SACLANTCEN REPORT
serial no: SR-285

SACLANT UNDERSEA RESEARCH CENTRE REPORT

AN APPROACH TO ROBUST MAP GENERATION FROM MULTIBEAM BATHYMETRIC DATA

G. Canepa, O. Bergem

October 1997

The SACLANT Undersea Research Centre provides the Supreme Allied Commander Atlantic (SACLANT) with scientific and technical assistance under the terms of its NATO charter, which entered into force on 1 February 1963. Without prejudice to this main task – and under the policy direction of SACLANT – the Centre also renders scientific and technical assistance to the individual NATO nations.

SACLANT Undersea Research Centre
Viale San Bartolomeo 400
19138 San Bartolomeo (SP), Italy

tel: +39-187-540.111
fax: +39-187-524.600

e-mail: library@saclantc.nato.int

NORTH ATLANTIC TREATY ORGANIZATION

SACLANTCEN SR-285

An approach to robust map generation from multibeam bathymetric data

Gaetano Canepa, Oddbjørn Bergem

The content of this document pertains to work performed under Project 033-2 of the SACLANTCEN Programme of Work. The document has been approved for release by The Director, SACLANTCEN.

Jan L. Spoelstra
Director

intentionally blank page

An approach to robust map generation from multibeam bathymetric data

Gaetano Canepa, Oddbjørn Bergem

Executive Summary: High frequency, shallow water swath bathymetry systems have significant potential in mine countermeasures. The method described represents a necessary first step towards the realization of a digital terrain model.

During the last twenty years, many multibeam bathymetric sonars have been produced. The instrumentation is usually accompanied by a system able to produce a seafloor map from the sonar data. There are also several public domain systems, which can be used to obtain a map from the data. All these systems produce a gridded map that must be filtered in order to reproduce the original seafloor surface because of noise on the bathymetric data. Both the gridding and the filtering algorithms introduce a source of error that is not easily controlled. Moreover, gridded maps may use significant storage space for a small amount of information. Finally, at present no systematic solution with realistic run-time requirements has been given to the problem of identification and elimination of bad data (outliers).

We present here an algorithm able to fit bathymetric data and to automatically deal with outliers. The most important characteristics of the algorithm are: production of a triangulated map of uniform accuracy irrespective of seafloor features; a map resolution which depends on the local data noise amplitude; automatic elimination of outliers and low computing cost even on large data files.

The algorithm can be used to reduce the operator intervention during bathymetric data mapping. Raw bathymetric data are directly analyzed by the algorithm which automatically and robustly eliminates outliers and produces a map the parameters of which can be finely tuned by the user (number of nodes, smoothing, level of data cleaning, *etc.*). The algorithm has been implemented, exhaustively tested on synthetic and real data and fully documented.

intentionally blank page

An approach to robust map generation from multibeam bathymetric data

Gaetano Canepa, Oddbjørn Bergem

Abstract: During the last twenty years, many multibeam bathymetric sonars have been produced. The instrumentation is usually accompanied by a system able to produce a seafloor map from the sonar data. There are also several public domain systems, which can be used to obtain a map from the data. All these systems produce a gridded map that must be filtered in order to reproduce the original seafloor surface because of noise on the bathymetric data. Both the gridding and the filtering algorithms introduce a source of error that is not easily controlled. Moreover, gridded maps may use significant storage space for a small amount of information. Finally, at present no systematic solution with realistic run-time requirements has been given to the problem of identification and elimination of bad data (outliers).

We present here an algorithm able to fit bathymetric data and to automatically deal with outliers. The most important characteristics of the algorithm are: production of a triangulated map of uniform accuracy irrespective of seafloor features; a map resolution which depends on the local data noise amplitude; automatic elimination of outliers and low computing cost even on large data files.

The algorithm can be used to reduce the operator intervention during bathymetric data mapping. Raw bathymetric data are directly analyzed by the algorithm which automatically and robustly eliminates outliers and produces a map the parameters of which can be finely tuned by the user (number of nodes, smoothing, level of data cleaning, *etc.*). The algorithm has been implemented, exhaustively tested on synthetic and real data and fully documented.

Keywords: seafloor map, map fitting, scattered data fitting

Contents

List of Figures

List of Tables

Introduction

Background

The purpose of the algorithm presented here is to create an accurate map from multibeam bathymetric data. The same algorithm can also be applied to other fields when a high number of measurements is present which contain corrupted samples in the data. In the geological field, for example, it could be applied to petroleum exploration (map of layers of sandstone, shale, limestone, *etc.*), and geological maps (Schumaker 1976).

Data for seabed mapping are mainly obtained using a multibeam echo sounder. These data are usually corrected for sound speed variation in the water column, tide, *etc.* The result fits in a table of three columns containing latitude, longitude and depth of a set of seafloor points. These data are affected by a certain amount of nearly white noise (due to uncertainty in the determination of the seafloor depth). This type of noise can be optimally treated using a Least Squares algorithm, as will be seen in Section 3. In the same Section it will also be shown that a normal low-pass filter is not adequate for the seafloor mapping problem, because it do not preserve the vertical and horizontal dimensions of the seafloor features.

Moreover, some points may be corrupted because an anomalous bottom detection was performed. In such a case, the datum or the set of data must be rejected either manually or using an adequate rejection algorithm.

Finally, the mapping problem is complicated because of the amount of data: it may consist of more than one million points and any available mapping algorithm is unrealistically time consuming on such a data set.

Problem overview

The problem of building a map can be stated in a general form as follows:

Definition 1 *Let D be a domain in the R^2, and suppose F is a real-valued function defined on D. Suppose the values $F_i = F(x_i, y_i)$ are given, with a certain error, in a set of points $P = \{p_i = (x_i, y_i) \in D \; i = 1, \cdots, N\}$. Find a function f defined on D that reasonably approximates F.*

When noise affects data the map cannot be simply an interpolation of the data points: data fitting is required.

During or before the data fitting it is necessary to perform an outlier elimination step that consists of the elimination of the points that have a low probability to be "correct". These may be generated during the acquisition phase (Du 1995, Du *et al.* 1996).

Once the outliers are rejected, the scattered noisy data must be analyzed in order to obtain a map. The currently available software tools (for example GMT-System, Appendix A) are only able to triangulate, contour, and plot all data without fitting, or (for example MB System, Appendix A) are only able to interpolate data on a regular grid basis. As a consequence, the dimensions of the map for a large area can be too high for practical purposes. Other packages (such as TRISMUS, Appendix A) are able to work on bathymetric data, integrating both the MB-System and GMT-System functionality, but still work on regularly gridded maps. Siscat (Appendix A), can be used to model data using maps that optimize storage space. However, it works on a pre-defined "correctly" gridded map or on scattered data that are supposed to be exact. More information on other software toolkits for mapping of scattered data can be found in Appendix A and in (Mayer *et al.* 1997, Tyce *et al.* 1997).

Outlier elimination

Two approaches exist in bathymetric data mapping to detect outliers: interactive outlier elimination, by means of a computer aided graphical tool, or software automatic elimination. Most commercial software uses a hybrid approach consisting of a software detection of "possible outliers" followed by an interactive session to confirm the elimination (Ware *et al.* 1992). These "outlier elimination" sessions are boring and time consuming, but also introduce a variable that is not always negligible: the researcher bias. Within the software for automatic outlier detection different approaches exist:

- Ware *et al.* (1990) divide the data set in cells: for each cell they estimate 4 statistical values; then, they use these values to classify the data as valid or outliers. This algorithm can work efficiently only if the number of bathymetric data points is three times higher than usual (Ware *et al.* 1990).
- Guenther and Green (1982), Grim (1988), and Wells *et al.* (1989) proposed a method for outlier elimination (COP) based on the comparison of the data set with the nearest neighbours. They reported that a certain number of outliers were still present in the data after the algorithm was applied to the data. Greenburg (1987) proposed a method that worked better but which is unrealistically time consuming.

- Du *et al.* (Du 1995, Du *et al.* 1996) proposed a method based on the statistical analysis of the data in a working window, the dimension of which starts from the whole data set and shrinks or increases during data analysis. The outlier elimination on the working windows is based on data clustering given the assumption that the noise in the data follow the Uniform distribution.

Of the above methods, only the last could be reasonably considered for automatic outlier detection because the data analysis can be realized in realistic time even though the algorithm elaborated by Du (1997) "may be far away from practical applications".

Our algorithm

The algorithm presented here, named **NR-COMPRESS**, attempts to address two problems that are not solved in the above cited packages: outlier elimination and storage space minimization. The characteristics of this algorithm are:

- it produces a map that optimizes the storage space;
- it is able to robustly deal with outliers;
- it can be used to interpolate data produced by any echosounder that can produce a simple data set consisting of coordinates and depths;
- it is reasonably fast even when extremely large data files must be mapped.

The fitting (mapping) algorithm described here is based on a triangulated grid: as it is shown in the following sections, such a choice minimizes the number of nodes necessary to describe the map and minimize the loss of information due to the gridding algorithm. The map that is produced is more dense where the second derivatives of the seafloor are higher and the resolution depends on the noise level on the data.

The basic idea behind the outlier detection method described here is derived from the Du *et al.* method. The most important change being that the outlier elimination phase (OEP) is carried out during the mapping phase. The OEP consists of choosing a cell the *center of which* could be a node of the seafloor map. For the data points included in the cell, a first robust outlier elimination phase, only used to eliminate far outliers, is carried out. Then, a second phase of *quasi*-robust outlier elimination is carried out, based on a statistically robust fitting criterion. The complete method is robust and efficient in the sense that it is not sensible to the presence of an even great number of outliers, and that it eliminates only a minimum amount of outliers

from the data, without appreciably changing the variance of the noise of the cleaned data.

Organization of the report

The report is organized as follow: in the first section a description of the bathymetric data sets is given; in the second section a comparison between triangulated and uniformly gridded map is given; in the third section the basic concepts of the fitting algorithm are given; in the fourth section the outlier elimination algorithm is described; in the fifth section the results of the algorithm on a series of interesting cases are presented and discussed. Some conclusions follow.

1

Data characteristics

The data characteristics are fundamental for the choice of a fitting algorithm. Bathymetric data have a series of characteristics which render harder the definition of the fitting (mapping) function.

The first of these characteristics is that the number of (x,y,z) data points in a bathymetric set may be higher than 1 million. This high number of points implies that each calculation of the cost function, or data selection on the complete data set is very time consuming.

The second characteristic of the bathymetric data is that they are not gridded (see Figure 1). This fact implies that the data cannot be placed on a regular grid without a loss of information (see subsection 3.2). Again, data that are not regularly gridded, cannot be scanned using fast algorithms.

The last, very important characteristic of bathymetric data is that they are noisy. The noise is made of two components: a nearly white component (due to error in the determination of the depth and to the knowledge of the position that is not perfect), and a noteworthy amount of outliers due to multiple paths, and reflection from fish shoals, sub-bottom layers, or abnormal water columns (Du *et al.* 1996, Du 1995). The nearly white component can vary across the data set due to depth changes or to changes in the characteristic of the seafloor (presence of sea grass, change of bottom type, *etc.*). A fitting algorithm should use more nodes in low-noise areas (where a higher accuracy can be obtained because the noise variance does not hide the seafloor features) and fewer nodes in high-noise areas.

The chronic presence of outliers requires two possible strategies: to pre-filter the data for outlier elimination or to implement a robust fitting algorithm (possibly with outlier elimination). The outlier elimination algorithm can be much more robust and efficient if it is realized taking into account the seafloor shape. As a consequence, the outlier elimination algorithm is implemented here in connection to the seafloor fitting algorithm (see Subsection 4).

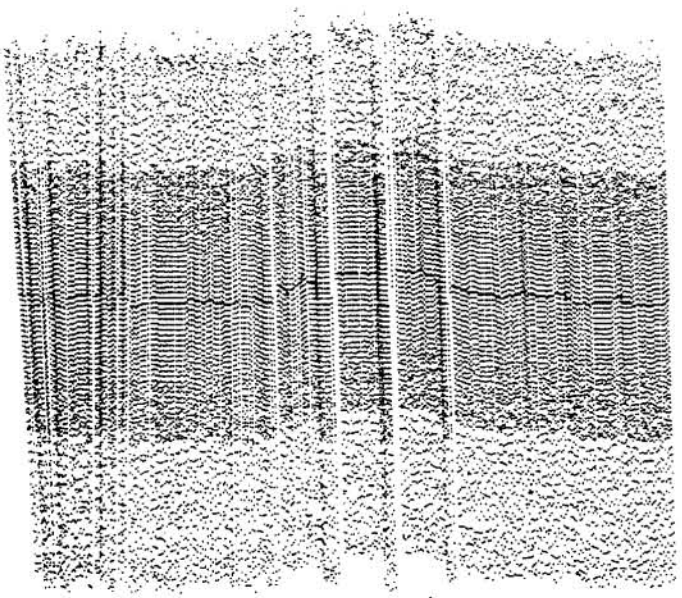


Figure 1 *Typical spatial distribution of bathymetric data acquired using an Atlas® Hydrosweep MD™ multibeam sonar (80 beams) from the NATO Research Vessel (NRV) Alliance. The data were acquired while the ship was travelling from East to West for the upper track and from West to East for the lower track. A large over-lap (the darker zone) is visible between the two tracks.*

1.1 An example of mapping

In this subsection an example will be given of the application to bathymetric data of three standard filtering techniques:

- The data are gridded using the mean depth (see Fig 2.**a**): the gridding introduce a first level of smoothing. This is a plot of a grid made of 51×51 pixels (2601 nodes). This technique is not robust (it is sensitive to outliers) and produces a result that lacks accuracy.
- The gridded data are then filtered along the North-South direction using the central moving average of order 3. Then, the resulting surface is filtered again along the East-West direction using the same filter (see Fig 2.**b**). The result is obtained using the same grid as before.
- The gridded data are filtered (in the two direction) using the central moving average of order 5 (see Fig 2.**c**).

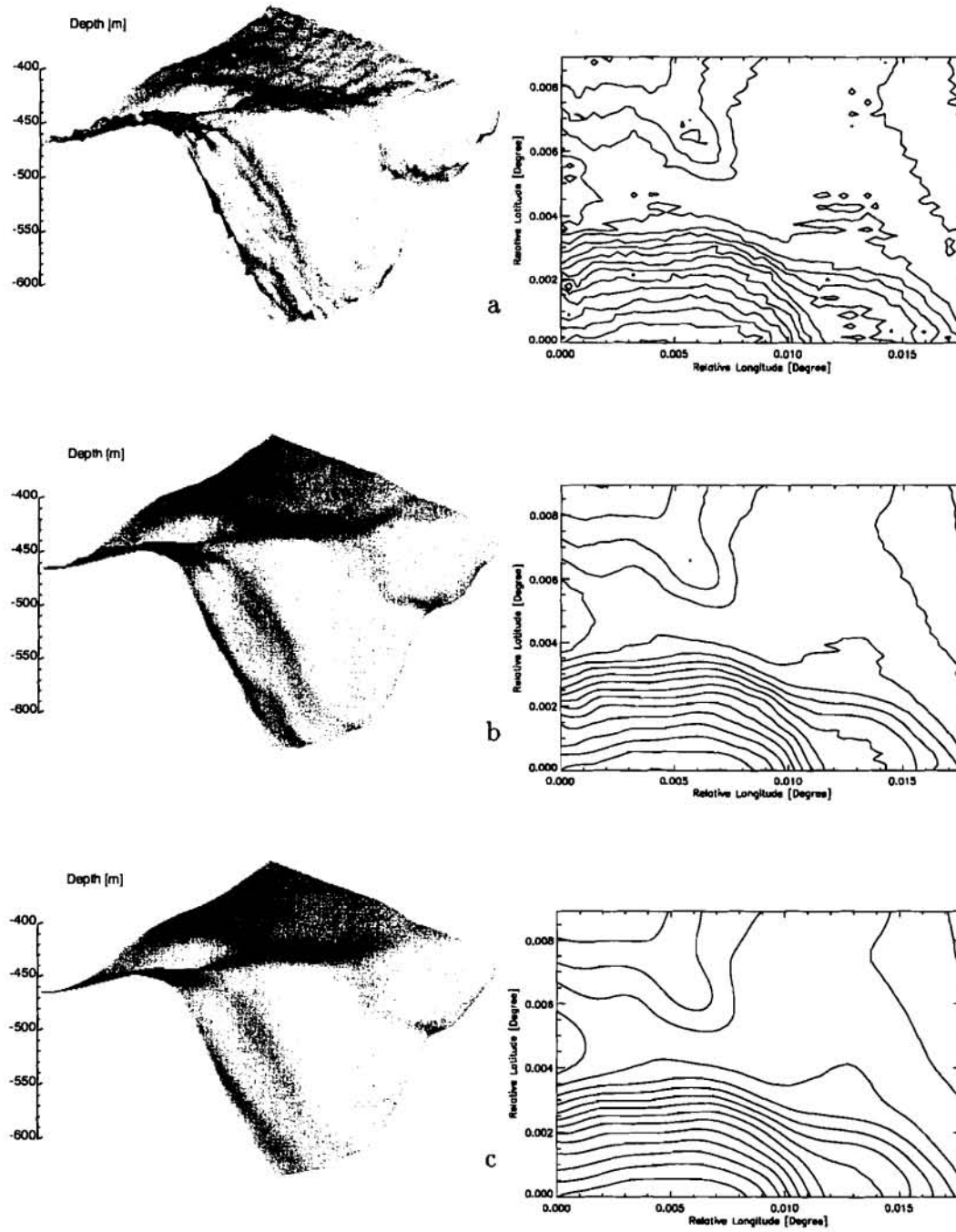


Figure 2 *Example of at sea data analysis from an area off Sestri Levante acquired using a Atlas® Hydrosweep MD™ multibeam sonar (80 beams) from NRV Alliance.* **a** *shows the results of applying a mean filter, while* **b** *and* **c** *show the application of a central moving average filter with decreasing low-pass spatial frequency.*

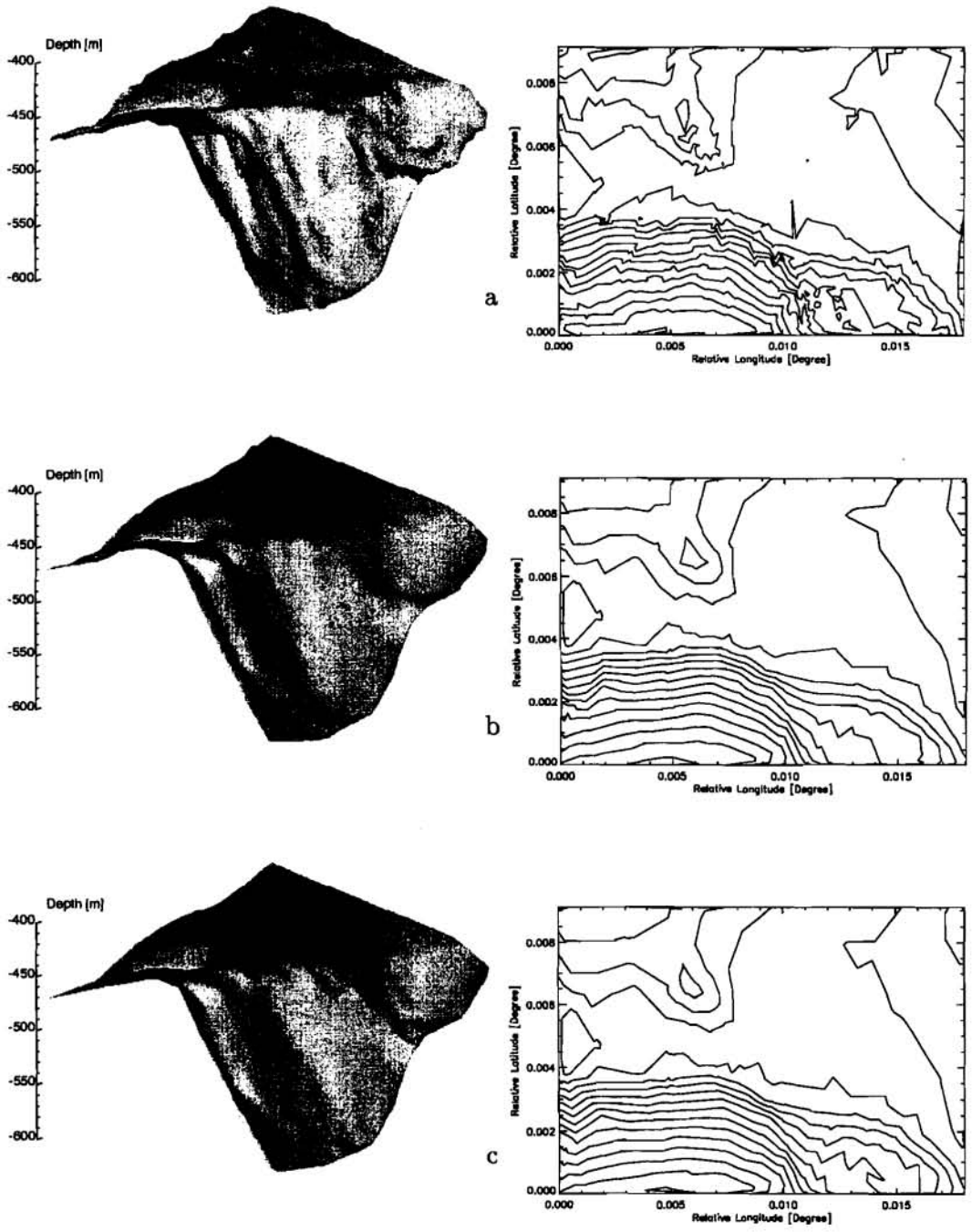


Figure 3 *The three graphs show the result of the application of the* NR-COMPRESS *algorithm to the data from Sestri Levante area.* **a** *shows the application of the algorithm using a small number of points for the approximation and a low number of nodes for the map;* **b** *and* **c** *show two map calculated using the same smoothing parameter but using more* **b** *and less* **c** *nodes for the map (see Section 5 for a better explanation.)*

