Preface

This report was written to fulfil part of the requirements for my internship that is part of the curriculum of my study in Geodesy at the Faculty of Civil Engineering and Geosciences at Delft University of Technology in The Netherlands. This internship took place in spring 2002 at the Geodynamics Department of The National Survey and Cadastre (Kort og Matrikelstyrelsen, KMS) in Copenhagen, Denmark.

I would like to thank Rene Forsberg for giving me the opportunity to do my internship at KMS and also for his help as my supervisor. I would also like to thank everyone else at the Geodesy and Geodynamics Department for their support, especially Kristian Keller, Nynne Sole Dalå, Klaus Schmidt, Sine Munk Jacobsen and Gabriel Strykowski. Then I would like to thank Rune Carbuhn Andersen and Brian Pilemann Olsen of the Topographic Department for their help on this research and my fellow student Maarten Vermeij for all his help on computer programming.

Further more, I would particularly like to thank Axel, Jacob, Lars, Mathilde, Susanne and Tina for making me feel at home in Copenhagen. And last, but certainly not least, I would like to thank Bianca for being patient with me being away for three months.

Copenhagen, May 24th 2002
Călin Arens

Abstract

This report presents some examples on the use of laserdata by using the Danish national topographic database (TOP10DK) and digital orthophotos. Laserdata from commercial surveys (Fotonor) have been used in a number of areas in Jutland. Laserdata from Møn is also used and is acquired by KMS's own low-cost laser-scanning set-up. The goal of this report is to give a general idea about the suitability of laserdata for certain applications, and hopefully also to be used as a starting point for further research. The focus in this report is on quality control and visualisation of the laserdata, as well as evaluation of the height determination accuracy.

The report starts out with a short overview of airborne laser scanning. Then it provides some practical information about the geographical data used in this research. It also shows how to make 3D visualisations of a certain area by using laserdata and aerial pictures. The main portion treats the height determination accuracy. Comparing the laserdata with the existing height model, some GPS heights and the building heights from the TOP10DK test this. There is also some information about the use of multiple recordings of the returning laser pulse and the amplitude data.

The overall impression of the height determination accuracy is that the best results are in flat and slightly hilly areas. The errors grow in areas with multiple slopes and especially in areas with forest. To give a more accurate description of the laserdata accuracy some research needs to be done on filtering this data and also on checking the planar accuracy. It shows that the multiple recordings are of little value at KMS. The amplitude data could be useful, but more research is needed to reveal this. An important conclusion is that the KMS laser scanning system can be used in most of the applications. This is interesting because this system has a cheap (compared to commercial systems) set-up, and in combination with cheaper software like Surfer (as opposed to 3D Analyst), there is a complete cheap alternative for the commercial systems.

Table of contents

1 Introduction

Airborne laser scanning has become a widespread research topic in the past couple of years (see e.g. [11], [12] or [13]). Until now, commercial laser scanning data have not been available to KMS. With this report I will present some examples on the use of laserdata by using the Danish national topographic database (TOP10DK) and digital orthophotos. Laserdata from commercial surveys (Fotonor) have been used in a number of areas in Jutland. Laserdata from Møn is also used and is acquired by KMS's own low-cost laser-scanning set-up, using a laser scanner from Riegl Gmbh, Austria. This system, which is designed especially for ice sheet applications in Greenland, is based on a Honeywell INS and kinematic GPS and the software is still being developed [10]. Results from the KMS system in Møn are therefore preliminary.

The goal of this report is to give a general idea about the suitability of laserdata for certain applications, and hopefully also to be used as a starting point for further research. Furthermore, there will be some comments on low-budget alternatives for professional laser scanning hardware and software. The focus in this report is on quality control and visualisation of the laserdata, as well as evaluation of the height determination accuracy.

In chapter 2 a short overview of airborne laser scanning is provided for readers that have little knowledge about this measurement technique. Chapter 3 contains a description of the data used in this research, and the pre-processing this data needed. Chapter 4 treats the 3D visualisation thoroughly, and chapter 5 consists of a series of analyses and results on the use of laserdata. Finally, chapter 6 sums up the conclusions and recommendations.

2 Airborne laser scanning

This chapter contains a short description of the principle of airborne laser scanning. The first paragraph introduces the general principle and the second paragraph goes a little deeper into the use of laser pulses.

2.1 Principle

A pulsed laser is directed out of an aircraft and when an earth feature intercepts the emitted laser pulse, it is reflected back to the aircraft (figure 2.1.1). A rotating mirror ensures that the laser over sweeps a larger part of the terrain. Laser pulses may, for example, reflect off trees or penetrate the canopy and be reflected back from the ground. The time interval between the laser pulse is emitted from the aircraft and the return of the terrain-reflected-pulse to the sensor is measured precisely. This gives the range between the aircraft and the terrain. During the flight the position as well as the attitude of the aircraft are also measured with respectively the Global Positioning System (GPS) and an Inertial Navigation System (INS). In post-flight data processing, these measurements are combined with the laser range and the angle of the mirror to compute the height of the terrain [9]. The typical accuracy of commercial systems is around 10 cm, but except for well-defined areas, this accuracy is hard to maintain. This is due to the problem that you do not exactly know where the laser pulse is reflected.

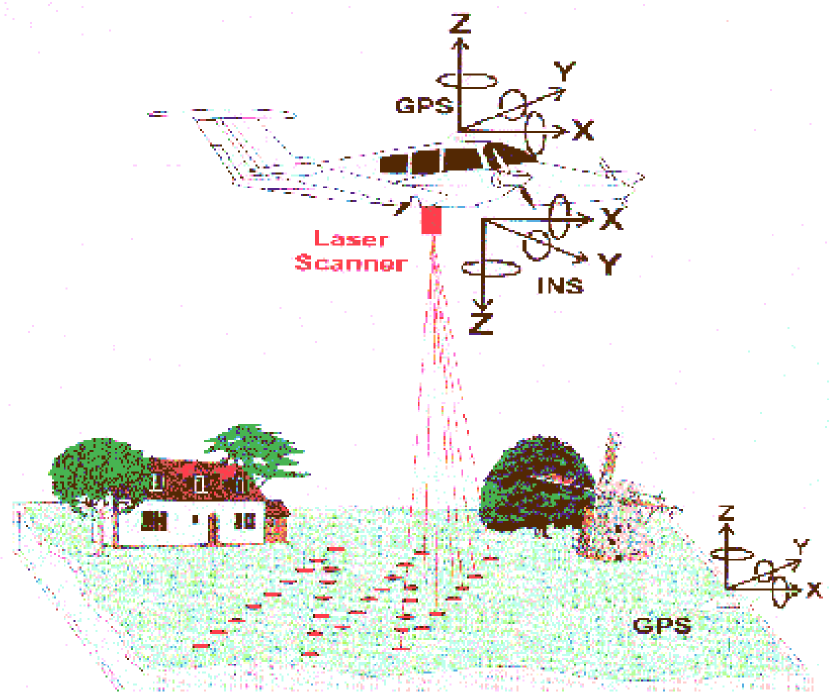


Figure 2.1.1: Principle of airborne laser scanning (source: [1]).

The laserdata also contains information about the reflectivity of the terrain target indicated by the amplitude of the reflected signal.

2.2 Laser pulses

Figure 2.2.1 shows a laser beam in red. It is easy to see that not all the light will reflect on the same feature. This is especially the case when there is vegetation in the area. Some part of the light will reflect on the tree canopy and other parts of the light will reflect from some of the lower branches or even penetrate to the ground.

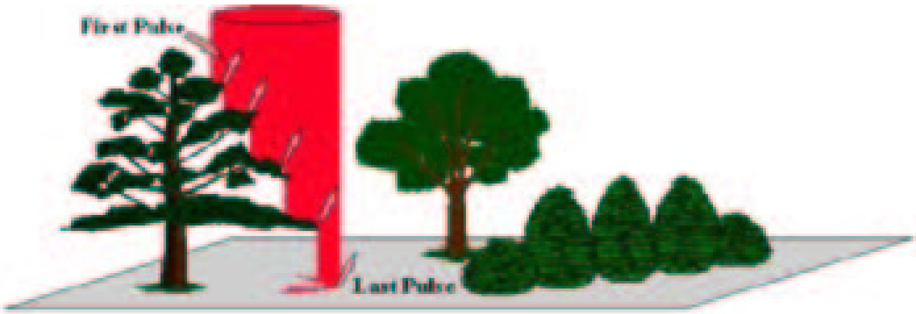


Figure 2.2.1: Reflecting behaviour of a laser pulse (source: [2]).

Some laser scanning systems are able to record this behaviour. In this case, one of the systems only recorded the last significant part of the returning pulse and the other system recorded the first and the last part. There are also some systems on the market that are able to record the whole returning spectrum. In figure 2.2.2 the emitted pulse is shown in purple and the return pulse in green. The laser scanning systems record a returning pulse based on its amplitude.

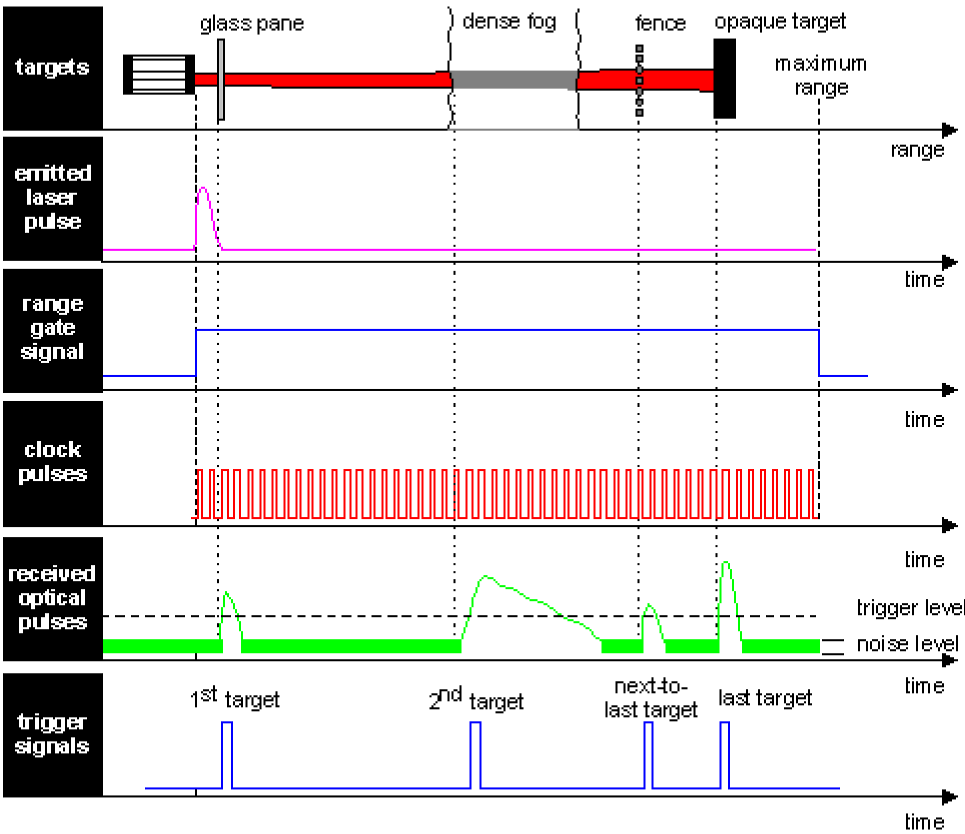


Figure 2.2.2: Laser pulses, the outgoing pulse is shown in purple, the returning pulse in green (source: [3]).

To be able to record two pulses, in some systems, there needs to be a certain minimum height difference (2m.) between these returning two pulses. You can use this information to detect vegetation (see Chapter 5).

3 Data collection and processing

Before starting the analysis of the laserdata, a lot of data preparation is needed. Not only must the laserdata be processed into a readable form for ArcView (the main analysis tool used here), but additional data also needs to be processed. This additional data consists of the Danish national topographic database (TOP10DK), aerial photographs that are mainly used for ground truth and visualisation, and GPS data that is used to check all the other data. For this report data collection is concentrated on 7 test areas with different characteristics on which the analysis will find its place.

In the first paragraph the location and characteristics of the test areas are shown. The following paragraphs contain information about the data that is used and how it is processed. The last paragraph deals with the problems that arise with the process of data fusion.

3.1 Test areas

For the analysis of the laserdata there are seven test areas, each with their own characteristics (see Appendix C for exact coordinates). Four of them are located in Foulum, two of them in Hvorslev and one of the areas is in Møn (figure 3.1.1). The Foulum and Hvorslev laserdata are from Fotonor, using an Optech laser scanner. The Møn data were acquired by the KMS Riegl laser system.

Figure 3.1.1: Denmark with the test areas near the arrows.

