Terrain Modelling via Triangular Regular Networks

J. C. Agüero[a,†] **A. Feuer**[b] **G. C. Goodwin**[a]

[a] School of Electrical Engineering & Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia.

[b] Department of Electrical Engineering, Technion, Israel.

[†]jaguero@ecemail.newcastle.edu.au

Abstract: This Paper describes an application of Structured Total Least Squares method to the interpolation of terrain data. We briefly review the ideas of Least Squares, Total Least Squares, and Structured Total Least Squares. We illustrate the use of Structured Total Least Squares in the approximation of terrain surfaces using a novel discrete surface, the Triangular Regular Network. The Structured Total Least Squares algorithm allows us to deal with data corrupted by noise in every coordinate (x, y, z).

Keywords: *Errors in variables, Structured total least squares, Terrain modelling.*

1. INTRODUCTION

Least Squares (**LS**) is a very well known procedure for regressing one measured output onto a set of regressors. It is used in diverse fields including Biology, Economics, Geophysics, and Control Engineering. Underlying this procedure is the implicit assumption that the regressors are known exactly whereas the measured variable is corrupted by noise (Kendall and Stuart, 1967). **LS** can be interpreted as a Maximum Likelihood estimation procedure by hypothesing that the noise has a Gaussian distribution.

An interesting issue is that in some applications, the regressors themselves, may be measured with errors. One may still apply ordinary Least Squares. However, the resultant parameter estimates can be biased. The implications of bias are that the estimated parameter depends on the particular data set and hence there may be difficulties when it is used to predict the measured output under alternative conditions.

In this context, Total Least Squares (**TLS**) is a procedure that allows for errors in all variables leading to unbiased (i.e. data independent) estimates. A special form of **TLS** is the Structured Total Least Squares (**STLS**) algorithm which allows one to impose special constraints on the model structure. These algorithms can also be interpreted via Maximum Likelihood (**ML**) estimation (Kendall and Stuart, 1967).

To illustrate these ideas, we will utilize a simple example motivated by terrain modelling where the data has errors in all 3 coordinates i.e. in x, y, and z. We introduce a particular model structure (called a Triangular Regular Network **TRN**) and show how the model can be fitted to data using the **STLS** algorithm.

To fit the specific application into context , we note that land surveying data is becoming increasingly available in digital form. This fact has caused a rapid evolution of Geographic Information Systems (**GIS**) and has increased the interest in terrain modelling (Mascardi, 1998).

One challenging area of research in this area is to find a way to reduce the amount of data to be stored in **GIS**. This research stream is aimed at several end-user requirements, e.g. decreasing the amount of information to be transmitted over a communication line, saving disk storage space or loading of a terrain from disk into memory (De Floriani *et al.*, 2000). The key problem is that there are usually many redundant measurements, and it is thus possible to reduce the amount of data by selecting the most significant data without interfering with the accuracy of the underlying representation.

There are many existing algorithms that fit a discrete surface to terrain data. These algorithms assume that the measurements are exact, and then apply a data reduction method to select the most important data (Heckbert and Garland, 1997; Mascardi, 1998; Longley *et al.*, 1999). However, this error free measurement assumption is rarely met in practice. For this reason, these models are more useful for representing a surface than for interpolation purposes.

Indeed, for the class of problems in which we are interested, the data may have spatial dispersion at every coordinate. This feature is central to the method that we have developed. To account for different errors we represent each data point by a different matrix of errors. Thus, for each measurement, $\phi_i = (x_i, y_i, z_i)$, on the surface $\Omega : \Xi \to \mathbb{R}$, we assign a different error covariance matrix ($C_i \in \mathbb{R}^{3\times3}$). In this paper we describe a discrete surface to represent a given terrain, which we call a Triangular Regular Network (**TRN**). **TRN** has an underlying data reduction procedure, which is inherent in the way we estimate the height for every sample in the grid (see Figure 1). One can then envisage that there are many data points inside every triangle, but we only store the vertices of every triangle. A typical example is shown in Figure 4. One important feature of the data reduction procedure in **TRN** is that it can be applied to different representations (such as a Triangular Irregular Network), and in consequence, to improve the interpolation procedure for the representation chosen. We use **STLS** to fit the **TRN** to given data from terrain.