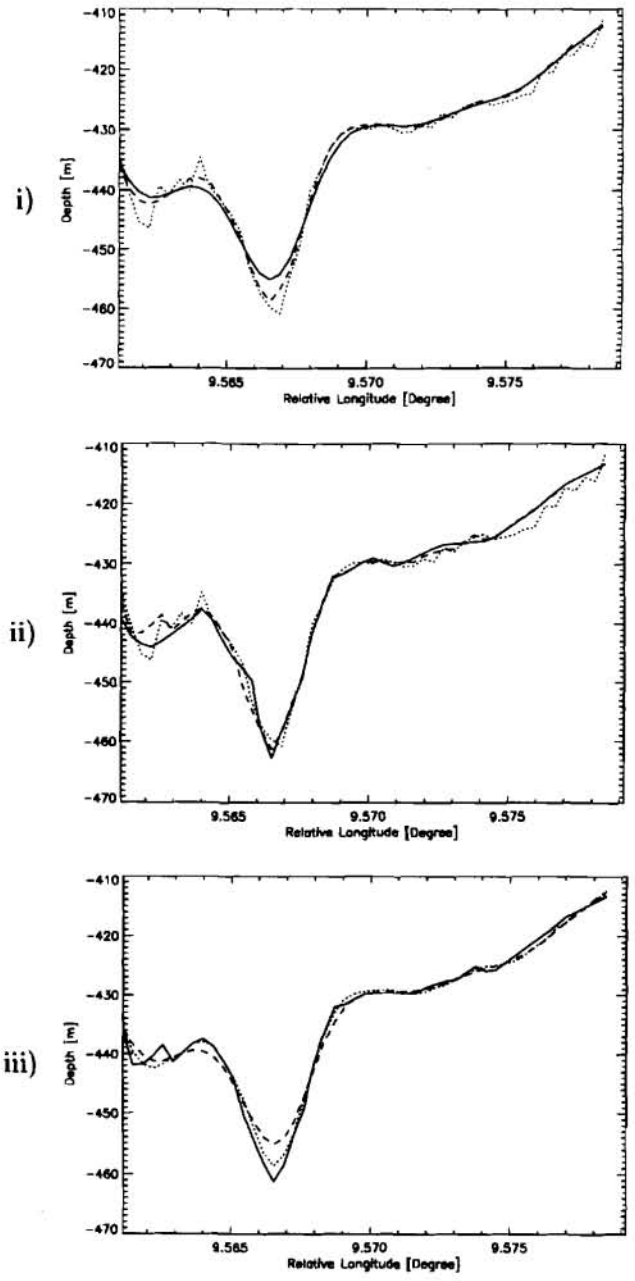


Figure 4 *Plot i) shows the results of a cut at constant latitude of the three surfaces shown in Fig 2: the dotted trace comes from surface* **a**, *the dashed trace from surface* **b**, *and the continuous trace from surface* **c**. *Plot ii) are the traces from the three surfaces of Fig. 3: the dotted trace comes from surface* **a**, *the dashed trace from surface* **b**, *and the continuous trace from surface* **c**. *Plot iii) shows the traces from surface 2.***b** *(dotted trace), from surface 2.***c** *(dashed trace), and from surface 3.***b** *(continuous trace)*

Figure 3 shows the results of applying the NR-COMPRESS algorithm to the same data, with an increasing low-pass effect. The results are shown calculating the value of the triangulated map on the usual grid. From these figures it seems that the results of the two fitting methods are similar, but a more detailed look shows that there are differences (Fig. 4). Here, the surfaces shown in Figs. 2 and 3 have been cut at a fixed relative latitude (0.00661°) and the resulting traces are shown. Plot 4.i, shows the three traces of the surfaces of Fig. 2. Plot 4.ii shows the traces to the surfaces of Fig. 3. Plot 4.iii shows the traces to the surface in 2.**b**, in 2.**c**, and in Fig. 3.**b**. It is now clearly visible, that the effect of the filtering procedure is to decrease the seafloor local depth variations and to enlarge the seafloor features.

It must be pointed out that the NR-COMPRESS algorithm is not sensitive to outlier but the mean and moving average filter is very sensitive. The filtering characteristics of NR-COMPRESS are similar to the moving average filter with a relatively low cut-off spatial frequency. It will be shown in Section 5, that the NR-COMPRESS algorithm produces accurate results also in the presence of discontinuity and in the presence of narrow seafloor features, while the application of the mean and central average filter to these causes an enlargement of the features and a reduction of their height (see Subsection 3.2).

The number of nodes of the maps produced by NR-COMPRESS (Fig. 3) is 454 for map **a**, 281 for map **b**, and 230 for map **c**, while the regularly gridded maps consist of 2,601 nodes (Fig. 3). The data set and the analysis shown in Fig 3 are explained more extensively in Subsection 5.4.

2

Data mapping: gridded *versus* triangulated maps

2.1 Introduction

An important aspect of the fitting algorithm is the decision as to the kind of map we intend to realize: regularly gridded or triangulated map.

The question is related to map storage space and data information content conservation. The bathymetric data are not gridded: as a consequence any data gridding algorithm implies a data pre-elaboration that can destroy some data information (Subsection 3.2). When a regular grid is considered, the spatial gridding interval is an important parameter that must be carefully taken into account: if the grid is too dense, storage becomes an important factor; on the other side, some maps (*e.g.* map of flat region) can be described with a very coarse grid. While most of the sea maps are of flat sites, the most important maps are those of transition zones, in which a flat region abruptly meets an underwater sink or canyon. In such a case, a regular grid map shows all its limits (Fremming *et al.* 1997).

A triangulated map can be built without loss of information and can be realized in such a way that regions where canyons or sinks are present are described by a higher number of points. This objective can be obtained using a number of nodes significantly lower than the one necessary for a gridded map. For these reasons, the mapping algorithm described here realizes a triangulated map.

2.2 Formal definitions

A triangulation T is defined by the following (Dyn *et al.* 1990):

Definition 2 *Let $\Omega \supset V$ be a region with a polygonal boundary $\partial\Omega$ with all vertices in V. A set T of non degenerate open triangles T_i is a triangulation of Ω if the following conditions hold:*

1. *V is the set of all vertices of triangles in T;*
2. *Every edge of a triangle in T contains only two points from V, namely its endpoints;*

3. $\bar{\Omega} = \bigcup_{i=1}^{t} \bar{T}_i$ *(t is the number of triangles in T);*

4. $T_i \cap T_j = \emptyset$ *for all* $i \neq j$;

In other words, a triangulation consists of dividing a given plane with triangles the vertexes of which are in a set of points. A particular kind of triangulation is suitable for seafloor mapping: the Delaunay triangulation. It must be built taking into account the local variations of the data. Where the seafloor changes are more important (where the surface second derivatives are higher), the number of triangles is higher, in order to obtain a better description of the seafloor. A Delaunay triangulation is defined by the following:

Definition 3 *For each triangle* $T_i \in T$ *a value* σ_i*, which is the minimum of the three interior angles of* T_i*, is assigned. The vector* N_T *is a vector of length t containing the* σ_i *values. Furthermore, suppose that* N_T *is ordered in a non-decreasing manner. The Delaunay criterion (or maxmin angle criterion) imposes the following ordering of triangulation:* $T' < T$ *means that* $N_{T'}$ *is lexicographically larger than* N_T*. A Delaunay triangulation is the maximum T according to the Delaunay criterion.*

This definition means that a Delaunay triangulation is a triangulation that has the triangles with the largest internal angle (ideally equilateral).

2.3 Examples

Figure 5.**a** shows an example of a possible seafloor $\left(f(x,y) = \frac{\tanh(9x-9y)+1}{2}\right)$. If this seafloor is mapped using a 10×10 regular grid (100 nodes) (Fig. 5.b) the maximum error ($\varepsilon_{\max}$) between the map and the real surface is 1.72 m, the mean of the absolute value of the errors ($\overline{\varepsilon}$) is 0.096 m, and the standard deviation (σ) is 0.172 m. If a triangulated map is used, (Fig. 5.c and 5.d), only 16 nodes are necessary to obtain a smaller $\varepsilon_{\max}$ (1.65 m) even if $\overline{\varepsilon}$ and σ are higher (0.276 m and 0.401 m, respectively). Incrementing the number of nodes of the map the mean error decreases: Figure 5.e and 5.f show the seafloor surface described using a triangulation of 46 points: in this case $\varepsilon_{\max} = 0.618$ m, $\overline{\varepsilon} = 0.093$ m, and $\sigma = 0.124$ m.

Figure 6 shows a map realized using a fitting algorithm that follows all the recommendation described here. At the top, the rendered sea surface is shown. At the bottom, the triangulation used to describe the map is shown: of interest is the difference in resolution between zones that are approximatively flat and the edge regions.

Some remark to take into consideration:

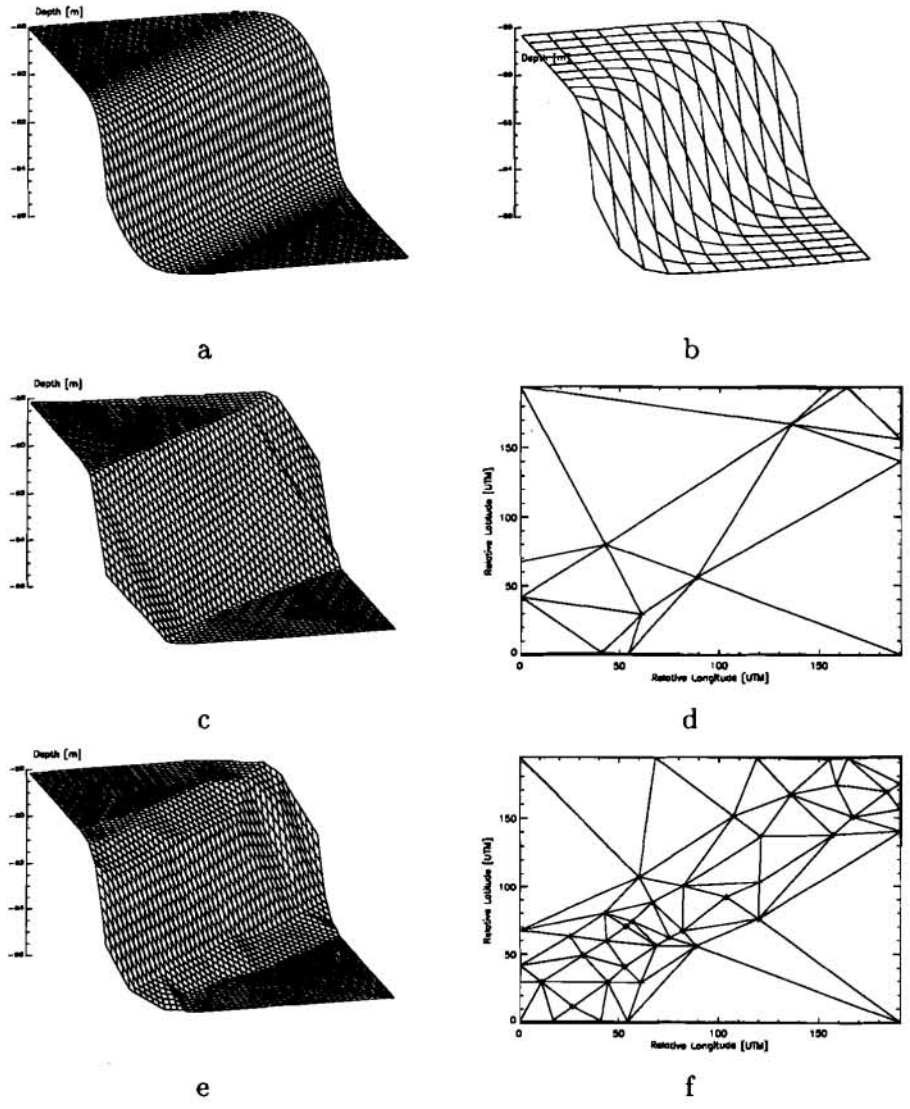


Figure 5 *An example of a how a triangulated map improves storage efficiency.*

- The choice of a triangulated map implies that some of the most popular fitting and filtering algorithms cannot be used, because they are only defined for regularly gridded data points.

- An advantage of a gridded map is that the access to the map data is fast. The triangulation algorithm must include a localization algorithm able to localize a point in the map in a realistic (short) time interval.

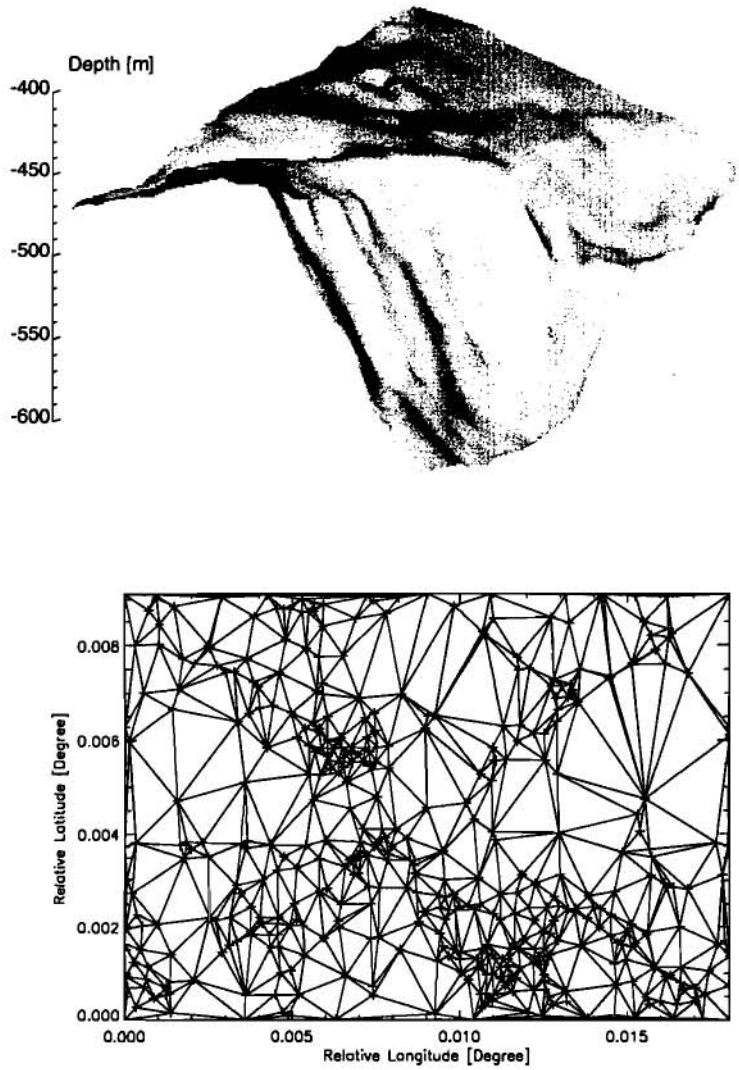


Figure 6 *An example of a map from an area off Sestri Levante: at the top, the rendered sea surface, at the bottom, the triangulation used to describe the map. Of interest is the difference in resolution between zones that are nearly flat and the edge regions.*

2.4 Data driven triangulation

In Subsection 3.2 we point out that, to obtain a map that optimizes the storage space, it is necessary to avoid the use of regular gridded maps. A triangulated map is able to optimize the information content if some condition are verified. If, for example, the map has a higher resolution where the second derivatives are higher

and lower elsewhere (see Fig. 5 and 6), the triangulated map will produce a result that, using the same number of points, is more informative than a regular grid map (Dyn *et al.* 1990, Rippa 1992). The criteria for a more informative triangulation are based on the characteristics of the data and are called: *data driven triangulation*. For example, in (Rippa 1992) the approximation of a surface is described, based on the triangulation on a subset of scattered data; the author defines an algorithm called COMPRESS that can be simplified to the following basic scheme:

1. Let Ω be a region with a polygonal boundary $\partial\Omega$ consisting of M vertices from V, where $V = \{v_i = (x_i, y_i) \in R^2, i = 1, \cdots, N\}$ are the point where the values of the function $F(x_i, y_i) = F_i$ are known. Be $V^{(M)}$ the set of M vertices of ∂M. Construct an initial $V^{(M)}$ triangulation T of Ω.
2. Construct $f_{T,V}$, the approximating surface, and compute the errors $E_i(f_{T,V}) = |F_i - f_{T,V}(x_i, y_i)|$ and let $E_{f_{T,V}} = \max_{1 \le i \le N} E_i$.
3. If $E_{f_{T,V}} \le \epsilon$, end the procedure else go to the next step.
4. Select a point $v_k = (x_k, y_k) \in V \backslash V^{(M)}$ for which $E_k(f_{T,V})$ is maximal and add it to $V^{(M)}$: $V^{(M+1)} = V^{(M)} \cup \{v_k\}$.
5. Update the triangulation T to include the new point, set $M = M + 1$ and go to step 2.

This procedure chooses the nodes of the triangulation using the available data: it is therefore a *data driven triangulation*. This procedure is able to produce a good map using a smaller number of nodes than a regular grid map (see Fig. 5). For example, when V is the regular gridded map, the number of points used to define T is always much smaller in the case of a reasonably smooth function. This method could be applied to a regular grid map generated by bathymetric data using programs such as MB-System. This method is fast but cannot deal with the outliers: in fact, MB-system will not generate a good map if outliers are present. Moreover, information is lost during the gridding phase. The COMPRESS scheme, however, cannot be applied directly to data acquired at sea because of the presence of noise (in the COMPRESS algorithm F_i are supposed to be error free samples). A modification of the COMPRESS algorithm is therefore proposed (Noise Resistant COMPRESS, `NR-COMPRESS`) to enable it to deal with outliers and noise (see next Sections).

3

Mapping multibeam bathymetric data

3.1 Introduction

Scattered noisy data fitting in $D \subset R^2$ consists of constructing a function (which will be called map in the following) $f = f(x, y)$ such that, given $E(i, f) = |f(x_i, y_i) - F_i|$, it minimizes a given cost function $C(E(i, f))$ for $i = 1, \cdots, N$. In the preceding definition F_i is the noisy value associated with the coordinate (x_i, y_i), and $P = \{p_i = (x_i, y_i) \in D,\ i = 1, \cdots, N\}$, is the set of noisy data coordinates. An important class of schemes for noisy data fitting is based on a regular grid of the map: here, an effort is made to compress the final map storage space and emphasis is placed on fitting with a triangulation scheme. This means finding a set of points $V \in D$, define on those nodes a triangulation T and define on the triangulation a map $f_{T,V}(x, y)$.

