TECHNICAL REPORT NO. 14

Data Assimilation for Updates of Digital Terrain Models

Thomas Knudsen

Danish Ministry of the Environment
National Survey and Cadastre

Thomas Knudsen
Data Assimilation for Updates of Digital Terrain Models

National Survey and Cadastre—Denmark, technical report series number 14
ISBN 978-78-92107-40-4
Technical Report
Published 2011-12-18
This report is available from http://www.kms.dk

Contents

Chapter 1

Introduction

1.1 DK-DEM

DK-DEM, the national Danish digital elevation model, consists of three primary products (Dalå et al., 2009):

1. A gridded digital terrain model (DTM)
2. A gridded digital surface model (DSM)
3. A set of terrain contour curves

Supplementary products include a digital terrain model with bridges included (for orthophoto production), but specifically *not* the raw data (i.e. the *point cloud*) used for computing the grids.

The gridded models (DSM and DTM) have a grid ground sample distance (GSD) of 1.6 m (i.e. $\approx$ 0.4 point/m^2) and were based on airborne LiDAR observations with a similar mean density.

The LiDAR data sets used were collected by the companies BlomInfo and Scankort in the time frame 2005–2007. So while DK-DEM was a big improvement compared to what was available prior to its introduction in 2009, it was already at that time slightly dated.

For many purposes, DK-DEM is still perfectly adequate, but for other purposes (most obviously the ones exceeding the original scope of DK-DEM) it has shown necessary to collect new data. These data, typically collected by public institutions with special tasks in limited areas, could be put to good use in the process of updating DK-DEM. The aim of this report is to take some steps towards a practical realization of just that.

1.2 Scope and aim

Since the original point cloud data sets behind the DK-DEM grids are not available, all updates must be carried out by combining the existing grids with new data, which may be either grids or point cloud data.

A few years ago, Joachim Höhle (Höhle, 2009) presented a very systematic approach to DK-DEM updates, suggesting the use of photogrammetric methods, systematically utilizing aerial photos already collected for mapping, to generate new height grids with an accuracy approaching that of LiDAR. These height grids will then fully replace the existing LiDAR based grids.

In a sense, this report takes the opposite approach to Höhle: rather than *generating height data* from sytematically collected *data of opportunity*, we are aiming for *utilizing existing height data*, collected (in potentially non-coordinated or even non-systematical, ways), for *opportunistic updating* of the existing height model.

Combining the two approaches, we may be able to put all available data into optimum use and, not the least, gain improved insight and confidence in the precision and accuracy of the updated model.

It is not just in its approach, but also in its scope and aim, the work presented here differs from that of Höhle (2009): Höhle presents and evaluates a practical study based primarily on the use of commercial implementations of algorithms and methods that have been developed through more than a decade of research by Höhle and his colleagues in international surveying and photogrammetry laboratories.

This report, on the other hand, presents work based on methods for gravity data analysis, originally developed and used since the 1960s by the physical geodesy community. The application of these methods to elevation data is unconventional, so in order to further develop the methods and ideas within a *controlled framework*, all the work presented here is based on simulated data (albeit data derived from a real DTM covering a 1 km $\times$ 1 km test site).

Since the results are encouraging and the implementation straightforward, I hope, in future work, to be able to follow Höhle's example of evaluating full scale experiments based on real world data (cf. section 5.1).

1.3 Data assimilation

Data assimilation is a term primarily used in numerical weather prediction, where it covers the process of combining newly arrived atmospheric observations with the current model forecast, in preparation for the next forecast cycle. Apparently this has nothing to do with height models, so how did data assimilation make it into the title of this report? For two reaons, really:

First, *data assimilation* in the form of optimum interpolation is an idea for which the time was ripe in the 1960s: Largely similar approaches: *optimum interpolation* in dynamic meteorology, *kriging* in mining engineering, and *least squares collocation* in geodesy[1], were published within a few years time. So the term *data assimilation* hints at all these ways of integrating spatial and/or spatio-temporal observations in a way that optimizes the recovery of the physical signal (modelled as a stochastic process) behind the data.

Second, *data assimilation*, ethymologically speaking, hints at a process of "making similar" (*to assimilate*). And making data similar is exactly what we need when taking the opportunistic-synergistic approach of making as much use as possible of whatever data that happens to come our way.

Getting better data for one area does not in any way make it possible for us to say much new about a neighbouring area. Hence, we must make existing and new data fit together – make them similar.

Sometimes we may even find that the long wavelength accuracy of the existing model may be much better than the new data, while the new data still have much higher accuracy at short wavelengths. [2] This is especially the case for corridor mapping data sets, where long, essentially one dimensional, areas around elongated features (power lines, railroads, highways) are mapped using just one flight line.

Hence the term data assimilation – to hint at a two way process that in one direction aims at updating an existing model by incorporating new data (as in a human cognitive process), and in the other direction makes the new data more similar to the existing, by propagating a splash of prejudice/prior knowledge/existing state, to the observations before incorporation.

Chapter 2

Prerequisites

2.1 Introductory remarks

This chapter presents some of the more important mathematical and geophysical prerequisites for the experiments presented in chapter 4. The presentation is neither extensive nor complete, it is simply a brief reminder that may be safely skippped by readers well informed in these matters.

But before skipping on to the experiments, the reader is encouraged to consider a few points about updating of height models which may put the experiments in a different perspective:

1. If we have new and "perfect" data for an area that reveals a bias in the heights of the old model, what should we do at the border between old and new data?
    - Introduce the new data directly in the model, and live with the step inevitably introduced on the border between the biased and the unbiased data?
    - Arbitrarily modify the old data near the border to get a smooth transition?
    - Gradually introduce the bias of the old data into the new data, as we get closer to the border?
2. The difference between updating and *bringing up to date*: what should be done in areas where we have new data, but also know that these data are already outdated by even newer developments?
3. New observations having error bars falling entirely within the error bars of the old model may actually not bring any new information to the table. Do we update the models anyway (e.g. to get a local reduction of the error bars)?
4. What should we do if we get new, but technically inferior data (e.g. more noisy and/or lower resolution than the existing model) for an area where we *know* that changes have happened.

Any competent practitioner will have good answers or opinions about these questions.

But when updating a national height model we are changing an essential piece of the geospatial infrastructure. A piece that may be in use in unknown and unexpected ways in various institutions. Hence, updating/changing the model may break existing applications in interesting, but expensive and disrupting, ways.

This means that a large number of relevant stakeholders may have differing opinions on the subject. Opinions that may even be mutually exclusive.

A technically simple way to deal with this could be to operate with a conservative model that is only updated at predictable and agreed intervals, and a progressive model semi-automatically incorporating all available new data, including their potential errors—*Bleeding Edge, Blunders Included!*.

But even in the case of stakeholder consensus on a purely conservative model, one should not underestimate the value of a process of continuously integrating new data, even though the improved model will not be distributed: in the case of continuous integration, one gets a much better feeling of the actual quality of the existing model, which in turn may lead to improved metadata.

2.2 Spatial autocovariance

Spatial autocovariance (or simply spatial covariance) is a concept describing how well a physical observable is represented by a nearby measurement. To define the autocovariance we start from the variance of a set of n spatial observations z_i:

$$C_0 = \sum \frac{z_i^2}{n}. \tag{2.1}$$

Readers expecting a different expression are referred to the note on means versus models below.

We now define the *lag*, d as the distance between two observations. For any given d, we compute the variance-like expression

$$C_d = \sum \frac{z_i z_j}{n_d} \tag{2.2}$$

where the sum is understood to run over all n_d pairs $\{z_i, z_j\}$ having a mutual distance of d (or, in most practical cases, a mutual distance of *approximately* d).

Computing C_d for a range of different lags results in a discrete set of numbers known as the *empirical covariance*. To be able to estimate the covariance for any d, we fit a continuous model to the discrete set.

The Hirvonen covariance model

One of the simplest and most useful covariance models was published by Hirvonen (1962). Hirvonen's model is isotropic, i.e. assuming that the covariance is a function of distance only (which we also did implicitly in the description above). The Hirvonen model is defined as:

$$C_H(d) = \frac{C_0}{1 + \left(\frac{d}{L_d}\right)^2} \tag{2.3}$$

where C_0 is the variance of the data set (equation 2.1), and L_d is the lag for which the covariance C_d (equation 2.2) has dropped to $C_0/2$.

C_0 and L_d can both be read directly from a plot of the discrete empirical covariance values. But be aware that C_0 and L_d are not just properties of the data set. *They are properties of the physical field investigated.*

Hence, one should be very sceptical if a new data set exhibit covariance values that differ much from the existing. *In other words, estimation of a covariance model is a natural early step in the acceptance check of any new data set.*

A note on means versus models

In geodesy it is common practice to work on anomalies, rather than raw physical values. Anomalies (or more generically speaking: residuals) are computed with respect to a model, essentially separating the deterministic part of the signal from the stochastic. This enables us to use the right tool for each job: Physical reasoning for the deterministic part, and geostatistical methods for the stochastic part.

When *not* working on anomalies, it is common practice to model the deterministic part of a signal as the mean of the observations. Hence the well known expression

$$\sigma^2 = \sum \frac{(z_i - m)^2}{n-1} \tag{2.4}$$

for the variance of a set of n observations z_i with mean value m.

When subtracting an independently derived deterministic model from the observations, we really subtract something that is potentially more meaningful than the mean (i.e. a local model value, rather than a global mean).

In computations involving anomalies, the last step is to add back the value of the appropriate deterministic model, all in all a scheme known as the *remove-restore principle*, cf. e.g. Hofmann-Wellenhof and Moritz (2006, pp. 379–381).

Hence, the implied use of the remove-restore principle in equations 2.1–2.2, plays the same role as the removal of the mean, m, in equation 2.4 [3].