2. BACKGROUND TO LS, TLS, AND STLS

To introduce the ideas of **LS**, **TLS**, and **STLS** we consider the simple problem of regressing an "output" variable y on a regressor x. Thus, we want to fit a model of the form

$$y_i^o = \alpha x_i^o, \qquad i = 1, \ldots, N \tag{1}$$

(i) **LS:** We assume that x_i^o is measured exactly but y_i^o is measured in noise ($y_i = y_i^o + \Delta y_i$). In this case, we can estimate α via **LS** as

$$\hat{\alpha} = \frac{\sum_{i=1}^{N} x_i^o y_i}{\sum_{i=1}^{N} (x_i^o)^2} \tag{2}$$

However, we might ask what happens if y_i^o is measured exactly but x_i^o is measured in noise ($x_i = x_i^o + \Delta x_i$). This could be thought of as an alternative "regression" problem:

$$x_i^o = \frac{1}{\alpha} y_i^o = \beta y_i^o \tag{3}$$

Then the **LS** estimate of β is

$$\hat{\beta} = \frac{\sum_{i=1}^{N} x_i y_i^o}{\sum_{i=1}^{N} (y_i^o)^2} \tag{4}$$

(ii) **TLS:** Say now that both x and y are measured with noise. Then it is probably clear that neither results (2), nor (4) are appropiate. Instead, we would like some "combination" of those methods. This is provided by **TLS**. A generalization of the model (1) is to write

$$\phi_i^o \theta = 0 \tag{5}$$

where

$$\phi_i^o = [x_i^o, y_i^o] \tag{6}$$

Now, say that ϕ_i is a noisy measurement of ϕ_i^o with errors in both variables. If we assume that these errors are Gaussian distributed with zero mean and a 2×2 covariance matrix C_i, then the **ML** for this problem is

$$\hat{\theta}_{ML} = arg \max_{\theta} \; p(\{\phi_i\}|\{\hat{\phi}_i\}, \theta)$$

$$p(\{\phi_i\}|\{\hat{\phi}_i\}, \theta) = \prod_{i=1}^{N} \frac{exp\left[-\frac{1}{2}(\phi_i - \hat{\phi}_i)C_i^{-1}(\phi_i - \hat{\phi}_i)^T\right]}{(2\pi)|C_i|^{1/2}} \tag{7}$$

where

$$\hat{\phi}_i \theta = 0 \tag{8}$$

Remark 1. The above ideas can be generalized to 3 dimensions, as is the case in Terrain Modelling; in which case C_i is a 3×3 covariance matrix. Indeed, higher dimension problems are also possible. ▽▽▽

Optimizing (7) subject to (8) leads to the **TLS** algorithm. Also, if additional structure (in the form of linear constraints on the different entries of the vector $\hat{\phi}_i$) is added to the model, we get a so called **STLS** problem. Earlier work on this problem has been presented by De Moor (1993). This latter paper considers a special case in which the covariance matrices, C_i, are equal to the identity matrix. We follow the development of De Moor (1993) making the necessary extensions to allow us to cover general C_i matrices and to deal with the specific structure that we have (Goodwin *et al.*, 2001).

Algorithms to solve this problem can be found in (De Moor, 1993; Jiang, 1998; Lemmerling, 1999; Mastronardi *et al.*, 2000; Van Huffel and Lemmerlimg, 2002). It is also possible to directly maximize (7) subject to the constraints. This approach is called Constrained Total Least Squares (**CTLS**) (Van Huffel and Lemmerlimg, 2002).

3. DATA ASSUMPTIONS

Suppose that samples $\phi_i = \{(x_i, y_i, z_i), 1 \le n \le N\}$ (corrupted by noise) of a surface $\Omega : \Xi \subset \mathbb{R}^2 \to \mathbb{R}$ are collected.

