} \tag{3}$$

Here, C_0 denotes the maximum of the covariance function, defined by $C_0 = \sum_i (z_i - \overline{z})^2 - \sigma^2$ in case the nugget effect is not considered. By σ the measurement accuracy is denoted, while p_i

equals the weight of each observation, which is adapted during the iterations, as will be described later. The nugget effect is σ^2/p_i. The values V_{p_i} are the diagonal elements ($\tilde{C}_{ii}$ in Equation (2)) in the covariance matrix $\mathbf{C}$ of the Kriging system and can be different for each observation, i.e. each observed height.

Obtaining a smooth surface, including all observations.

The nugget effect is only considered in the Kriging matrix, but not in the vector $\mathbf{c} = \tilde{C}_{i0}$ of correlations with the interpolation position, compare Equation (2). That is, if the position of the prediction coincides with the position of observation z_i, the nugget effect is considered in the entry $\tilde{C}_{ii}$ in the covariance matrix $\mathbf{C}$ but not in the entry $\tilde{C}_{i0}$ in the vector of correlations $\mathbf{c}$. This leads to a continuous surface, where the random measurement errors have been filtered out. If all p_i are one, than the distances from the interpolated surface to the observations (i.e., the residuals) are in the order of σ.

The correlation length ξ influences the smoothness of the final surface. The value ξ can either be determined by analysis of the empirical variogram, but a more practical approach is to set ξ to the average point distance in the area. This works satisfactory, if i) the point distribution is homogenous, and ii) the oversampling of the surface is not too dense with respect to the measurement accuracy. In other words, the measurement accuracy must be sufficiently good or the sampling distance sufficiently large, so that the difference between observations (in our case heights) from adjacent locations (xy-positions) is mainly due to surface characteristics and not to measurement noise.

The heights $\hat{z}_i$ of all observation positions are estimated by the Kriging system, with $p_i \equiv 1$ for all observations, leading to $V_{p_i} = C_0 + \sigma^2$. The vector of residuals $\mathbf{r} = \hat{\mathbf{z}} - \mathbf{z}$ of the differences between the estimations $\hat{\mathbf{z}}$ and the observations $\mathbf{z}$ is given by

$$\mathbf{r} = \sigma^2 \cdot \begin{pmatrix} 1/p_1 & & \\ & \ddots & \\ & & 1/p_n \end{pmatrix} \cdot \mathbf{C}^{-1} \cdot \mathbf{z}$$

The iteration.

These residuals are used to compute weights in the range [0,1], indicating if a point shall have a low or high influence on the surface run in the next iteration. For this purpose the following weight function is used:

$$p_i = \frac{1}{1 + (ar_i)^b} \tag{4}$$

The values a and b are the parameters of the weight function: $1/a$ is the half-width value, while b controls steepness. With these weights new nugget effects are assigned to the individual points and the computation of the surface and the residuals is repeated. During these iterations erroneous points get less and less weight. A lower weight means less influence on the run of the surface and leads to a larger residual. Points neighboring outliers, which may have a big residual after the first iteration, get higher weights again, as their 3D position is not in contradiction to the neighboring points and fit well into the surface. After a pre-described number of iterations or when the maximum change of the residuals drops below a threshold value, the algorithm terminates and all observations with a residual exceeding a critical value are eliminated.

Computational aspects.

Determining the correlation length takes either $O(n^2)$, in the experimental covariogram case, or $O(n)$ time, the average point distance case. Most time consuming is taking the inverse of the covariance matrix $\mathbf{C}$, e.g. $O(n^3)$ time for Gaussian elimination if all observations are incorporated, which has to be done in every iteration step, as the diagonal elements of **C** are updated at each step. Therefore it is essential that the number of points treated at once is not too big.