The first six test areas (figure 3.1.2) are chosen for their terrain characteristics. These areas each feature a different kind of terrain (table 3.1.1). Because the quality of laserdata is dependent on features like the slope of the terrain and the presence of forest, the chosen areas either have or do not have a combination of these characteristics.

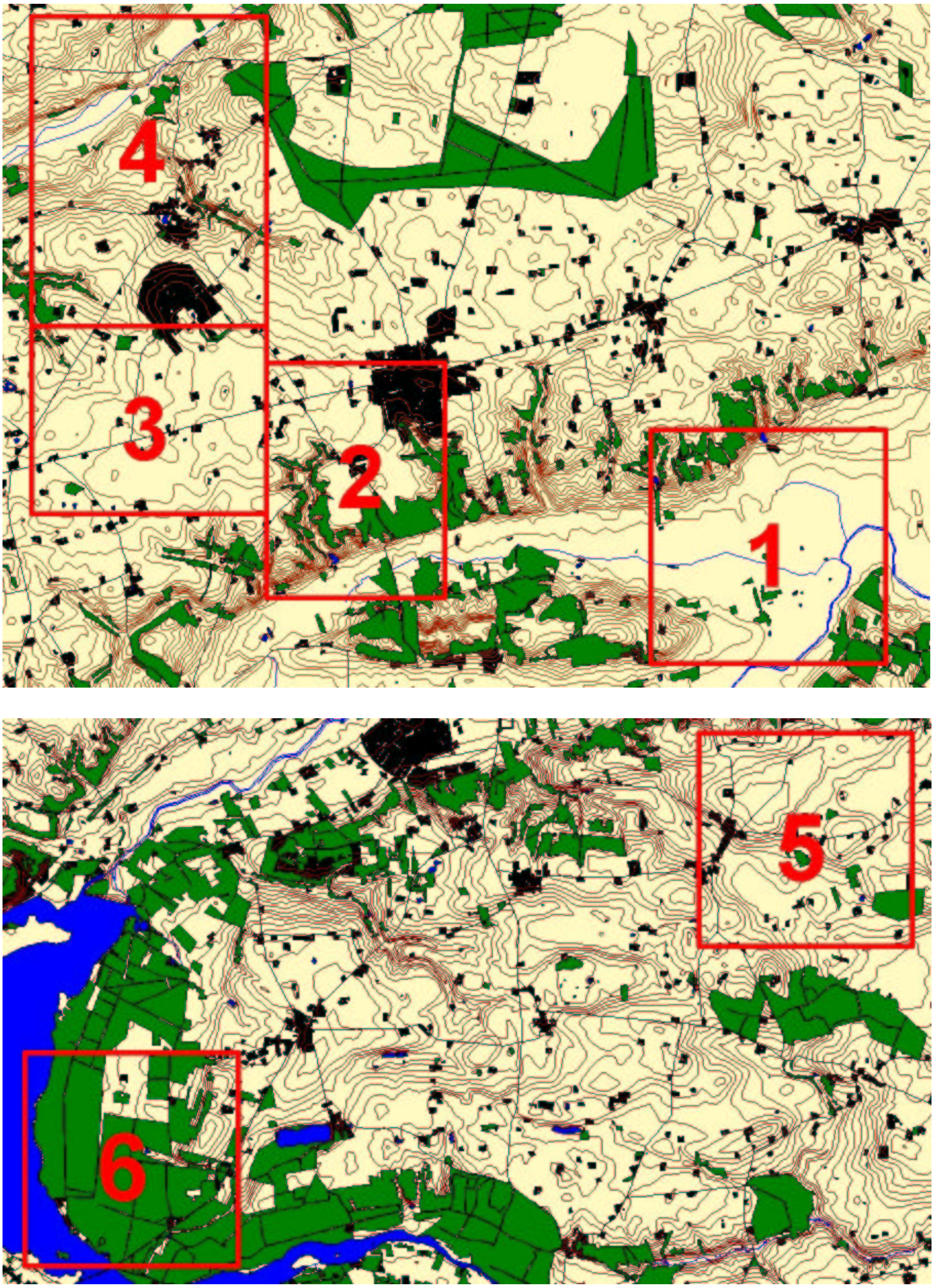


Figure 3.1.2: Six test areas in Foulum (top) and Hvorslev (bottom). The areas in red cover an area of about 2,5 x 2,5 km each. The height contour interval is 2,5 meters.

Table 3.1.1: Terrain characteristics test areas.

Area	Terrain characteristics
1	Very flat – river valley
2	Very steep – one slope
3	Slightly hilly
4	Very hilly – multiple slopes
5	Smooth hills
6	Forest - plantation

The Møn area is a special area, because it until recently was the only area where KMS has its own laserdata. The area includes a beach, a swamp, some forest and a little town with a harbour (figure 3.1.3).

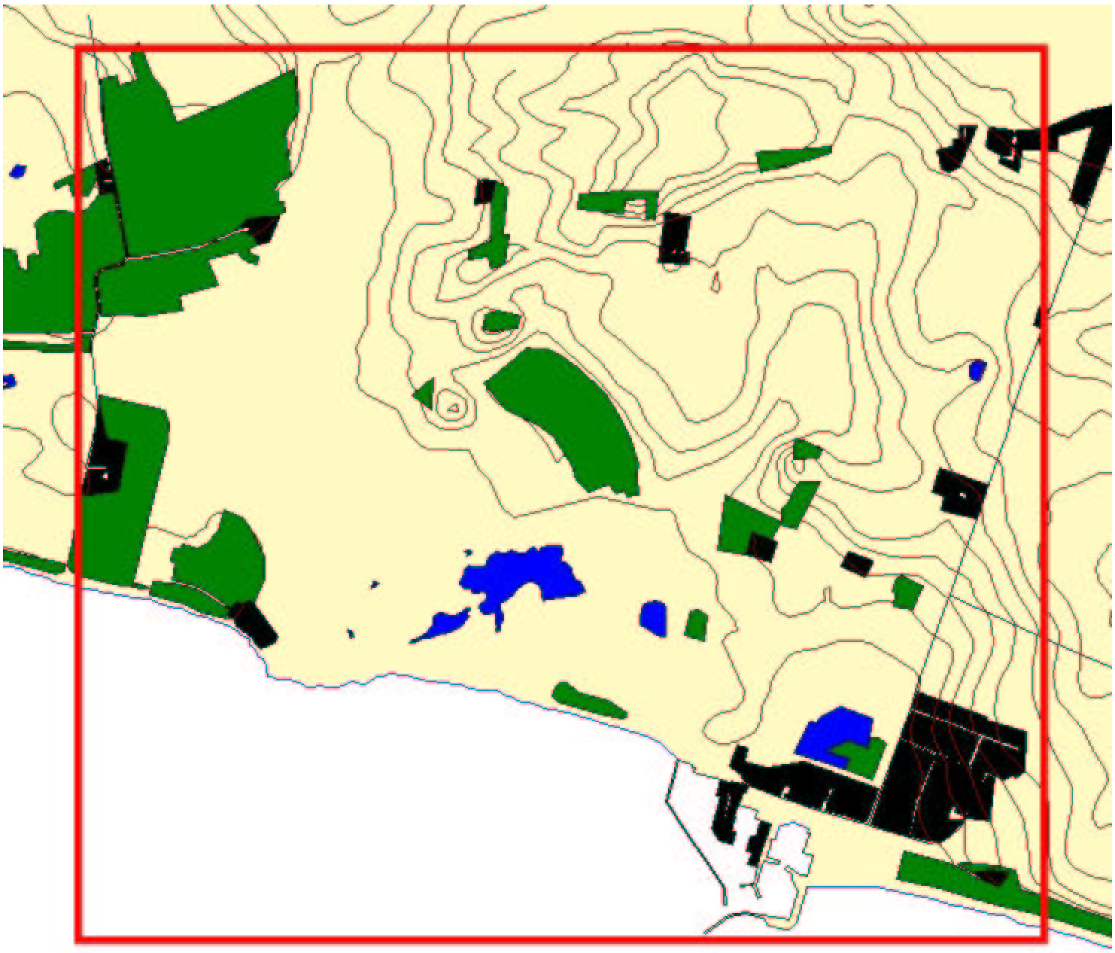


Figure 3.1.3: Møn test area with Klintholm Harbour on the right. The red area covers 2,5 x 2,5 km. The contours are placed every 2,5 meters.

All test areas have a size of about 4-5 km^2. Doing analysis on areas of that size takes quite some time. That is why for the initial analysis (where a lot of the work has to be redone, because of larger and smaller errors), a smaller area of about 500 x 500 m has been selected. This area is located in the centre of area 5 and contains some forest, some buildings, some roads and a little lake. A second subset out of area 6 has also been chosen. This smaller area contains some steeper slopes and more vegetation.

3.2 TOP10DK

The TOP10DK is a digital topographic base map, which covers the whole area of Denmark on a 1:10.000 scale. It holds basic information regarding landscape, municipal boundaries, names etc. (figure 3.2.1). The TOP10DK is constructed as a vector map, which makes it suitable for use in a GIS. The TOP10DK has been established with the purpose of forming a topographical basis, to be a frame of reference for other geographical registration and to form a basis for the topographical map production.

Figure 3.2.1: TOP10DK in standard colours.

The source of the data is mainly new measurements from photogrammetry, and if necessary existing technical and topographic maps were used. The accuracy is about 70 centimetres in the plane as well as the heights.

Clipping the right areas from bigger files generated the TOP10DK data for the test areas. There are multiple files for the different categories, like a file with all the roads, all the water lines, all the buildings, etc. The files are in the ESRI 3D shapefile format and they use the UTM 32 WGS84 reference system.

The TOP10DK is much alike the Dutch TOP10vector, but because Denmark does not have a real countrywide equivalent to the Dutch Large Scale Base map (GBKN), there are more features in the TOP10DK than in the TOP10vector. Further more, there are some minor differences. An example: the roads in TOP10DK are not available in polygon form.

3.3 Laserdata

The laserdata is simply a dataset with a lot of points containing x-, y- and z-coordinates. The heights of those points are measured with an airborne laser scanner (see Chapter 2). The laserdata can be used to generate a Digital Elevation Model (DEM), which can be used for many applications, like forest management, floodplain management, shoreline surveys and visualisation (figure 3.3.1).