Our aim is to find an approximation to this surface. We introduce the following assumptions:

- The surface Ω defined by $\Omega = \{(x, y, z)|z = z(x, y)\}$ can be modelled by a particular kind of surface called **TRN** (this concept will be explained below).
- The nominal value (error free) of each measurement ϕ_i is $\hat{\phi}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i)$ which belongs to the surface Ω.
- The measurement error $\{\tilde{\phi}_i = \phi_i - \hat{\phi}_i\}$ is a sequence of independent random variables, jointly Gaussian with zero mean and covariance matrix C_i.
- The domain Ξ of the surface Ω is the region defined by $\Xi = \{(x, y)|0 < x < \overline{X}, 0 < y < \overline{Y}\}$.
- Sufficient data exists in all the regions of Ξ.

Note that the first assumption allows us to transform this approximation problem into an estimation problem.

4. TRN SURFACE

The **TRN** that we propose is basically a regular grid representation that uses a grid such as the one shown in Figure 1 . Using every three neighboring grid points we can define a triangle to represent the surface. The rectangles are combined into four components. (See Figure 3). Notice that we have constrained the triangles so that the surface is continuous. Repeating this over all rectangles, a surface which can be viewed as an approximation of Ω, is generated.

4.1 The Grid

By way of illustration, we choose the grid shown in Figure 1. Using this grid, we next have to decide the number of rectangles to be used and the dimensions of the rectangles in order to cover all of the domain Ξ of the surface Ω.

The number of subdivisions of the edges satisfy $N_x = \overline{X}/\Delta x$ and $N_y = \overline{Y}/\Delta y$. Also, the total number of samples represented by the grid is $N = 2N_xN_y + N_y + N_x + 1$. The samples are enumerated from left to right, row by row and, bottom to top, as illustrated in Figure 1.

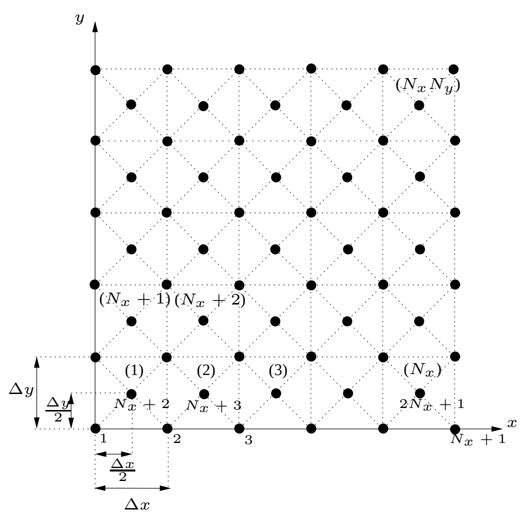

Fig. 1. Finite Sampling Grid

Given the heights, $Z_n = z\left(X_n, Y_n\right)$ at the nth sample we use a two dimensional plane to approximate the surface and refer to this approximation in the sequel as *Triangular Regular Network*. Specifically, every three neighboring samples, as described in Figure 1, define a plane so that the surface approximation consists of connected triangular plane sections as demonstrated in Figure 3.

For reasons which will become clearer later, we also enumerate the rectangles created by the neighboring sample points which are integer multiples of the sampling intervals. Here too, we go from left to right, bottom to top and have $K = N_xN_y$ rectangles (see the numbers in parenthesis in Figure 1). Consider the kth rectangle, as described in Figure (2). Then, as we can observe from the figure, the samples are enumerated again according to the rectangle(s) they belong to. Clearly, any given sample can belong to, at most, four rectangles. We wish to identify the five samples of the kth rectangle in the previous enumeration. So, given $k = 1, 2..., K$ define

$$k_y = \text{floor}(k/N_x) + 1$$
$$k_x = k - (k_y - 1)N_x$$

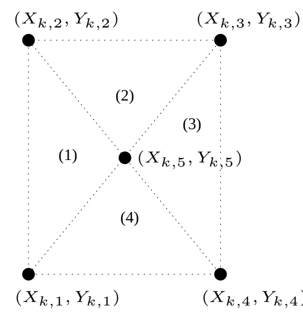

Fig. 2. The k-th rectangle of samples.