4 Filtering the data sets.

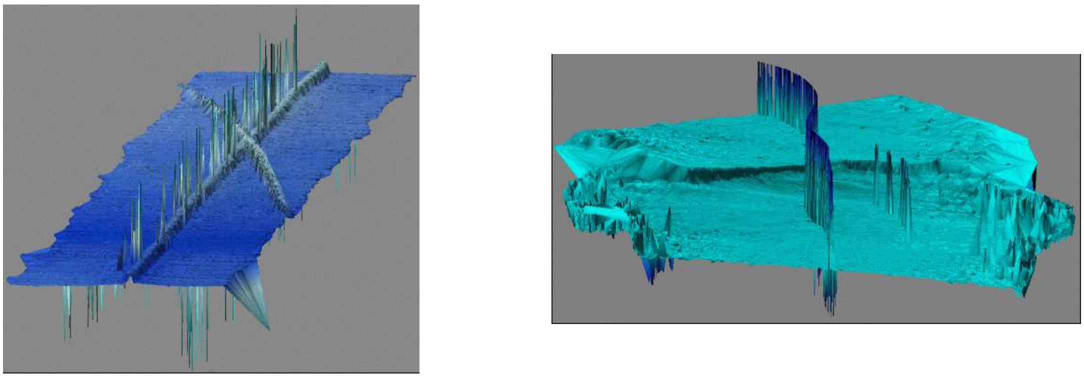

Figure 3: Data sets 1 and 2 before processing.

For testing the different methods, four data sets are used. We will concentrate on the first two however, that were both obtained by a multi beam system mounted on a ROV, a Remotely Operated Vehicle, operating at about 15 m above the sea floor.

The first data set has been acquired using a Reson Seabat 9001 multi beam system. This system has 60 beams and a ping rate of 7 pings a second. The average depth is around 145 meters and the distance between consecutive pings is approximately 0.03 m and the approximate distance between two adjoining beams is 0.1 m.

The second data set has been acquired with the Reson Seabat 8101 multi beam system, which has 101 beams and a ping rate of 9 pings per second. Here the average depth is approximately 13 meters in the first half of the data set and then 23 meters in the second half. The approximate distance between consecutive pings is 0.13 m and between two adjoining beams 0.45 m. This data set does not cover pipelines but represents a short steep sloop. In Figure 3 one can clearly see a belt of spikes. This indicates a defect beam since it is only one sounding thick.

The third and fourth data set have an average depth of around 300 meters and approximate distance of 0.12-0.14 m between consecutive pings and 0.19-0.24 m between adjoining beams. In both data sets a pipe and some templates are present.

Parameter choices for the different methods.

In the 2D method it was possible to vary the number of pings processed at once and to select the number of neighbors. The data presented here were obtained by processing 10 pings at a time while 8 neighbors were used for the cross validation. Although the exact numbers change, similar results were obtained with different parameter choices.

The robust method was run with seven iterations. The measurement accuracy was $\sigma = 10cm$, the half-width was set on $1/a = 2\sigma$, while $b = 2$.

4.1 The results.

Unfortunately it is not very clear what distinguishes good soundings from bad soundings, or, what soundings should be removed. The methods discussed above all divide the soundings objectively in one of these two categories. In most cases it is (subjectively) clear, by a simple visualization, if a wrong decision is made. Often, [8], the following two types of wrong decisions are distinguished:

Type I error: a sounding is rejected, although it is correct.
Type II error: a sounding is accepted, although it is wrong.

In Table 1 the number of removed outliers are given. Clearly, the 1D method finds a lot of 'outliers'. As the 1D method is designed as a data thinning procedure this is an essential part of the algorithm. In the case of data set 1 however, a lot of type I errors are made, due to the presence of the pipelines. This was the reason to consider alternative methods. In general the 2D method removes less soundings, but as we will see later, still type I errors are made. The robust method removes the smallest number of points and seems to perform the best, by minimizing both the number of type I and type II errors.