It is, however, not uncommon to further fit and subtract a low order spatial polynomial from the anomalies. Essentially this amounts to modelling (as a low order trend surface) effects unresolved by the deterministic reference model. Evidently, when selecting a polynomial of order zero, this is equivalent to removing the residual mean, as in equation 2.4.

The use of n in the denominator of equations 2.1–2.2, rather than the $n-1$ used in equation 2.4, comes from the fact that $n-1$ signifies the loss of one degree of freedom by the computation of the mean from the same sample used to estimate the variance. When obtaining the "mean equivalent" from an independent model, this loss does not occur.

2.3 A DEM filtering scheme

When applying to heights methods that were developed for use with gravity data, we must cut some corners and cannot expect to gain the same level of conceptual rigor as in the original field.

In the present work, this is most evident in the case of splitting the deterministic part of the signal from the stochastic. The deterministic part is constructed from the original DTM in a meaningful, but openly heuristic, method dubbed *Bimorphologically Constrained Filtering* (BCF).

The method is documented by a code snippet in appendix A.1, but in brief it is based on iterative application of still wider gaussian filters then, for each grid point, selecting the maximum degree of filtering still keeping the change below a predefined threshold.

This threshold is then relaxed for singular outliers, which we do not want to consider part of the deterministic signal. The relaxation is based on operators from the field of mathematical morphology (Haralick et al., 1987). In other words, the filter is constrained by both the landscape morphology and by mathematical morphology. Hence, the *Bimorphologically Constrained...* moniker.

An example of BCF in action is shown in section 4.1.

2.4 Spatial interpolation

In the experiments (chapter 4), we will need to carry out spatial interpolation in various ways. For information, we quote below (without derivations, but with some comments), some of the main results derived in the admirably clear, compact, and highly recommended lecture note by Nielsen (2009). For a very different (but equally clear) approach, see Bourke (1999)

In general, interpolation is carried out by computing a weighted mean of observations in the vicinity of a point of interest (POI). Essentially interpolation schemes differ only in how they define vicinity, and how they assign weights.

Nearest neighbour interpolation

In nearest neighbour interpolation, the observation nearest to the POI is assigned the weight $w = 1$. All other observations are assigned the weight $w = 0$.

Global mean interpolation

In global mean interpolation, all N observations are assigned the weight $w = 1/N$. This also means that all POIs will get the same value.

Local mean interpolation

In local mean interpolation, all n observations within a given search radius r of the POI, are assigned the weight $1/n$. For large values of r local mean interpolation tends toward global mean interpolation.

Inverse distance weighting

In inverse distance weighting, the weights are constructed such that observations close to the POI gets higher weights. Let d_i denote the distance from the POI to observation number i. Then weights are assigned as:

$$w_i = \frac{1/d_i}{\sum_{j=0}^{N} 1/d_j}$$

Which is trivially generalized to weights based on powers of the inverse distance:

$$w_i = \frac{1/d_i^p}{\sum_{j=0}^{N} 1/d_j^p} \tag{2.5}$$

Often $p = 2$ is used (*inverse square distance weighting*). Presumably inspired by the inverse square nature of gravitational and electomagnetic force fields. There is, however, nothing magical about $p = 2$, so if using inverse distance weighting, the p factor should be selected in a way commensurable with the autocovariance of the phenomenon at hand.

For $p = 0$, $d_i^p = 1$ for all i, turning inverse distance weighting into global mean interpolation (or local mean if the sum is restricted to points within a certain distance from the POI).

For $p \to \infty$, the weight function drops off more and more sharply, so in this case inverse distance weighting tends toward nearest neighbour interpolation. For most practical purposes, $p = 10$ is sufficiently close to infinity to make this happen. This feature arguably makes 10 one of the smallest infinities in common use!

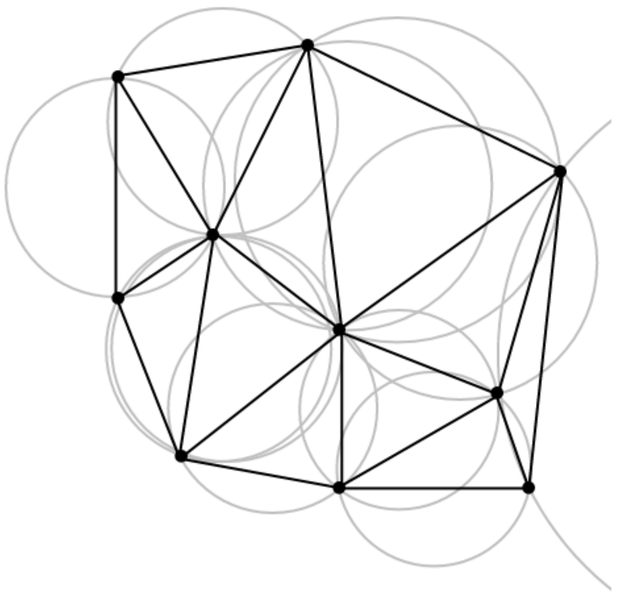


Figure 2.1: The Delaunay triangulation for a set of points: no point of the set is *inside* the circumcircle of any triangle. Delaunay triangulations tend to avoid long skinny triangles since they maximize the minimum angle of all the angles in the triangulation.

Kriging

In kriging (named after the South African mining engineer *Danie Krige*), the prediction weights are designed to result in a *central* estimator with *minimum estimation variance*. This is a cryptic way to say two things.

First that since we do not know the actual value at the POI, we cannot know the size of the prediction error (i.e. the difference between the actual value and the predicted). But by designing the estimator appropriately, we can make sure that the *statistical expectation value* of the prediction error is zero, i.e. that the mean error of a large number of predictions is zero. This is what *central estimator* means.

Second that once we have designed the estimator to result in zero *mean* prediction errors, we also want to have optimum confidence that any *individual* prediction error is as small as possible (we do not want to achieve a zero mean by delicately balancing huge errors with alternating signs). Assuming that errors are approximately normally distributed, the chance of running into a large error grows with the variance of the distribution. Hence, by designing the weights to result in minimum error variance, we maximize the chance that the error of any individual prediction really *is* very close to zero.

To make this happen in real life, we need to estimate a good covariance model for the physical field we are studying. Having obtained a variance model, we must compute variance values C_{Pj} for the distances between the POI and the observations, and C_{ij} for the distances between the individual observations. The weights can then be found by solving the set of linear equations:

$$\begin{bmatrix} C_{11} & \cdots & C_{1N} \\ \vdots & \ddots & \vdots \\ C_{N1} & \cdots & C_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} C_{P1} \\ \vdots \\ C_{PN} \end{bmatrix} \tag{2.6}$$

The prediction is then given by:

$$Z_0 = \begin{bmatrix} w_1 & \cdots & w_N \end{bmatrix} \begin{bmatrix} Z_1 \\ \vdots \\ Z_N \end{bmatrix} \tag{2.7}$$

The prediction variance is, in turn, given by

$$\sigma_0^2 = C_0 - \begin{bmatrix} w_1 & \cdots & w_N \end{bmatrix} \begin{bmatrix} C_{P1} \\ \vdots \\ C_{PN} \end{bmatrix} \tag{2.8}$$

To compute the set of N weights, we need to solve a set of N linear equations. But the processing power needed to do this is proportional to N^3, which quickly makes it prohibitively expensive in computer time to solve for more than just a few weights.

As a simple example consider the ratio between $5^3 = 125$ and $3^3 = 27$, indicating that it takes almost 5 times as long to solve for 5 weights than for 3 weights.

This means that even for moderately large computations, we need an efficient way to select a small, relevant subset of observations for any POI at hand.

To this end, the Delaunay triangulation comes to the rescue.

2.5 Delaunay triangulation

The Delaunay triangulation was introduced by the Russian mathematician Boris Delaunay (Delaunay, 1934). Any given set of points can be organized as a area partitioning set of triangles (see figure 2.1). The Delaunay triangulation is defined as the (unambiguous) partitioning that globally maximizes the minimum angle of the entire triangulation.

For any triangle in a Delaunay triangulation, no other points than its three corner points will be situated inside the circumcircle of those three points. Hence, elongated triangles are essentially avoided, since they correspond to excessively large circumcircles.

Due to its many practical uses, much effort has been put into the derivation and development of very fast algorithms for constructing the Delaunay triangulation. In this work, the *QHULL* algorithm (Barber et al., 1996) is used. QHULL is widely used in closed- as well as open source software. The *Triangle* algorithm by Jonathan Shewchuk (Shewchuk, 1996) is another excellent implementation, but Triangle is less used than QHULL, arguably due to a more restrictive licencing policy.

Once the Delaunay triangulation is constructed for a given set of points, it is easy to find the triangle surrounding any given POI, and hence obtain a small set of 3 observations that may not necessarily be the nearest neighbours of the POI, but which are close to the POI and spatially distributed in a way making them good candidates for a robust estimation of the value needed at the POI.

Chapter 3

Test site and test data

3.1 Test site location and topography

The 1 km × 1 km test site used for the experiments in chapter 4, is situated just northwest of the village of Vejby in Helsinge Municipality, North-Zealand (figure 3.1). The landscape of the Vejby area is glacially shaped and gently undulating. Due to its beauty, the area was the subject of a large number of paintings from the 1840s by the National Romantic painters J.Th.Lundbye (cf. figure 3.2) and P. C. Skovgaard (Jørgensen, 1995).

The test site is characterized by relatively large and sometimes steep, height variations. The land use/land cover (LULC) includes hedgerows, paths, dirt roads, farms, wetlands, lakes, a small forest, farmland, a closed down and only partially refilled, clay pit, and (in the northern end) a cottage area from around 1960. All in all a landscape that is not only beautiful, but also challenging and hence highly interesting from a height modelling point of view.