Then we have the following relationships

$$\begin{aligned}
(X_{k,1}, Y_{k,1}) &= (X_{n-N_x-1}, Y_{n-N_x-1}) \\
(X_{k,2}, Y_{k,2}) &= (X_{n+N_x}, Y_{n+N_x}) \\
(X_{k,3}, Y_{k,3}) &= (X_{n+N_x+1}, Y_{n+N_x+1}) \\
(X_{k,4}, Y_{k,4}) &= (X_{n-N_x}, Y_{n-N_x}) \\
(X_{k,5}, Y_{k,5}) &= (X_n, Y_n)
\end{aligned} \tag{9}$$

where

$$n = (2N_x + 1)\, k_y + k_x - N_x$$

and

$$\begin{aligned}
X_{k,1} &= X_{k,2} = \Delta x\,(k_x - 1) \\
X_{k,3} &= X_{k,4} = \Delta x k_x \\
X_{k,5} &= \Delta x \left(k_x - \frac{1}{2}\right) \\
Y_{k,1} &= Y_{k,4} = \Delta y\,(k_y - 1) \\
Y_{k,2} &= Y_{k,3} = \Delta y k_y \\
Y_{k,5} &= \Delta y \left(k_y - \frac{1}{2}\right)
\end{aligned} \tag{10}$$

Having the heights at the rectangle samples, $Z_{k,l} = z\left(X_{k,l}, Y_{k,l}\right)$, $l = 1, 2, 3, 4, 5$ the surface section above the rectangle is approximated as shown in Figure 3.

4.2 TRN Parameterization

Let us assume we have I data points each consisting of three values $\phi_i = (x_i, y_i, z_i)$ measured relative to

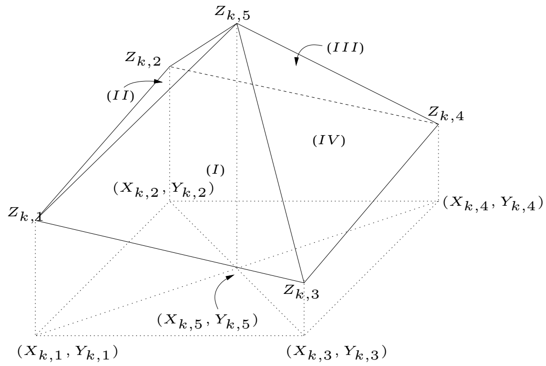

Fig. 3. TRN at rectangle k

the same origin. We hypothesise that these data points are of the form

$$(x_i, y_i, z_i) = (\hat{x_i}, \hat{y_i}, \hat{z_i}) + (\Delta x_i, \Delta y_i, \Delta z_i)$$

where $\tilde{\phi_i} = (\Delta x_i, \Delta y_i, \Delta z_i)$ are the error terms in each data point, assumed jointly Gaussian with zero mean and covariance matrix $C_i \in \mathbb{R}^{3\times 3}$, and $\hat{\phi_i} = (\hat{x_i}, \hat{y_i}, \hat{z_i})$ is the nominal value on the TRN surface.

Typically, the data collected from landscape surveying is irregular and contains errors. Furthermore, since it is likely that the data comes from a number of different surveys, it may represent different accuracies for latitude, longitude and height measurements. Moreover, the accuracies may differ from location to location (i.e. different data points). Hence, we will allow for the possibility of different covariance matrices at different data points.

To formulate the above surface fitting problem as an optimization problem, let us assume that the data is sorted according to (x_i, y_i) belonging to triangle l in rectangle k with $I_{k,l}$ data points in each such set and $I = \sum_{k=1}^{K}\sum_{l=1}^{4} I_{k,l}$. Algorithms, such as those presented in (Goodwin *et al.*, 2001), can be used to determine to which rectangle k, and triangle l that the data (x_i, y_i, z_i) belongs. A feature of this problem, which makes it difficult, is that there is a lot of structure in the way that the surface is parameterized. Actually, it is this structure that leads to the need for **STLS**. We will capture this structure by parameterizing the problem via a set of basis functions. Specifically, let us define:

$$\begin{aligned} B_x^i &= e_i^I \left(e_{4i-3}^{4I}\right)^T \in \mathbb{R}^{I\times 4I} \\ B_y^i &= e_i^I \left(e_{4i-2}^{4I}\right)^T \in \mathbb{R}^{I\times 4I} \\ B_z^i &= e_i^I \left(e_{4i-1}^{4I}\right)^T \in \mathbb{R}^{I\times 4I} \\ B_0^i &= e_i^I \left(e_{4i}^{4I}\right)^T \in \mathbb{R}^{I\times 4I} \end{aligned}$$

where e_i^I is the ith column of the I dimensional identity matrix.