Two important points to take into account are the node choice and the stopping criteria of the fitting algorithm: many criteria are available depending on the particular application. One of the criteria, that fits well with the seafloor data mapping, depends on the local data noise (LDNC): the triangulation produced using this criterion is more dense where the data noise is lower and coarse where the noise is higher. The triangulation produced using the LDNC is also more dense where the second derivatives of the seafloor are high, that is where the surface "changes more."

Global fitting of the seafloor surface can be very time consuming. For each step of the optimization, the error between the data points and the actual approximating surface must be computed. Considering that the number of data points easily exceeds 1 million, it is easy to understand that the cost function can be excessively time consuming. It will be more efficient to divide the problem into a number of smaller problems in which the global solution can be efficiently found and then assemble the local solutions to generate a global one. Considering that the approximating surface described here is based on triangulation, the local solution can be found on triangular sub-domains: the union of such subdomain generates the global map.

There are a number of possible ways to construct a map. Four choices have to be made:

1. The analytical expression of the approximating surface: for example, a piecewise linear approximating function, a radial basis function, a cubic or quintic

spline function, *etc.*

2. The choice of global *versus* local algorithm.
3. The cost function (error criterion) to optimize to find the best among the infinite possible approximating surfaces.
4. The algorithm used to optimize the cost function.

These four choices are related and must also take into account computer memory space and computing speed.

Most fitting algorithms (for example Magestic, Appendix A) work on functions of a single variable. When they can work on function defined on a R^2 domain, they usually can deal with a limited number of parameters (<100), using a Least Squares method, extremely sensitive to outliers. A problem of normal data fitting software is that the calculation time of the error function can easily overwhelm any normal fitting algorithm based on a nonlinear optimization algorithm. As a consequence, even good software solutions, such as the multidimensional data fitting algorithm based on Radial Basis Function implemented in RBFpack (Appendix A), cannot work on the global seafloor fitting problem (the complete map).

Analytical expression of the approximating surface

The fitting of the seafloor must have realistic time and memory requirements. In some applications it is necessary to prepare a printout of the map in real-time. As a consequence of this consideration, a piecewise linear approximating surface is chosen here: the simplest C^0 (continuous) function. In fact, any other fitting function will increase the computational time of the fitting algorithm.

Global versus *local algorithm*

The number of data points is very high: as a consequence, the calculation of a global cost function is excessively time consuming. Moreover, the calculation time of the global cost function increases as the number of nodes of the map increases. Finally, the global optimization algorithm will require more time when the number of nodes of the map increases (*e.g.* the optimization of a function of 500 variables). Some tests performed using global fitting algorithms confirmed that they are excessively time consuming (Subsection 5.5). As a consequence, the possibility of using a global fitting algorithm must be rejected. Once the global fitting algorithm choice is discarded, it is necessary to determine the characteristics of the local fitting algorithm.

It must use only a part of the total data points for the surface fitting and must be, at least, C^0 extendable to the global map.

To obtain a fitting algorithm that uses only a part of the bathymetric data is simple: the complete map region is divided into a certain number of smaller triangular sub-maps, each with a number of points that can be dealt with in a reasonable computing time. Then, the mapping algorithm is applied to each sub-map. The algorithm for the division of the map into such smaller domains is as follows:

1. The algorithm starts making a triangulation of the vertex which are the points that delimit the map region (contour points, Subsection 3.3).
2. A test is performed to check if a triangle exists, in the triangulation, where more than a given number of data points (n_p) is present. If not, the map sub-division is complete.
3. A point must be added to the triangulation in the middle of the longest side of the triangle in which the highest number of data points lies. A new triangulation is performed on the new points set. Then, the algorithm restarts from point 2.

The local fitting algorithm must also be C^0 extendable to the global map: this is possible only if the values at the boundary of two adjacent sub-maps are the same (Figure 7 shows an example of local fitting function that cannot be C^0 extended to global fitting).

3.2 Surface fitting: smoothing

The realization of a seafloor map is connected with the problem of filtering 2D scattered data. In particular, it is interesting to study the smoothing properties that are required to produce a map in the case of a seafloor. We are interested in finding a technique able to produce a triangulated smooth surface using *all* the available data information and including as much as possible of the data information in the resulting smooth map. It is also necessary to define filter parameters which determine the information the smoothed map must contain.

The first important characteristic of bathymetric data filtering is that the usual filter algorithms cannot be easily applied because they are not gridded and, as already remarked, gridding bathymetric data will always cause information loss. Moreover, even if a filter is able to analyze scattered data (for example the median and the Gaussian algorithm of the `mbsmooth` routine in the MB-System package, Subsection 1.1), it usually produces a gridded result or a value for each data point: the

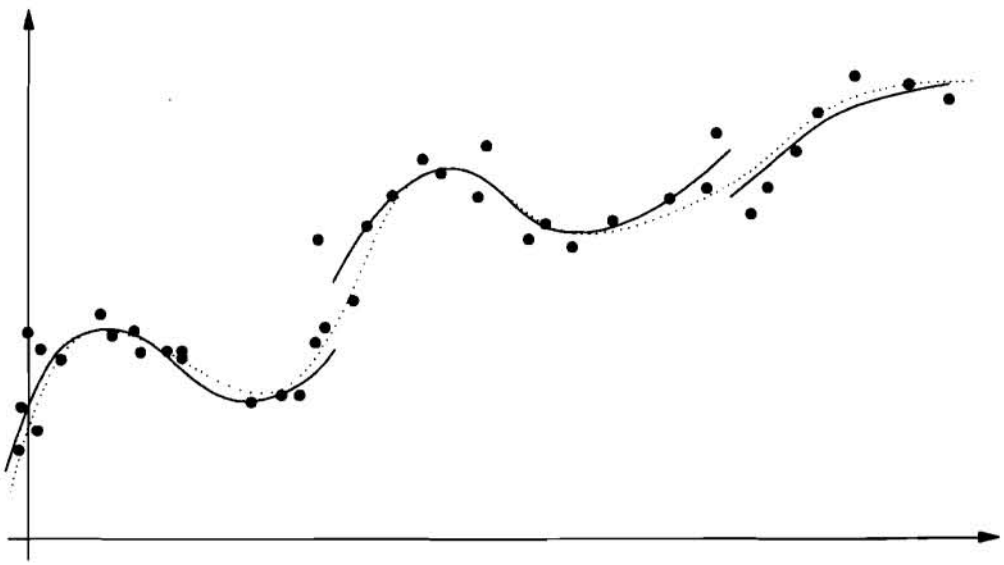


Figure 7 *A 2D example of local fitting functions that are not C^0 extendable to global fitting.*

last option produces too much information for the seafloor approximation. Finally, the analysis of each datum requires a computing time that is impractical.

The way in which a smoothing filter deals with seafloor features is an important characteristic to consider when a filter procedure for bathymetric data must be selected. The effect of normal low pass filters (which are usually used in smoothing of bathymetric data) is that of diminishing the height and enlarging the horizontal extension of the seafloor features (see Fig. 8). For example, a moving average filter actually always reduces the amplitude of a local maximum of the surface. This is also the case, for a simple gridding technique based on the average or the median of the points nearest to the grid node. Such effects are not advisable when mapping a seafloor: the edge position and the depth of each seafloor feature should be conserved by smoothing. Some filtering techniques exist that are able to maintain such characteristics although losing some smoothing power. They are low-pass filters, well-adapted for data smoothing, and termed variously least-squares, DISPO (Digital Smoothing Polynomial), and Savitzky-Golay filters (Press *et al.* 1992). A least-square filters is able to preserve height and edge position of functions with non-zero second derivative. The idea is to approximate the underlying function within the moving window not by a constant (the estimate of which is the average), but by a polynomial of higher order, typically quadratic or quartic. For each point $p_i = (x_i, y_i, z_i)$, the least-square polynomial fit $P_{p_i,N_f}(x, y)$ to the N_f points in the neighbourhood of p_i is performed and the smooth value (SV) of the function $z_i^* = P_{p_i,N_f}(x_i, y_i)$ is taken as the value of the polynomial fit at (x_i, y_i): $g_i = (x_i, y_i, z_i^*)$. Then, the filter is moved to the next point until all the smoothing function points are calculated (see Fig 9). This kind of filtering can be performed using fast algorithms on gridded

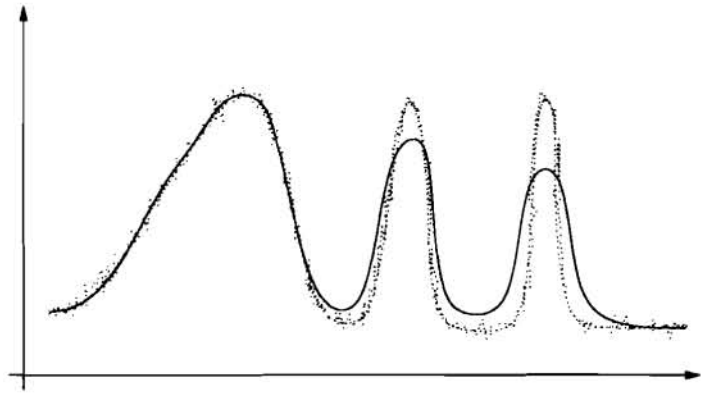


Figure 8 *An example of the result of the application of a* normal *low-pass filter to seafloor features.*

data, while it is very slow if they are not. As a consequence, it is not possible to filter all the bathymetric data using this powerful technique.

3.3 Surface fitting: the algorithm in one dimension

Before formally describing the seafloor fitting algorithm that takes into account preceding discussions, a less formal description of the algorithm, on a simple one dimensional case will be given.

The samples of the function of a single variable $f(x)$, shown in Figure 10.**a**, are given. The algorithms start determining the region where the map must be built: the boundary points x_0 and x_1 given by the user (Fig. 10.**a**). Two fits, using the N_f points with the x coordinate nearer to x_0 and x_1, are computed and the values of the fitting curves at the x_0 and x_1 points are taken as smoothed values (y_0 and y_1). The first approximation of the seafloor map is given by the linear function joining $p_0 = (x_0, y_0)$ to $p_1 = (x_1, y_1)$ (Fig. 10.**a**). Then, the sample point with the maximum error to the first approximation of the map is chosen, with coordinate x_2, a fitting function is built and its value at the abscissa x_2 is taken (y_2). Then the new seafloor map is taken as the linear function from p_0 to $p_2 = (x_2, y_2)$ and from p_2 to p_1 (Fig. 10.**b**). The approximation proceeds (Fig. 10.**c**) until the stop criterion is verified (Fig. 10.**d**).

3.4 The stop criterion

A very important point in the fitting algorithm is the stop criterion. For bathymetric data mapping the attention is focused on a set of four stop criteria useful in different

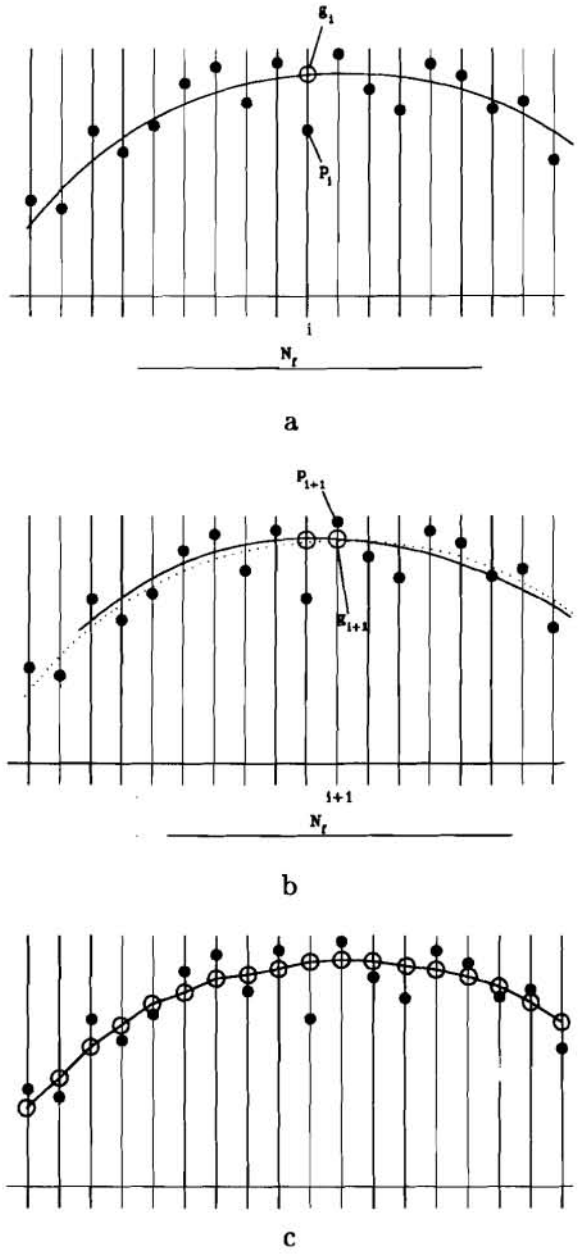


Figure 9 *An example of how a least squares filter works.* **a** *the point i is considered: the nearest N_f points are considered and a least-squares quadratic polynomial is calculated. Then, the value of the polynomial in i is considered as the smoothed value.* **b** *The next point is considered.* **c** *The result of least-squares filtering all the function points.*

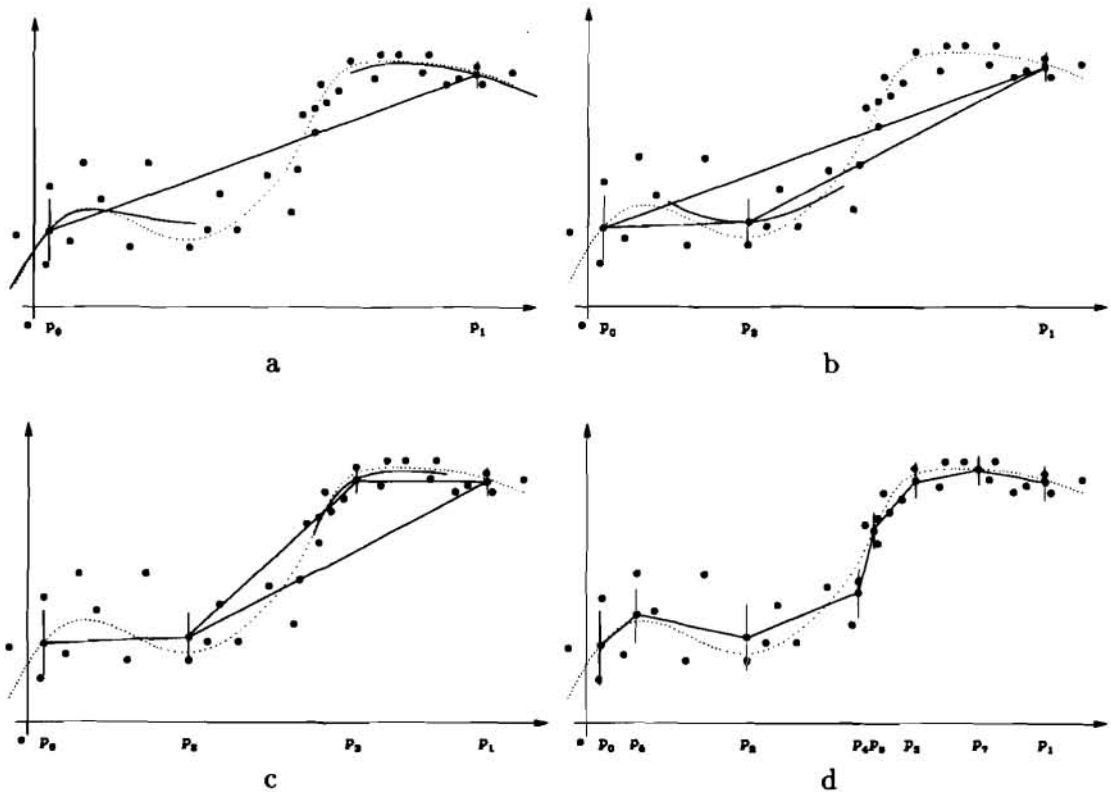


Figure 10 *An 1D description of the fitting algorithm.*

situations:

1. The maximum error criterion, that stops the algorithm when the maximum error between the data points and the fitted surface is lower than a given value.

2. The mear error criterion, that stops the algorithm when the mean absolute error between the data points and the fitted surface is lower than a given value.

3. The maximum number of nodes criterion, that stops the algorithm when the number of nodes in the map is higher than a given value.

4. The local maximum criterion, that compares, locally, the maximum error between the data points and the surface with an approximation of the local standard deviation. Local, means in the 2D case, that the standard deviation and the maximum error are calculated for each single map triangle (in the 1D case, local means in each segment composing the fitting line). If the local maximum error is less than a certain number of times the local standard deviation for each triangle of the map, the algorithm is stopped.

Criteria 1 and 2 can be applied only if the noise level on all the map region is known and constant in all the data set. In particular, criterion 1 can produce unwanted results: it can create a map made of a very high number of points. Criterion 2 can easily omit narrow but tall features that do not change appreciably the mean error on a large region. Criteria 1 and 2 also do not take into account the possibility that the data has different levels of noise in different parts of the seafloor.

Criterion 4 is able to deal with these kind of situations. An example is given in Figure 10: the points on the left part of the figure are more affected by noise than the point on the right. The error bar over each SV point is a possible measure of the local standard deviation: it is the standard deviation of the residual from the polynomial fitting function, used to calculate the smoothed points. If $g(x, y)$ is the fitting function, the residuals are defined as the values given by $\varepsilon_i = z_i - g(x_i, y_i)$, where the (x_i, y_i, z_i) are the N_f points used to determine the fitting surface. If the local maximum error is lower than k times the lower local standard deviation, that point is not added to the set of node of the map. If no more points can be considered, the algorithm stops. That is the reason why there is a greater number of point on the left than on the right of Figure 10.d. Criterion 4 cannot be deceived by tall but narrow features that do not appreciably change the mean error: the tall feature will have points the error of which is higher than k times the mean error and they will not be neglected. Moreover, changing the k value, it is also possible to change the number of nodes used for the map. The lower the value of k, the higher the number of nodes in the map.

The value of k can be determined using an algorithm parameter called k_σ. k_σ is the probability (in percent) that a good point of a Gaussian distribution is farther than k times the approximated standard deviation $(\sqrt{(s^2)/(N_f - 2)})$ of the data residuals from the approximated mean. The value of k is estimated using the student distribution, the number of points used to fit the SV (N_f) and the value of k_σ.

The value of the local standard deviation is calculated in the following way: let ε_a, ε_b, and ε_c be the three standard deviation of the residual between the polynomial fitting functions, used to calculate triangle vertex smoothed points, and the N_f fitting points. The local standard deviation ε is the median of ε_a, ε_b, and ε_c.

The local standard deviations ε should be quite resistant to the outliers, because it is calculated eliminating the outliers (Section 4). The median of the three values is chosen to enhance the resistance of the algorithm to the variations in the calculation of the standard deviations at the SV.

Criterion 4 was tested in a number of synthetic and real situations: the results, always satisfactory even in the case of multiple mapping, are given in Section 5. An important feature of criterion 4 is that it is not necessary to know the data noise level. Moreover, if the noise level in a certain location is high, the number of nodes

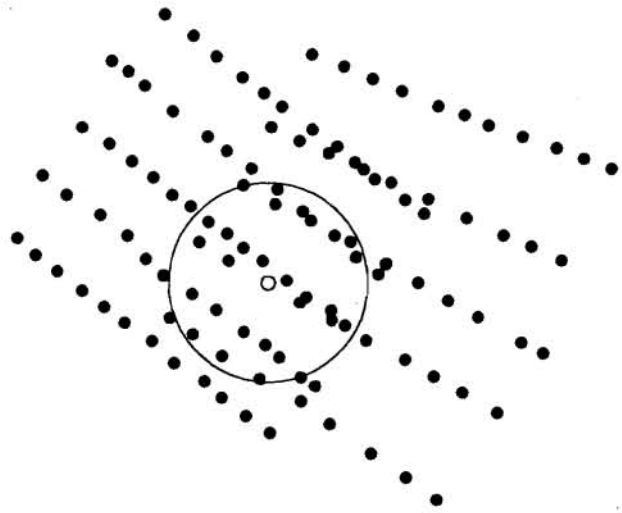


Figure 11 *Data points selection for data fitting.*

approximating the seafloor in that region will be sufficient to reduce to an acceptable level the maximum error between the data points and the fitting function.