To establish a well controlled framework for our DTM updating experiments, we introduce *synthetic changes* in the existing DTM (section 3.2), and use the modified model as the new *ground truth*.

This ground truth is in turn used as target for a *synthetic LiDAR flight* (section 3.3), generating new synthetic observations of the changed terrain.

Finally, in chapter 4, we combine the original (unchanged) DTM with the new synthetic LiDAR observations, attempting to reproduce the synthetic ground truth.

3.2 A DTM with synthetic changes: The new ground truth

The 1 km × 1 km test area is represented by a 626 × 626 grid stored in an ESRI ascii format file (figure 3.1). We introduce change in the form of a synthetic road spanning the grid rows numbered 523–527 (marked in white on figure 3.1).

The "road" is constructed as follows: First we remove any across track slope by computing the columnwise mean of the set of 5 rows.

Then we carry out an along track gaussian filtering of the mean row, i.e. easing the road for the cyclists by cutting hilltops and filling valleys (figure 3.3). Finally the filtered mean row is copied back into the original five rows 523–527.

The resulting grid (figure 3.4) is now considered the new ground truth for the experimental work.

3.3 Synthetic LiDAR observations

The synthetic LiDAR observations are intended to simulate the result of a single flight strip in a corridor mapping effort, mapping a strip centered on the new road. The observations are generated by this little snippet of Octave code, which notably does not take any roll/pitch/yaw irregularities into account:

```
% rand: uniform, randn: gaussian randomness
N = 2*100*1000;
north = 6218000 + 160 +  100*(rand (N, 1) - 0.5)
east  =  694000 + 500 + 1000*(rand (N, 1) - 0.5)
z = interp2 (h.x, h.y, g, east, north);
% add gaussian noise with a variance of 5 cm
noise = sqrt (0.05) * randn (N, 1);
z += noise;
```

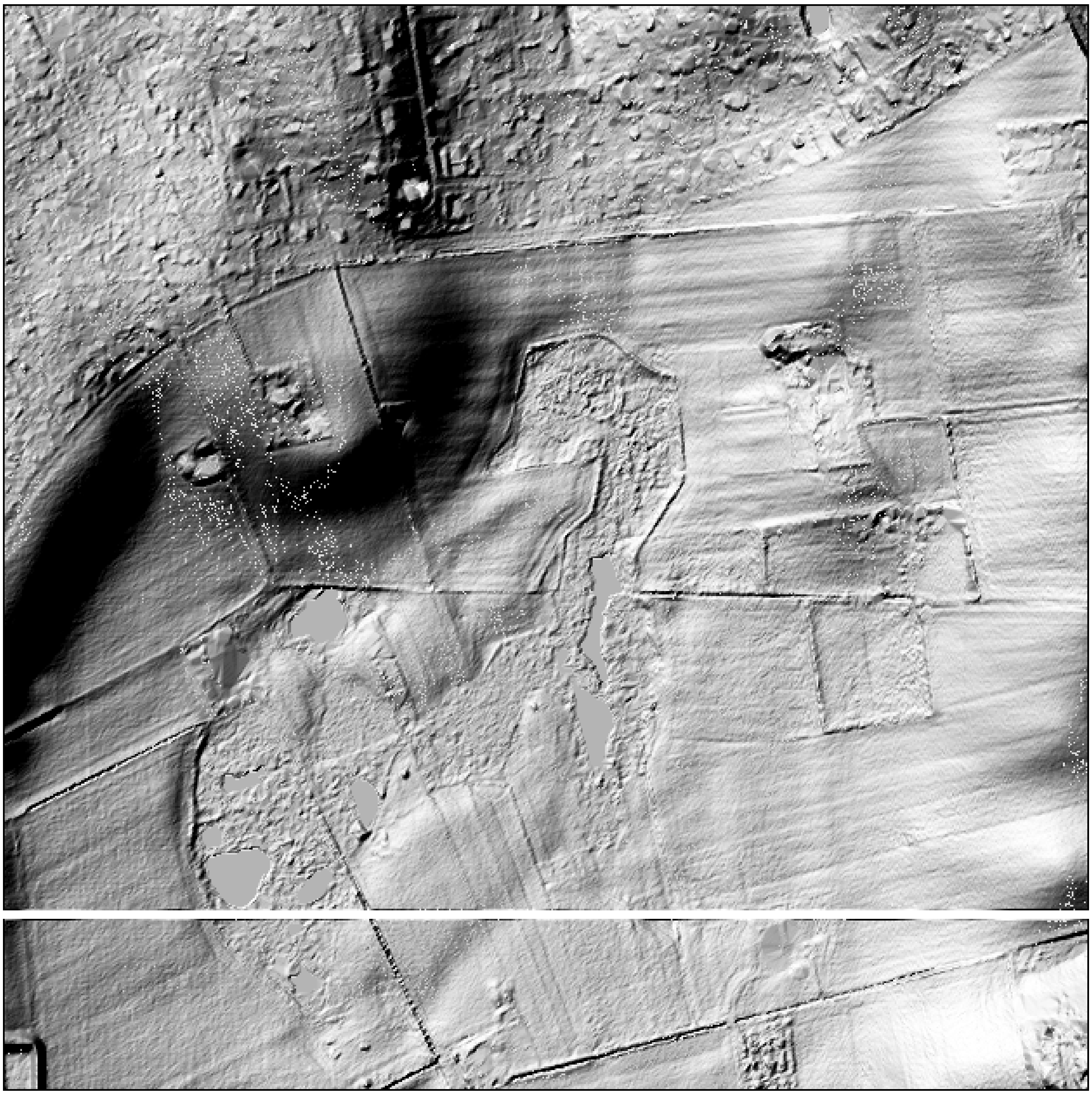
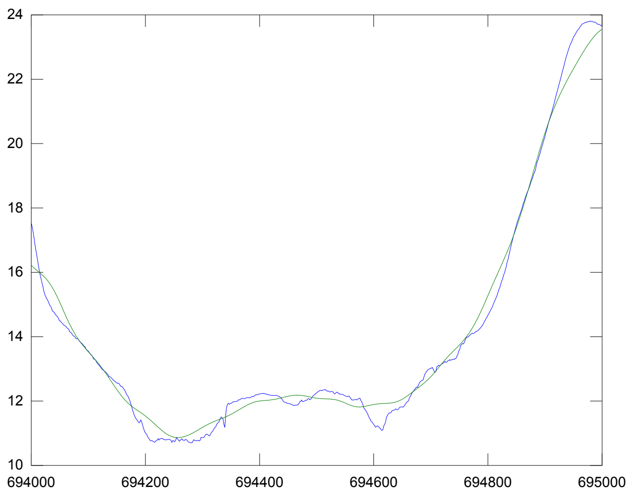


Figure 3.1: The test site covers a range of 6218000 m–6219000 m northing and 694000 m–695000 m easting in UTM zone 32/ETRS89 coordinates. Heights are in the range of 10.4 m–35.3 m (referred to the Danish vertical datum DVR90) with a mean value of 18.3 m and a median of 15.6 m. The white strip in the lower part of the figure indicates the position of the synthetic road introduced for the DTM update experiments (cf. section 3.2).

Figure 3.2: Johan Thomas Lundbye: Landskab fra Vejby (Landscape from Vejby), 1843.



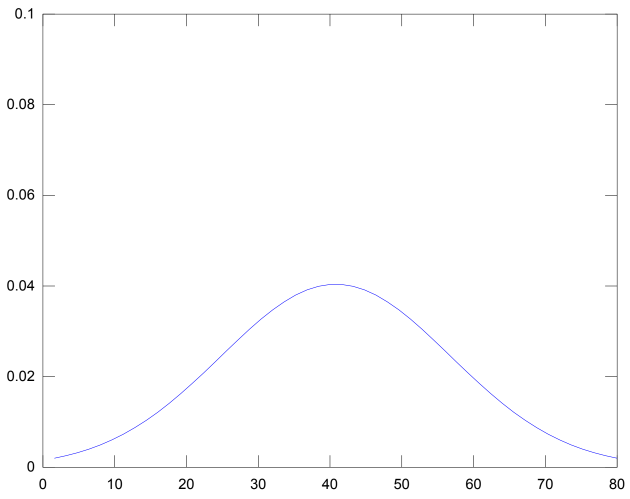


Figure 3.3: **Left:** The raw road profile (blue), and the final along track filtered road profile (black). **Right:** The 50 point gaussian filter used to even out the bumps in the synthetic road.