	1D method	**2D method**	**Robust method**
# MBES 1	104 040	111 476	111 476
# outliers	3 149	1 394	193
% outliers	3.03	1.25	0.17
# MBES 2	125 245	130 641	130 641
# outliers	3 853	1 746	1 407
% outliers	3.08	1.34	1.08
# MBES 3	297 958	303 014	303 014
# outliers	4 364	3 084	256
% outliers	1.46	1.02	0.08
# MBES 4	215 557	219 217	219 217
# outliers	1 955	3 479	1 154
% outliers	0.91	1.59	0.53

Table 1: Overview of numbers and percentages of removed outliers by the different methods.

Computation time.

It was reported that especially the determination of the covariance function for the 2D method took a lot of computational effort on a standard PC (several hours), while the 1D method is a bit faster. The robust method took 2'23" on a notebook with a Centrino 1.7 GHz CPU for the filtering of the largest example (300 000 soundings). Here it should be noted that the robust method can be considered to be professionally implemented, while for the 1D and 2D method only an experimental Matlab program was available.

Data set 1.

In Figure 4 DTM's of data set 1 are presented, after the three methods were applied. Both the 2D method and the robust method give normal looking DTM's. In Figure 5 however, the removed soundings are clearly visible in red. Even in the 2D method a lot of soundings on both pipelines were marked as outlier, although the DTM of the robust method looks very good with these outliers included. We conclude that most of the reported outliers for the 1D and 2D approach are removed unwantedly and should be considered type I errors. The number of outliers found by the robust method is much lower, while the DTM generated from the remaining points has no anomalies. Therefore we conclude that in this case, the robust method gives the best result.

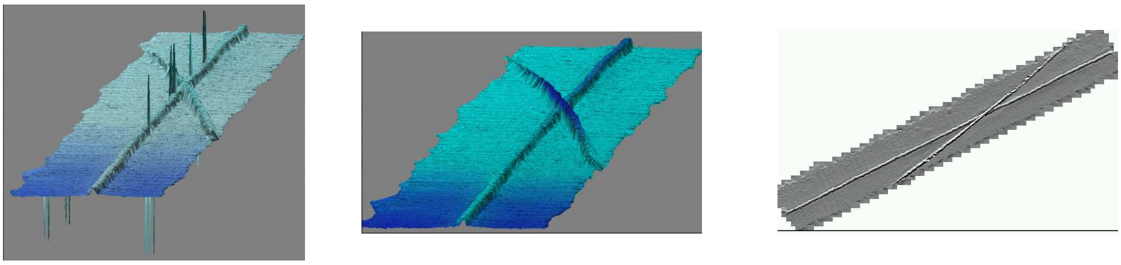

Figure 4: Data set 1 after processing with the 1D, the 2D and the robust approach.