Using these basis functions we can succinctly parameterize the problem using

$$\begin{aligned} \Phi &= \sum_{i=1}^{I} \left(x_i B_x^i + y_i B_y^i + z_i B_z^i + B_0^i\right) \in \mathbb{R}^{I\times 4I} \\ &= diag([\,\phi_1 \;\; 1\,], [\,\phi_2 \;\; 1\,], \cdots, [\,\phi_I \;\; 1\,]) \end{aligned} \tag{11}$$

Next we define a vector $\underline{\theta} \in \mathbb{R}^{4I}$ whose entries are a set of parameters (a_i, b_i, c, d_i) for every data point $i = 1..I$.

$$\underline{\theta} = \left[\, a_1 \;\; b_1 \;\; c \;\; d_1 \;\; a_2 \;\; b_2 \;\; c \;\; d_2 \;\; \cdots \;\; a_I \;\; b_I \;\; c \;\; d_I \,\right]^T \tag{12}$$

Notice that the vector of parameters contains the coefficients of all the planes in the **TRN** surface. In order to obtain a continuous surface some constraints should be added to this problem. Also, note that the parameter c is the same for every plane. This constraint will be necessary to develop Lemma 1.

If we replace the entries of Φ by their nominal values, we will call the resultant matrix $\hat{\Phi}$. Now, since each data point lies in a plane, we can describe these planes by the relationship $\hat{\Phi}\underline{\theta} = 0$. Of course, it will be generally be the case that several data points will lie in the same plane, so many of the entries in $\underline{\theta}$ will be the same. One way to reduce the number of parameters to estimate is to define a new vector compounded by the vertex height of every triangle.

4.3 TRN Reparameterization

For future use in the reparameterization procedure, we define the following matrices for any $k = 1, 2, ..., K$

$$D_{k,1} = \begin{bmatrix} X_{k,1} & Y_{k,1} & 0 & 1 \\ X_{k,2} & Y_{k,2} & 0 & 1 \\ 0 & 0 & 1 & 0 \\ X_{k,5} & Y_{k,5} & 0 & 1 \end{bmatrix}; \quad D_{k,2} = \begin{bmatrix} X_{k,2} & Y_{k,2} & 0 & 1 \\ X_{k,3} & Y_{k,3} & 0 & 1 \\ 0 & 0 & 1 & 0 \\ X_{k,5} & Y_{k,5} & 0 & 1 \end{bmatrix}$$

$$D_{k,3} = \begin{bmatrix} X_{k,3} & Y_{k,3} & 0 & 1 \\ X_{k,4} & Y_{k,4} & 0 & 1 \\ 0 & 0 & 1 & 0 \\ X_{k,5} & Y_{k,5} & 0 & 1 \end{bmatrix}; \quad D_{k,4} = \begin{bmatrix} X_{k,4} & Y_{k,4} & 0 & 1 \\ X_{k,1} & Y_{k,1} & 0 & 1 \\ 0 & 0 & 1 & 0 \\ X_{k,5} & Y_{k,5} & 0 & 1 \end{bmatrix}$$

$$E_{k,1} = \begin{bmatrix} \left(e_{n-N_x-1}^{N+1}\right)^T \\ \left(e_{n+N_x}^{N+1}\right)^T \\ \left(e_{N+1}^{N+1}\right)^T \\ \left(e_{n}^{N+1}\right)^T \end{bmatrix}; \quad E_{k,2} = \begin{bmatrix} \left(e_{n+N_x}^{N+1}\right)^T \\ \left(e_{n+N_x+1}^{N+1}\right)^T \\ \left(e_{N+1}^{N+1}\right)^T \\ \left(e_{n}^{N+1}\right)^T \end{bmatrix}$$

$$E_{k,3} = \begin{bmatrix} \left(e_{n+N_x+1}^{N+1}\right)^T \\ \left(e_{n-N_x}^{N+1}\right)^T \\ \left(e_{N+1}^{N+1}\right)^T \\ \left(e_{n}^{N+1}\right)^T \end{bmatrix}; \quad E_{k,4} = \begin{bmatrix} \left(e_{n-N_x}^{N+1}\right)^T \\ \left(e_{n-N_x-1}^{N+1}\right)^T \\ \left(e_{N+1}^{N+1}\right)^T \\ \left(e_{n}^{N+1}\right)^T \end{bmatrix}$$