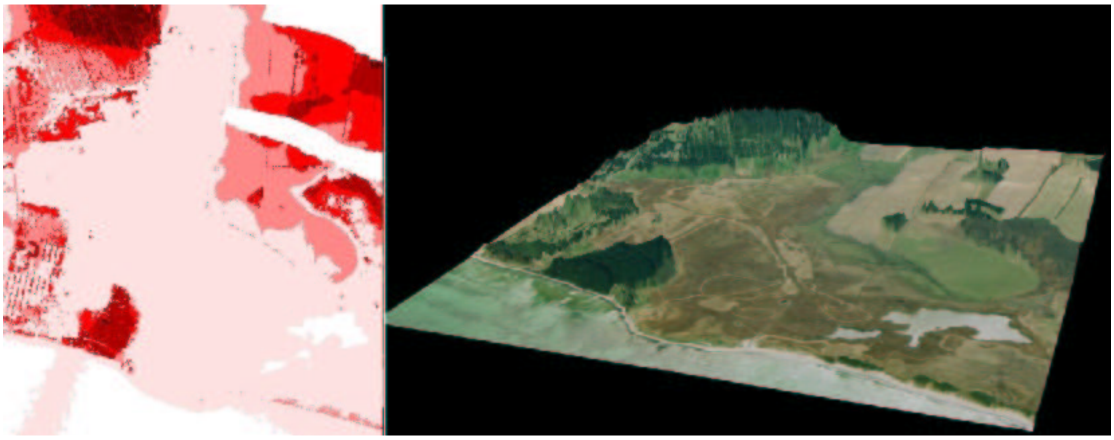
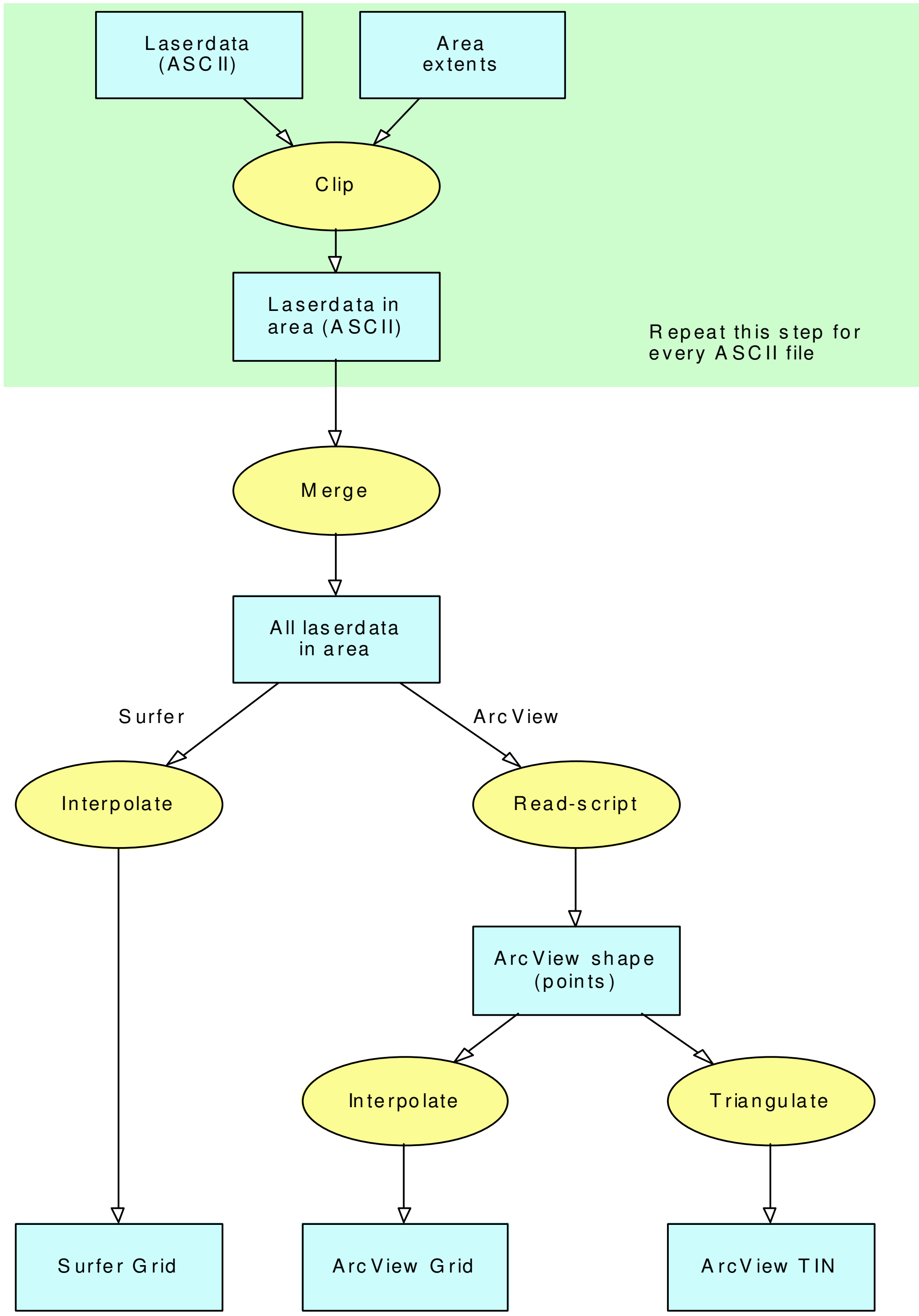


Figure 3.3.1: Laserdata showing heights (left) and a 3D visualization with an aerial picture draped on top of the laserdata (right) in the Møn area.

The laserdata was delivered in ASCII files. These files contain the x, y and z-coordinate of the points in UTM 32 WGS84 and additional information about the return signal (amplitude). Because of the many points in the datasets, processing takes a long time. First the ASCII files were converted from ellipsoidal to orthometric heights using the Danish geoid model DKGEOID98. Two programs that subtract the local geoid heights from the ellipsoid heights did this (Geoip and KMSTrans, see Appendix B.3 and B.4). Then all the ASCII files were read into the ESRI shapefile format with the help of Avenue scripts (see Appendix A.1 and A.2).

There were actually two different types of laser systems used. The first, for the Møn dataset, is developed by KMS itself. It records the last significant part of the returning pulse and the points also contain information about the number in scan line. This dataset contains about 1.5 points/m^2. The second dataset, from an Optech system operated by Fotonor A/S, and covers the Jutland test areas. This system records the first and the last significant part of the returning pulse. This gives us information about the existence of vegetation in the areas, because in those areas there is a difference between the heights derived from the first pulse (reflected on the tree canopy) and the second pulse (reflected from the ground or from branches). The shapefiles derived from these data were extended with a field containing the difference between the two heights and also an extra field for the second height (figure 3.3.2). These datasets contain about 0.2 points/m^2.



Flowchart 3.3.1: Pre-processing the laserdata.

Attributes of Lidarsel.shp

Shape	Id	East1	North1	Eheight1	Nheight1	Amp1	East2	North2	Eheight2	Amp2	Hghtdiff	Nheight2
Point	128293.000000	545799.30	6246372.70	103.33	64.05	52	545799.39	6246372.55	103.87	52	-0.54	64.59
Point	128293.000000	545798.41	6246373.15	103.36	64.08	27	545798.50	6246372.99	103.95	27	-0.59	64.67
Point	128293.000000	545799.53	6246370.61	103.35	64.07	31	545799.62	6246370.46	103.90	31	-0.55	64.62
Point	128293.000000	545798.87	6246364.18	103.46	64.18	49	545798.96	6246364.04	104.01	49	-0.55	64.73
Point	128293.000000	545797.55	6246367.12	103.46	64.18	44	545797.64	6246366.98	103.97	44	-0.51	64.69
Point	128293.000000	545796.22	6246370.09	103.38	64.10	46	545796.31	6246369.93	103.97	46	-0.59	64.69
Point	128293.000000	545794.90	6246373.02	103.43	64.15	47	545794.99	6246372.88	103.95	47	-0.52	64.67
Point	128293.000000	545794.17	6246373.13	103.29	64.01	24	545794.25	6246372.98	103.82	24	-0.53	64.54
Point	128293.000000	545795.26	6246370.67	103.43	64.15	24	545795.36	6246370.51	104.02	24	-0.59	64.74
Point	128293.000000	545796.43	6246368.01	103.33	64.05	24	545796.52	6246367.86	103.90	24	-0.57	64.62
Point	128293.000000	545797.58	6246365.43	103.47	64.19	23	545797.67	6246365.28	104.03	23	-0.56	64.75
Point	128293.000000	545798.71	6246362.88	103.50	64.22	23	545798.80	6246362.74	104.05	23	-0.55	64.77

Figure 3.3.2: Part of the Fotonor AS dataset containing two heights in each recording.

In Denmark, airborne laser scanning is a fairly new technology. The National Survey and Cadastre (KMS) has since August 2000 developed a laser scanning system around a Riegl laser, primarily for applications in Greenland. KMS has only flown 4 Danish test areas, with only Møn processed until now. The Danish company Kampsax operates a commercial laser scanning system (TopoSys) as well, with emphasis on urban applications, e.g. telecommunication companies use that data to plan their mobile networks. In Holland there is a laser dataset covering the whole country. It is called the AHN. There are also regular surveys for coastal monitoring.

3.4 Aerial Photographs

There are aerial photographs covering all Denmark. The aerial photographs of the test areas are available as scanned photos in the JPG format (figure 3.4.1). Their scale is 1:10.000 (For the Møn area: 1:25.000). These images are not geo-referenced yet and also need to be converted to orthophotos (flowchart 3.4.1). To complete this task, I used OrthoEngine 6.3 Airphoto Edition developed by PCI Geomatics.

Figure 3.4.1: Scanned aerial photograph with the fiducial marks along the sides.

In order to create the orthophotos, you need a Digital Elevation Model (DEM). I derived this DEM from the laser scanning data. First, the points were converted to a 10-meter grid and then a file with 5-meter height contours was derived (figure 3.4.2). This file was exported to the DXF format, so that it can be used in OrthoEngine. This part of the step was completed with ArcView 3D Analyst. To convert the contours to a raster DEM you can simply use the appropriate function in OrthoEngine. This strange conversion from grid to contours to grid is done as a workaround as it was difficult to import the grid directly into OrthoEngine.

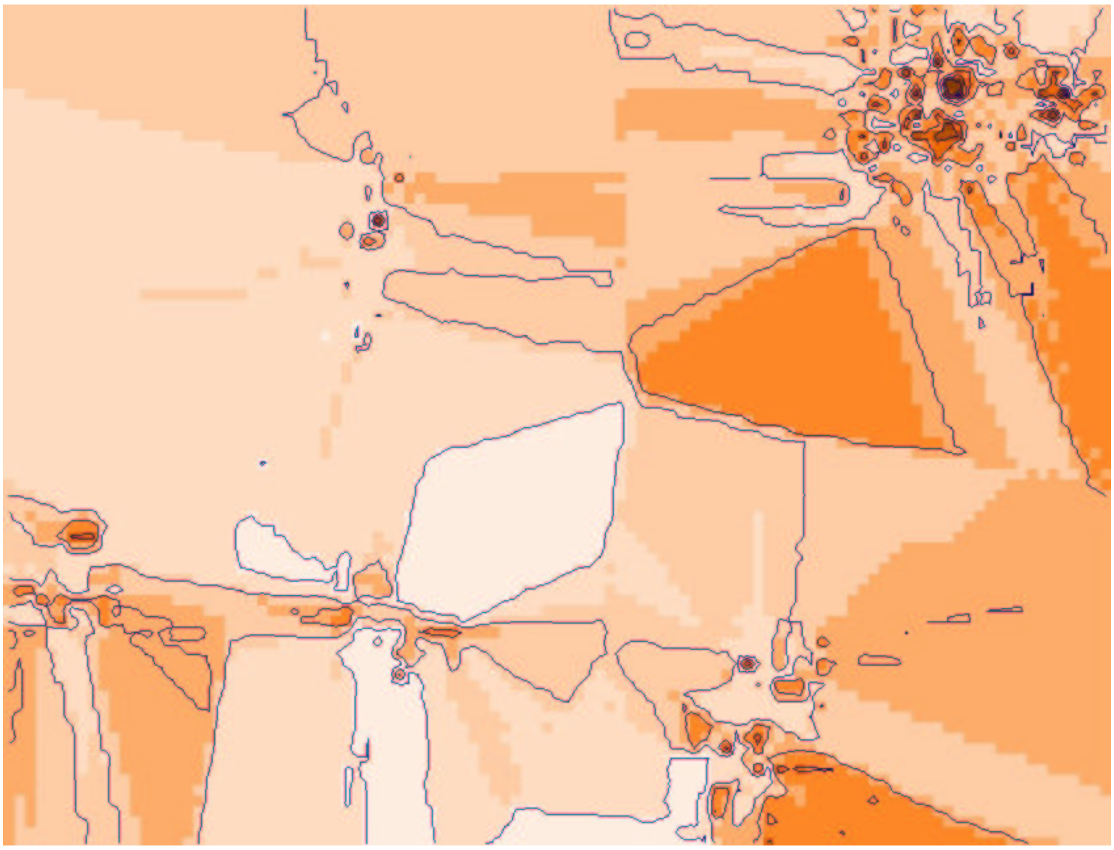
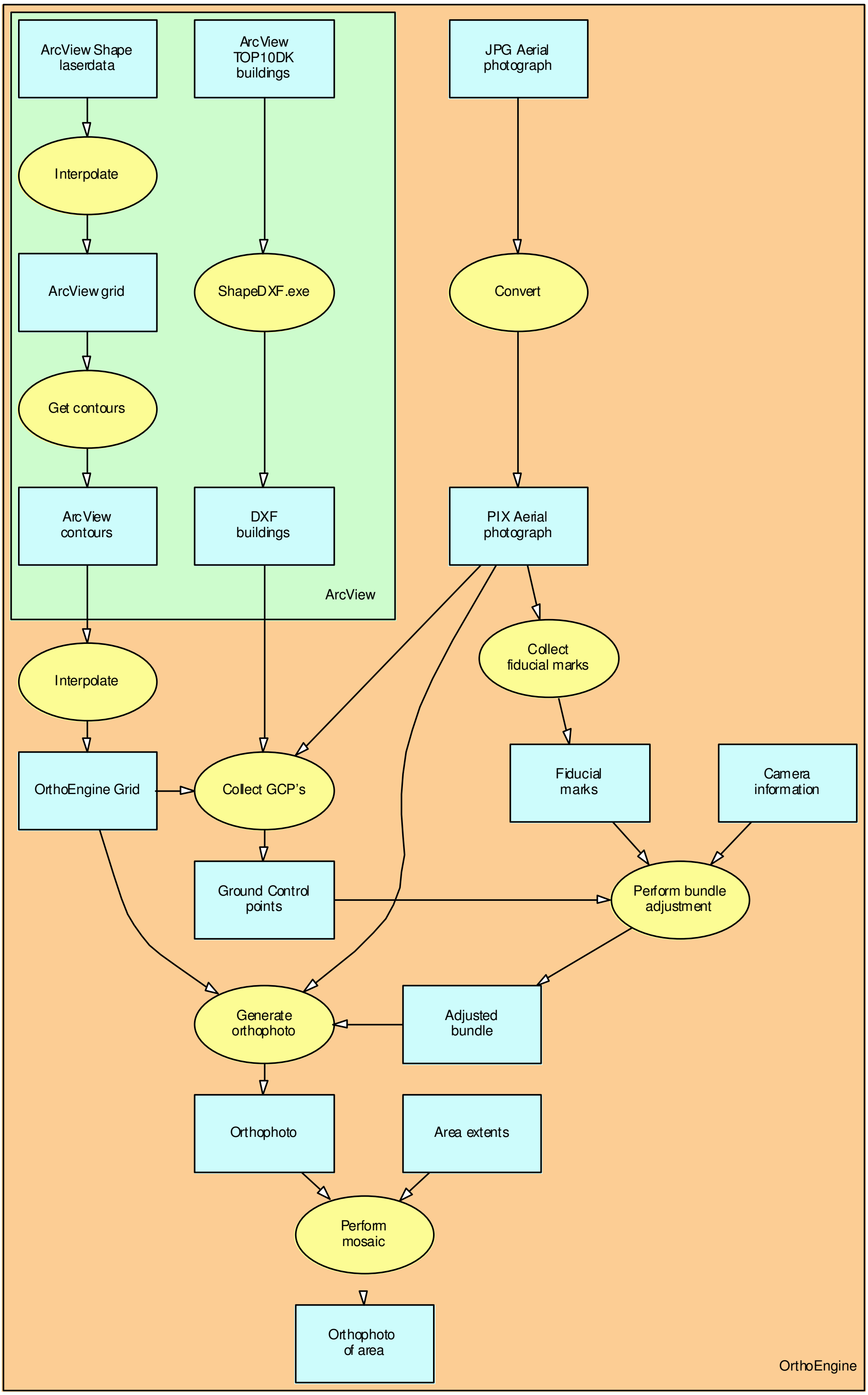


Figure 3.4.2: DEM (orange tints) with height contours (blue).