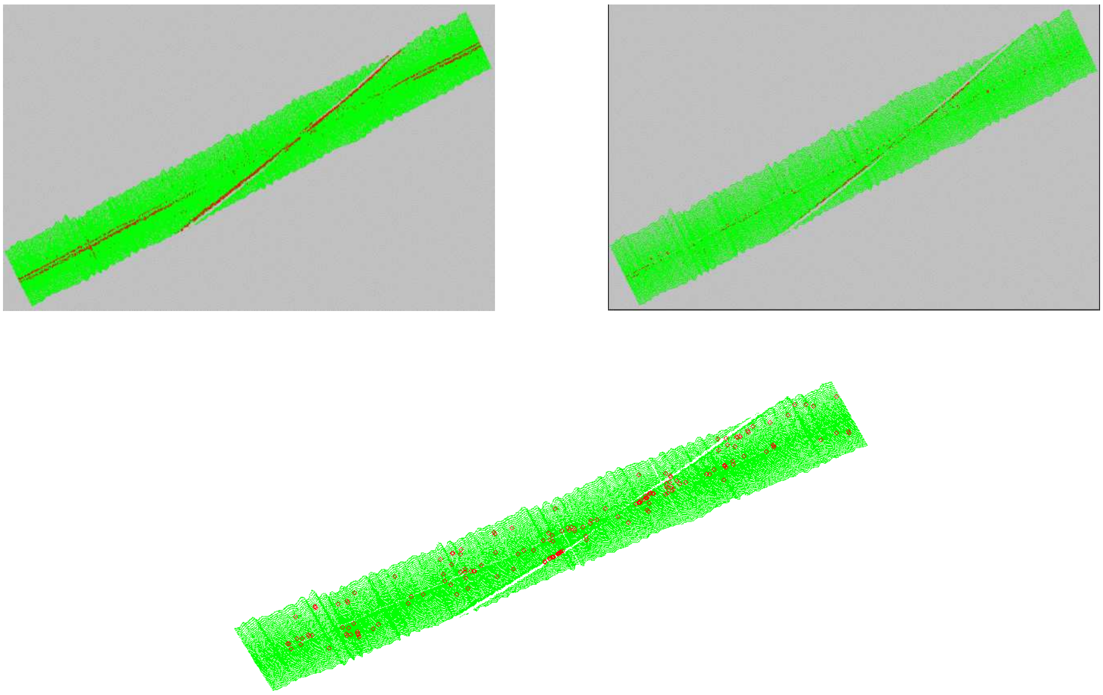

Figure 5: Data set 1: in red the data removed by the 1D, the 2D and the robust approach.

Data set 2.

Again, the 1D approach finds the highest number of outliers. Many of these outliers are on the borders of the data set, where the point density is lower. Most of these border points are removed by the 2D algorithm and the robust one as well, compare also Figure 7. An explanation could be that the variation between neighboring points increases as the point density decreases, implying that the cross validation test is easier failed. Other outliers removed by the 1D approach are on a common ping and are probably removed in the ping cross validation step. The resulting DTM's do still contain anomalies on the borders probably due to local low point densities. Again the robust method removes the smallest number of points, although the resulting DTM looks the best.

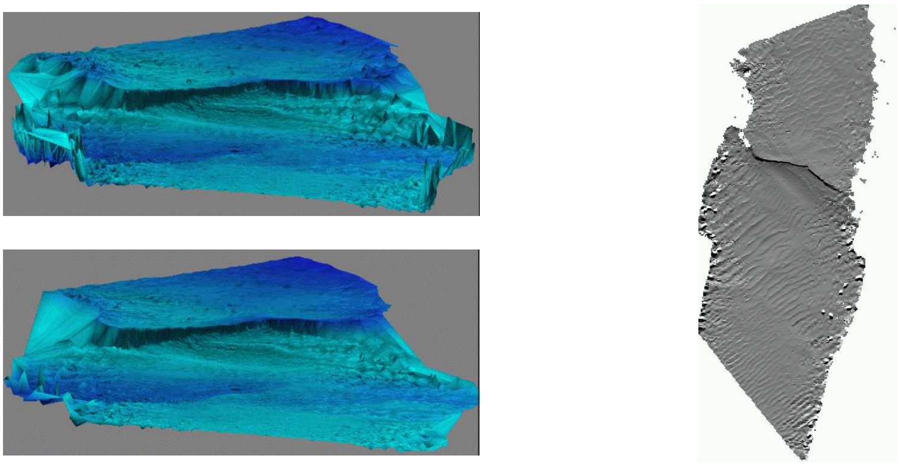

Figure 6: Data set 2 after processing with the 1D, top left, the 2D, bottom left and the robust approach.