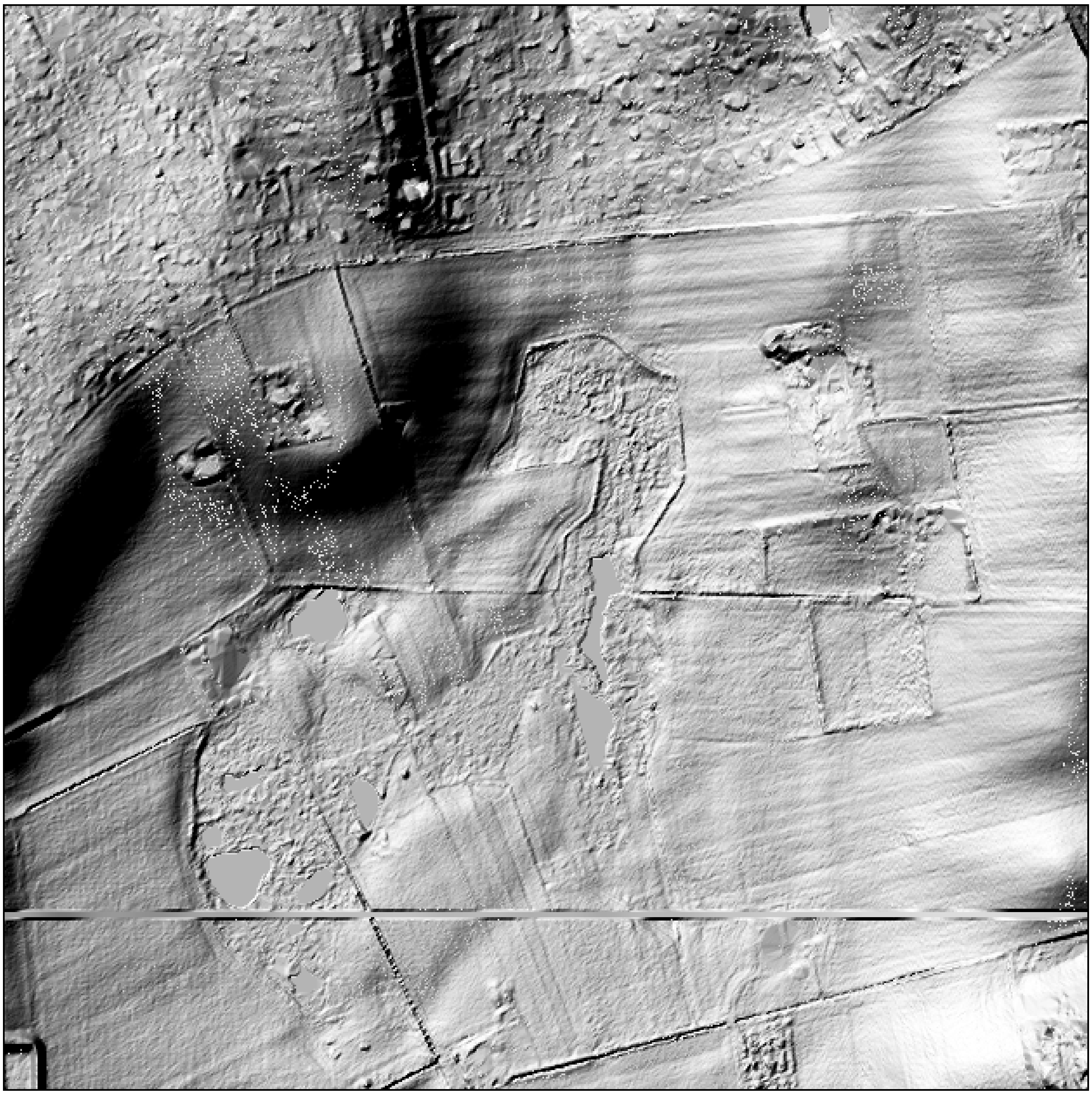


Figure 3.4: The final synthetic road introduced into the original grid.

The call to **`interp2`** interpolates new height values from the ground truth grid **`g`**.

The constants of the code snippet are to be interpreted as follows:

`(6218000, 694000)` Northing/easting of the lower left corner of the test site.

`(160, 500)` The center of the LiDAR covered area is situated 160 m north and 500 m east of the loweer left corner.

`(100, 1000)` The flight strip is 100 m wide (north/south) and 1000 m long (east/west).

`N=2*100*1000` We need 2 observations/m^2 for each of the 100 m $\times$ 1000 m covered, i.e. a total of 200 000 observations.

Chapter 4

Experiments

4.1 Experiment 1: Geostatistical characterization

The first thing to do with a newly acquired geodetic data set is to get a general feeling of its characteristics. For data which can be plotted in meaningful and straightforward ways (e.g. imagery, simple time series), plotting is the obvious first action. But in our case of irregularly sampled observations, a geostatistical characterization is the obvious first action.

Actually the first step of the geostatistical characterization is not geostatistical at all, but strictly statistical (*sans* geo-). The first step is to plot the histogram of the data, to get an idea of what kind of distribution is behind the data.

The upper left panel of figure 4.1 shows the histogram for the heights of the DTM of the test area (figure 3.1). It is not evident what kind of distribution might fit this histogram, but it certainly isn't a normal (Gaussian-) distribution. This is even more clear from the QQ-plot in the upper right panel of figure 4.1, where the quantiles of the distribution (i.e. the lower quartile, median, upper quartile, and their unnamed brothers and sisters for other values than 25%, 50%, and 75% of the distribution mass) is plotted against the corresponding quantiles of the standard-normal distribution (i.e. the normal distribution with parameters $(\mu, \sigma^2) = (0, 1)$). In a QQ plot, normal distributions will be depicted as a straight line, with the slope determined by the σ^2 parameter, and the offset determined by the μ parameter of the actual distribution. It is evident from the QQ-plot that the distribution of the raw heights is not normal at all.

Suspecting that the non-normality is due to autocorrelation effects induced by causal (deterministic, non-stochastic, etc.) processes, we go on to construct a "deterministic height surface", as described in section 2.3, using the code presented in appendix A.1. The resulting smooth surface, and its corresponding anomalistic surface are shown in figure 4.2.

The histogram for the anomalies are shown in the upper left panel of figure 4.3. Despite a marked "knee" around $\delta h = 1m$ and a minor wart around $\delta h = -1m$, the histogram looks much more normal than the corresponding histogram for the raw heights. The normality is confirmed by the QQ-plot in the upper right panel.

Covariances

The covariances of the raw terrain model and the anomalies, along with the corresponding Hirvonen covariance models, are shown in the lower right panels of figures 4.1 and 4.3, respectively.

The first thing to note is that removing the "deterministic height model", really reduces both variance and correlation length: Where the raw DTM (with mean height removed, since it is not an anomaly, as per the discussion in section 2.2) corresponds to a Hirvonen model with parameters $(C_0, L_d) = (85 \text{ m}^2, 182 \text{ m})$, the corresponding parameters for the anomalies are $(C_0, L_d) = (0.57 \text{ m}^2, 44.8 \text{ m})$. In other words: by removing the "deterministic model", the covariance scale is reduced by a factor of almost 150, and the correlation length is reduced by a factor of more than 4.

It is also striking how well the Hirvonen model fits the actual data – especially in the interval $[0 \ldots L_d]$, which is really the interval we usually would need[4].

One should, however, bear in mind that the empirical covariances in these cases are based on very simplified computations based directly on

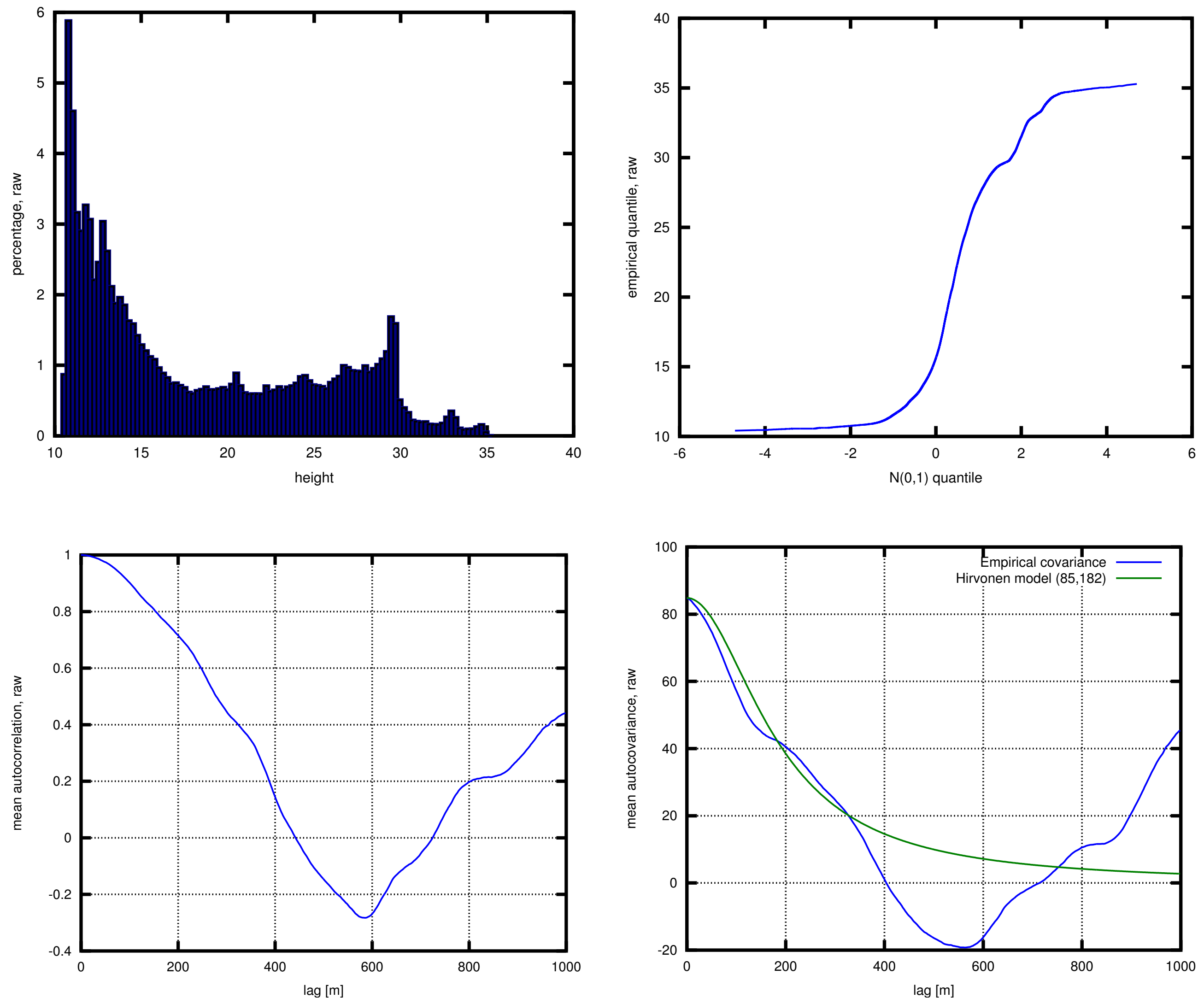


Figure 4.1: Raw DTM geostatistics. Upper left: Histogram. Upper right: Quantile plot of data quantiles vs. quantiles from a standard normal distribution. Lower left: Empirical autocorrelation. Lower right: Empirical autocovariance. Note that the autocorrelation only changes very slowly from 1. In other words, far away points are as important as very nearby points in predicting the height at any given point.