We also define

$$L_{k,l} = \begin{bmatrix} D_{k,l}^{-1} E_{k,l} \\ D_{k,l}^{-1} E_{k,l} \\ \vdots \\ D_{k,l}^{-1} E_{k,l} \end{bmatrix} \in \mathbb{R}^{4I_{k,l}\times(N+1)} \tag{13}$$

and finally

$$L = \begin{bmatrix} L_{1,1} \\ \vdots \\ L_{1,4} \\ \vdots \\ L_{K,1} \\ \vdots \\ L_{K,4} \end{bmatrix} \in \mathbb{R}^{4I\times(N+1)}$$

Note that, in equation (13), we use the inverse of the matrices $D_{k,l}$. This is possible because all of these matrices are non-singular. In fact, if we use the definitions of the sample grid (equation (10)), we obtain that the determinant of the matrices $D_{k,l}$ is non zero, specifically,

$$|D_{k,l}| = -\frac{\Delta x \Delta y}{2};\; k = 1, 2, \cdots, K;\;\; l = 1, 2, 3, 4$$

With the above definitions, we can reparameterize all the parameters in (12) in terms of the grid point heights. This is formally established in the following Lemma:

Lemma 1. Consider the vector

$$\theta = \left[-cZ_1 \; -cZ_2 \; \ldots \; -cZ_N \;\; c \right]^T \tag{14}$$

and define $\left[a_{k,l} \;\; b_{k,l} \;\; c \;\; d_{k,l} \right]^T = D_{k,l}^{-1} E_{k,l}\theta$. Then, the plane defined by $a_{k,l}x + b_{k,l}y + cz + d_{k,l} = 0$ is the lth triangular plane section in the kth rectangle.

Proof. We will prove the result for $l = 1$ and any k. For $l = 2, 3, 4$ the proof is similar. The $(k, 1)$ th triangular plane section is the plane passing through the three points $(X_{k,1}, Y_{k,1}, Z_{k,1})$, $(X_{k,2}, Y_{k,2}, Z_{k,2})$, $(X_{k,5}, Y_{k,5}, Z_{k,5})$. Let us then test the result. Specifically, we have

$$\begin{bmatrix} X_{k,1} & Y_{k,1} & Z_{k,1} & 1 \\ X_{k,2} & Y_{k,2} & Z_{k,2} & 1 \\ X_{k,5} & Y_{k,5} & Z_{k,5} & 1 \end{bmatrix} \begin{bmatrix} a_{k,1} \\ b_{k,1} \\ c \\ d_{k,1} \end{bmatrix}$$

$$= \begin{bmatrix} X_{k,1} & Y_{k,1} & Z_{k,1} & 1 \\ X_{k,2} & Y_{k,2} & Z_{k,2} & 1 \\ X_{k,5} & Y_{k,5} & Z_{k,5} & 1 \end{bmatrix} D_{k,1}^{-1} E_{k,1}\theta$$

$$= \begin{bmatrix} X_{k,1} & Y_{k,1} & Z_{k,1} & 1 \\ X_{k,2} & Y_{k,2} & Z_{k,2} & 1 \\ X_{k,5} & Y_{k,5} & Z_{k,5} & 1 \end{bmatrix} D_{k,1}^{-1} \begin{bmatrix} -cZ_{n-N_x-1} \\ -cZ_{n+N_x} \\ c \\ -cZ_n \end{bmatrix}$$

Recalling the relationships between the two indexing system for the sampling points (equation (9)) we have

$$\begin{bmatrix} -cZ_{n-N_x-1} \\ -cZ_{n+N_x} \\ c \\ -cZ_n \end{bmatrix} = \begin{bmatrix} -cZ_{k,1} \\ -cZ_{k,2} \\ c \\ -cZ_{k,5} \end{bmatrix} \tag{15}$$