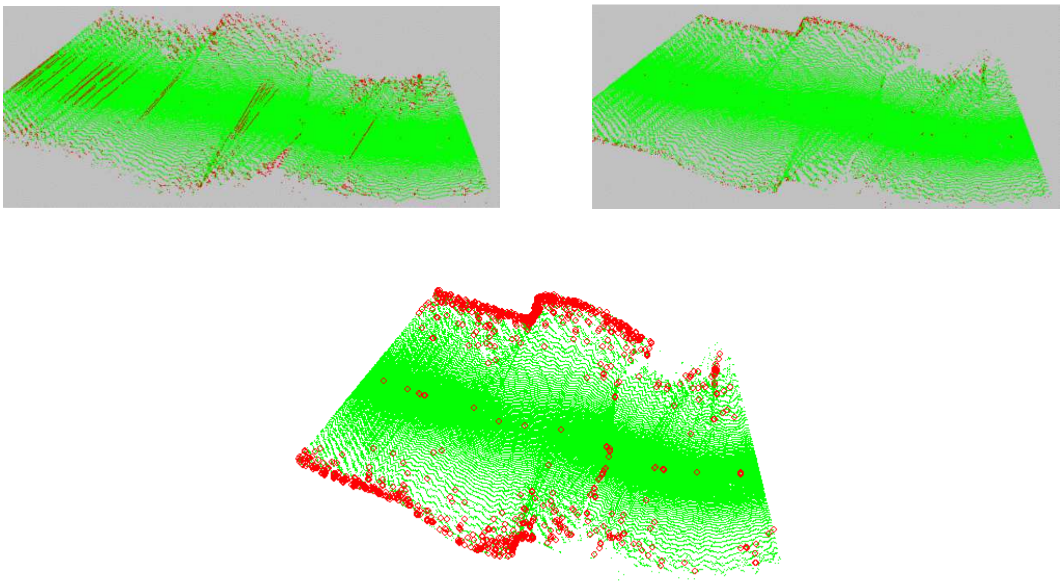

Figure 7: Data set 2: in red the data removed by the 1D, the 2D and the robust approach.

5 Conclusions.

Both the 1D and 2D method are forced to filter away good points by the 5% confidence level. If the confidence level drops, the number of Type II errors will however increase. Before applying these methods one should have a rough idea on the expected number of outliers.

In the 2D method, forced including of soundings of different pings is applied. But these included soundings do not get automatically a relevant interpolation weight: for the first two data sets, the ratio of ping width versus beam width is 1 : 3. Therefore the 2D method is to some extent still a '1D method'.

Both the 2D method and the robust method can be very time inefficient if implemented without consideration. The parameters of the covariance function should preferably be determined in a heuristic way while the number of soundings processed at once should not become too big. The number of soundings included in the Kriging system should be strongly limited while in selecting these small number of neighbors one could try to use the structure of the data file.

Finally it would be interesting to test these method on sea floors containing periodic bed formations like sand waves and mega ripples. Depending on the size of these structures compared to the data resolution, these methods could run into problems.

Acknowledgments.

The UT Delft Research Theme Earth is thanked for financial support enabling the useful visit of Christian Briese to the Delft University of Technology. Fugro Intersite BV and namely Cees de Jong are thanked for raising the problem, hosting Natasha Hennis and for the data sets.

References

[1] P. Bottelier, R. Haagmans, and N. Kinneging. Fast reduction of high density multibeam echosounder data for near real-time applications. *The Hydrographic Journal*, 98:23–28, 2000.

[2] J.-P. Chilès and P. Delfiner. *Geostatistics: modeling spatial uncertainty.* Wiley Series in Probability and Statistics. John Wiley & Sons, New York, 1999.

[3] C. D. de Jong, G. Lachapelle, S. Skone, and I. A. Elema. *Hydrography.* Delft University Press, Delft, 2002.

[4] P. Kitanidis. *Introduction to Geostatistics.* Cambridge University Press, Cambridge, 1997.

[5] K. Kraus. *Photogrammetry - Advanced Methods and Applications*, volume 2. Dümmler Verlag, Bonn, 1997.

[6] K. Kraus and N. Pfeifer. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53:193–203, 1998.

[7] N. Pfeifer, P. Stadler, and C. Briese. Derivation of digital terrain models in the SCOP++ environment. *OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Digital Elevation Models*, 2001.

[8] P. J. G. Teunissen. *Testing theory; an introduction.* Delft University Press, Delft, 2000.