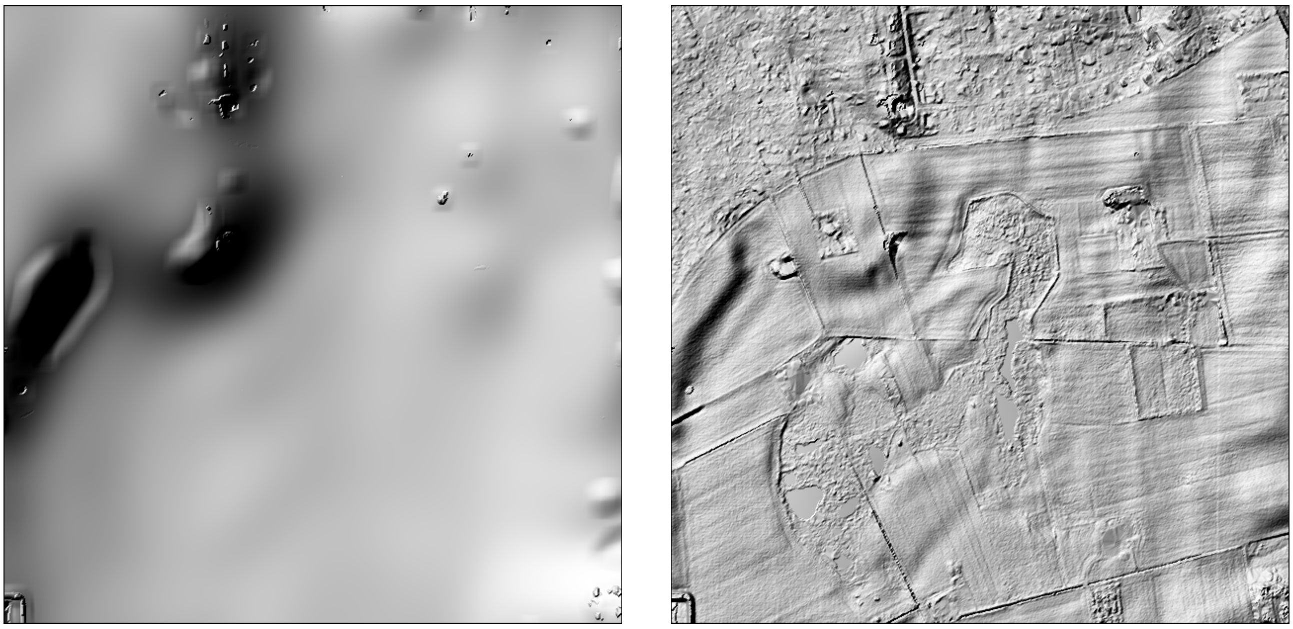


Figure 4.2: Left: filtered “deterministic” terrain model. Right: corresponding anomalistic (residual) terrain model.

the gridded structure of the data: covariances are computed for two directions only (the directions of the northing and easting axes), and using the boundary values of the grid as starting points.

Hence these data are based on 1252 sets of empirical covariance values for each of the two directions, each set consisting of 626 values covering the lags $d \in [0 \text{ m} \ldots 1000 \text{ m}]$ at steps exactly corresponding to the grid GSD of 1.6 m.

While being a simple and totally sensible approach, one should expect more smooth covariance values than what would be the case for the the more complex algorithm necessary for arbitrarily distributed data.

The more complex algorithm has been used for preparation of figure 4.4: Here, we work on a dataset consisting of 63 grid lines centered on the center line of the artificially introduced road (cf. section 3.2), i.e. grid line number 525. In other words, we work on a set of $63 \times 626 = 39438$ grid points which we in this case treat as a non-structured point cloud. Obviously, we might have used the synthetic LiDAR data set covering the same region (cf. section 3.3), but to keep in line with the data used above, we stay with the orginal grid values.

Now, we randomly select 5 million point pair combinations and sum up their products according to equation 2.2. The sums are computed in bins of width 1.6 m, centered on the set $[1.6, 3.2, 4.8, \ldots, 1000]$ (see Nielsen (2009) for details of the construction). In other words, rather than the *almost isotropic* data used above, each bin now gets contributions from point pairs at different bearings and different distances (except for the very first few bins, where only a few combinations of distance and bearing are possible)

Not unexpected, the data in figure 4.4 look rather more noisy than the previous plots. Also note that the restriction to a smaller area has reduced both the variance and the correlation length of the raw values as well as the anomalies. This is also as expected, as we now compare a more uniform data set (especially with respect to the raw heights, where we now avoid the effects of the large north–south height undulation).

4.2 Experiment 2: Updating with simple kriging

Figure 4.5 shows the original grid (figure 3.1), improved with data predicted using the *simple kriging* method described in section 2.4, and the synthetic LiDAR data presented in section 3.3.

There really is not much to say about the results: in comparison with the artificial ground truth (figure 4.5), the most striking differences are the (expected) effects of the noise added in

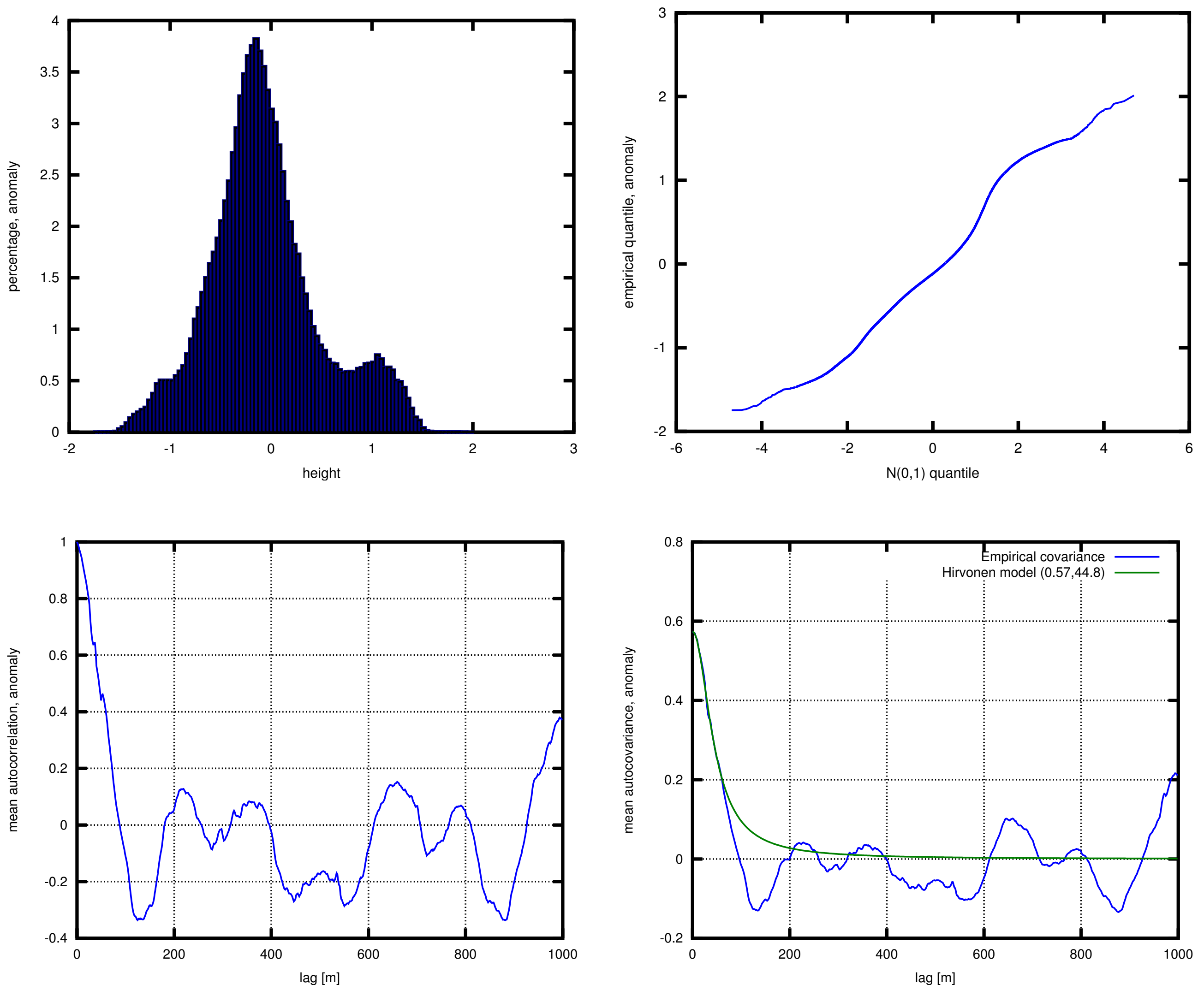


Figure 4.3: Anomaly DTM geostatistics. Upper left: Histogram. Upper right: Quantile plot of data quantiles vs. quantiles from a standard normal distribution. Lower left: Empirical autocorrelation. Lower right: Empirical autocovariance. Note that the autocorrelation falls rapidly from 1. In other words, far away points are not as important as very nearby points in predicting the height at any given point.