3.5 Polynomial fitting in 2D

The polynomial fit described in Subsection 3.3 was performed using the polynomial of a single variable (quadric). When mapping the seafloor, it is necessary to use a polynomial in two variables (conic) of the following form:

$$f(x,y) = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 \tag{1}$$

Given a point, the fitting algorithm finds the N_f nearest points (see Fig. 11). Then, the fitting surface $g(x,y)$ is found using a Singular Value Decomposition (SVD) algorithm (Press *et al.* 1992). This algorithm is both very stable, when the fitting problem is ill-conditioned, and fast. Once the SVD procedure returns the coefficient of the fitting function, the smooth value $z_k^* = g(x_k, y_k)$ at the given point $p_k = (x_k, y_k)$ is calculated, together with the local standard deviation $\sigma_k = \sqrt{\frac{\sum_{N_f} |z_k - g(x_k,y_k)|}{N_f - 2}}$ between the fitting function and the N_f fitting points.

3.6 Definition of the bathymetric data fitting procedure

The bathymetric fitting procedure is a kind of COMPRESS scheme modified in such a way that can be applied to noisy data.

1. Determine a convex polygonal contour $C^{(M)}$ of M vertices, $c_j = (x_j, y_j)$ inside which the map must be realized. For each point in $C^{(M)}$, find the SV using the nearest N_f fitting points, together with the associated standard deviation. Let Ω be a region with a polygonal boundary $\partial\Omega$ consisting of the C^M vertices from V, where $V = \{v_i = (x_i, y_i) \in R^2, i = 1, \cdots, N\}$ are the points where the noisy values of the seafloor $F(x_i, y_i) = F_i$ are known plus the contour vertex c_j. Be $V^{(M)} = C^{(M)}$. Construct an initial $V^{(M)}$ triangulation T of Ω.

2. Construct $f_{T,V}$, the approximating linear piecewise surface, and compute the errors $E_i(f_{T,V}) = |F_i - f_{T,V}(x_i, y_i)|$ and let $E_{f_{T,V}} = \max_{1 \le i \le M} E_i$. Also compute the global mean and maximum of the absolute errors and the local standard deviation.

3. If one of the stopping criterion is verified, end the procedure else go to the next step.

4. Select a data point $v_k = (x_k, y_k) \in V \backslash V(M)$ for which $E_k(f_{T,V})$ is maximal and add it to $V^{(M)}$: $V^{(M+1)} = V^{(M)} \cup \{v_k\}$.

5. Set $M = M + 1$, update the triangulation T to include the new point, and go to step 2.

$C^{(M)}$, the first $V^{(M)}$ set (that will be from now addressed as the initial contour) must be defined from the algorithm user.

The data point p is selected only if the number of points in the triangle is at least N_t.

This algorithm is useful if the number of data points is small, because it is a global method and the calculation of $E_{f_T,V}$ can be very time consuming when the number of data points is large. In this case the whole contour region can be divided into triangular regions where no more than a given number of points are situated (Subsection 3.1). Using each sub-map, the points in the complete data set are selected in such a way that only the data points that are in or near the given sub-map are used for the mapping. Then the global fitting algorithm is applied to each sub-map using the selected data points; the resulting sub-maps are added together eliminating duplicate points. The resulting map is C^0 because the SV point on the sub-map contour, obtained using the least-squares procedure, are produced using the same fitting points thanks to the data set selection procedure. This algorithm has given very good results both for global and sub-mapping situations (Section 5).

4

Outlier elimination

4.1 Introduction

Outliers are considered in a different way depending on the scientific area in which they are encountered. In physics, experiments with a high level of outliers are rare, in other fields, such as economics and demography, the presence of outliers is normal. During bathymetric data analysis, outliers are frequent (Du *et al.* 1996): "outliers occur in multibeam echo sounding data due to the malfunction of electronic unit components, multiple paths, strong reflection from side lobes, and reflection from fish shoals, sub-bottom layers, or abnormal water columns." Du *et al.* (1996) reported the presence, in their bathymetric data, of 10 % outliers in EM1000 Simrad® echosounder data and of 10 to 15 % in Seabat® 9001 echosounder data. During the test of our algorithm, the number of reported outliers was, on average, 5 % on Atlas™ Hydrosweep MD© system.

A high level of outliers is a rule in bathymetric data: in such cases many authors prescribe (Launer and Wilkinson 1979, David 1979) that a better estimation of the surface location is obtained if an algorithm for outlier elimination is used. Every method for outlier elimination is based on a *significance level*, the probability that good data could be discarded assuming the data are distributed according to a given probability density function. For high levels of contamination (10 %), many authors in (Launer and Wilkinson 1979) prescribe the use of methods where the probability threshold is high: it is preferable to discard a certain percentage of good data rather than use outliers for the surface location estimation. In this case, the elimination of a small amount of good data permits a *robust* and *efficient* location of the surface. *Robust* means that the algorithm is able to find an approximation of the real location (with a limited error) even if a high number of outliers is placed far from the real location. *Efficient* means that if outliers are not present, the location estimation has a variance that is near to the variance of the least-squares mean (average).

The algorithm developed here, is intended as a *robust* and *efficient* algorithm for outlier elimination. The starting idea was the algorithm developed by Du *et al.* (1996) for outlier elimination: using this method "... the matches between the two classification" (automatic *versus* manual) "of the soundings data set are greater than 0.95." They analyzed the data for outliers without extracting at the same time the seafloor surface: if the surface fit is known, the outlier elimination may be more

powerful, as it will be clear in the following.

The outlier elimination algorithm presented here is based on the application of two algorithms:

- a *robustness-inducing* algorithm and
- a *quasi*-robust algorithm.

A *quasi*-robust algorithm is an algorithm that cannot be applied when outliers lay indiscriminately far from the real location. Anyway, if outliers are reasonably near to the real location, a *quasi*-robust algorithm is able to estimate it and to identify the outliers. The presence of the *robustness-inducing* algorithm is clear: it is used to identify the presence of "far" outliers and of eventual seafloor discontinuities. In fact, the seafloor is one of the places, in nature, where a step can be encountered and a continuous fitting function cannot be always used: the presence of such discontinuities is taken into account in the *robustness-inducing* part of the outlier elimination algorithm.

4.2 *The* quasi-*robust algorithm*

The *quasi*-robust algorithm is based on a suggestion that can be found in the manual of the **loess** program (Cleveland *et al.* 1992): a program of local regression[1]. **loess** is an interesting program for data mapping when the number of data points is not as high as that of a bathymetric data file. With so many data points the **loess** method is too slow. The *quasi*-robust method is based on a so called M-estimator which minimizes functions of deviations of the observations from the estimates that are more general than the sum of squared deviation or the sum of absolute deviations. In this way the class of M-estimators includes the mean and the median as special cases. It is reasonable to expect that a suitably chosen M-estimator will have good robustness and efficiency on large samples. A simple reformulation of M-estimators yields to a weighted mean in which the weights depend on the data: the resulting estimators, called W-estimators, provide a straightforward way of modifying the familiar least-squares method, particularly in regression problems, where they are the basis for the technique of iteratively reweighted least-squares.

The regression method used here is an iteratively reweighted least-squares method. It works on long-tailed distribution, but it has a high efficiency in the Gaussian case. It is applied in the mapping algorithm, when the N_f fitting points are already collected, during the fitting phase. A first estimation of the fitting surface is calculated $\hat{g}(x, y)$ using the N_f fitting points and the SVD algorithm. The residuals

[1] A program in the **dloss** package, Appendix A.

$\hat{\varepsilon}_i = z_i - \hat{g}(x_i, y_i)$ for all N_f fitting points (x_i, y_i, z_i) are computed. Let

$$B(u,b) = \begin{cases} (1-(u/b)^2)^2 & \text{for } 0 \le |u| < b \\ 0 & \text{for } |u| \ge b \end{cases} \tag{2}$$

be the *bisquare weight* (also *Tukey's biweight*) function, and $m = \text{median}(|\hat{\varepsilon}_i|)$. The *robustness weights* are $r_i = B(\hat{\varepsilon}_i, km)$, where k is a parameter. An updated estimate, $\hat{g}(x_i, y_i)$, is computed using a scaled SVD algorithm using the multiplicative *robustness weights*; thus, points with large residuals receive reduced weight. Then new residuals are computed and the procedure is repeated. The final robust estimate (the "fitting surface") is the result of updating the initial estimate n_c times. After n_c cycles, the fitting surface is identified and the data points farther then km from the fitting surface are eliminated from the data set as outliers. Then, some neighbouring data points are added to the fitting points set, to reach again N_f, and the fitting procedure is repeated till all the fitting points are "good" points.

The same weight system is suggested by Goodall (1983) as the *Tukey's biweight* robust W-estimator, for robust data fitting. Using different values for k it is possible to obtain[2] asymptotic variance near to 1: it is 2.102 for $k = 3$, 1.094 for $k = 6$, and 1.018 for $k = 9$. An asymptotic variance near to 1 is an indication of an efficient estimator. As said before, in presence of a great number of outliers, it is better to discard more outliers (smaller k) even if the asymptotic variance is greater than the optimal one (with asymptotic variance 1). A value of k between 5 and 6 is usually appropriate. In the **NR-COMPRESS** algorithm the value is chosen starting from the *significance level* (in percentage), k_{MAD}: given this probability, the Student distribution for $N_f - 2$ degree of freedom is used to find the wanted value for k. The 6 degrees of freedom are the degree of freedom of the fitting surface.

4.3 The robustness-inducing algorithm

The iteratively Tukey's reweighted least-squares can have problems in two situations connected to bathymetric data: very far outliers and seafloor discontinuities. To avoid such situations, a *robustness-inducing* algorithm is applied before applying the W-estimator. The algorithm is applied during the N_f fitting points selection phase. During that phase, a fitting of the surface is not available: as a consequence the points depth will not, in general, be Gaussian distributed. The idea is to use the *fourth-spread* range (based on an unknown Gaussian distribution, see Emerson (1983) for a description of the test), as a test to eliminate only the "far outliers": the complete elimination will be performed by the *quasi*-robust algorithm. Du (Du

[2]The asymptotic variance is defined as the limit, as n becomes infinite, of $n\text{var}(T_n)$ (if the limit exists); here, T_n is the location estimator. The asymptotic variance for the mean of observations with variance σ^2 is: $n\text{var}(\overline{x}) = n\sigma^2/n = \sigma^2$. The asymptotic variance is usually referred to observation with unitary variance: therefore, the asymptotic variance of the mean is 1.

1995, Du *et al.* 1996) suggests the use of an outlier elimination criterion based on the Uniform distribution[3]. In our case the number of considered points (N_f) is small and the region where these points lie is quite small. As a consequence, the statistics of the points is better approximated by a Gaussian distribution. Moreover, the use of the Gaussian distribution implies criteria that are more conservative with respect to the ones derived from the Uniform distribution.

The *fourth-spread range test* is based on the ordering of a stochastic variable such that $X_0 < X_1 < \cdots < X_{N_f-1}$. The *fourth-spread* d_F is defined as $d_F = F_U - F_L$ where F_U and F_L are the *upper* and *lower fourth*. The *lower* (*upper*) *fourth* is the point under (over) which approximately a quarter of the data lie. The depth (index) of the *lower fourth* is given by

$$\text{depth of fourth} = \frac{[\text{depth of median}] + 1}{2}$$

and the depth of the median is given by

$$\text{depth of median} = \frac{\text{number of samples} + 1}{2}$$

where the brackets in $[x]$ stand for the largest integer not exceeding x. So the rule for finding the depth of a *fourth* says, "Drop any fraction from the depth of the median, add 1, and halve." If the depth of the *lower fourth* is an integer, say k, the value of the *lower fourth* is $F_L = X_k$; else, the value is $F_L = \frac{X_{[k]}+X_{[k]+1}}{2}$. It is clear that the *fourths* are robust statistical values, and that the *fourth-spread* is a robust estimation of the scale of the data. Using F_L, F_U, d_F, and a *significance level* it is possible to define the range of the valid data as $[F_L - K_{d_F} d_F, F_U + K_{d_F} d_F]$. The value of K_{d_F} depends on the *significance level* and is chosen using conservative criteria, because only the *far outliers* must be identified using this technique. K_{d_F} is determined using the Student distribution, taking into account N_f and the *significance level.*

In the *robustness-inducing* algorithm, to avoid the elimination of a point that is on a seafloor steep edge, the point is considered an outlier only if it is the point that the fitting algorithm must smooth. Otherwise, the given point is only considered a *quasi*-outlier and it is removed from the N_f set of points and substituted with a neighbour point. This procedure is intuitive: the outliers, far from the seafloor surface, are the first points to be considered as possible nodes of the map. They are outlier only if near all the neighbour points have a different value. The *robustness-inducing* procedure is iterate until N_f "good" points are find. Under normal conditions, such a set is always found.

The *fourth-spread range test* is also used for discontinuity detection. The largest interval between the ordered values $X_0 < X_1 < \cdots < X_{N_f-1}$ is calculated. If

[3]He claims that a Uniform distribution is more appropriate to describe the statistic of the data if this is dominated by the seafloor variation: this hypothesis could be correct only if the considered points come form a large region that shows significant height variation.

this interval is not inside the *fourth-spread range* a discontinuity is considered to be present. In such cases, all the points on the opposite side of the gap with respect to the point being smoothed are regarded as *quasi*-outliers and are removed from the N_f set. Neighbour points are then inserted into the set of fitting points and the procedure is iterated untill N_f points are found. If a discontinuity is present in the data, the smoothing of a point is thus performed using only points that are "on the same side" of the discontinuity, enhancing in such a way the precision of the mapping algorithm. This procedure is applied only if the number of points on the same side is higher that N_{of}: this condition is added to avoid considering clusters of outliers as discontinuities which are included in the map.

Both the outliers and the discontinuity test are used contemporarily in the *robustness-inducing* algorithm. During all the tests of the algorithm shown in Section 5, if the testing surface does not have discontinuity, the algorithm does not recognize any discontinuity. On real data, some abnormal condition can be caused by the presence of a "compact cloud" of outliers that can produce two effects: it can block the algorithm (if the number of outliers in the cloud is smaller than N_f), or produce a *mesa* effect (if the number of outliers in the cloud is greater than or equal to N_f) that should be identified by the user when the map is produced. By changing some parameters of the algorithm it is possible to eliminate such problems when the presence of a cloud of outliers is identified by the user (Section 5).

4.4 Final elimination step

After the mapping algorithm stops, a last cleaning step for outlier elimination is performed. All points whose error is greater than K times the local standard deviation, are eliminated. The K value is calculated using the Student distribution and the *significance level* parameter $k_{\sigma,o}$.

The outlier elimination phase can be applied to the mapping algorithm without substantial changes. The only changes are in how the "fitting points" are collected and on the substitution of the SVD algorithm with the *quasi*-robust fitting algorithm. It is also clear that the points declared as outliers can be eliminated by the data set after the map is produced.

Some tests on the outlier elimination algorithm are presented in Section 5: they all show good outlier elimination properties.

5 Algorithm testing

The mapping procedure described in the preceding sections is tested here using both synthetic and real data.

5.1 Parameters of the algorithm

The values selected for the algorithm parameters during the test phase, shown in Table 1, are a good trade of different exigencies, but they can not be optimal to each problem. The best way to describe parameter choice is to show the effects of the parameters on various situations. A better description of some of the parameters can be found on Section 3; other parameters are explained in this section.

k_σ Approximate percentage of points used to build the map: the higher this number, the more the points of the map (Subsection 3.4). This parameter is used to vary the number of data points used to create the map.

$k_{\sigma,o}$ Approximate percentage of points eliminated as outlier *after the map is built*: the higher this number, the higher the number of points eliminated as outlier (Subsection 4.4). This parameter is used to determine the sensibility of the algorithm to the outliers.

$k_{\mathbf{MAD}}$ Approximate percentage of the points eliminated as outliers during the fitting phase: the higher this number, the higher the number of points eliminated as outliers (Subsection 4.2). This parameter is used to determine the sensibility to the outliers of the algorithm.

k_{d_f} Approximate percentage of the points eliminated as far outliers during the collection of the fitting set: the higher this number, the higher the number of points eliminated as outliers (Subsection 4.3). This parameter is used to determine the sensibility to the far outliers of the algorithm.

n_c Number of iterative cycles in the fitting algorithm (Subsection 4.2).

$\overline{\varepsilon}_{\max}$ Maximum mean error under which the algorithm is stopped (Subsection 3.4).

$\varepsilon_{\max}$ Maximum error under which the algorithm is stopped (Subsection 3.4).

N_f	k_σ %	$k_{\sigma,o}$ %	k_{MAD} %	k_{d_F}	n_c	$\bar{\varepsilon}$	ε_σ	$\varepsilon_{\max}$	N_{of}	N_t	n_p
30	0.5	k_σ	0.1	0.001	5	0	0	0	$0.25N_f$	0	10,000

Table 1 *Table of the selected values for the* **NR-COMPRESS** *algorithm.*

N_f	10	15	20	25	30	40	50	60	100
ε_f [m]	.042	.045	.058	.069	.085	.095	.118	.120	.170

Table 2 *Value of the fitting error for various N_f: data are 10,000 samples of F_4 with no noise added.*

ε_σ Maximum standard deviation under which the algorithm is stopped (Subsection 3.4).

N_f Number of fitting points: the higher this number the smoother the map (Subsection 3.3 and 3.5).

N_{of} Number of points that can be considered as outliers in the fitting set (Subsection 4.3).

N_t Minimum number of points in a map triangle: this parameter is used to reduce the number of nodes of the generated map when a discontinuous seafloor region is encountered (Subsection 5.3).

n_p Maximum number of points lying in a sub-map (Subsection 3.1 and 3.6).

5.2 Fitting error

The fitting procedure acts as a low pass filtering procedure resulting in a smoothing of the surface that can be regarded as an error. Using the synthetic data without noise (samples from the F_4 function, see the next subsection), a test was performed on the value of the maximum fitting error, ε_f, as the number N_f of the fitting points increases (the low-pass effect increase). Table 2 gives the result of the test when 10,000 "non noisy" samples of the F_4 function were used. A smoother test function will produce a lower fitting error. Using the table it is possible to select (for synthetic data) the value of N_f. In fact, it is better to use the value of N_f for which the fitting error is lower than the noise on the data. When analyzing real data, the value of N_f must be selected by the user depending on data noise and on the required smoothing effect.

5.3 Test on synthetic data

Test functions

A set of nine test functions was chosen to conform to (Dyn *et al.* 1990, Franke 1979, Lyche and Morken 1987, Rippa 1992). These functions are standard for testing interpolation and fitting algorithms. Only the following four functions were chosen to show the results as no additional information was given by the remaining functions: the results of the algorithm on the functions not in the table were comparable with the results on the four functions used.

$$\begin{aligned} f_1(x,y) &= -\left(1-\frac{x}{2}\right)^6\left(1-\frac{y}{2}\right)^6 - 1000(1-x)^3x^3(1-y)^3y^3 \\ &\quad -y^6\left(1-\frac{x}{2}\right)^6 - x^6\left(1-\frac{y}{2}\right)^6 \end{aligned} \tag{3}$$

$$f_2(x,y) = \frac{\tanh(9y-9x)+1}{9} \tag{4}$$

$$f_3(x,y) = \exp\left(-\frac{81}{4}\left((x-0.5)^2+(y-0.5)^2\right)\right) \tag{5}$$

$$f_4(x,y) = \begin{cases} 1 & \text{if } y-\xi \geq \frac{1}{2} \\ 2(y-\xi) & \text{if } 0 \leq y-\xi \leq \frac{1}{2} \\ (\cos(4\pi r(\xi,y))+1)/2 & \text{if } r(\xi,y) \leq \frac{1}{4} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $r(\xi,y) = \left((\xi-\frac{3}{2})^2+(y-\frac{1}{2})^2\right)^{\frac{1}{2}}$, and $\xi = 2.1x-0.1$.