Because the JPG format uses a lot of compression for the scanned photos, OrthoEngine processes these very slowly. The solution for this problem is to convert the JPG images to PIX images. PIX is the format that is used by OrthoEngine. You need a lot of disk space for this, because a typical scanned aerial photograph in the JPG format, scanned at 1200 DPI, is around 40 MB in size and the file size in the PIX format is 350 MB. The processing speed of the PIX format is worth the file size though.

It is also necessary to prepare a DXF file with the buildings and roads derived from the TOP10DK. With this file it is easier to collect the Ground Control Points (GCP's). This is also easily done with ArcView. You can simply use the ShapeDXF.exe program in the BIN-directory.



Flowchart 3.4.1: Orthophoto generation.

Now, the processing can begin. First you need to specify some information for your project, such as the name and the output projection. I used UTM zone 32 WGS84. The output pixel spacing is the same as the scanning resolution, i.e. 1200 DPI or about 0.300 micron. The camera calibration is supplied in a document that was included along with the aerial images. Among others, you need to enter the focal length and the radial lens distortion.

The next step is to open the scanned images and collect the fiducial marks. This is a simple process of mouse clicking. After that, the GCP's for each image are collected, using the DXF vector file and the raster DEM file for respectively the planar and height coordinates. Collecting the tie points works the same way. The last step in this phase is to perform the bundle adjustment, which can be done by one mouse click and is computed automatically.

Then you can generate orthophotos. This task just requires you to push a button and wait for a while. The final step is putting the orthophotos together in a mosaic of the test areas (figure 3.4.3).

Figure 3.4.3: Orthophoto with twists on both sides as result of the processing.

The result picture is then converted to the TIFF format, which can be used in ArcView. The photograph is now corrected and can be used as a source for ground truth (i.e. for checking the results of the analysis) and for visualisation.

3.5 GPS

The GPS data may be used to check the heights in the other datasets, as the GPS measurements are more accurate than any of the other measurement techniques. The GPS data for the Møn test area was already available in an ASCII file containing x-, y- and z-coordinates. These data were provided by Carl Bro company and the local county (Storstrøms Amt). The same procedure as with the laserdata is used to convert the data to an ArcView readable format. There was no GPS data available for the other 6 test areas, so we decided to do some measurements ourselves (figure 3.5.1). We picked out parts of area 2, 3, 4 and 6 to do the survey. This ensured that there were GPS measurements available in all kinds of terrain. The measurements consist of two planes and a lot of profiles. The locations are chosen to contain different landscapes.

Figure 3.5.1: GPS reference station.

To get the GPS measurements in ArcView the same procedure is followed as with the laserdata. The Avenue script needed a little adaption, which is shown in Appendix A.3.

3.6 Problems with data fusion

All the processing on the above datasets is needed to fuse the data, or to make it work with each other. The most important thing is to have the data in the same reference system. In this case the UTM zone 32 WGS84 with orthometric heights is chosen, because most of the data was already in this form, or could easily be transformed to this reference system.

Then there is the problem of differences in accuracy. The data will not perfectly overlap each other, because of differences in the used measurement techniques. Table 3.6.1 shows the accuracy of the different datasets.

Table 3.6.1: Accuracy of different datasets.

Dataset	**Measurement technique**	**Accuracy**
TOP10DK	Mostly derived from aerial photographs	± 70 cm.
Laserdata	Airborne laser scanner	± 15 cm.
Orthophotos	Aerial photo camera	Depends on DEM and GCP's
GPS data	GPS	± 5 cm.

The trouble with visible differences in the datasets is always which dataset to 'believe'. In this case one should believe the dataset with the most accurate measurement technique.

Then there can be an inconsistent accuracy within a dataset. The laserdata for instance, shows a decrease in accuracy in areas with steep slopes or forest.

Finally, the time of recording is very important. The landscape changes, and seasonal change in vegetation will affect the laserdata. There can be new buildings, or the forested areas can change. This shows how important metadata (i.e. information about the dataset, like the reference system and the time of survey) is.

4 3D Visualisation

Airborne laser scanning delivers a point cloud. These points in 3D can be used to visualise the data and with that, visualise the area that is surveyed. This 3D visualisation can be used for a large amount of reasons. The first paragraph of this chapter contains information about the 3D Data and its data structures. In paragraph 4.2 the most important 3D visualisation are shown, i.e. draping an aerial picture on top of the laserdata. The last paragraph includes some other forms of 3D visualisation.

4.1 3D Data

Once the data is collected and processed, it is possible to create any visualisation within a short time, depending on the size of the dataset. There are basically three forms in which 3D data exists:

Point cloud

A point cloud is a (large) collection of points consisting of an x-, y- and z-coordinate. Figure 4.1.1 shows a point cloud. It is easy to see that this is not an ideal representation of the outside world. This is because there is not a surface, just a large number of individual points.

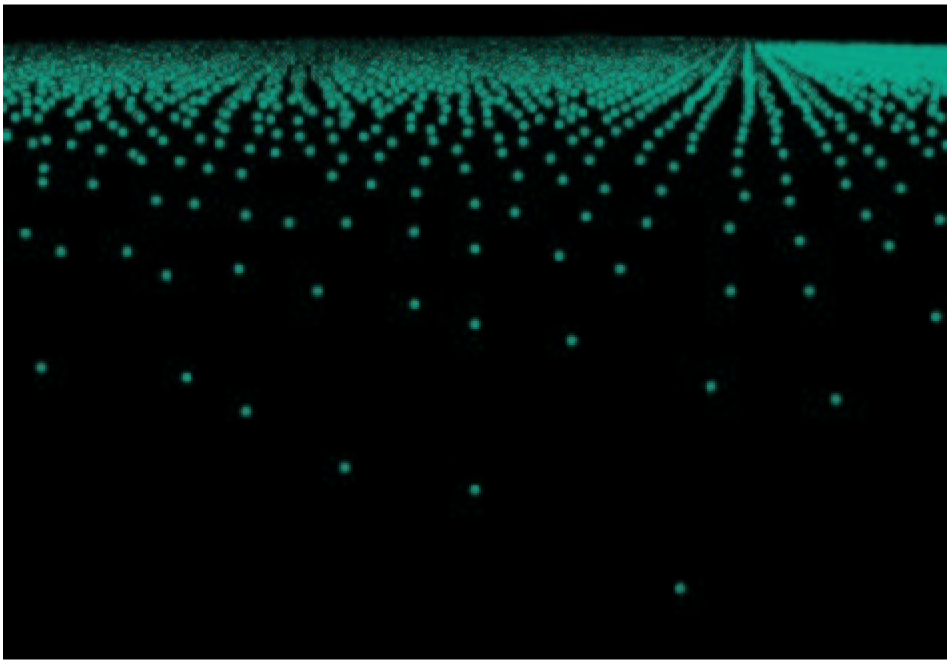


Figure 4.1.1: 3D Point cloud

Grid

Grids can be constructed from point clouds. First you choose a grid cell size and then the heights from the point cloud will be interpolated into this grid. Grids can be used in many applications. In this research the main tool is ArcView. Surfer 8 is also used. The interpolation methods consisted of respectively the Inverse Distance Weighted method and Kriging. In figure 4.1.2 you can see a picture that is made by connecting the surface heights from a grid.

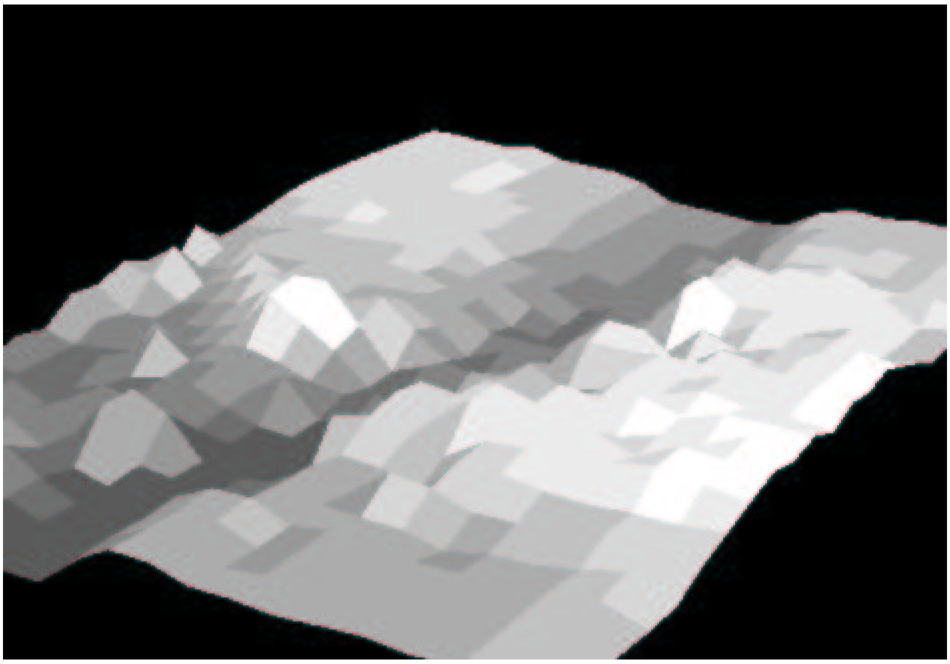


Figure 4.1.2: A visualisation made from a grid.

Triangular Irregular Network (TIN)

As the name indicates, a TIN is a network set up by irregular triangles. With the point cloud as a basis, forming triangles within the original point data forms the TIN. This gives better detail than a grid, but is also computational more expensive. Figure 4.1.3 shows a TIN.

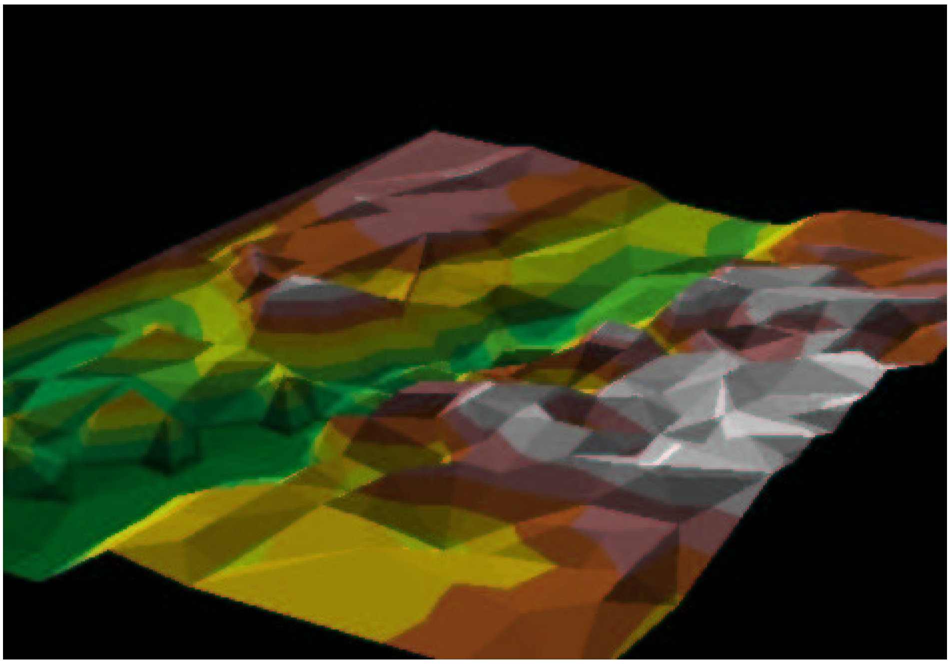


Figure 4.1.3: A colour-coded TIN.