Additionally,

$$\begin{bmatrix} X_{k,1} & Y_{k,1} & Z_{k,1} & 1 \\ X_{k,2} & Y_{k,2} & Z_{k,2} & 1 \\ X_{k,5} & Y_{k,5} & Z_{k,5} & 1 \end{bmatrix} D_{k,1}^{-1}$$

$$= \left(\begin{bmatrix} X_{k,1} & Y_{k,1} & 0 & 1 \\ X_{k,2} & Y_{k,2} & 0 & 1 \\ X_{k,5} & Y_{k,5} & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & Z_{k,1} & 0 \\ 0 & 0 & Z_{k,2} & 0 \\ 0 & 0 & Z_{k,5} & 0 \end{bmatrix} \right) D_{k,1}^{-1}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & Z_{k,1} & 0 \\ 0 & 0 & Z_{k,2} & 0 \\ 0 & 0 & Z_{k,5} & 0 \end{bmatrix} \tag{16}$$

Then, the product of the last two terms ((15) and (16)) yields:

$$\begin{bmatrix} X_{k,1} & Y_{k,1} & Z_{k,1} & 1 \\ X_{k,2} & Y_{k,2} & Z_{k,2} & 1 \\ X_{k,5} & Y_{k,5} & Z_{k,5} & 1 \end{bmatrix} \begin{bmatrix} a_{k,1} \\ b_{k,1} \\ c \\ d_{k,1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

which completes the proof. ■

Using Lemma 1 we obtain

$$\underline{\theta} = L\theta \tag{17}$$

which allows to reparameterize the **TRN** surface in terms of the vertex heigth.

Remark 2. Note that this Lemma can be generalized for other kinds of representation with a given grid.

▽▽▽

5. MAXIMUM LIKELIHOOD

We utilize the general idea described above but with $3D$ data and with the particular model structure briefly explained above.

5.1 Fitting the TRN via optimization

If the covariance matrices C_i are known (or approximated) instead of maximize (7), we can minimize (18) subject to the same constraint, i.e.

$$(\underline{\theta}_{ML}, \{\hat{\phi}_i\}_{ML}) = \underset{\theta, \underline{\theta}, \{\hat{\phi}_i\}}{arg\,min} \sum_{i=1}^{I} \left(\phi_i - \hat{\phi}_i\right) C_i^{-1} \left(\phi_i - \hat{\phi}_i\right)^T$$

subject to:

$$\hat{\Phi}\underline{\theta} = 0, \;\; \underline{\theta} = L\theta \tag{18}$$

In particular, we optimize (18) with respect to $\left\{\hat{\phi}_i\right\}_{i=1}^{I}$, $\underline{\theta}$ and θ, where $\hat{\Phi}$ satisfies the structural constraints imposed by eqn. (11), namely

$$\hat{\Phi} = \sum_{i=1}^{I} \left(\hat{x}_i B_x^i + \hat{y}_i B_y^i + \hat{z}_i B_z^i + B_0^i \right)$$

Note that with this structural constraint, Lemma 1 implies that $\{(\hat{x}_i, \hat{y}_i, \hat{z}_i)\}_{i=1}^{I}$ are all on the TRN surface as determined by θ and eqn. (14).

It is clear that the vector of parameters, θ, is not unique, since we can scale all parameters by a constant. Thus, we can add $||\theta||_2 = 1$ as a new constraint to this optimization problem.

6. AN ILLUSTRATIVE EXAMPLE

Synthetic elevation data has been generated by means of evaluating a function given by:

$$\bar{z} = 2sin^2(2\bar{x} + \bar{y})$$

Then noise was added to the coordinates of every data point as follows: $x = \bar{x} + \epsilon_x$, $y = \bar{y} + \epsilon_y$, $z = \bar{z} + \epsilon_z$

where ϵ_x, ϵ_y, ϵ_z are uniform i.i.d. in the interval $[-0.05, 0.05]$.

Note that, the function used to generate the data is not a **TRN** since it is not composed of triangles. Thus, our procedure can be thought of as solving an approximation problem where we aim to find the "closest" **TRN** surface to the given data. Of course, this corresponds to practical reality since all **DEM** is an approximation problem. However, when the data is generated by a **TRN** (corrupted by noise) this problem is an estimation problem and the Maximum Likelihood procedure asymptotically achieves the Cramer Rao lower bound, which is a measure of the estimator efficiency (Kendall and Stuart, 1967).