A fifth function was added to the set to test the algorithm on a discontinuous surface:

$$f_5 = \begin{cases} 1-2.3(x-y)^2 & \text{if } 1.8(1-y)^2 < x-0.3 \text{ and } 1-2.3(x-y)^2 > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The first function f_1 is a polynomial surface of degree 12 (Fig 12.**a** and **b**); function f_2 simulates a sharp rise running diagonally (Fig. 12.**c** and **d**); f_3 is a Gaussian hill (Fig. 12.**e** and **f**) and f_4 represent a "mountain" on a plane and a ramp leading to another plane (Fig. 12.**g** and **h**). It is a function with discontinuous first derivatives. The last function, f_5, represents a mountain on a plane with a cliff (Fig. 12.**i** and **j**). Equations (4) to (7) produce data in a small interval of depth; moreover, the x and y range are intended to be [0,1]. To obtain data files comparable with those of a multibeam echosounder, the functions f_i have been normalized in such a way that their range of variation was equal to one ($\overline{f}_i$). The following transformation was applied to $\overline{f}_i$:

$$F_i = d - \text{range}_d \overline{f}_i\left(\frac{x-\text{lat}_0}{\text{range}_{\text{lat}}}, \frac{y-\text{lon}_0}{\text{range}_{\text{lon}}}\right) \tag{8}$$

where:

lat_0 is the starting latitude in UTM

lon_0 is the starting longitude in UTM

$\text{range}_{\text{lat}} = \text{range}_{\text{lon}} = 200$ are the range of variation of latitude and longitude

$d = 65$ m, is the depth

$\text{range}_d = 7$ m is the maximum depth variation of the data (with no noise)

The value of lat_0 and lon_0 was used in order to compare the results with collected data the coordinates of which were given in UTM: using this transformation "natural" variations of latitude and longitude result in "natural" variations of depth. The functions were sampled using a random Uniform sampling: the resulting sample distribution is similar to the bathymetric data distribution, but the data are ordered differently. The order of the data does not affect the results of the mapping algorithm but does affect calculation time which is higher if the samples are randomly distributed in the map area. During the fitting phase, the error calculation is based on a search algorithm that is much faster if the point queried is near to the preceding point.

Figures 12.**b**, **d**, **f**, **h**, **j** are contour plots obtained using 10,000 random samples of F_1 to F_5: the number of samples used to test the global algorithm when not specified differently.

Synthetic noise

Gaussian uncorrelated noise is added to all the synthetic data sets. To simulate different multibeam sonar two values of noise variance are used: $\sigma_m = 0.5$ m and $\sigma_n = 0.05$ m. Figure 13 shows an example of the effect of the noise on data sampled by the F_4 function.

Some tests were also done using uniform noise. As the algorithm performed as in the Gaussian case, an example is not reported.

To simulate real data, outliers are added, in some tests, to the synthetic data. To simulate outliers, a fixed error ($\pm k_o\sigma$, where σ is the Gaussian noise variance) is added to 5 % of the data. For example, in the case of the standard 10,000 points data set:

- to 9,500 data was added gaussian noise with variance σ,
- to 500 data was added the fixed error $\pm k_o\sigma$.

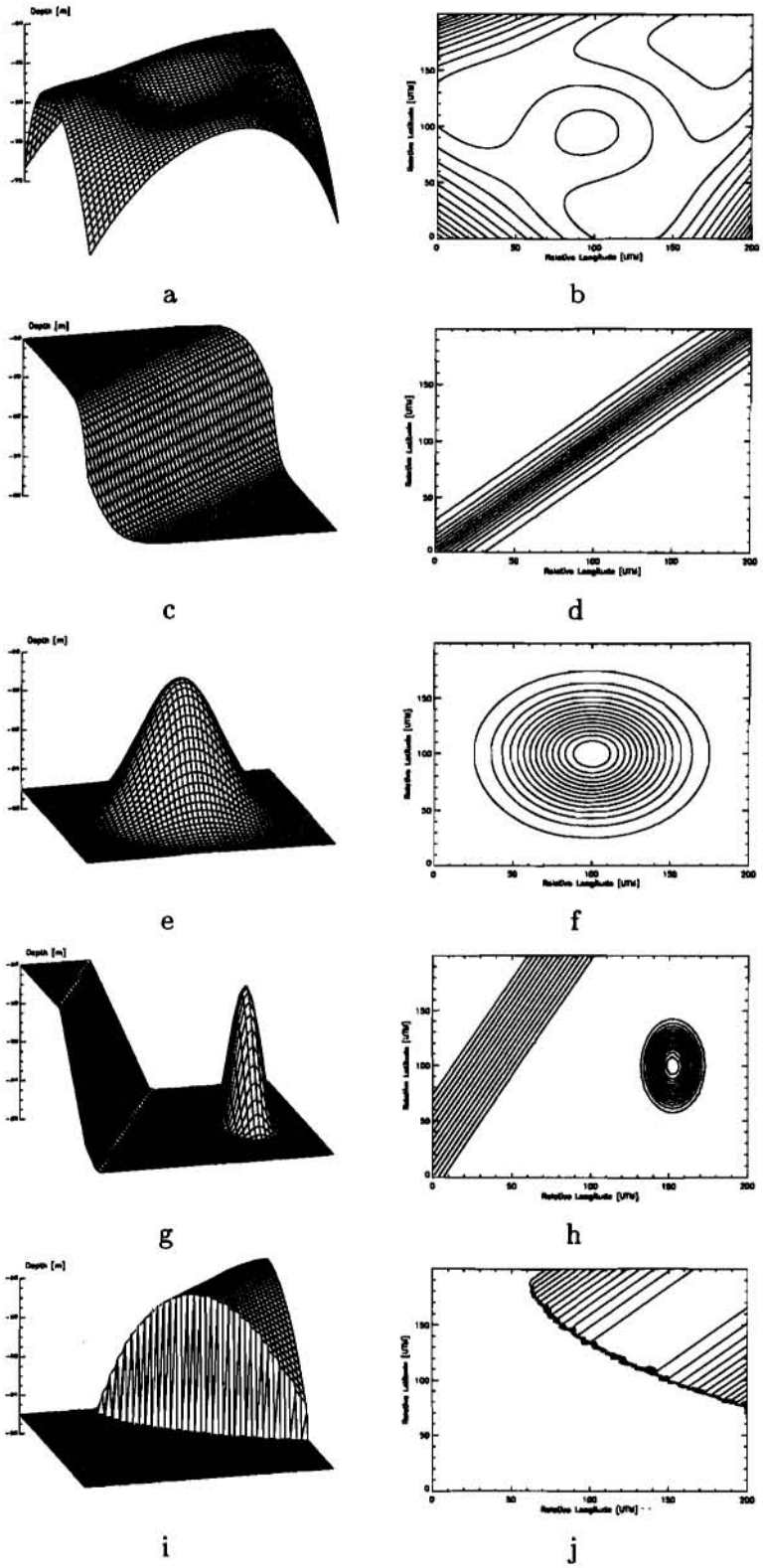


Figure 12 *The synthetic test functions.*

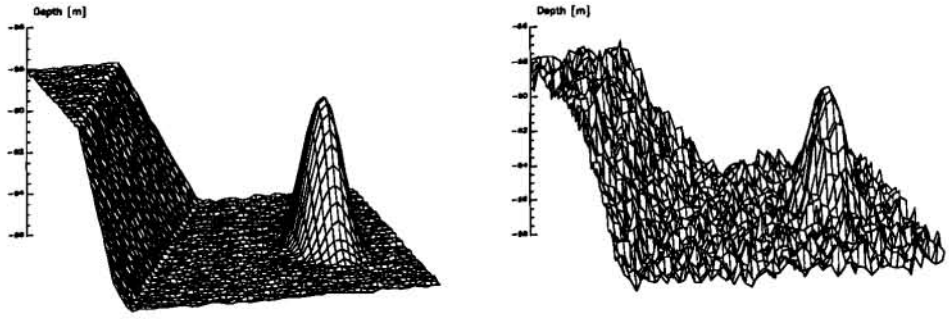
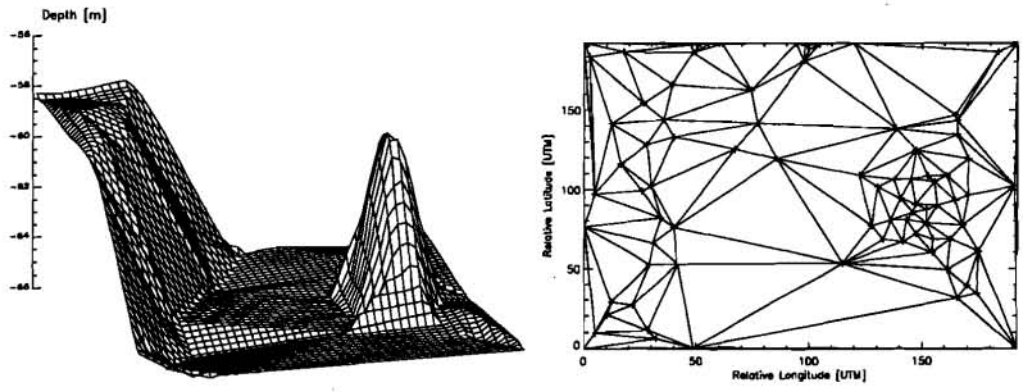


Figure 13 *The effect of the noise added to the F_4 function: the variance of the noise applied to the data is, on the left, σ_n and, on the right, σ_m.*

Tests on synthetic data with Gaussian noise and no outliers

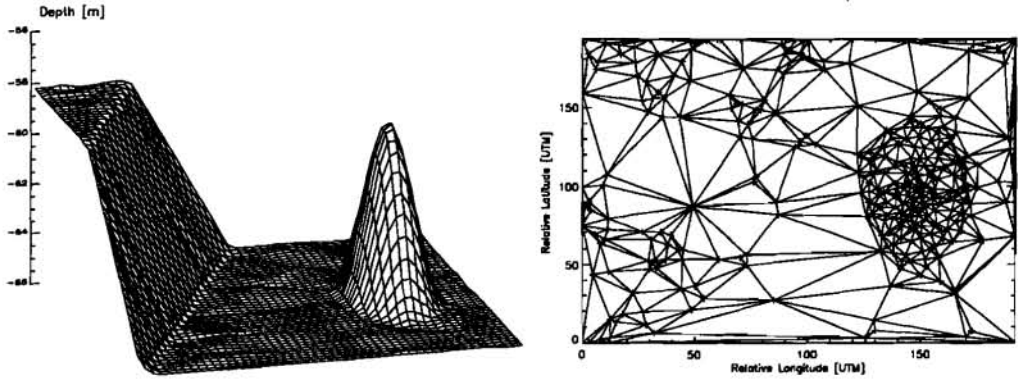
The first test was performed on a synthetic data set: the map of these data (with no outliers) is produced using parameters which differ to the selected parameters for the algorithm (see Table 1). In fact, these parameters (intended for at sea data) tend to eliminate a high percentage of points with the philosophy that is better to eliminate some good points rather than fitting using bad points (David 1979). As in this case it is known that outliers are not present, conservative parameters are used which do not eliminate too many points and which produce maps from a small number of nodes ($k_\sigma = 0.5$ and $k_{\mathrm{MAD}} = 0.01$). The data are obtained from function F_4 with superimposed Gaussian noise with variance σ_m (see Fig. 13). Figure 14 shows the results in terms of reconstructed surface, triangulation, parameters and statistics. Considering the high level of noise, the surface is well reconstructed. Finally, it is interesting to note how the standard deviation and the mean of the absolute deviation of the data from the reconstructed surface are near to the respective quantities for the data with respect to the "real" synthetic surface (respectively $\sigma_F = 0.5$ m, and $\overline{\varepsilon} = 0.4$ m). In the the table in Fig. 14, and in the following tables, the columns labelled N_o and N_n give, respectively, the number of outliers in the data set lying in the map and the number of nodes of the map. The column labeled σ_F gives the value of the standard deviation of the noise added to the synthetic functions samples.

When the algorithm is applied to a low noise data set (see Fig. 13) more points are necessary to obtain a result in which the map is sufficiently accurate to have a variance from the data of the same order of the variance of the noise on the data. For example, in Subsection 2.3 the minimum standard deviation obtained using a map of 46 triangles was 0.102 m. Using σ_n as the level of noise (a low level in a map) it will require a much higher number of points to obtain a standard deviation between the map and the data of the order of σ_n. The N_f value was reduced to



N_f	σ_F [m]	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	$\varepsilon_{\max}$ [m]	N_o	N_n
30	.5	0.5	0.01	.518	.415	1.63	38	83

Figure 14 *The results of the fitting algorithm applied to data characterized by a high level of noise.*



N_f	σ_F [m]	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	$\varepsilon_{\max}$ [m]	N_o	N_n
15	.05	0.5	0.01	.053	.043	.176	289	346

Figure 15 *The results of the fitting algorithm applied to data characterized by a low level of noise.*

15, to reduce the fitting error (Subsection 5.2). Using 15 fitting points, $k_\sigma = 0.5$, $k_{\mathrm{MAD}} = 0.01$ and the selected values of the other parameters, the values of the statistics of the obtained map (see Figure 15) are higher than the values of Gaussian noise, but the number of nodes and outliers found by the algorithm are lower than those obtained using higher values for k_σ. It is noteworthy, in Fig. 15, that the mapping algorithm uses a high resolution where the surface second derivative is high, and a low resolution where it is zero.

F	σ_F	σ [m]	$\bar{\varepsilon}$ [m]	εmax [m]	N_o	N_n
F_1	σ_m	.515	.412	1.73	36	69
F_2	σ_m	.522	.418	1.68	30	56
F_3	σ_m	.522	.418	1.68	31	69
F_4	σ_m	.518	.415	1.63	38	83
F_1	σ_n	.053	.042	0.173	137	209
F_2	σ_n	.052	.042	0.181	189	309
F_3	σ_n	.053	.043	0.180	221	313
F_4	σ_n	.053	.043	0.176	289	346

Table 3 *Results of* **NR-COMPRESS** *on the first four test functions. The only parameters of* **NR-COMPRESS** *different from the selected ones are* $k_\sigma = 0.1$*, to reduce the number of generated points, and* $N_f = 15$*, when the noise level is* σ_n*.*

N_f	15	30	50	100	200	300	400
$\lvert\varepsilon\rvert/\sigma_m$	2.00	1.22	0.96	0.66	0.20	0.18	0.17

Table 4 *Relative error realized by* **NR-COMPRESS** *in the mapping of a constant depth synthetic seafloor data with a Gaussian noise of variance* σ_m*. This is a value calculated using the result of a single realization, not a medium on more realizations.*

The triangulation from Fig. 14 is influenced only slightly by the shape of the surface. A stronger effect on the map structure is shown in Fig. 15 which shows a much higher number of triangles where the "mountain" is. This is an effect of the level of the noise on the data: when it is low, the algorithm attempts to find a much accurate solution. By changing the k_σ parameter it is also possible to obtain a less accurate map.

Table 3 shows the results of the application of **NR-COMPRESS** to the sets of data obtained from the first four test functions (the parameters were $k_\sigma = 0.5$ and $k_{\text{MAD}} = 0.01$).

5.3.1 *Limits to the map accuracy*

An interesting test was applied to a synthetic data set of noise with constant variance σ_m (i. e. constant depth). Table 4 reports the result of a single run of the **NR-COMPRESS** in terms of error betweeen the reconstructed surface and the real one. It is easy to see that the error decreases when the number of fitting points increases (in the case of constant deep). When the number of fitting points is below or equal to 30 points, the error is quite high: this is a statistical limit for **NR-COMPRESS**. Two seafloor surfaces realized from data from the same site could therefore have a difference equal to twice the error reported in Table 4.

The triangulation

It is interesting to see the maps (and the triangulations underlying the maps) produced to realize Table 3. These maps are shown for $\sigma_F = \sigma_n$ and $\sigma_F = \sigma_m$ in Fig. 16 and 17: the triangulation is more dense where the second derivatives of the functions are higher. The regularity of the contour levels even in presence of a high noise is also noteworthy (compare Figs. 16 and 17 with Fig. 12).

Another interesting test is shown in Fig. 18, which is a map elaborated from a data set obtained from the F_2 function using varying noise. The top part of the seafloor has low noise variance (σ_n), while the bottom part has high noise variance (σ_m). This simulates the change due to the presence of, for example, sea weed in a part of the map (sea weed can be present at a given depth and not at another one). The algorithm, using the local noise criterion, recognizes the variation of the data noise and produces a map that is more accurate where the noise is low, and coarser where the noise is high. Fig. 18 shows the maps produced when the data noise is low or high on all the data set, and the map produced when it changes depending on the site.

Tests on synthetic data with Gaussian noise and outliers

The test of the algorithm with outliers is realized using the synthetic data set with outliers previously described (the underlying function was F_2). Table 6 shows the result of applying **NR-COMPRESS** using conservative parameters (the selected parameters): these settings do not remove many "good" data but, at the same time, they render the algorithm sensible to outliers only when k_o is very high. When k_o is equal to 4 the number of identified outliers (among the 500 added to the 9,500 noisy data points) is only the 57.8 % (see column N_o % of Table 6). N_o % is referred to the percentage of identified outliers among the 500.

If less conservative parameters are used (the selected parameters and $k_{\mathrm{MAD}} = 0.2$ %), more "good data" are eliminated as outliers (Table 7) but at $k_o = 4$ the number of identified outliers is already 97.2 % of the total outliers. When $k_o = 5$ all outliers are recognized and eliminated.

An explanation must be given about the dependence of σ (and of the other statistical estimators) with k_o. Data point error variance varies with k_o. 9,500 data have a noise variance of 0.5 and 500 have varying noise amplitude. The column σ_F in the Tables 6–7 gives the real variance value as k_o varies from 0 to 5. At the same time the variance of the result is calculated using all the data exept the part of data eliminated as outliers. The resulting dependence of σ is the combination of both phenomena.