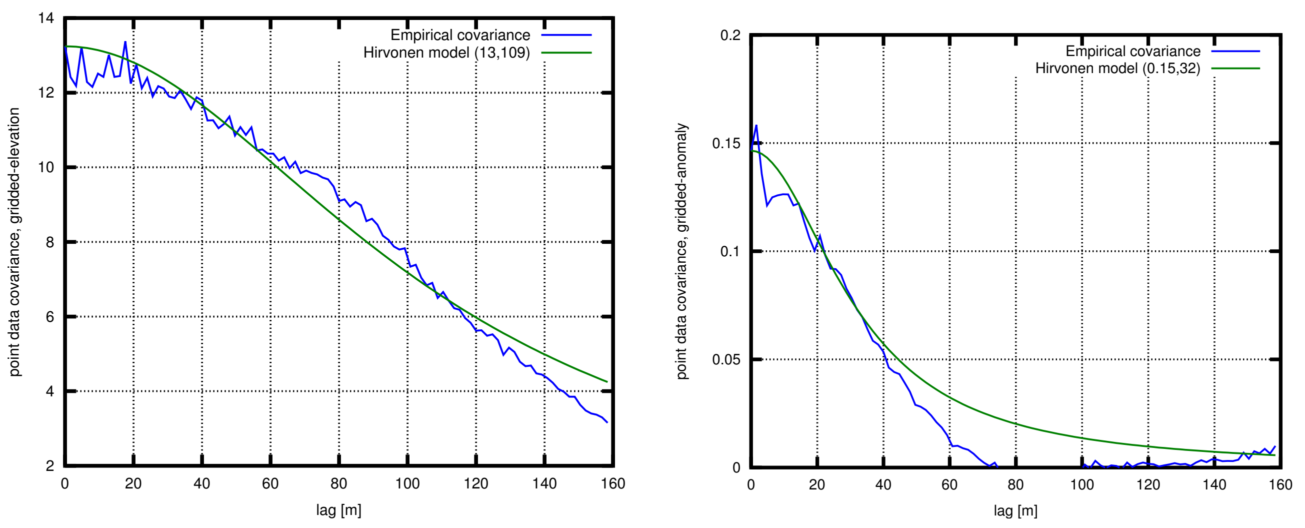


Figure 4.4: Covariances for the narrow region around the new road. Note that the lag-axis only covers the first 160 m. Left: Full elevations. Right: Anomalies

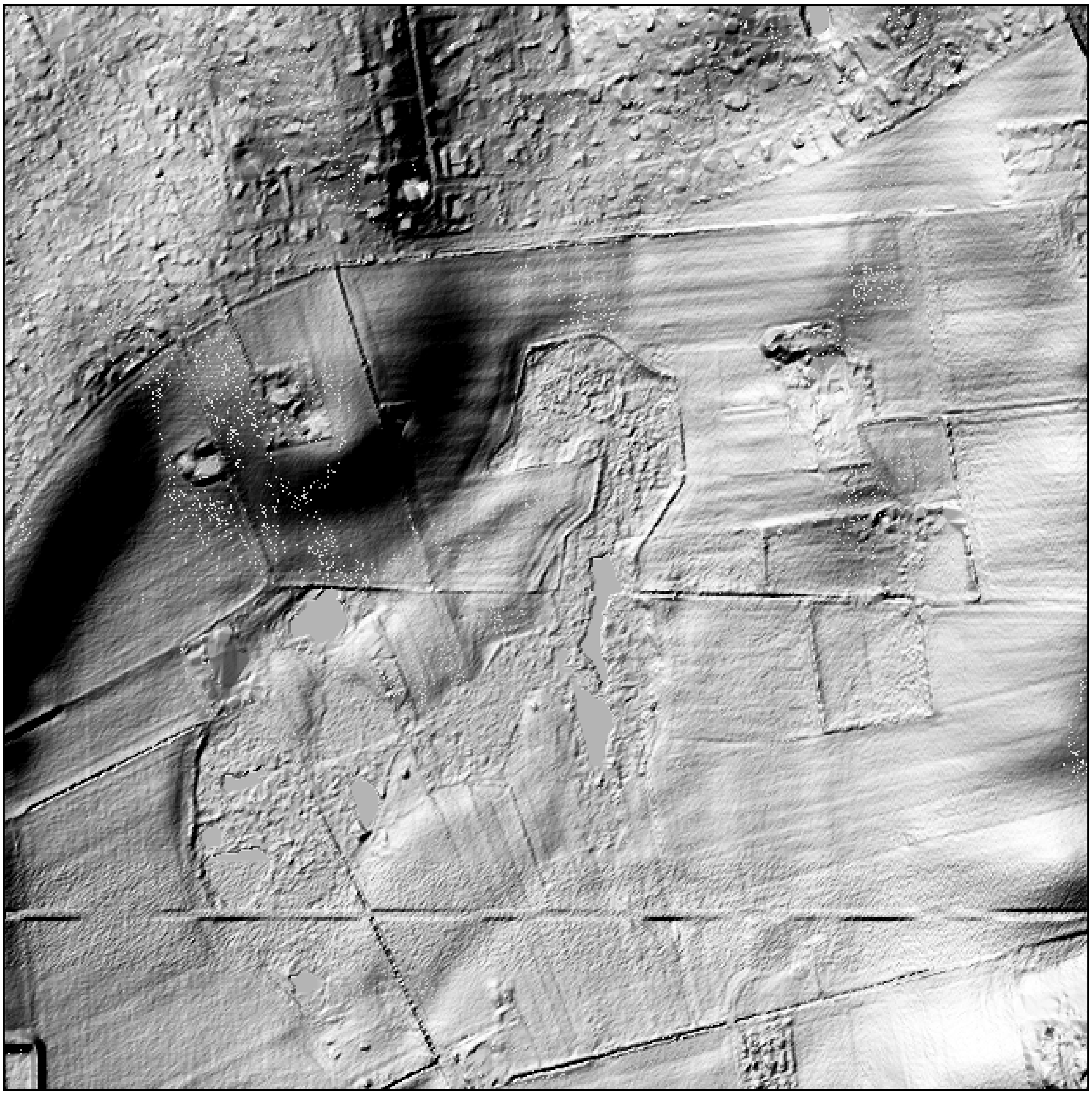


Figure 4.5: The test site modified with updated values (computed by simple kriging) in the area around the artificially introduced road. The effect of the deliberately introduced noise is clearly seen.

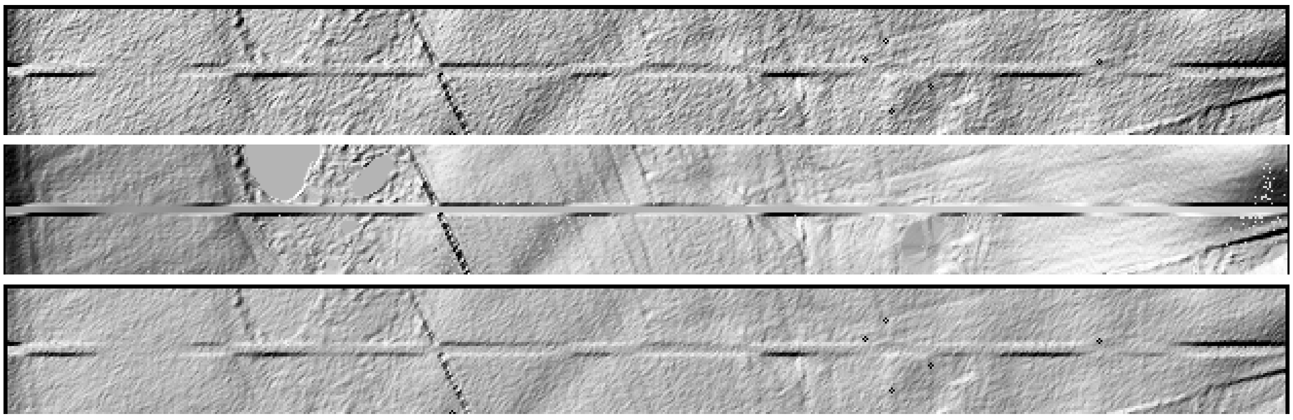


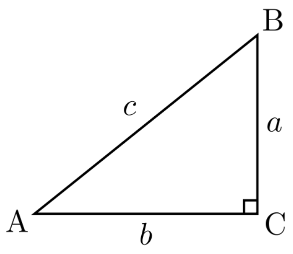
Figure 4.6: **Upper panel:** The updated values computed by simple kriging in the area around the artificially introduced road. **Centre panel:** The artificial ground truth (cf. section 3.2). **Lower panel:** The updated values computed by simple kriging with noise reduction applied (cf. section 4.3).

the production of the synthetic LiDAR data.

In the next section we will show how this noise can be reduced. Not by filtering the gridded data, but directly as a part of the prediction process for the updated grid values.

4.3 Experiment 3: Handling observational noise

Consider a height field described by a Hirvonen covariance model with the parameters $(C_0, L_d) = (0.5\ \text{m}^2, 5\ \text{m})$, and consider the configuration shown in this sketch,



with 3 observations made at the 3 corners A, B, C of a right triangle with side lengths $a = 3$, $b = 4$, and $c = 5$. Furthermore, let us place the origin of our system at A, and the POI, P at the barycenter of the triangle (i.e. at the intersection of its medians). In this case, the code in appendix A.2, solving the simple kriging system, equation 2.6, results in the weight vector

$$w_0 = \begin{bmatrix} 0.30420 \\ 0.29346 \\ 0.49874 \end{bmatrix}$$

Now consider that the observation at point A was influenced by a noise characterized by $\sigma^2 = 0.1\text{m}^2$. If we add that noise term to the diagonal element for A, and solve once again for the weights, we get

$$w_a = \begin{bmatrix} 0.23022 \\ 0.30178 \\ 0.53773 \end{bmatrix}$$

i.e. the weight for the noisy observation is reduced (and so is the total sum of weights).