4.2 Drapes

Drapes are an often-used form of 3D visualisation. The idea is to let an image (in our case aerial photographs) follow the surface of the laserdata. You can use either a grid or a TIN and drape an aerial photograph on top of the surface. This is an easy process, because this function is build in ArcView 3D Analyst (flowchart 4.2.1) as well as Surfer 8 (flowchart 4.2.2). Figure 4.2.1 gives an example.

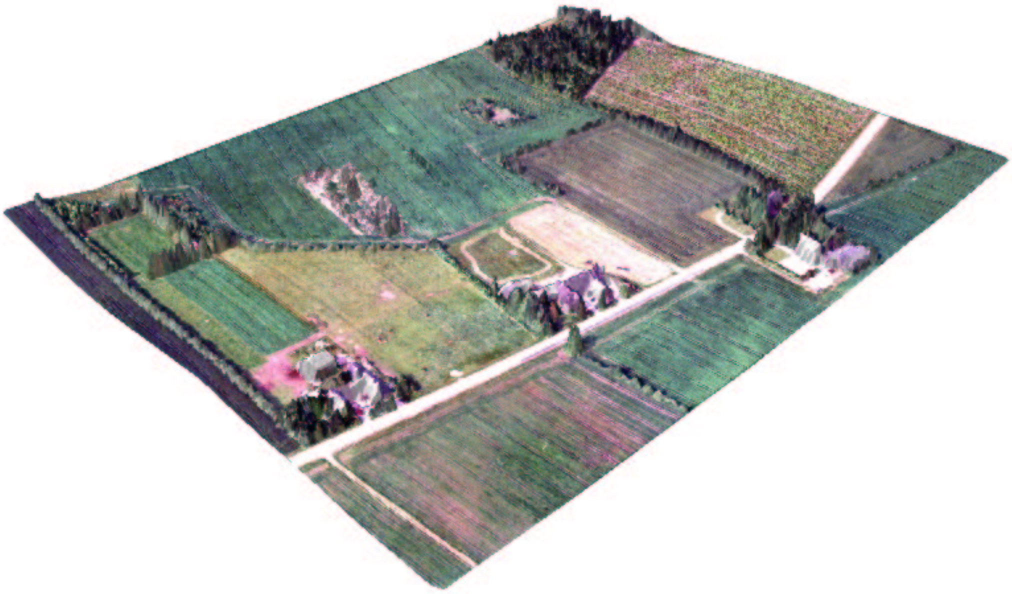
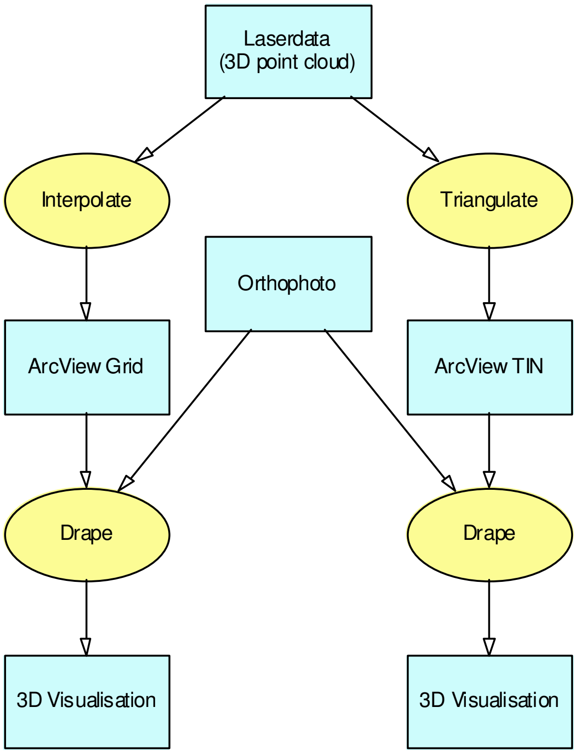


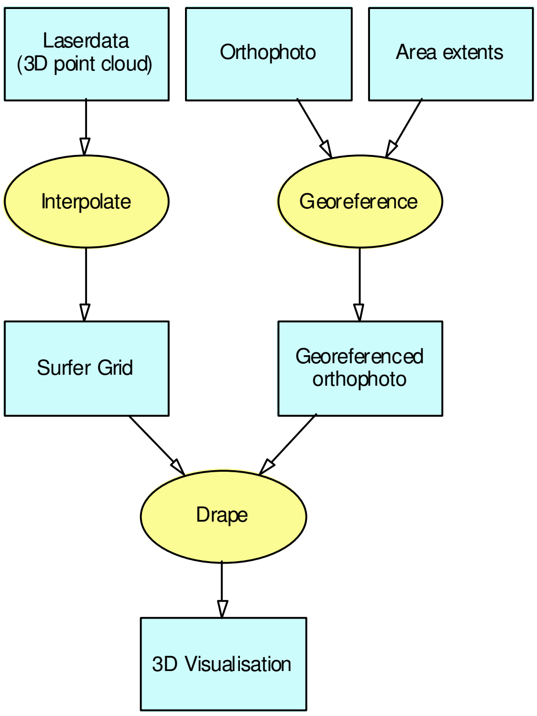
Figure 4.2.1: An aerial picture draped upon a TIN.

Using a grid or a TIN gives slightly different results. A grid usually works faster but has fewer details. A TIN is a lot slower, but provides better results, especially on vegetation and rooftops.



Flowchart 4.2.1: 3D Visualisation in ArcView.

Both ArcView 3D Analyst and Surfer 8 give the possibility to exaggerate the heights to make the 3D visualisation look better. Another option is to make the scene more real by adding light and shadows.



Flowchart 4.2.2: 3D Visualisation Surfer.

4.3 Other output

Once you have made a 3D visualisation (either with or without a draped image), there are two more possibilities of distributing it:

Movie

It is possible to export the 3D visualisation to an image (most commonly in the Jpeg format). If you have a series of these images, you can show them consecutively thereby making a movie. Exporting the images requires a lot of patience. The way to go is to turn your 3D visualisation just a bit and then export an image, and repeat this step for a (large) number of times. I used Slide Show Movie Maker [4] to make an AVI movie from Jpeg's. This program is freeware. The user interface is not great, but after 15 minutes of trying, it turns out that it is very easy to make a movie.

It is also possible to extent ArcView or Surfer with the function to do this automatically. For ArcView 3 there is already a download available [7]. This Flyby Animation Builder will let you specify a path of motion and then exports the 3D visualisation to an Mpeg movie that follows this path.

VRML

VRML stands for Virtual Reality Modelling Language [5]. It describes virtual worlds that can be browsed in an Internet browser with a certain plug-in. I used Cosmo Player [6], but there are many more plug-ins available. The ArcView 3D Analyst gives you the possibility to export the 3D visualisation to a VRML world. This will create a VRML file (Wrl format) and some Jpeg images. If you load this VRML world, you have the ability to move freely through your 3D visualisation.

5 Analyses and results

With all data collected and processed, using 3D visualisation techniques we have a good idea of what the terrain looks like in reality. In this section I will do some analyses on the data and determine by analysing the results how suitable the laserdata is for certain applications.

In paragraph 5.1 I will compare the laserdata with the existing height model of Denmark (based on digitised contours of older maps). In paragraph 5.2 the inconsistencies in the accuracy of the laserdata are discussed. The following paragraphs are about the use of multiple recordings and the amplitude of the returning laser pulse. This chapter ends with a brief comparison between the laserdata and the elevated TOP10DK features.

5.1 Laserdata versus old height model

The Danish height model is based on old measurements, dating back to the last century. I will here evaluate the height model by comparing it with laserdata (flowchart 5.1.1).

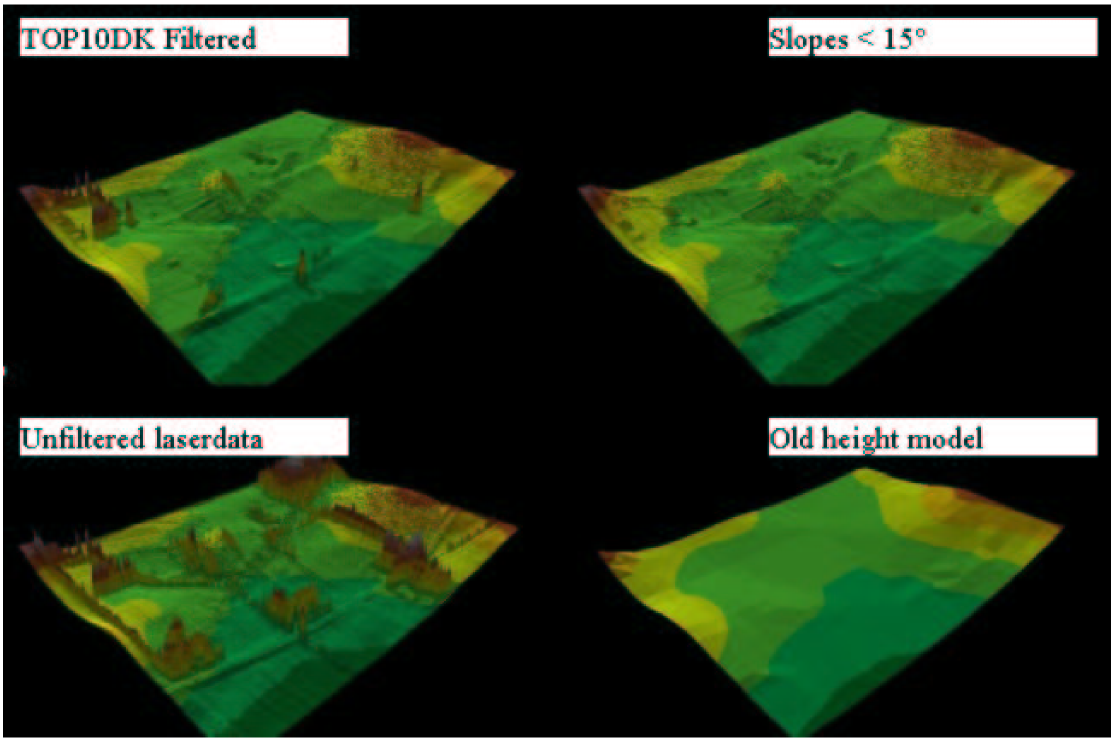
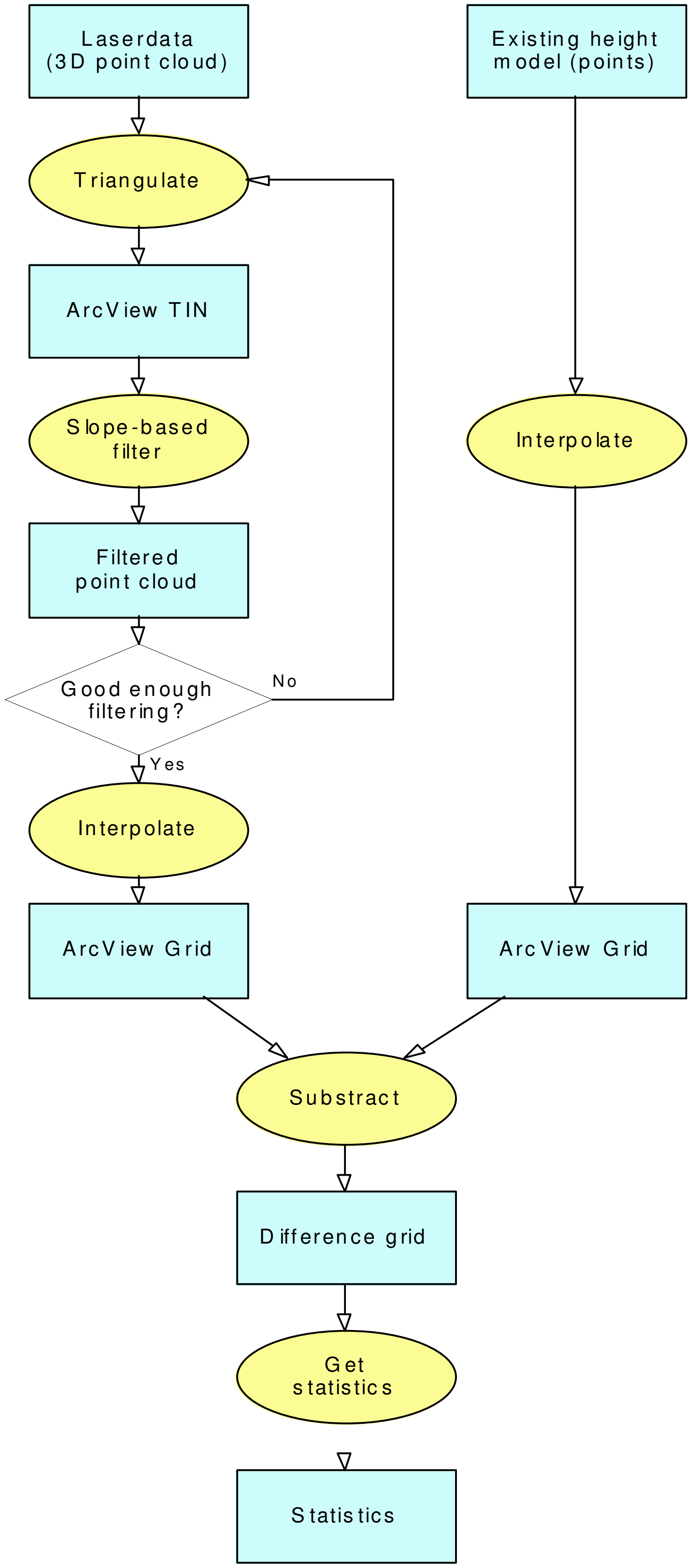


Figure 5.1.1: The steps needed to compare the laserdata with the old height model.