In Figure 4 we can see the data, and the TRN approximation obtained by **CTLS**. The maximum deviation (in height) of the **TRN** from the "true" surface is 0.15 when the true height is 1.94.

In this example, the number of data points was 441 each comprising 3 measurements. This gives 1323 scalar variables. The number of samples on the **TRN** grid is 41, each comprising 3 coordinates. However, the x, and y components lie in a regular grid which does not need to be stored. Hence, the end result of the **TRN** fitting is that the original 1323 degrees of freedom have been replaced by 41 degrees of freedom to describe the surface.

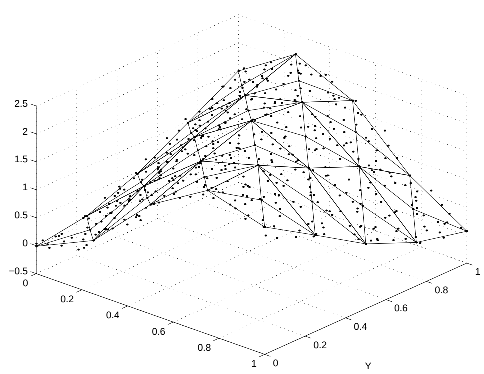

Fig. 4. Digital Elevation Modelling.

7. CONCLUSIONS

This paper has described **LS**, **TLS**, and **STLS** motivated by the Terrain modelling problem. This procedure has been applied to one particular kind of surface which we call Triangular Regular Network. However, the same procedure can be applied to other kinds of representation over a given grid, e.g. non-uniform grids. The benefits of applying the **STLS** approach to the terrain modelling problem is that it is possible to deal with errors (in x, y, and z) in all the measurements. Since **TRN** is basically a regular grid representation, they share the same advantages and disadvantages. It is possible to apply the ideas of errors in variables (**TLS**, **STLS**) to more complex surface representations.

8. REFERENCES

Agarwal, P., M. de Berg, P. Bose, K. Dobrint, M. van Kreveld, M. Overmars, M. de Groot, T. Roos, J. Snoeyink and S. Yu. The complexity of rivers in triangulated terrains. *Proc. 8th Canad. Conf. Comput. Geom.* pp. 325–330, 1996.

De Floriani, L., P. Magillo and E. Puppo. Compressing triangulated irregular network. *Geoinformatica*, 2000.

De Moor, B. Structured total least squares and l_2 approximation problems. *Linear Algebra and its Application* **188,189**, 163–205, 1993.

Garland, M. and P. Heckbert. Fast polygonal approximation of terrains and height fields. *Technical Report, Carnegie Mellon University*, 1995.

Goodwin, G. C., J. C. Agüero and A. Feuer. An overview of digital elevation modelling. *Technical report,The University of Newcastle*, 2001.

Heckbert, P. and M. Garland. Survey of polygonal surface simplification algorithms. *Technical report C.S. Dept. Carnegie Melon University*, 1997.

Heywood, I., S. Cornelius and S. Carver. An introduction to geographical information systems. *Longman*, 1998.

Jiang, E. Information Retrieval and Filtering Using the Riemannian SVD. PhD thesis. The University of Tennessee at Knoxville. USA, 1998.

Kendall, M. and A. Stuart. *The advanced theory of statistics*. Vol. 2. C. Griffin. London, 1967.

Lemmerling, P. Stuctured Total Least Squares: Analysis, Algorithms and Applications. PhD thesis. Katholieke Universiteit Leuven. Belgium, 1999.

Longley, P., M. Goodchild, D. Maguire and D. Rhind. *Geographical information systems*. Vol. 1. John Wiley, 1999.

Mascardi, V. Extraction of significant terrain features from RSG and TIN: A survey, 1998.

Mastronardi, N., P. Lemmerling and S. Van Huffel. Fast structured total least squares algorithm for solving the basic deconvolution problem. *SIAM, Journal on Matrix Analysis and Applications* **22,2**, 533–553, 2000.

Van Huffel, S. and P. Lemmerlimg. *Errors in Variables Modelling: Analysis, Algorithms and Applications*. Kluwer Academic Publishers. Philadelphia, 2002.