But what if all 3 observations were equally influenced by noise? Let us add the same noise term to the other diagonal elements, and solve once again for the weights:

$$w_{abc} = \begin{bmatrix} 0.29204 \\ 0.29875 \\ 0.41855 \end{bmatrix}$$

Now, the two weights originally largest are reduced, while the smallest weight is slightly enlarged. All in all moving the estimator more in the direction of equal weights, which would improve the suppression af random noise.

In figure 4.6, we show the result of adding the artificially generated noise source (section 3.3) to the diagonal elements of equation 2.6 for an entire grid prediction experiment.

The result is quite convincing, although one should consider that only part of the noise reduction is due to the blurring: another part is due to the relative downweighting of the anomaly compared to the deterministic part.

One may see this as an advantage or a disadvantage. In either case, the downweighting with respect to the deterministic part would not happen if using Ordinary Kriging (OK), rather than Simple Kriging (SK) as used here: in OK, the sum of the weights is forced to unity, and no prior assumptions are made with respect to the mean value of the signal, which is implicitly reestimated for every prediction, hence, perhaps eliminating the need for computing the "deterministic" surface.

The use of individual noise estimates for each observation will probably become possible as access to full waveform reflection data become more common. When that happens, including the noise term in the prediction process will make even more sense.

4.4 Experiment 4: Eliminating drift by draping

Airborne LiDAR observations depend heavily on a well functioning inertial navigation system (INS) providing the pointing of the platform: The combination of position information from the GPS, and pointing from the INS is essentially what makes it possible to convert LiDAR reflection timings to ground elevations.

But INS tend to drift – a feature that can often be corrected for by cross over adjustments with neighbouring flight strips. But in the case of corridor mapping, few or no neighbouring strips are recorded. In such cases the technique of draping (cf. e.g. Strykowski and Forsberg (1998)) comes handy. Draping builds on the assumption that if two datasets that are supposed to have identical long wavelength parts tend to drift from each other anyway, then we must construct a correction surface.

We may think of the two datasets as an old, stable but noisy model, and a newer drifting, but less noisy one.

We construct the correction surface by low-pass filtering both signals and subtracting them. The draping operation is then carried out simply by applying the correction surface to the newer dataset.

The new dataset is then said to have been *draped* over the old, inheriting the old dataset's long wavelength accuracy, while keeping its own higher short wavelength accuracy.

Figure 4.7 shows a somewhat exaggerated example based on simulated data: An old terrain model of moderate quality ($\sigma^2 = 0.5$m) is to be updated by new data of much higher quality ($\sigma^2 = 0.05$m). Unfortunately, the new data suffers from a huge drift of almost 8 m along the 100 km track. The draping process (see code in appendix A.3) improves the RMS between the "true" landscape and the new data from a ghastly 4.45 m, to the more acceptable 0.24 m

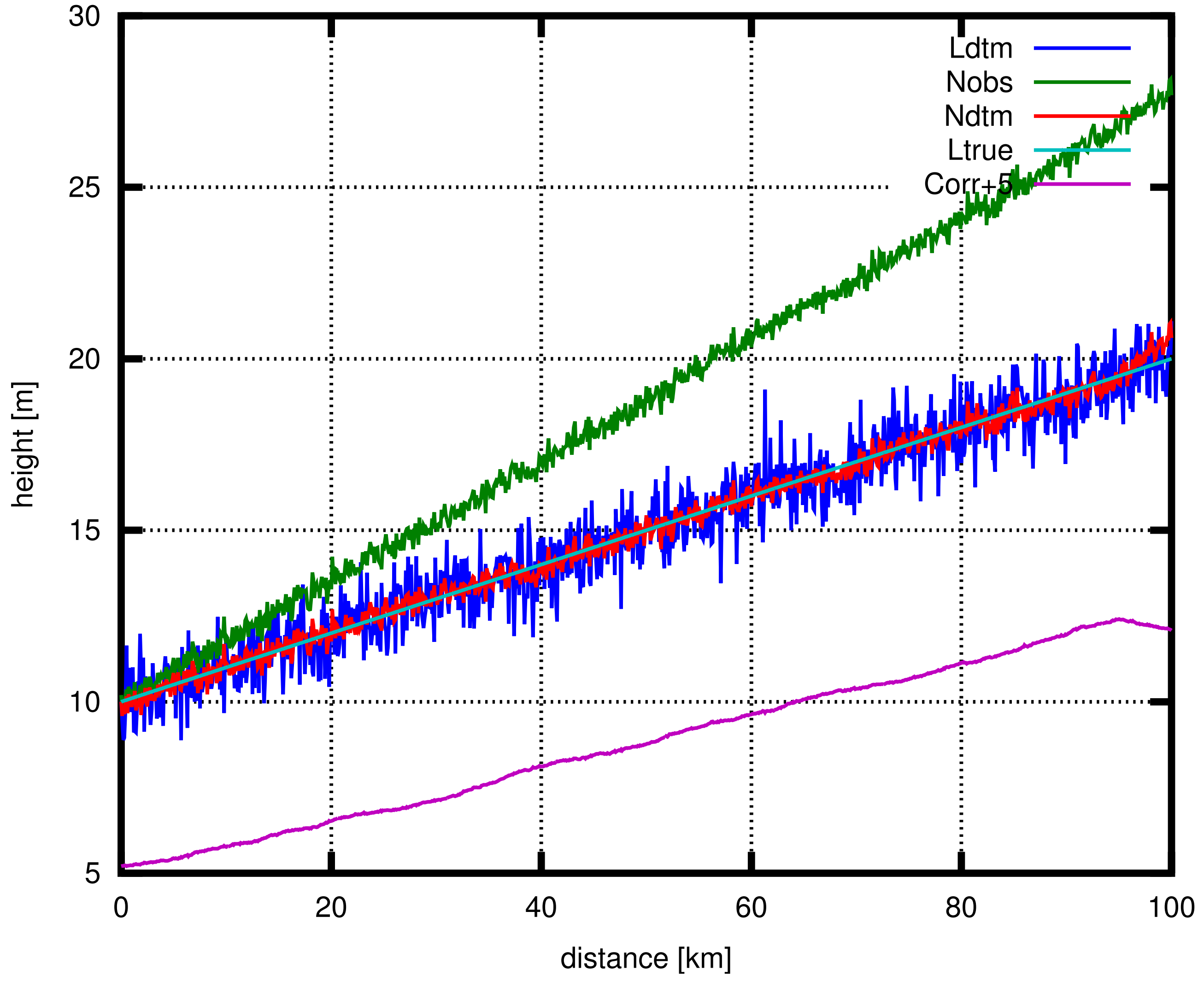


Figure 4.7: Removing drift by draping (cf. section 4.4. Except for the last 5 km, where some boundary effects become visible, the draping process is very succesful in removing the drift. **Cyan:** The true landscape. **Blue:** The old, noisy terrain model. **Green:** The new, less noisy but drifting observations. **Purple:** The correction surface (biased by 5 m to make it fit onto the plot). **Red:** The new terrain model, consisting of the new observations (green) corrected by the correction surface (purple) / draped onto the old model (blue).

Chapter 5

Outroduction

5.1 Future work

The fields of geostatistics and geodesy can provide plenty of useful methods and techniques for updates of height models.

The next obvious step in the work will be to use some of the methods presented here with new real data, rather than the simulated data used in this report.

Other interesting work will be the (fairly simple and ordinary) switch from Simple Kriging to Ordinary Kriging, with its implicit reestimation of the mean value for each prediction. This may also eliminate the need for the deterministic surface in the predictions (while it may still be of use for visualisation and contour line generation).

Also the case of non-isotropic covariance models, which has been left out of the scope for this report, needs to be handled. A particularly simple (conceptually, not implementation-wise!) example of this is the handling of breaklines by modifying covariances for vectors crossing the breakline.

Another interesting way of handling breaklines is through the direct inclusion of the breakline into the Delaunay triangulation (i.e. a so called constrained Delaunay triangulation).

Finally, there is the problem of surface models. In the case of surface models, we do not have a geostatistically uniform area to model. This is a significant complication, which will probably lead on to new and interesting problems!

There's plenty of work to take up!

5.2 Conclusion

For fear of sounding too optimistic, speaking on the basis of purely synthetic data, I have refrained from quoting numerical results in this report. But the examples presented have shown to be quite convincing.

Hence, taking the geostatistical/geodetic route seems to be a viable option, with plenty of new opportunities that should be further explored.

Acknowledgements

Figure 2.1 was produced by Wikipedia user *Nü es*. The original version is available from http://commons.wikimedia.org/wiki/File:Delaunay_circumcircles.png, and is distributed under the Creative Commons Attribution-Share Alike 3.0 Unported license.

Figure 3.2 *Johan Thomas Lundbye: Landskab fra Vejby (1843)* was provided by Wikimedia Commons (http://commons.wikimedia.org/wiki/File:Johan_Thomas_Lundbye_-_Landskab_fra_Vejby.jpg), where it was uploaded by user *Rlbberlin* and marked as being "in the public domain in the United States, and those countries with a copyright term of life of the author plus 100 years or fewer". A somewhat better reproduction, but with less clear reproduction right statements can be found at the web site of the National Gallery of Denmark *(Statens Museum for Kunst)*, http://soeg.smk.dk/VarkBillede.asp?objectid=10236.

The right triangle illustration in section 4.3 was produced by Wikipedia user *Gustavb* (http://en.wikipedia.org/wiki/User:Gustavb). The original version is available from http://en.wikipedia.org/wiki/File:Rtriangle.svg, and is distributed under the Creative Commons Attribution-Share Alike 3.0 Unported license.