Flowchart 5.1.1: Comparing laserdata with the existing height model.

The way to do this comparison is to subtract two grids, one containing the old height model, the other containing the laserdata heights. Here the first problem arises. The old height model consists of measurements on the ground and laserdata contains measurements on top of buildings, vegetation, cars, etc. So these surface elements have to be filtered out first. Because filtering is a whole research subject in itself, I did the filtering with a visual check. In areas with a few buildings and little forest it is possible to cut out these parts of the laserdata with the features from TOP10DK and interpolate new values in these areas. For city and forest areas this will not work, because there is no data left to interpolate. Here I used a slope-based filter [8] that simply filters out points that have a large height difference compared to their neighbouring points (see Appendix A). To get the right slope value, you can make a slope map and make two classes: one above this value and one below. Then look at the aerial photograph and see what slope value would be best. The steps of this paragraph are in figure 5.1.1. Note that the heights are exaggerated five times in the images to emphasise the details.

Now, the laserdata can be interpolated to a grid (the filter requires the data to be in a TIN). If we subtract the grid of filtered laserdata from the grid of the old height model, we get the image as in figure 5.1.2. The heights are quite similar, showing the high quality of the old data. Only the red areas are significantly different - other differences are within the measurement accuracy. These errors are likely due to the different resolution of the laserdata and the old height model, or due to errors in the filtering algorithm.

The problem with testing the accuracy on grids is that you get a more overall accuracy rather than point accuracy. It might be better to work directly with TIN's, but for this research it is sufficient to work with grids, because the old height model is already a grid and the mean value for the difference still gives a good idea about the real accuracy.

Figure 5.1.2: Old height model minus laserdata heights. See Appendix C (area 5) for the geographical limits of this area.*

To give a single value for this comparison I use the mean and standard deviation of the resulting grid. This had to be done by an Avenue script (see Appendix A), because ArcView standard only computes neighbour statistics. The particular values for this area were:

- Mean: -0,075
- Standard deviation: 0,410

This implies that the overall height model accuracy is pretty much the same. Laserdata often have a similar error value, because of the presence of low grass for example. These are the results in a relatively flat and easy to filter area.

When moving to an area with steep slopes and forest, it is first of all more difficult to filter. The TOP10DK cannot be used, because there will not be any data left. The slope-based filter should not filter out too many slopes. The solution is to let the filter run multiple times until the surface looks smooth. In figure 5.1.3 the filter is run 2 times with a maximum slope of 30º. The third run did not reject many more points.

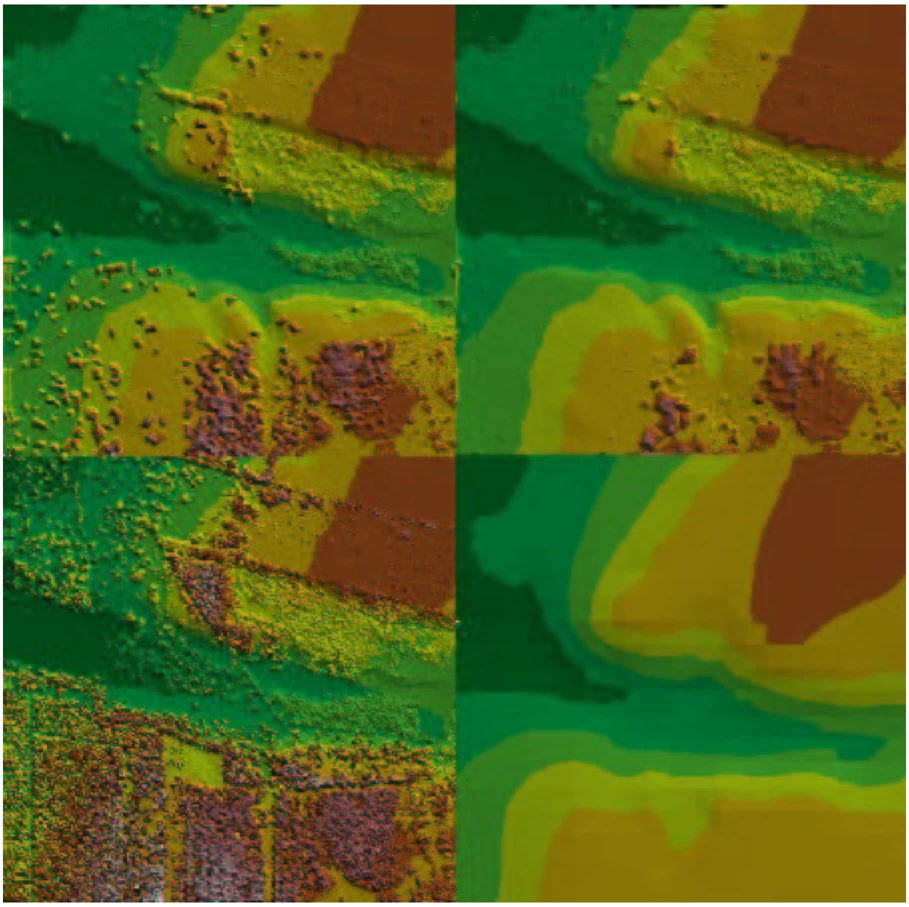


Figure 5.1.3: Clockwise, starting at '6': unfiltered laserdata, first run, second run, old height model.

There is still some noise after the second run, but this data will be interpolated to a grid and be compared with the old height model grid. This results in figure 5.1.4. This does not look good at all. This area needs a new filtering technique to do the comparison.

- Mean: 0,126
- Standard deviation: 2,855

The mean still looks good, but the standard deviation is way too high in this area. In the next section some more information about the accuracy of laserdata in forest and steep-sloped areas are given.

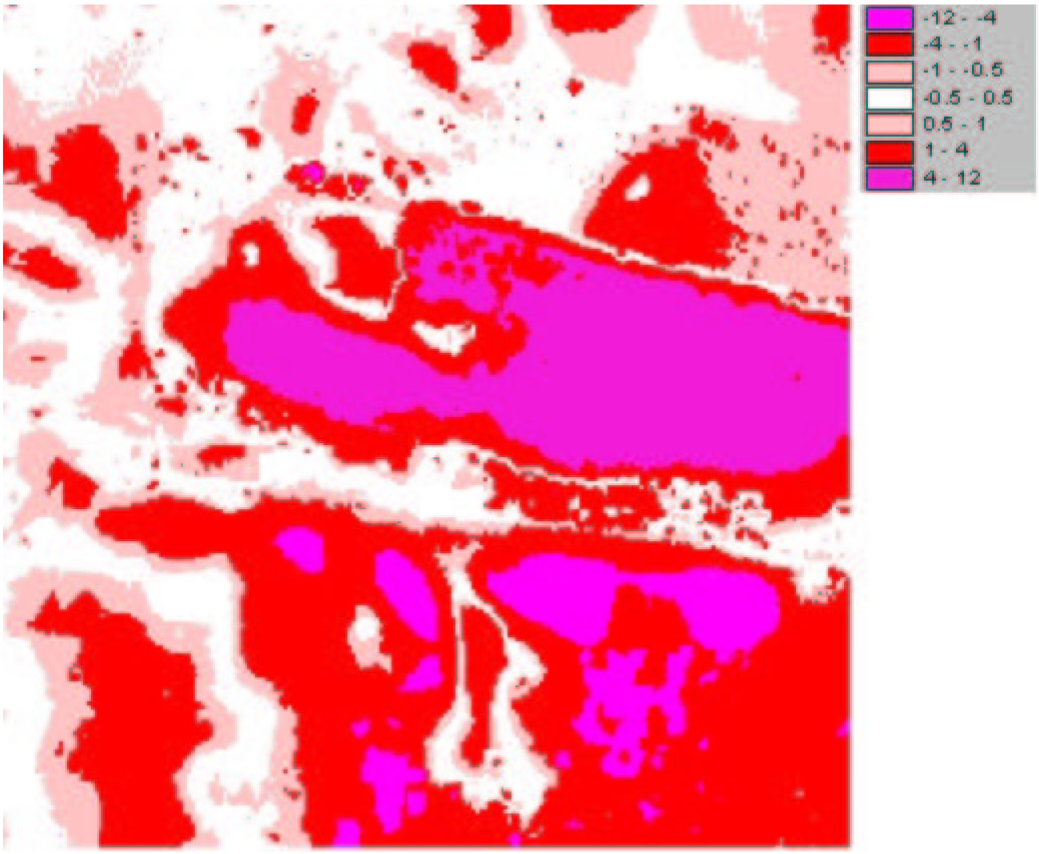


Figure 5.1.4: Old height model minus laserdata heights in forest and steep sloped area. See Appendix C (area 6) for the geographical limits of this area.*

Table 5.1.1 shows the results of the other areas. The steps that have been taken for the above two areas are the same. The table shows the minimum (negative) and maximum difference, and the mean and standard deviation for each area.

Table 5.1.1: Results from the comparison of the existing height model with the laserdata.

Area	Min. diff.(m)	Max. diff. (m)	Mean (m)	St. deviation (m)
1	-18,3514	22,4383	-0,2437	2,0159
2	-23,1420	10,5007	-0,7281	2,4848
3	-15,5542	7,5141	0,1837	0,8949
4	-24,9735	25,7921	-0,8206	4,8229
5	-17,2044	3,1387	-0,6613	0,9417
6	-18,4955	16,0161	-0,7905	1,7077

The low negative values for the minimum differences are all due to filter errors. The large maximum differences are probably also caused by the filter, which did its work slightly too good. The mean height differences are all within 1 meter. Because laserdata usually reflects on some low vegetation instead of the pure ground level, the values are negative. Only in area 3 there is a positive value. This is probably due to some new buildings where the terrain has been raised. The following paragraph also shows some offsets near that area.

Area 1 is a very flat area and shows a good result. With some better filtering the standard deviation will probably drop too. Area 2 has a lot of trees in it. This causes most of the differences with the existing height model. Area 3 is slightly hilly, but shows good results. The only problem was a power station and some power lines that caused some height differences with the old height model. There is some laserdata missing in area 4 which caused some interpolation errors. The high buildings from the Foulum research centre also caused some of the filtering errors. That is why the standard deviation is so high. The hilly terrain causes the larger mean height difference. Area 5 has some more hills then area 3, this results into a higher mean, but because the filtering worked better in area 5, the standard deviation is much smaller. Area 6 has a lot of vegetation in it. This results in slightly higher mean height difference and, because of the filtering, a somewhat higher standard deviation.

Note that the old height model for Møn was not available in time. The results are expected to be more or less in the same order as the other areas. This can be verified in the other paragraphs of this chapter.

If I look at the area characteristics and the height differences with the old height model, it can be said that the differences are the smallest in flat and slightly hilly areas. The differences get larger in areas with multiple slopes and the largest differences are found in areas with forest. Implementing a good filtering technique may solve this.

5.2 Evaluation of laser scanning accuracy by GIS

It is commonly known that laserdata has an inconsistent accuracy. Especially in forest areas and steep areas the accuracy can turn out lower than the theoretical accuracy. With all the errors in the laser system, the GPS and the INS, this theoretical accuracy is typically in the range of 10-15 cm. Since we have GPS data in areas with different terrain types and different slopes, we can see how inconsistent this accuracy is. This will check the heights from the laserdata at the same time.