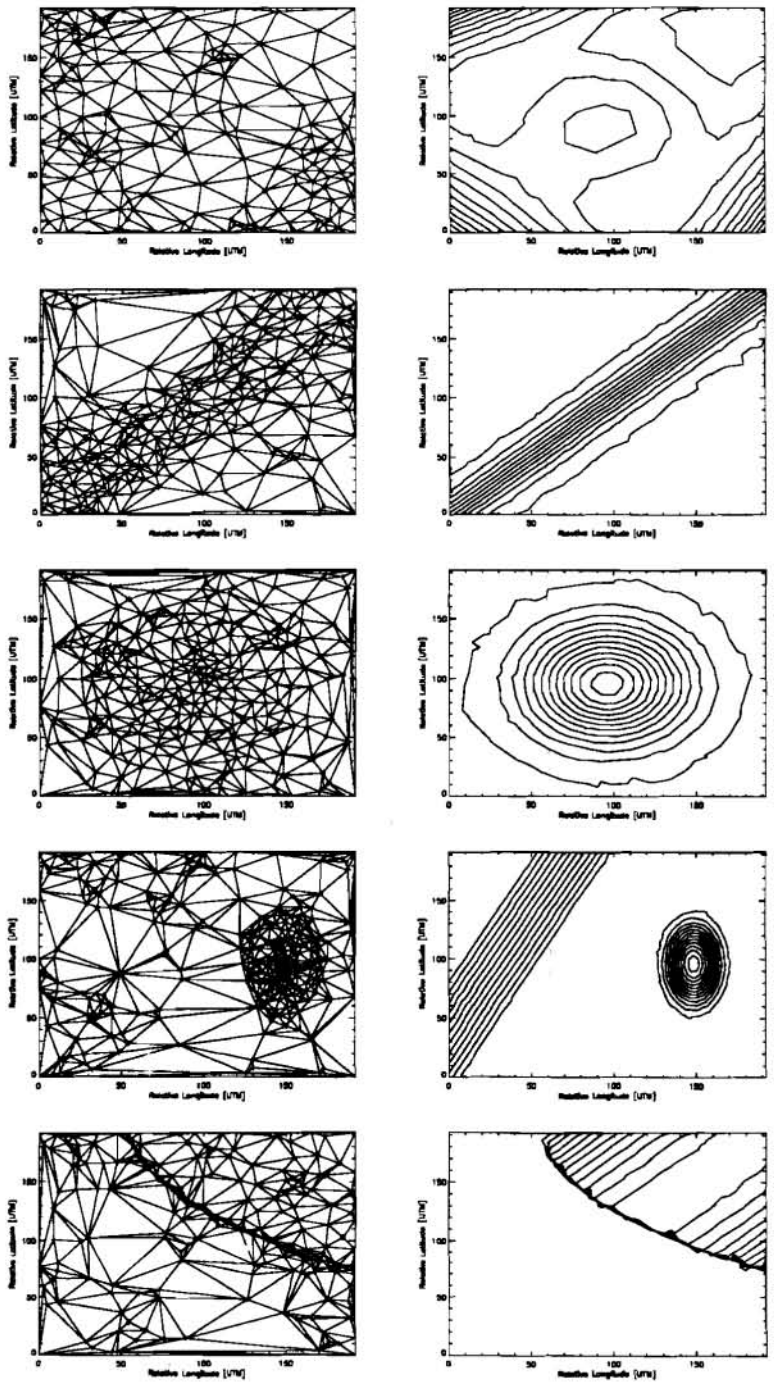


Figure 16 *The triangulation and contour graph of the maps produced from the synthetic data with low noise level (σ_n) and no outliers.*

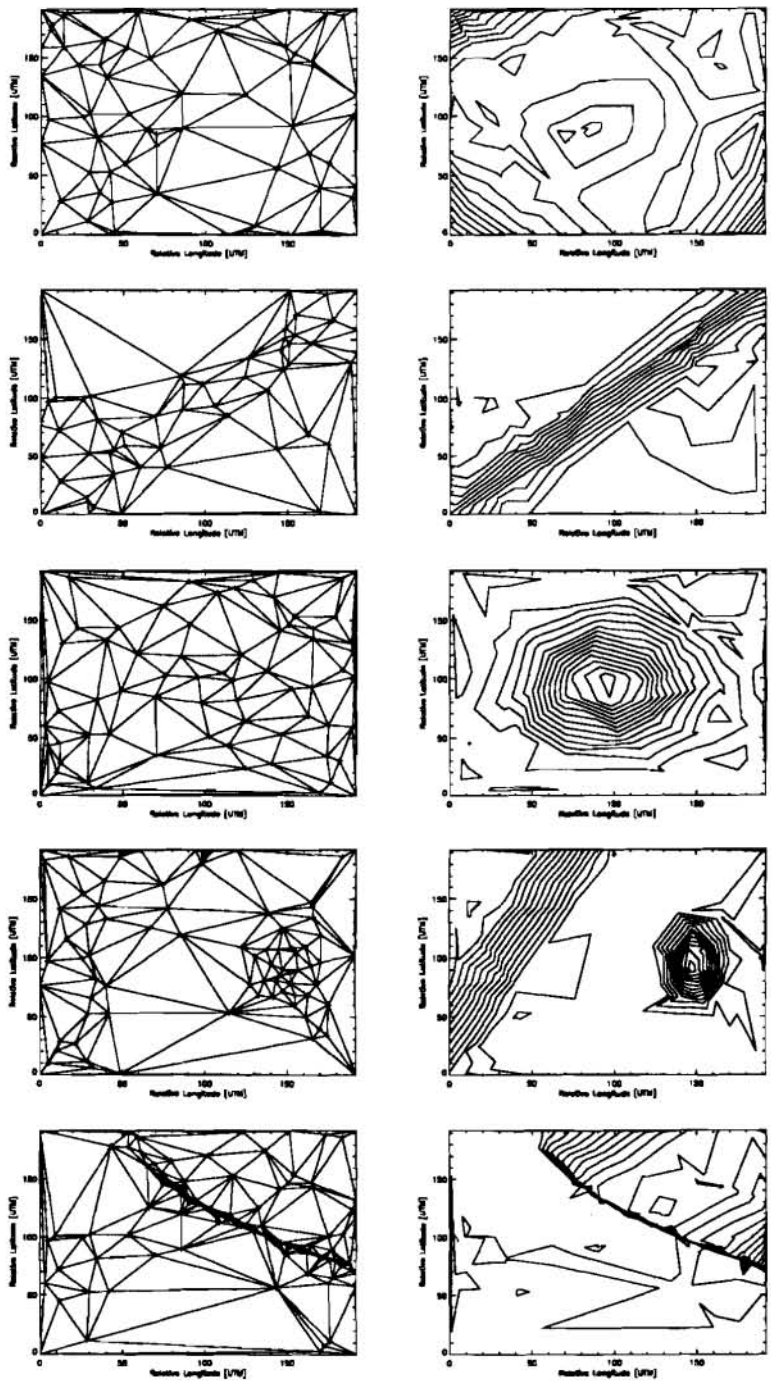
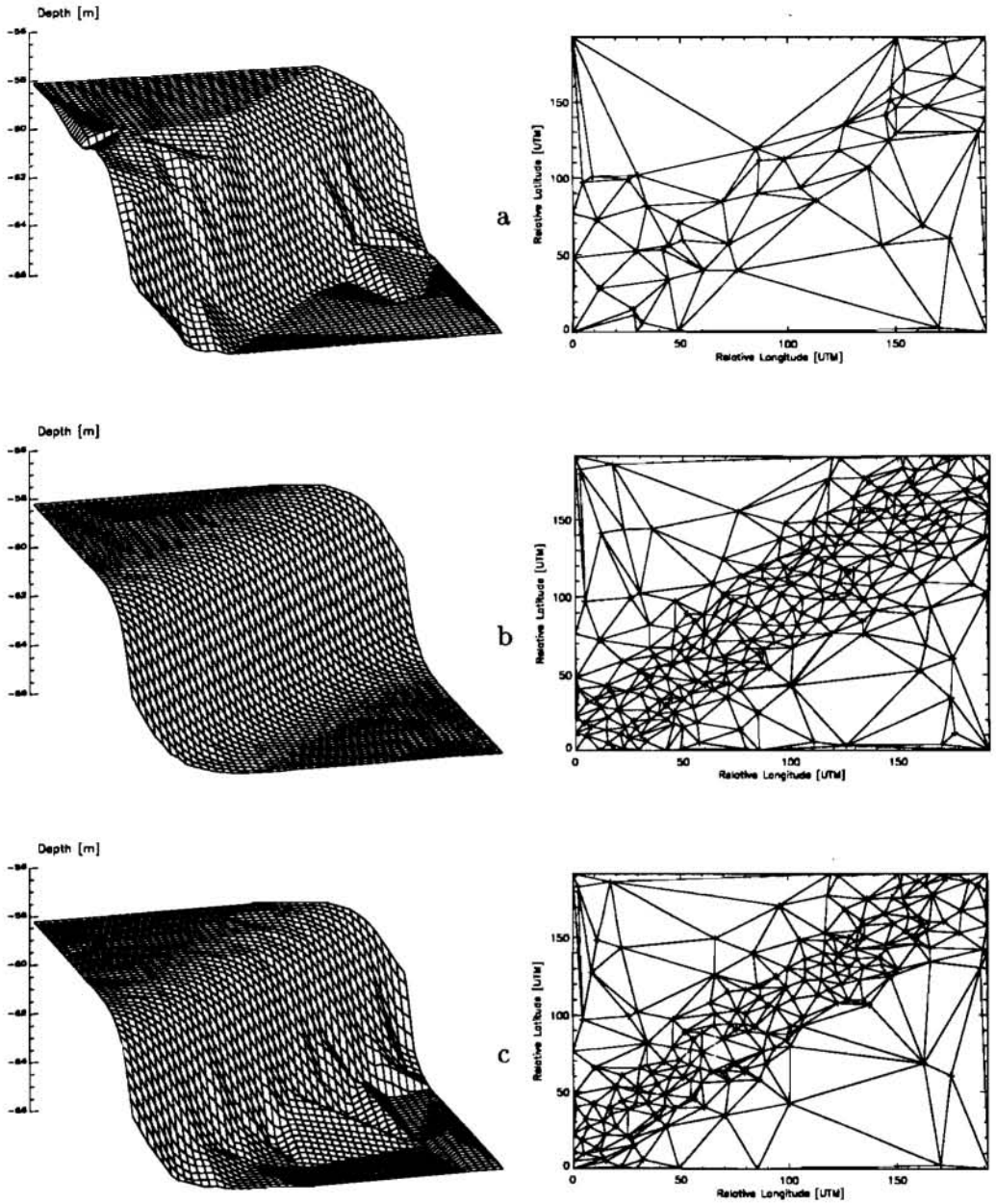


Figure 17 *The triangulation and contour graph of the maps produced from the synthetic data with high noise level (σ_m) and no outliers.*



N_f	σ_F [m]	k_σ %	k_{MAD} %	N_o	N_n
30	0.5 & 0.05	0.5	0.01	414	279

Figure 18 *The results of the fitting algorithm applied to data characterized by a change in the noise characteristics on the seafloor. Plots* **a** *show the case with constant high level of noise, plots* **b** *show the case with constant low level of noise ($N_f = 30$). Finally, plots* **c** *shows the case of varing level of noise (low level at 55 m depth and high level at 66 m). The ensemble means (σ etc.) are not given because they do not make sense in this case.*

k_o	0	1	2	3	3.5	4	4.5	5
σ	0.490	0.502	0.537	.591	.624	.660	0.698	0.739

Table 5 *Data variance changes with k_o: the noise variance on the 95 % of the data was σ_m.*

k_o	σ_F	σ [m]	$\bar{\varepsilon}$ [m]	εmax [m]	N_o	N_n	N_o %
0	0.490	0.505	0.399	1.64	43	61	0.0
1	0.502	0.524	0.423	1.63	27	66	0.0
2	0.537	0.564	0.452	1.76	20	58	0.2
3	0.591	0.602	0.466	1.89	42	79	5.4
3.5	0.624	0.612	0.471	1.98	147	102	25.6
4	0.660	0.608	0.474	2.12	312	116	57.8
4.5	0.698	0.611	0.480	2.23	398	90	74.6
5	0.739	0.609	0.484	2.28	482	66	90.8

Table 6 *Results of* NR-COMPRESS *on data with outliers (the normal noise on data has variance σ_m). The total number of outliers is 500 out of 10,000 (selected parameters).*

If the data are affected by a lower noise (σ_n), the algorithm (Table 8) tends to eliminate a much higher number of outliers (up to 17 %) to reach a good sensitivity at low value of k_o (using the selected parameters with $k_\sigma = 1$).

These two tests confirm the statement that, when data are affected by outliers it is better to eliminate an even high number of "good" data rather then to take into account too many outliers (see the difference between Table 6 and Table 7). In fact, the elimination of "good" data does not significantly affect the map reconstruction while to take into account "bad" data produce a big error in the resulting map.

An important fact that can be extrapolated from Table 8 regards outlier elimination when data noise is low. If most of the data with a residual much higher than the data set variance must be eliminated, it is also necessary to eliminate a substantial amount of "good" data. As a consequence, it is difficult to discriminate between outliers and some "good"data. This is true for automatic algorithms as for human outlier elimination. This problem is less important when dealing with higher noise levels.

Tests on discontinuous synthetic data

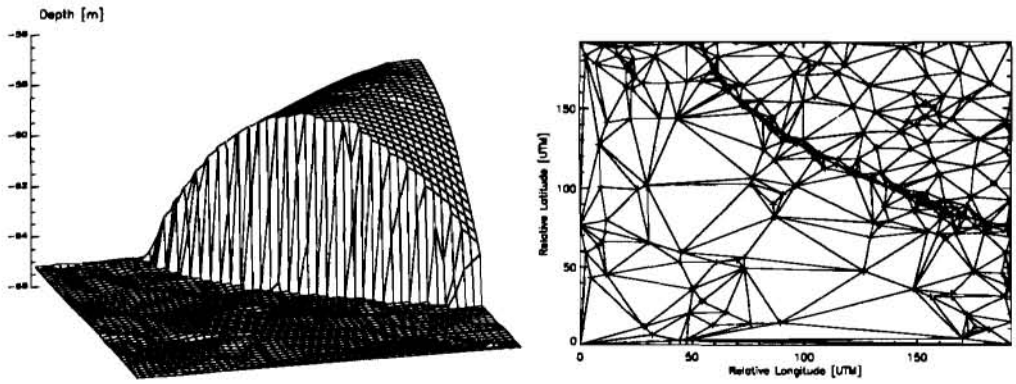
NR-COMPRESS was also tested on the function F_5 to verify it when in presence of a discontinuity of the seafloor. Figures 19 and 20 show the results of the algorithm using data with Gaussian noise variance σ_n and σ_m. The algorithm produces maps

k_o	σ_F [m]	σ [m]	$\overline{\varepsilon}$ [m]	εmax [m]	N_o	N_n	N_o %
0	0.490	0.484	0.384	1.55	223	69	0.2
1	0.502	0.501	0.407	1.58	292	77	1.4
2	0.537	0.537	0.431	1.62	219	72	6.0
3	0.591	0.560	0.442	1.75	460	87	47.8
3.5	0.624	0.550	0.436	1.86	745	94	75.4
4	0.660	0.522	0.417	1.73	685	72	97.2
4.5	0.698	0.515	0.413	1.69	697	74	99.6
5	0.739	0.516	0.413	1.64	659	71	100.0

Table 7 *Results of* NR-COMPRESS *on data with outliers. The total number of outliers is 500 out of 10,000 (the normal noise on data has variance σ_m). The parameter of* NR-COMPRESS *different from the selected ones is $k_{\mathrm{MAD}} = 0.2$.*

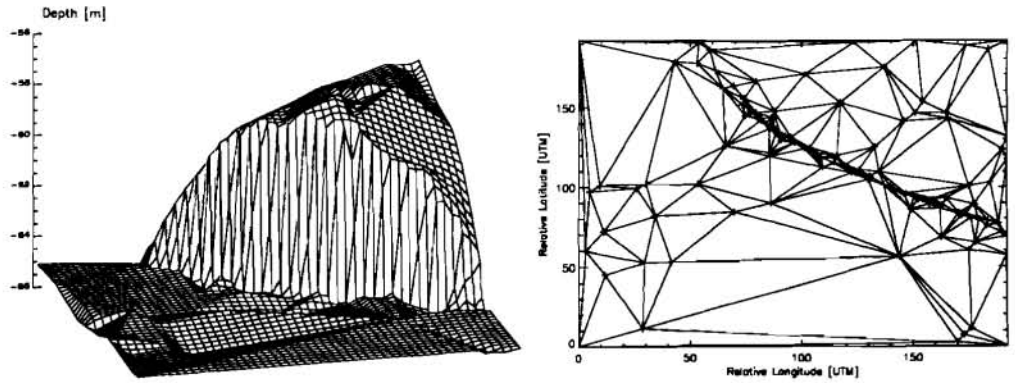
k_o	σ_F [m]	σ [m]	$\overline{\varepsilon}$ [m]	εmax [m]	N_o	N_n	N_o %
0	0.049	0.047	0.038	0.149	1334	414	5.2
1	0.050	0.049	0.0395	0.169	1184	400	10.6
2	0.054	0.054	0.043	0.173	971	359	20.8
3	0.059	0.057	0.045	0.204	1152	392	50.4
3.5	0.062	0.058	0.045	0.240	1093	365	63.8
4	0.066	0.060	0.048	0.203	1344	399	78.0
4.5	0.070	0.054	0.043	0.213	1498	420	91.0
5	0.074	0.0498	0.040	0.201	1766	373	98.8

Table 8 *Results of* NR-COMPRESS *on data with outliers. The total number of outliers is 500 out of 10,000 (the normal noise on data has variance σ_n). The parameter of* NR-COMPRESS *different from the selected ones is $k_{\mathrm{MAD}} = 0.01$.*



N_f	σ_F [m]	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	εmax [m]	N_o	N_n
15	.05	0.5	0.01	.053	.043	.173	195	263

Figure 19 *The results of the fitting algorithm applied to data obtained by a discontinuous function and characterized by a low level of noise.*



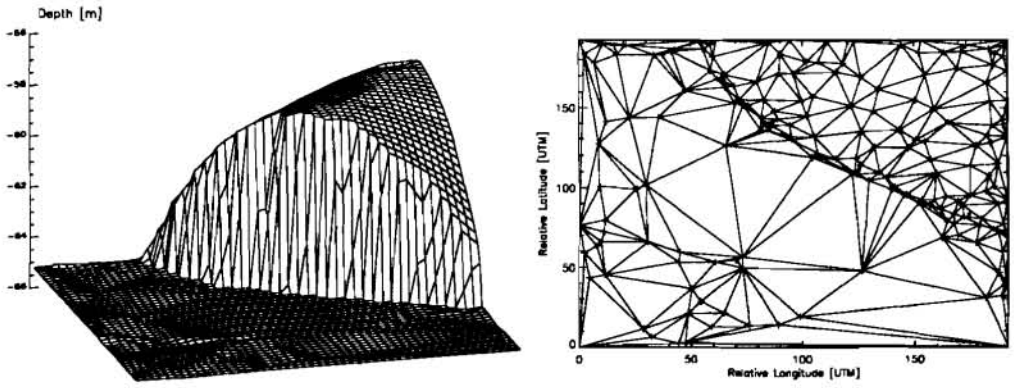
N_f	σ_F [m]	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	$\varepsilon_{\max}$ [m]	N_o	N_n
30	.5	0.5	0.01	.52	.42	1.68	54	127

Figure 20 *The results of the fitting algorithm applied to data obtained by a discontinuous function and characterized by a high level of noise.*

that have many nodes around the discontinuity: this result is not optimal. This is controlled by the parameter N_t as explained in Subsection 3.6. When the number of data points lying in a map triangle is below N_t, that triangle is not considered when **NR-COMPRESS** looks for a new map node. The selected value for N_t is zero: if its value is different, (*e.g.* 5 in Fig. 21), the number of triangles (nodes) along the discontinuity is lower, but a certain number of points is eliminated as outlier. This is usually not an important problem and it is possible to always put N_t to a number higher than zero. If N_t is different from zero, the level of σ, $\overline{\varepsilon}$, and $\varepsilon_{\max}$ is, usually, slightly higher than for $N_t = 0$.

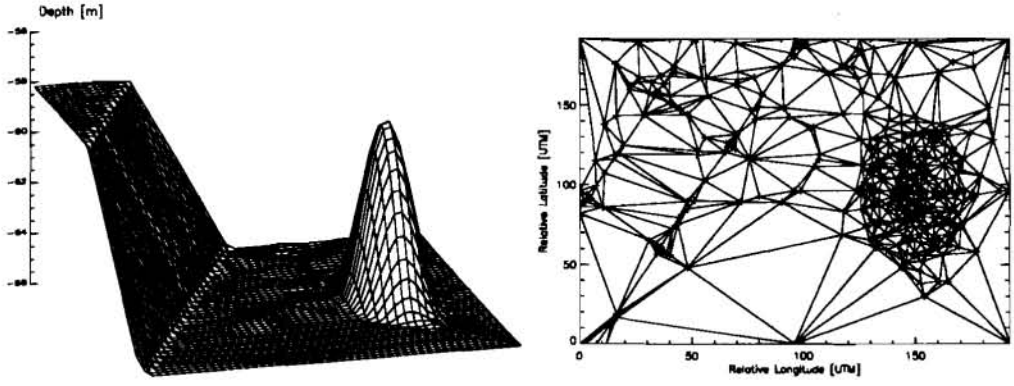
Tests on synthetic data multiple maps

After the tests of **NR-COMPRESS** as a global algorithm other tests were performed on synthetic data using the algorithm on sub-maps (Subsection 3.1). Two results on local application of the **NR-COMPRESS** algorithm are shown on Fig. 22 and 23. The data sets are made of 120,000 points and the algorithm was applied using the selected parameters but dividing the problem on maps with no more than 10,000 points each (16 maps in this case). As for the global problem solution, the map with lower noise is reconstructed with a higher resolution and lower error than the map with higher noise. The figures show that when the **NR-COMPRESS** algorithm works using sub-maps it produces results very similar to the global application of the **NR-COMPRESS** algorithm. The only difference is a small increase in the number of nodes and in the noise between the data and the reconstructed map.