Allan Aasbjerg Nielsen, Andrew Flatman,

Gitte Rosenkranz, Kai Sattler, Lars Stenseng, Niels Broge, Nynne Sole Dalå and Simon Lyngby Kokkendorff read and commented on the manuscript (it goes without saying that the author assumes full responsibility for any remaining errors).

My wife and kids suffered much through my many prolonged working days during the execution of this project. I am deeply grateful for their love, care and support.

Notes

[1] Least Squares Collocation is a considerably more general framework than the other geostatistical approaches mentioned. Cf. e.g. Krarup (1969) or Hofmann-Wellenhof and Moritz (2006), chapter 10

[2]Although one should rather talk about precision than accuracy in such cases

[3]One should, however, never understimate the potential for holy wars over this issue. In this author's totally subjective, and potentially uninformed, opinion, this is probably due to the empirical roots of statistics, making it a fruitful field of study and application alike—hence haunted by domain specific practitioners, as well as more mathematically inclined theorists. In discussions of (geo)statistics, these groups will often find themselves "divided by a common language".

[4]For exactly this reason, the parameter C_0 is computed directly as the variance of the data, and L_d is found through interpolation around the first lag corresponding to a variance of less than $C_0/2$. If doing an actual least squares fit, one would run a serious risk of putting way too much weight on fitting the fluctuations for large lags.

References

C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22:469–483, December 1996. ISSN 0098-3500. doi: http://doi.acm.org/10.1145/235815.235821. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.405. 2.5

Paul Bourke. Interpolation methods, 1999. URL http://paulbourke.net/miscellaneous/interpolation/. 2.4

Nynne Sole Dalå, Rune Carbuhn Andersen, Thomas Knudsen, Simon Lyngby Kokkendorff, Brian Pilemann Olsen, Gitte Rosenkranz, and Marianne Wind. DK-DEM: one name—four products. In Thomas Knudsen and Brian Pilemann Olsen, editors, *Proceedings of the 2nd NKG workshop on national DEMs, Copenhagen, November, 11–13 2008*, number 4 in Technical Report Series, page 4. National Survey and Cadastre (KMS), Copenhagen, Denmark, 2009. URL ftp://ftp.kms.dk/download/Technical_Reports/KMS_Technical_Report_4.pdf. 1.1

B. Delaunay. Sur la sphère vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800, 1934. Here quoted from the WikiPedia article "Delaunay Triangulation", http://en.wikipedia.org/wiki/Delaunay_triangulation. 2.5

R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 9(4):532–550, 1987. 2.3

R. A. Hirvonen. *On the statistical analysis of gravity anomalies*, volume 37 of *Publications of the Isostatic Institute of the International Association of Geodesy*. The Isostatic Institute of the International Association of Geodesy, Helsinki, Finland, 1962. 2.2

Bernhard Hofmann-Wellenhof and Helmut Moritz. *Physical Geodesy*. Springer, Wien/New York, Second corrected edition, 2006. 2.2, 1

Joachim Höhle. Updating of the Danish Elevation Model by means of photogrammetric methods. Technical Report 3, National Survey and Cadastre (KMS), Copenhagen, Denmark, 2009. URL ftp://ftp.kms.dk/download/Technical_Reports/kmsrep_3.pdf. 1.2

Jens Anker Jørgensen. Vejby i vore hjerter. In *Vejby-Tibirke årbog*, page 5. Vejby-Tibirke Selskabet, 1995. URL http://www.vejby-tibirke-selskabet.dk/aarbog/aarbog.asp?page=33. 3.1

Torben Krarup. *A contribution to the mathematical foundation of physical geodesy*, volume 44 of *Geodætisk Institut Meddelelser*. Geodætisk Institut, Copenhagen, Denmark, 1969. 1

A. A. Nielsen. Geostatistics and analysis of spatial data, oct 2009. URL http://www2.imm.dtu.dk/pubdb/p.php?5177. 2.4, 4.1

Jonathan Richard Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. URL http://www.cs.cmu.edu/~quake/triangle.research.html. From the First ACM Workshop on Applied Computational Geometry. 2.5

Gabriel Strykowski and René Forsberg. Operational merging of satellite, airborne and surface gravity data by draping techniques. In René Forsberg, Martine Feissel, and Reinhard Dietrich, editors, *Geodesy on the Move; Gravity, Geoid, Geodynamics and Antarctica*, pages 243–248. Springer Verlag, 1998. 4.4

Appendix A

Code

A.1 Bimorphologically constrained filtering

NOTE: this code is taken directly from the Octave input file. Some lengthy input/output operations have been edited out, and all subroutines called have been left out from the listing, as the code is provided for illustration only – mostly as pseudocode.

```
function ret = filtering (base, cmax, padsize)
    [h g] =  (read header and grid data here)

    kernelsizes     = [3 5 7 9 11 21 41 77]
    kernelvariances = [.1 .1 .1 .09 .07 .07 .04 .04]
    iter = prod(size(kernelsizes))
    f = zeros([size(g) iter]);

    f(:,:,1) = g;

    % filter


    for i = 2:iter,
        iteration = i
        kernel = gaussian(kernelsizes(i), kernelvariances(i));
        kernel = kernel*kernel';
        kernel /= sum(kernel(:));
        f(:,:,i) = filter2 (kernel, f(:,:,i-1));
    end

    % compute corrections going from filtered to plain signal
    d = zeros([size(g) iter]);
    for i = 2:iter,
        d(:,:,i) = f(:,:,i) - g;
    end

    maxd = max(d(:,:,iter)(:))
    mind = min(d(:,:,iter)(:))

    % for each grid node select the filter giving maximum smoothness without excessive adjustment
    % avoid some noise by morphological transformations
    n = iter * ones (size (g));
    c = d(:,:,iter);
    for i = iter-1:-1:1,
        mask = logical((c > cmax) | (c < -cmax));
        mask = bwmorph(mask, 'clean');
        mask = bwmorph(mask, 'open');

        c(mask) = d(:,:,i)(mask);
        n(mask) = i;
    end

    final = g+c;



    kernel = gaussian(19, 0.1);
    kernel = kernel*kernel';
    kernel /= sum(kernel(:));
    f = filter2 (kernel, final);

    c = f - g;
    mask = logical((c > cmax) | (c < -cmax));
    absolutely_final = f;
    absolutely_final(mask) = final(mask);

    (write reults here)

endfunction
```

A.2 A noise reduction experiment

This is the code referred to in section 4.3

```

% coordinates of observations
A = [0 0];
B = [4 3];
C = [4 0];

% coordinate of the POI
P = (A+B+C)/3;

Dp = [ sqrt(sum((P-A).^2))    % distance from P to A
       sqrt(sum((P-B).^2))    % distance from P to B
       sqrt(sum((P-C).^2)) ]; % distance from P to C



Dn = [  0  5  4      % distances from A to A, B, C
        5  0  3      % distances from B to A, B, C
        4  3  0  ];  % distances from C to A, B, C

% Hirvonen parameters
C0 = 0.5
Ld = 5

% Covariances between the observations
Cn =  C0 ./ (1 + (Dn./Ld).^2)

% Covariances between the POI and the observations
Cp =  C0 ./ (1 + (Dp./Ld).^2)

% Compute weights
w0 = Cn\Cp


% Assume observation at A is noisy
Cn(1,1) += 0.1;

% Compute new weights
wa = Cn\Cp

% Assume all observations are equally noisy
Cn(2,2) += 0.1;
Cn(3,3) += 0.1;

% Compute new weights
wabc = Cn\Cp
```

A.3 A draping experiment

This is the code referred to in section 4.4.

```
% true landscape 10 observations/km for 0..100 km:
X = [0:0.1:100];

Ltrue = 10 + 0.1 * X;

% existing DTM: noisy
Ldtm = Ltrue + sqrt(0.5)*randn(size(Ltrue));

% the inertial navigation system drifts slowly.
Inoise = sqrt(0.0001)*randn(size(Ltrue));
Idrift = cumsum(abs(Inoise));
max(Idrift)

% new observations: less noisy, but drift from the INS
Nobs = Ltrue + sqrt(0.05)*randn(size(Ltrue)) + Idrift;


% box filter kernel
B = ones(1,50)/50;

% Filter the DTM
w = Ldtm;
% pad with boundary values to reduce boundary effects of the filtering
u = [repmat(w(1), size(w)) w repmat(w(end),size(w))];
% do a symmetric filtering to avoid pushing the signal rightwards
f1 = filter(B, 1, u)(prod(size(w))+1:2*prod(size(w)));
u = u(end:-1:1);   % reverse raw signal
f2 = filter(B, 1, u)(prod(size(w))+1:2*prod(size(w)));
f2 = f2(end:-1:1); % reverse result
Fdtm = (f1+f2)/2;

% Then filter the new observations using the same code
w = Nobs;
u = [repmat(w(1), size(w)) w repmat(w(end),size(w))];
f1 = filter(B, 1, u)(prod(size(w))+1:2*prod(size(w)));
u = u(end:-1:1);
f2 = filter(B, 1, u)(prod(size(w))+1:2*prod(size(w)));
f2 = f2(end:-1:1);
Fobs = (f1+f2)/2;

% the correction factor is the long wavelength difference
Corr = Fobs - Fdtm;

% remove INS drift from New OBServations, creating the New DTM
Ndtm = Nobs - Corr;

plot (X', [Ldtm; Nobs; Ndtm; Ltrue; Corr+5]');
grid on;
xlabel ('distance [km]');
ylabel ('height [m]');
legend('Ldtm', 'Nobs', 'Ndtm', 'Ltrue', 'Corr+5');
```

National Survey and Cadastre
Rentemestervej 8
2400 Copenhagen NV, Denmark
http://www.kms.dk