The GPS heights are compared with the laserdata. For every GPS point, the laserdata heights within 3 meters of that point are selected and then the heights are averaged. The differences between the GPS height and the average laserdata heights are then accumulated together and they form a mean difference and standard deviation (see Appendix A.8).

There are two different sets of GPS data. The first is a collection of road profiles that are surveyed in areas 2, 3 and 4 and the second is a collection of small areas surveyed in areas 2, 4 and 6 and in Møn (table 5.2.1).

Table 5.2.1: GPS heights minus laserdata heights.

Area	Number of points	Mean height difference (m)	Standard deviation
Roads 2	1019	-1,8608	4,1895
Roads 3	350	-4,8138	0,3476
Roads 4	1441	-4,9359	7,2694
Patch 2	423	0,3326	0,1382
Patch 4	370	3,6524	0,6585
Patch 6	1645	-0,0088	0,9440
Møn	2474	-1,0480	0,9074

The road profiles show quite some difference. In area 3 the standard deviation is low, so there is an offset between the GPS heights and the laserdata heights. It is unclear what the reason of this offset is. The roads in area 4 are sometimes very steep, which is the probably cause of the large errors.

The patch in area 2 is a steep slope covered with grass. The results are good. Note that the GPS heights are higher than the laserdata heights. The reason for that is unknown. The patch in area 4 is a parking lot near a new building. The standard deviation is not very large, but the height difference is. This offset is probably due to changes in the terrain situation. Patch 6 (figure 5.2.1) is a sloped piece of terrain with some trees on it. The results are very good. The standard deviation is somewhat bigger, but this is probably due to the presence of high vegetation in the area.

Figure 5.2.1: Aerial picture of the patch in area 6 with in red the GPS points.

The Møn area shows some higher values for the laserdata. The data was recorded in the summer, which means the presence of vegetation. This is the probable cause of the height difference in that area.

It is difficult to use these values to form a general conclusion. It seems that the accuracy of laserdata degrades most in areas with trees. Single slopes do not seem a big problem, but when the area has a lot of different slopes the laserdata gets less accurate. Note that this paragraph (and research) only covers the accuracy of the height determination and not the accuracy of the planar coordinate determination.

5.3 Multiple recordings of the returning pulse

In section 2.2 some information was given about recording multiple parts of the returning pulse. This section will show how we can use this recorded data to our advance. In the KMS laserdata only the last significant part of the returning pulse is recorded, so we will not use that data here. In the Fotonor laserdata the first and the last significant part is recorded. Most of these pulses will show no height differences. This occurs in areas where the laser hits a solid object like bare terrain or a building. In areas with non-solid objects however, there will be a difference in height between the first and the last recording. Generally, this will happen in areas with vegetation.

The processed laserdata shows height differences in all points. This is probably because of the processing, because it is known that the measurement is only reliable when there is a minimal height difference of about 2-5 meters (depending on the system) between the two objects the laser pulse reflects on. This theory is supported by the data, for the data shown in figure 5.3.1 there were no points that had a height difference between –0.45 and 3.84 meters (the negative values seem strange, because the first pulse should always be higher than the second, but this is due to the automatic pre-processing by the surveying company).

This analysis basically selects all points where there is a height difference of at least 2 meters between them. These points are plotted in figure 5.3.1. Part of area 5 (see Chapter 3) is used here. If you compare the plot on the right with the aerial photograph on the left, you can see that the points form part of the vegetation. Still, there is some vegetation missing (and in section 5.4 it shows that this is not the result of changes in the terrain).

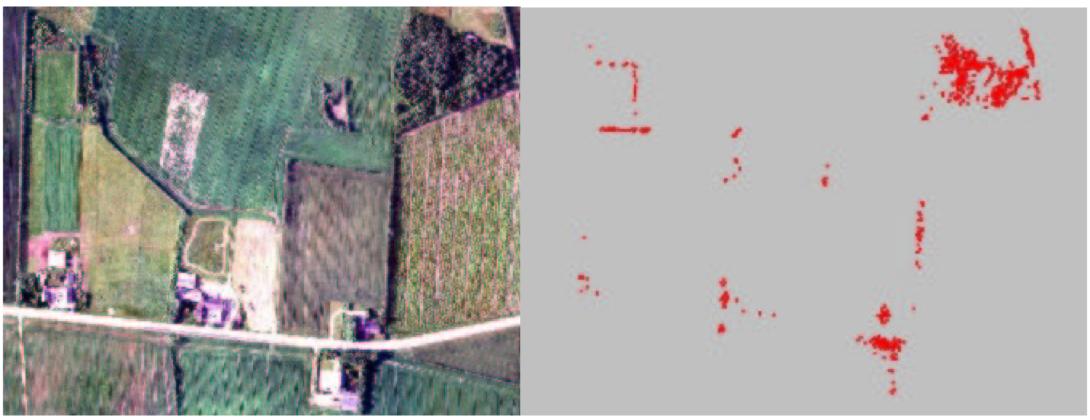


Figure 5.3.1: Left, an aerial photograph of the area on the right, where points are plotted in red with a height difference of at least 2 meters between the two pulse recordings.

Figure 5.3.2 shows the same analysis, but now in a more forested area (part of area 6, see Chapter 3). This shows basically the same result. Note that the lower vegetation cannot be shown on the plot. Again there are no points with a height difference between –0.41 and 3.84. This means that the value of 2 meters may actually be a little low for this airborne laser scanning system (Optech ALTM1210).

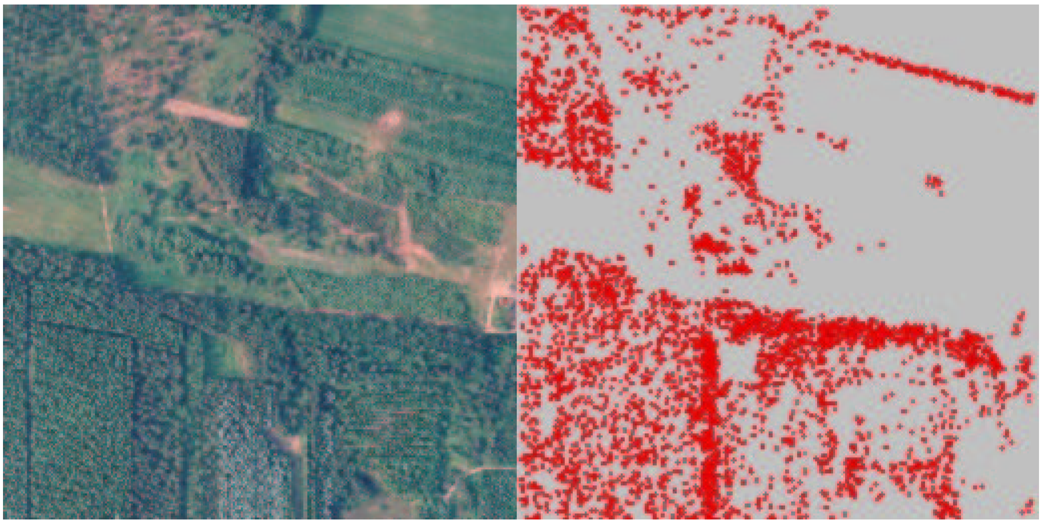


Figure 5.3.2: Left, an aerial photograph of the area on the right, where points are plotted in red with a height difference of at least 2 meters between the two pulse recordings.

Then figure 5.3.3 is used to check the results for a larger area (the whole of area 5, see Chapter 3). This is to form a general conclusion on this analysis. This dataset contains more than 1 million points and only has three points within the –0.40 and 3.84 interval. This time the TOP10DK vegetation features are drawn on top of the aerial photograph. This also shows that not every tree is in TOP10DK either. Especially in places where there are some buildings put together with trees, there is a discrepancy between the laserdata and the TOP10DK. Note that the green lines of the TOP10DK mainly consist of low vegetation, which means it is not visible in the right image.

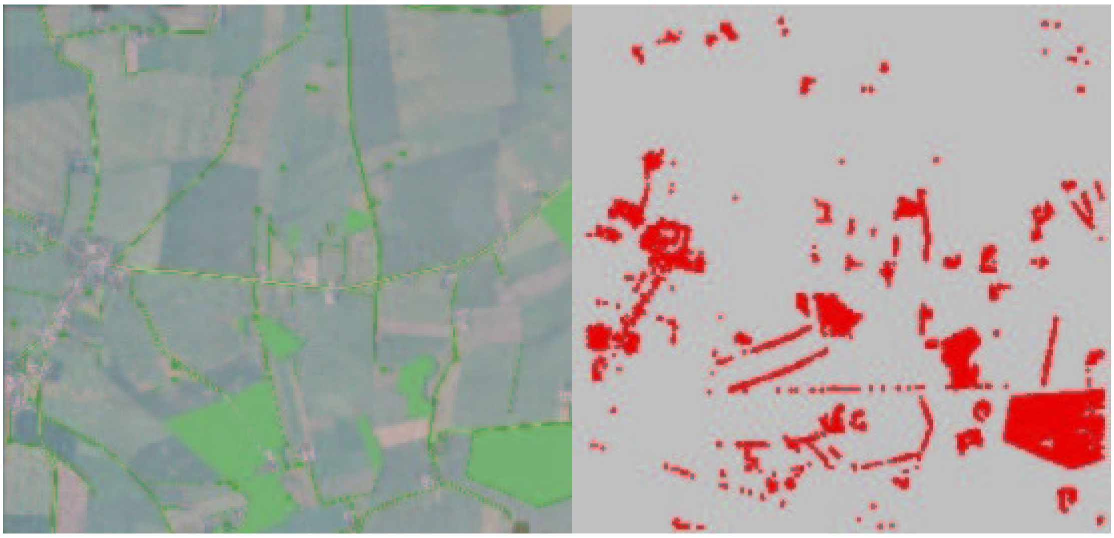


Figure 5.3.3: Left, an aerial photograph with TOP10DK vegetation of the area on the right, where points are plotted in red with a height difference of at least 2 meters between the two pulse recordings.

It is clear that the multiple recordings can give us an idea of where we can expect vegetation. In the KMS case we do not need this information, because we already got aerial pictures. On the other hand, the terrain may change in time. The data from the multiple recordings is not complete enough to do full change detection though. The third option is to derive tree heights. This application fails too, because there is no way to see on what part of the tree the laser reflected and if the laser reached all the way to the ground (see Section 2.2). An educated guess of the amount of biomass is possible though, and laser scanning has in general many potential applications in forestry.

5.4 Amplitude

Both the KMS laserdata and the Fotonor laserdata record the amplitude of the reflected laser pulses. Although all amplitude data seems to be (with a few exceptions) within the 0 to 100 interval, the amplitudes are not the same as reflectivity (i.e. the percentage of light that is reflected back). The amplitude data are typically logarithmic values.

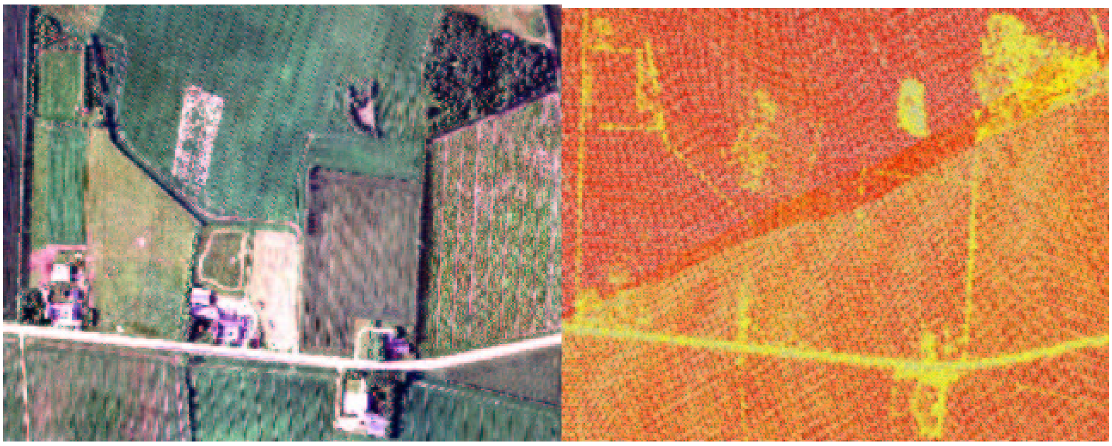


Figure 5.4.1: Aerial photograph on the left and amplitude data on the right (yellow is low).