N_f	σ_F [m]	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	εmax [m]	N_o	N_n	N_t
15	.05	0.5	0.01	.053	.042	.173	249	189	5

Figure 21 *The results of the fitting algorithm applied to data obtained by a discontinuous function and characterized by a low level of noise ($N_t = 5$).*

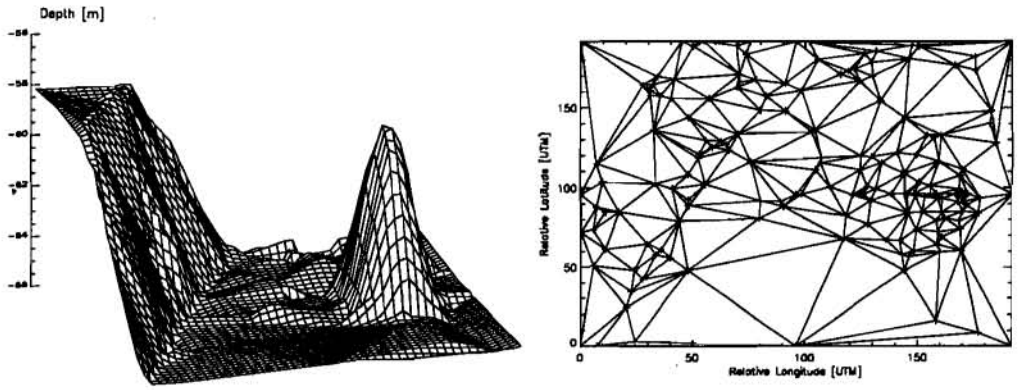


N_f	σ_F [m]	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	εmax [m]	N_o	N_n
15	.05	0.1	0.01	.053	.043	.202	156	444

Figure 22 *The results of* NR-COMPRESS *multimap algorithm applied to synthetic data obtained by the F_4 function and characterized by a low level of noise.*

5.4 Test on real data

The data collected at sea used to test the algorithm was obtained using an Atlas^{TM} HYDROSWEEP MD© multibeam echosounder (STN ATLAS Elektronik GmbH, Bremen, Germany).



N_f	σ_F [m]	k_σ	k_{MAD}	σ [m]	$\overline{\varepsilon}$ [m]	εmax [m]	N_o	N_n
30	0.5	0.1 %	0.01 %	.525	.420	1.94	75	178

Figure 23 *The results of* NR-COMPRESS *multimap algorithm applied to synthetic data obtained from the F_4 function and characterized by a high level of noise.*

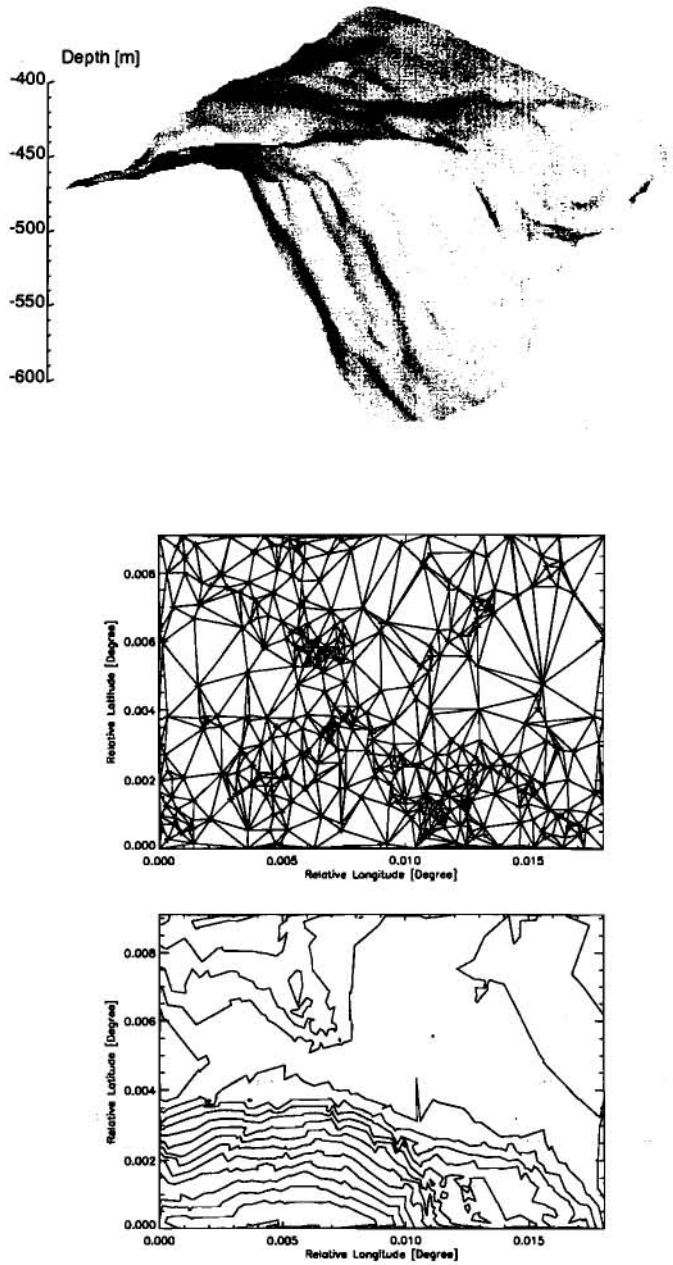
Small size data file

The first test on real data was performed on a deep-water area off Sestri Levante (Italy). Two sets of tracks were available to test the algorithm: North-South and East-West tracks. The tracks were nearly completely overlapped and a part of the beams (the 5 outer beams on each side) was eliminated. The number of acquired data points was low (36482), so the map could be obtained with the subdivision of the NR-COMPRESS problem in only 7 sub-maps ($n_p = 10,000$). The map from the data of all tracks (North-South and the East-West) is shown in Fig. 24.

Figure 25.**a** shows the fitting errors resulting from the algorithm when fitting the data to find the map nodes. This map is a kind of local noise map from which is possible to localize zones where the noise is anomalously high. The maximum noise variance in this map is approximately 16 m and is localized in some zones of the map. Considering the number of fitting points (N_f=30) the maximum error expected from the obtained map should be (from Table 4) $2 \times 0.96 \times 16 \approx 32$.

Figure 25.**b** shows the difference between the maps obtained from the North-South and the East-West tracks. The maximum value of the difference is, as expected, around 32 m. It is also very interesting to see that there is a direct correspondence between Fig. 25.**a** and Fig. 25.**b**.

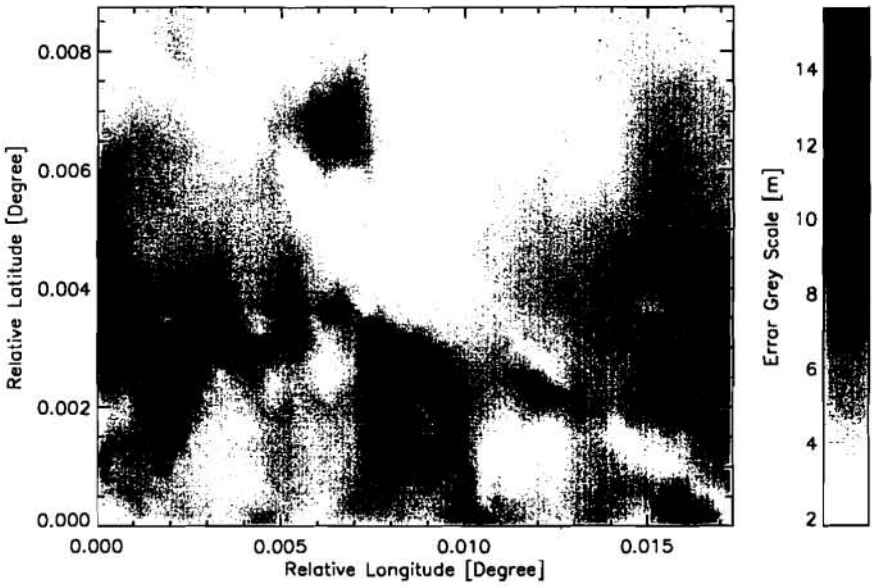
Figure 3.**a** is a replica of Fig. 24, while Figs 3.**b** and **c** are obtained from the same data of Fig. 24 using $N_f = 100$ and, respectively, $k_\sigma = 0.8$ and 0.5. Tables 9.**a** and **b** give the results of the two algorithms. Maps **b** and **c** in Fig. 3 are smoother



N_f	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	$\varepsilon_{\max}$ [m]	N_o	N_n	N_o %
30	0.5	0.1	5.96	4.72	29.45	2163	454	5.8

Figure 24 *The results of the application of the* NR-COMPRESS *algorithm to data from a deep-water real sea bottom. A combination of North-South and West-East track was used.*

a



b

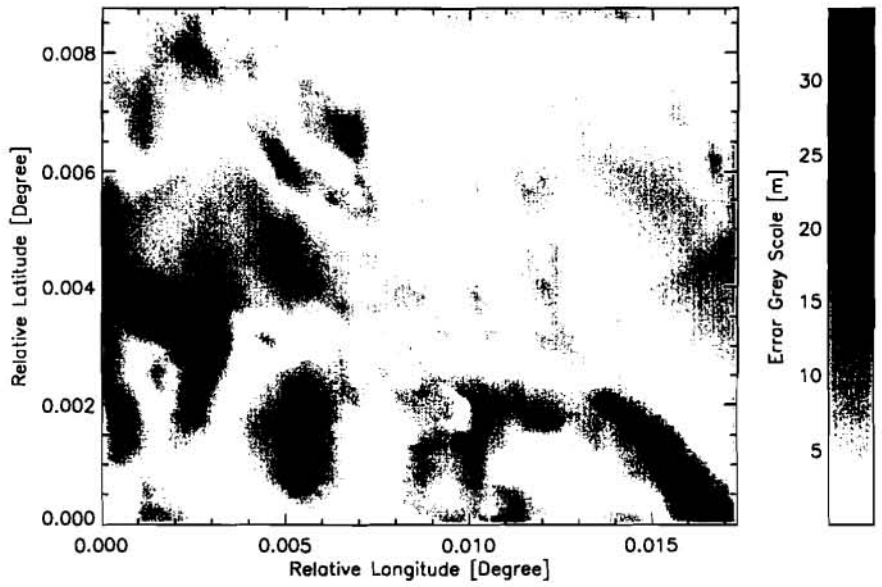


Figure 25 **a** *Plot of the local noise of the data set made of East-West and North-South tracks (deep water site off Sestri Levante);* **b** *plot of the difference between the maps obtained from the East-West and North-South tracks in the same site. Considering Table 4 and plot* **a** *the mapping error in* **b** *should be distributed in the same way with a maximum error of about twice the error in* **a**: *this is what effectively happened.*

a

N_f	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	εmax [m]	N_o	N_n	N_o %
100	0.5	0.1	6.08	4.79	31.8	1056	230	2.8

b

N_f	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	εmax [m]	N_o	N_n	N_o %
100	0.8	0.1	5.86	4.63	26.4	1271	281	3.4

Table 9 *The results of the application of the* NR-COMPRESS *algorithm with an increased low pass effect.*

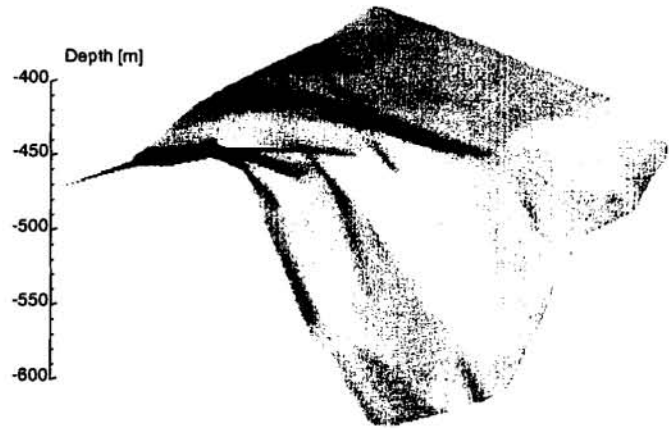
than the map obtained with $N_f = 30$ (Fig. 3.**a**) but, if the value of k_σ is not high enough, the error between the data point and the map increases. An example of what can happen in such a case is given in Fig. 26, where the map of the same sites are analyzed using a value of $k_\sigma = 0.1$ ($N_f = 30$). The resulting values of σ, $\overline{\varepsilon}$, and εmax show that the error increases dramatically. The number of nodes however becomes relatively low. It is possible to discern that a relatively flat zone visible in all preceding figures at the coordinate $(0.013, 0.0005)$, has almost disappeared. This is a clear indication that the value of k_σ was effectively too low.

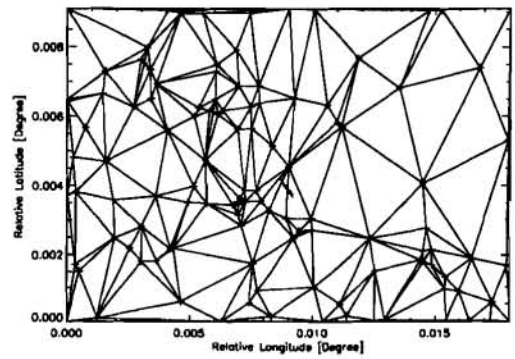
Medium size data file

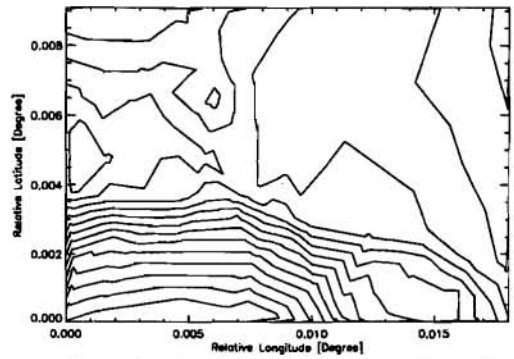
The second test on real data was performed on a flat shallow water area off Portofino (Italy) with a constant North-South slope of about 1 %. Two sets of tracks were available to test the algorithm: a set of five North-South tracks and a set of five East-West tracks. The tracks were overlapped of about 60 %, the mean water depth was 64 m and a part of the beams (the five outer beams on each side) was eliminated because their noise level were too high. Three results are shown from these data:

- the map resulting from the application of the algorithm using the combination of East-West and North-South tracks (Fig. 27): a low noise map is obtained without any hard filtering technique. The number of data points was 287,700 and the complete map was the results of the subdivision of the problem in 27 sub-maps. From Figure 27 it is possible to see that the NR-COMPRESS algorithm is trying to approximate the surface too accurately. Probably, fewer nodes should be used (lower value of k_σ).
- The error analysis (Fig. 28):
  - the plot of the local error calculated from map in Fig. 27
  - the plot of difference between the maps resulting from the analysis of the North-South and the East-West runs

  The maximum difference is coherent with the level of noise in the data. On the contrary with the data of the small dimension data file, there appears a







N_f	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	$\varepsilon_{\max}$ [m]	N_o	N_n	N_o %
100	0.1	0.1	6.81	5.29	31.5	601	118	1.6

Figure 26 *The results of the application of the* NR-COMPRESS *algorithm to the data from a deep-water sea bottom. A combination of North-South and West-East track was used.*

constant and low value of noise resulting in a constant error between the two maps.

- Once it is known, from Fig. 27, that the sea at that site is flat, it is possible to obtain a map with a small amount of points (Fig. 29). In such a case it is correct to fix the value of k_σ to a low value (0.05), while the number of fitting points is raised to a high value to obtain a better estimation of any map node (compare Fig. 27). Looking at the value of the errors in Fig. 29 it is possible to say that the map in Fig. 29 approximates well the map in Fig. 27.

The number of outliers (once the 5 most external beams were eliminated) was low (0.4–1.5 %).

Large size data file

The last test on real data was performed on a shallow to deep water area in the Black Sea. Only one direction tracks were available to test the algorithm, and they were slightly overlapped. No beams were eliminated from the acquired data. Because the number of data points was high (about 350,000), the map was obtained by subdividing the problem into 59 sub-maps.

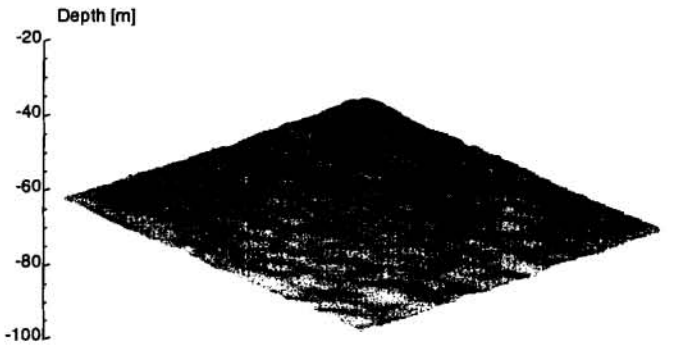
Figure 30 shows the seafloor surface obtained using $N_f = 30$. The number of nodes of the map is very high so another map was calculated using fewer nodes and a higher number of fitting points ($N_f = 150$ and $k_\sigma = 0.1$). The results are shown in Fig. 31: the number of nodes is halved and the error is not increased.

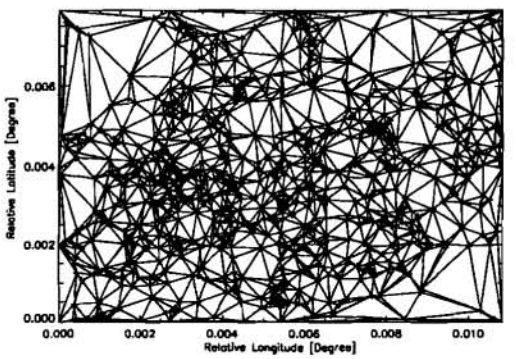
Finally, Fig. 32 shows the local noise obtained producing the map of Fig. 30: the black spot is a flag of possible problems during the acquisition of the data.

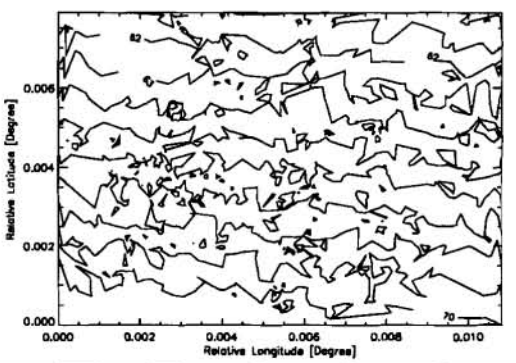
5.5 Run time examples

Run time tests were carried out to check the influence of some parameters on the computing cost of the global algorithm. The data used for the test are synthetic samples from a flat seafloor to which Gaussian noise with a standard deviation of 0.5 m was added. Figure 33 reports the result of the analysis.

Figure 33.**a** shows the time necessary for the algorithm to compute a map of 100 nodes from data sets consisting of an increasing number of samples. Figure 33.**b** gives the time necessary for the algorithm to compute a map with an increasing number of nodes using a data set of 10,000 samples. Figure 33.**c** and **d** gives the time necessary to compute a map of 100 nodes, using the data set. The varying







N_f	k_σ %	$k_{\sigma,o}$ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	$\varepsilon_{\max}$ [m]	N_o	N_n	N_o %
30	0.5	1	0.1	0.686	0.554	2.867	4395	1020	1.5

Figure 27 *The results of the application of the* NR-COMPRESS *algorithm to data from a shallow water sea bottom. The combination of North-South and West-East tracks was used.*

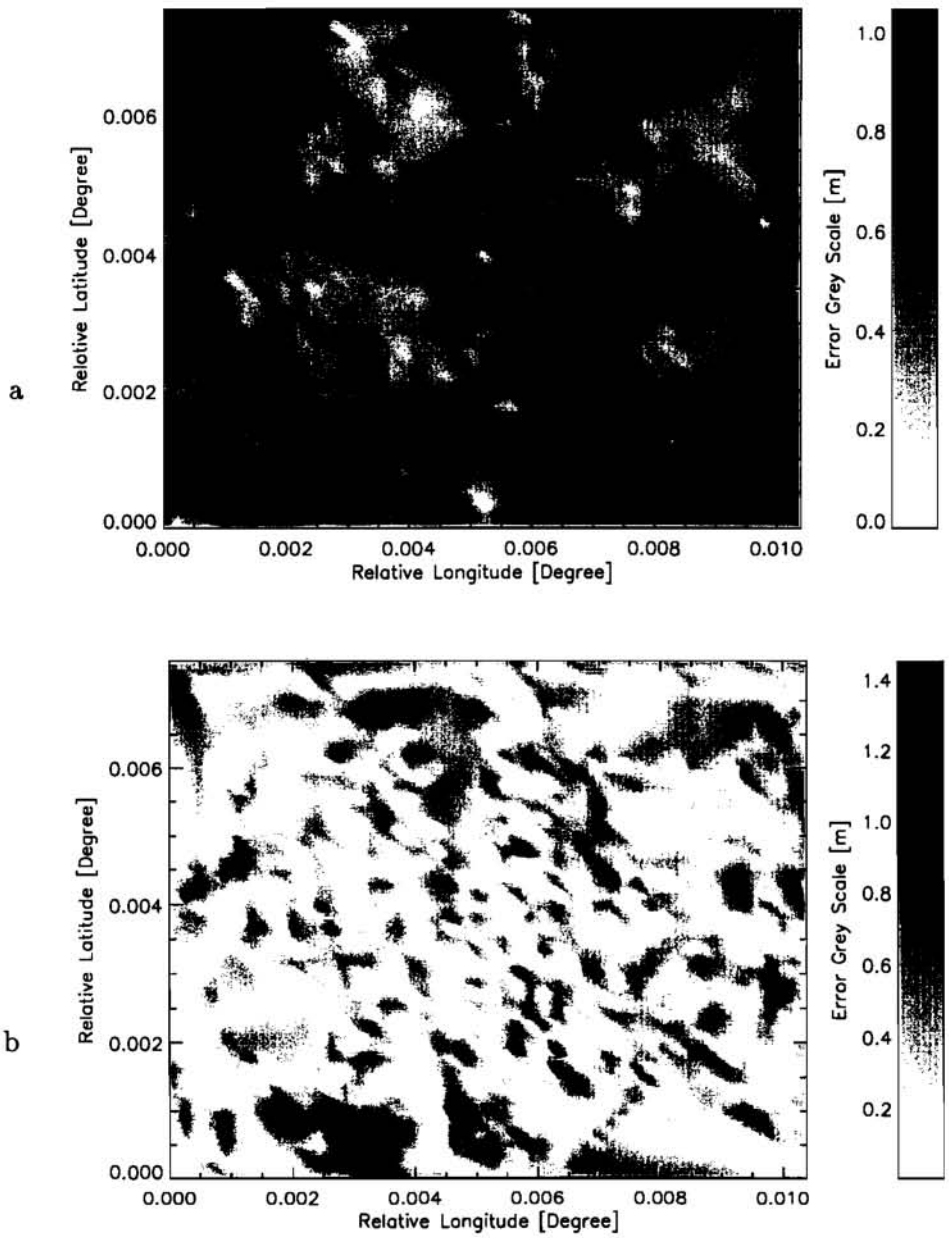
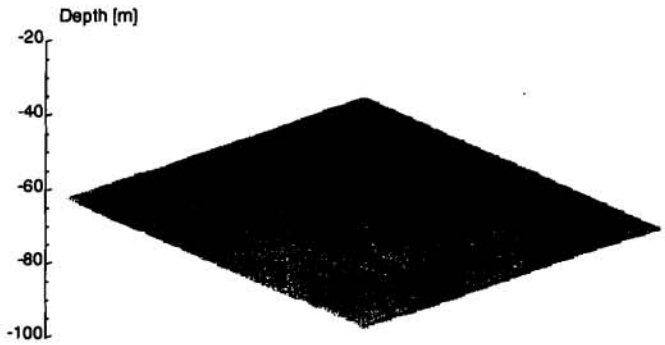
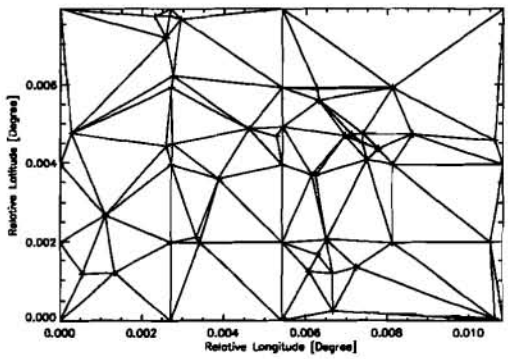
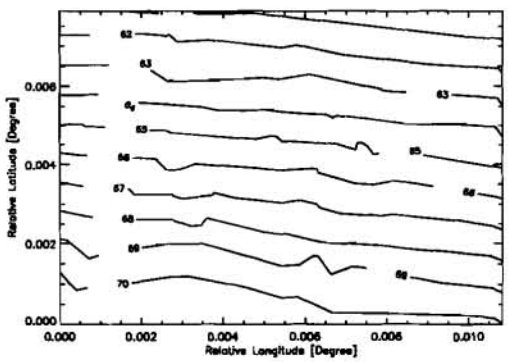


Figure 28 **a** *Plot of the local noise of the data set made of East-West and North-South tracks (shallow water site off Portofino);* **b** *plot of the difference between the maps obtained from the East-West and North-South tracks at the same site.*

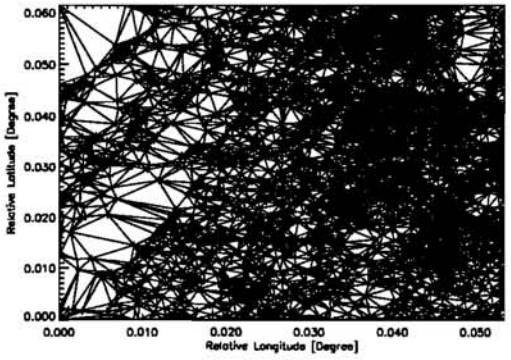


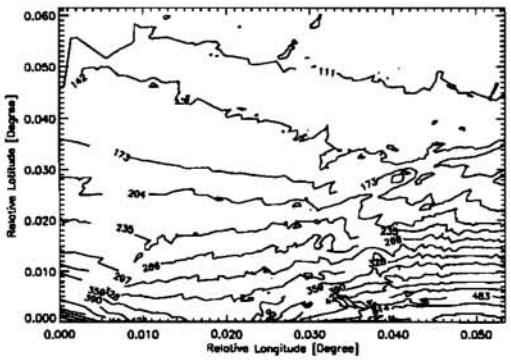




N_f	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	εmax [m]	N_o	N_n	N_o %
150	0.05	0.1	0.690	0.530	2.93	1272	63	0.4

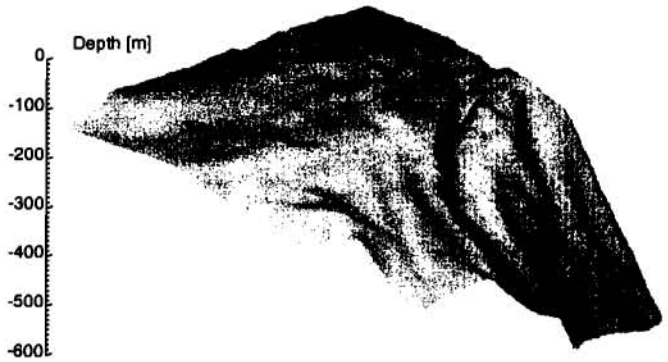
Figure 29 *The result of the application of* NR-COMPRESS *algorithm to a flat bottom.*

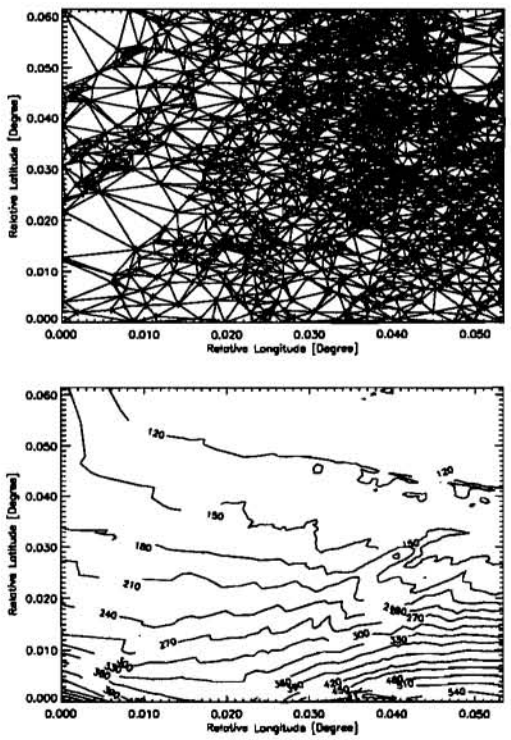




N_f	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	$\varepsilon_{\max}$ [m]	N_o	N_n	N_o %
30	0.5	0.1	1.92	1.38	34.46	25128	6773	7.2

Figure 30 *The results of the* NR-COMPRESS *algorithm applied to a very large data set.*





N_f	k_σ %	k_{MAD} %	σ [m]	$\overline{\varepsilon}$ [m]	$\varepsilon_{\max}$ [m]	N_o	N_n	N_o %
150	0.1	0.1	2.19	1.55	41.36	13585	2835	3.9

Figure 31 *The results of the* NR-COMPRESS *algorithm applied to a very large data set.*

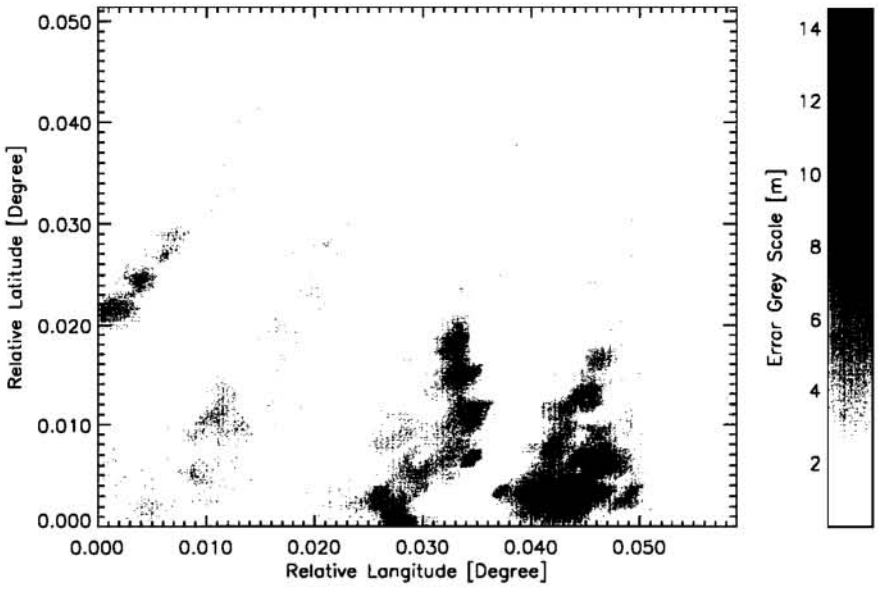


Figure 32 *Plot of the local noise of the data set acquired in the Black Sea.*

parameters were the number of outliers flagged by the algorithm and the number of fitting points.

In all the cases the time cost increases with the given parameter. In general, if the data set contains more samples, the map region will be described using more nodes and the number of outliers will be higher. In the best case, it will not be necessary to increase the number of nodes. As a consequence, it is possible to say that the computation time of the algorithm, as the number of data samples increases, is increased because the number of data samples is increased and because the number of outliers is increased. If the functions describing the time cost of the algorithm depending from the number of point of the data set (no outliers, 100 nodes in the map) were linear, the overall time cost should be at least quadratic. The function shown in Fig. 33.**a** indicates that the time cost increases more than linearly with the number of samples. To avoid the possibility of a quadratic (or even worse) time cost, the total map is divided in smaller sub-maps and the global algorithm is applied to the sub-maps using only a part of the complete data file. The results are added together to form a single map (Subsection 5.3.1): the efficiency of the algorithm is only minimally affected by this subdivision, but the possibility of a quadratic time consuming algorithm is avoided.

The algorithm computing time depends also on the number of fitting points. When the number of fitting points increases, the time necessary to select the nearest points increases concomitantly (see Fig. 33.**d**).

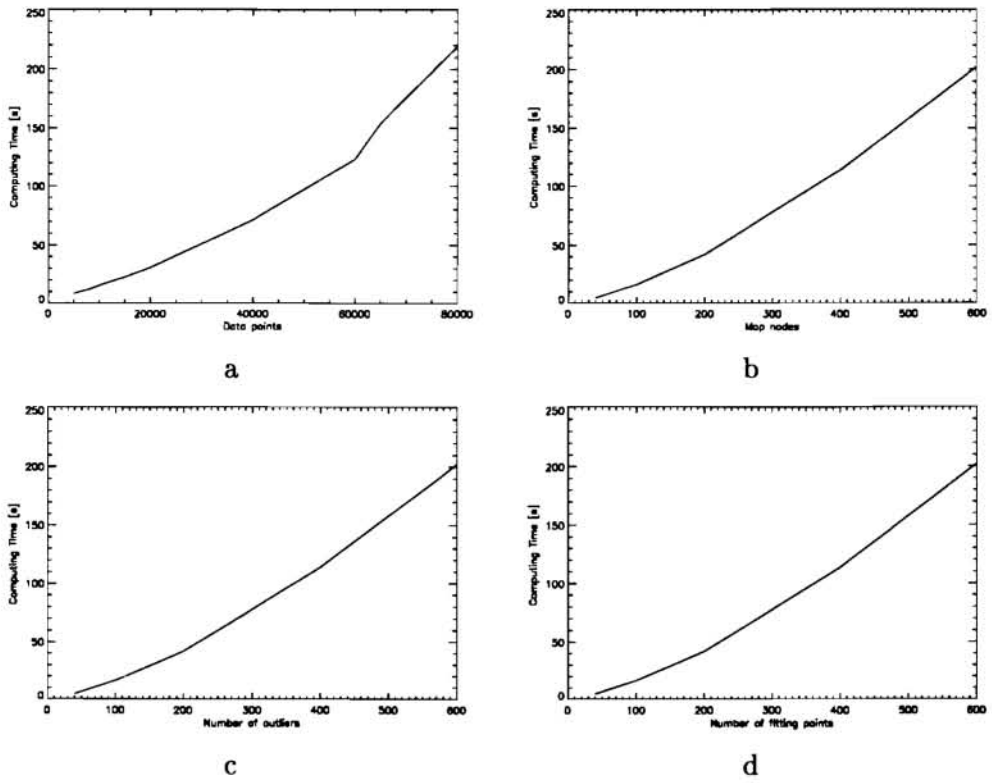


Figure 33 *Plot* **a** *shows the time necessary to the algorithm to compute a map of 50 nodes using an increasing number of data points ($N_f = 30$, N_n=100). Plot* **b** *gives the time necessary to the algorithm to compute a map with an increasing number of nodes using a fixed number of data points ($N_f = 30$, $n_p = 10,000$). Plot* **c** *gives the time necessary to compute a map when the number of outliers increases ($N_f = 30$, $n_p = 10000$, N_n=100). Finally, plot* **d** *give the time necessary to calculate a map when the number of fitting points increases ($n_p = 10000$, N_n=100). The times are calculated using a software implementation of* ***NR-COMPRESS*** *on a Digital AlphaStation 600^{TM} using* ***UNIX***© *operating system. The some execution times are obtained on a 133 MHz Pentium PC.*

The algorithm is slower on synthetic data point because the search time is higher due to the fact that the data are scattered. In a real bathymetric data file the data are placed in close proximity and the search algorithm finds the triangle where the point is in a faster time.

6
Conclusion

The purpose of this work was to elaborate and test an algorithm able to produce accurate maps from bathymetric data. The main characteristics of the algorithm are:

- Production of a triangulated map of uniform accuracy irrespective of seafloor features
- A map resolution which depends on the local data noise amplitude
- Automatic elimination of outliers
- Small computing cost even on large data files (more than 1 million points)

The algorithm was applied to synthetic data to understand its behaviour when parameters are changed. In particular, data mapping with and without outliers has been tested for both continuous and discontinuous synthetic seafloors. The algorithm has also been tested on real data. Some tests were performed on the triangulation engine behaviour, algorithm implementation speed, and fitting errors.

NR-COMPRESS can be used to reduce operator intervention during bathymetric data mapping. Raw bathymetric data are directly analyzed by the algorithm which automatically and robustly eliminates outliers and produces a map the parameters of which can be finely tuned by the user (number of nodes, smoothing, level of data cleaning, etc.). The algorithm has been implemented, exhaustively tested on synthetic and real data, and fully documented.

Report no. changed (Mar 2006): SR-285-UU

Document Data Sheet

Security Classification		Project No.
		033-2
Document Serial No.	**Date of Issue**	**Total Pages**
SR-285	October 1997	75 pp.

Author(s)

Canepa, G., Bergem, O.

Title

An approach to robust map generation from multibeam bathymetric data.

Abstract

During the last twenty years, many multibeam bathymetric sonars have been produced. The instrumentation is usually accompanied by a system able to produce a seafloor map from the sonar data. There are also several public domain systems, which can be used to obtain a map from the data. All these systems produce a gridded map that must be filtered in order to reproduce the original seafloor surface because of noise on the bathymetric data. Both the gridding and the filtering algorithms introduce a source of error that is not easily controlled. Moreover, gridded maps may use significant storage space for a small amount of information. Finally, at present no systematic solution with realistic run-time requirements has been given to the problem of identification and elimination of bad data (outliers).

We present here an algorithm able to fit bathymetric data and to automatically deal with outliers. The most important characteristics of the algorithm are: production of a triangulated map of uniform accuracy irrespective of seafloor features; a map resolution which depends on the local data noise amplitude; automatic elimination of outliers and low computing cost even on large data files.

The algorithm can be used to reduce the operator intervention during bathymetric data mapping. Raw bathymetric data are directly analyzed by the algorithm which automatically and robustly eliminates outliers and produces a map the parameters of which can be finely tuned by the user (number of nodes, smoothing, level of data cleaning, etc.). The algorithm has been implemented, exhaustively tested on synthetic and real data and fully documented.

Keywords

Seafloor map – map fitting – scattered data fitting

Issuing Organization

North Atlantic Treaty Organization
SACLANT Undersea Research Centre
Viale San Bartolomeo 400, 19138 La Spezia,
Italy

*[From N. America: SACLANTCEN
(New York) APO AE 09613]*

Tel: +39 (0)187 540 111
Fax:+39 (0)187 524 600

E-mail: library@saclantc.nato.int

Initial Distribution for SR 285

Ministries of Defence

DND Canada	10
CHOD Denmark	8
DGA France	8
MOD Germany	15
HNDGS Greece	12
MARISTAT Italy	9
MOD (Navy) Netherlands	12
NDRE Norway	10
MOD Portugal	5
MDN Spain	2
TDKK and DNHO Turkey	5
MOD UK	20
ONR USA	42

NATO Commands and Agencies

NAMILCOM	2
SACLANT	3
CINCEASTLANT/ COMNAVNORTHWEST	1
CINCIBERLANT	1
CINCWESTLANT	1
COMASWSTRIKFOR	1
COMMAIREASTLANT	1
COMSTRIKFLTANT	1
COMSUBACLANT	1
SACLANTREPEUR	1
SACEUR	2
CINCNORTHWEST	1
CINCSOUTH	1
COMEDCENT	1
COMMARAIRMED	1
COMNAVSOUTH	1
COMSTRIKFORSOUTH	1
COMSUBMED	1
NC3A	1
PAT	1

Scientific Committee of National Representatives

SCNR Belgium	1
SCNR Canada	1
SCNR Denmark	1
SCNR Germany	1
SCNR Greece	1
SCNR Italy	1
SCNR Netherlands	2
SCNR Norway	1
SCNR Portugal	1
SCNR Spain	1
SCNR Turkey	1
SCNR UK	1
SCNR USA	2
French Delegate	1
SECGEN Rep. SCNR	1
NAMILCOM Rep. SCNR	1

National Liaison Officers

NLO Canada	1
NLO Denmark	1
NLO Germany	1
NLO Italy	1
NLO Netherlands	1
NLO Spain	1
NLO UK	1
NLO USA	1

Sub-total	208
SACLANTCEN	30
Total	238