Figure 5.4.1 shows an aerial picture and a graduate colour-coded amplitude of the Fotonor laserdata. The area shown is the same as in figure 5.3.1. The points with low amplitude show something interesting. They form a classification of the high (no grass and crops for example) vegetation, the tarmac roads, the water and perhaps some other features. The visible division between red and orange is due to two overlapping flight swaths. Again, the laserdata has information about where to expect vegetation and this time it is complete. The downside is that you get the tarmac roads and the water too. This problem maybe solved by using some remote sensing software (like Idrisi) and do a classification after selecting some training areas. This is not part of this research though.

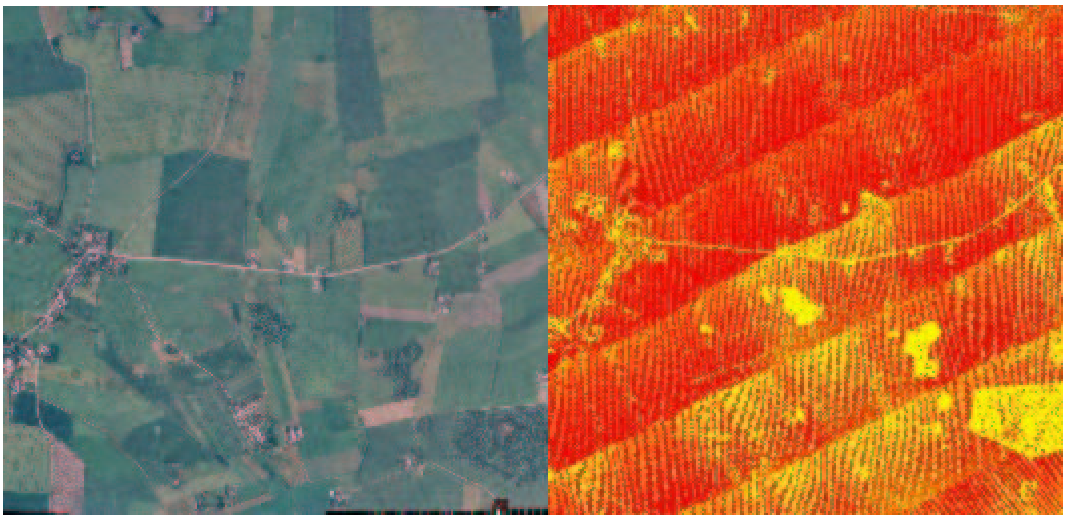


Figure 5.4.2: Aerial photograph on the left and amplitude data on the right (yellow is low).

Figure 5.4.2 shows the same analysis, but for a bigger area (again area 5). It is easy to see the flight swaths, the denser forest and the tarmac roads. We do the same analysis on the KMS laserdata in Møn. Figure 5.4.3 shows that there is little information in the amplitude data for the KMS laserdata. The problem is likely that the Riegl laser is less powerful (and less expensive) than the Optech laser used in the Fotonor laserdata. This results in no returning light at all for many features.

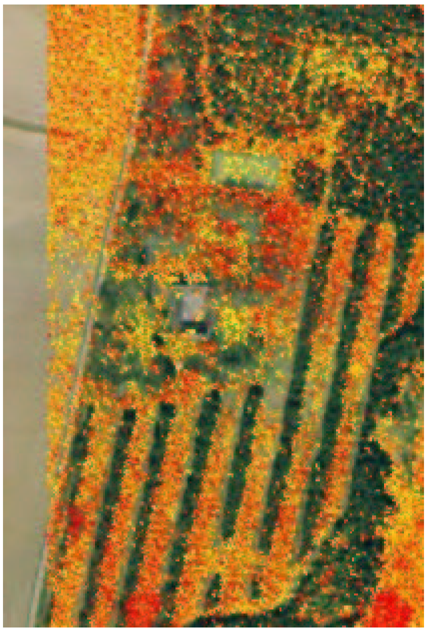


Figure 5.4.3: Amplitudes in the KMS laserdata.

The first part of this paragraph compared the amplitude with the terrain features. The amplitude says something about the data itself. You can expect that a point that is recorded with low amplitude is less reliable than a point that is recorded with high amplitude. Of course this is in connection with the terrain type (figure 5.4.4), but also with the scanning angle. It is too difficult to say what the lower limit on amplitude should be in order for the point to be reliably recorded. The larger the angle is, the lower the amplitude. This is not a large effect, and it is for the most part already filtered out in the stripe adjustment.

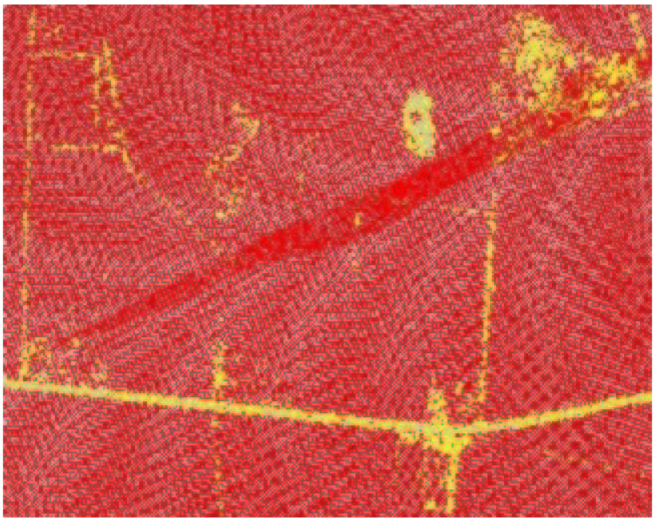


Figure 5.4.4: Amplitudes lower than 10 in yellow.

So, the amplitude can be used to say something about the terrain type. It requires some more research to see how much you can tell from the amplitude data. It shows that some terrain features, like vegetation (except for grass and crops), tarmac roads and water, do not reflect the laser light very well (or not at all in case of the KMS laserdata at the fairly high flight level used in Møn).

5.5 Laserdata versus elevated TOP10DK features

The features in the TOP10DK have heights assigned. In section 5.1 we already discussed the ground level heights, here we will check the elevated TOP10DK features that can be found in the buildings. To do this, we take the average height of the building and compare them with the average height of the laserdata points on top of them. Then we take the average for all the buildings. This can be done with an Avenue script (see Appendix A.6). The results are in table 5.5.1.

Table 5.5.1: Average height difference (TOP10DK minus laserdata)

Area	Number of buildings	Mean height difference (m)	Standard deviation
1	19	-0,6869	1,0250
2	413	-0,6187	0,7763
3	106	-0,4776	0,9941
4	248	-0,8732	1,0202
5	122	-0,7791	0,8423
6	42	-1,4102	1,2101
Møn	160	-1,8010	1,8718

The differences are all within 2 meters. I will not look at the amount of difference here, but the differences between the different areas. The height differences are likely due to the way the building heights are measured in the TOP10DK. All buildings are treated as if they have a flat roof. There are many buildings with gable roofs. The laserdata contains heights around the gable roofs, which explains the difference with the TOP10DK.

The best results are in area 3. This area is slightly hilly and has little high vegetation. In the more hilly (areas 4 and 5) and the more vegetated (area 6) areas the results are slightly worse. Especially in area 6 with a lot of forest the results are worse. It is very well possible that laser pulses get reflected on overhanging branches there.

The Møn laserdata shows larger differences. It is unknown what caused this, but since the system is still under development there might be better results in the future.

6 Conclusions and recommendations

In the introduction (see Chapter 1) is stated that the goal of this report is to give a general idea about the suitability of laserdata for certain applications, and hopefully also to be used as a starting point for further research. In this chapter that general idea will be given in the conclusions (see Section 6.1) and some recommendations for further research are in section 6.2.

6.1 Conclusions

The height differences between the existing height model and the laserdata are the smallest in flat and slightly hilly areas. The differences get larger in areas with multiple slopes and the largest differences are found in areas with forest. (see Section 5.1).

It seems that the accuracy of laserdata degrades most in areas with trees. Single slopes do not seem a big problem, but when the area has a lot of different slopes the laserdata gets less accurate. (see Section 5.2).

Recording multiple parts of the returning pulse can give us an idea of where we can expect vegetation. This might be useful when there are no aerial pictures available or if the terrain has changed over time. The data from the multiple recordings is not complete enough to do full change detection though. The tree heights can be derived from this information, but there is no way to see on what part of the tree the laser reflected and if the laser reached all the way to the ground. An educated guess of the amount of biomass is possible though (see Section 5.3).

The amplitude can be used to say something about the terrain type. It shows that some terrain features, like vegetation (except for grass and crops), tarmac roads and water, do not reflect the laser light very well or not at all in case of the KMS laserdata at the fairly high flight level used in Møn (see Section 5.4).

The differences between TOP10DK building heights and the laserdata heights are all within 2 meters. This amount is probably due to the way the building heights are measured in the TOP10DK (all flat roofs, some roofs are gable). In the more hilly and the more vegetated areas the results are slightly worse. It is very well possible that laser pulses get reflected on overhanging branches in highly vegetated areas. The Møn laserdata shows larger differences. It is unknown what caused this, but since the system is still under development there might be better results in the future (see Section 5.5).

Doing this research working with both a commercial dataset as the KMS laserdata shows that the KMS laserdata is fit for most applications. Only the amplitude data seems less reliable. Because the KMS laser scanning system has a cheap (compared to commercial systems) set-up, it could be interesting for other organisations to use this set-up. If this is combined with cheaper software like Surfer (as opposed to 3D Analyst), there is a complete cheap alternative for the commercial systems.

6.2 Recommendations

The filter used in this research did not fit all the needs, because it was designed for flat terrain. It is important to conduct research towards filtering techniques so that it is possible to derive good DEM's (see Section 5.1)

The problem with testing the accuracy on grids is that you get a more overall accuracy rather than point accuracy. It might be better to work directly with TIN's (see Section 5.1). Using TIN's for smaller test areas is therefore recommended.

This research is mainly concentrated on the height accuracy of the laserdata. It may be interesting to look at the planar accuracy of the laserdata too (see Section 5.2).

More research is required to illustrate how much you can tell from the amplitude data. It might be possible to do a classification with the amplitude data. This may be done by using remote sensing software (like Idrisi) and do a classification after selecting some training areas (see Section 5.4).

References

Websites visited in May 2002

[1] Meetkundige Dienst, *http://www.minvenw.nl/rws/mdi/home/*

[2] TopoSys, *http://www.toposys.de/*

[3] Riegl, *http://www.riegl.co.at/*

[4] Slide Show Movie Maker, *http://www.joern-thiemann.de/tools/*

[5] Web 3D Consortium, *http://www.vrml.org/*

[6] Cosmo Software, *http://www.cai.com/cosmo/*

[7] ESRI ArcScripts, *http://arcscripts.esri.com/details.asp?dbid=1642349600*

Books and articles

[8] Arens, C.A., *Filteroperaties in TIN's: Wat is het voordeel boven rasterstructuren?*, Delft: TU Delft, 2001

[9] Lemmens, M.J.P.M and E.H.W. Fortuin, *Fouten-analyse van vliegtuig-laseraltimetrie*, Delft: TU Delft, 1997

[10] Forsberg, R., K. Keller and S.M. Jacobsen, *Laser monitoring of ice elevations and sea-ice thickness in Greenland.* In: M.A. Hofton (ed), IAPRS Vol XXXIV Part 3/VV4 Com III, Annapolis, 2001, pp. 163-168

[11] Huising E.J. and L.M. Gomes Pereira, *Errors and accuracy estimates of laser altimetry data acquired by various laser scanning systems for topographic applications*, ISPRS JPRS, Vol. 53, no. 5 pp. 245-261, 1998

[12] Kraus, K. and N. Pfeifer, *Determination of terrain models in wooded areas with airborne laser scanner data*, ISPRS JPRS, Vol. 53, no. 4, pp. 193-203, 1998

[13] Vosselman, G., *Slope based filtering of laser altimetry data*, ISPRS, Vol. XXXIII, B3, Amsterdam, 2000

Appendix A Avenue scripts

This appendix contains all the scripts that are written in Avenue (i.e. the programming language of ArcView 3). The first four scripts look much alike, but there are some subtle differences. Paragraph 5 contains a script taken from [8]. The next script computes the average height difference of the buildings in TOP10DK and the laserdata. The last paragraph has a short script for showing the grid statistics in it. When copying these scripts, repair the comments (’) and the line breaks.

A.1 Read the KMS laserdata

```
'======================================================================
' Name:                 ReadLIDAR_KMS
'
' Headline:             Generate Theme from Text File (KMS data)
'
' Description:          Reads the coordinates plus amplitude and other
                        data from a text file adds them to a Point
                        Theme
'======================================================================
' Created by:           Calin Arens
' Date:                 Apr-22-2002
'======================================================================

'Get the Active View
theView = av.FindDoc ("View1")

'Locate the Points Coordinates File
theFileName = FileDialog.Show("*.txt","Text File (*.txt)","Locate the
LIDAR file")
theLIDARFile = LineFile.Make(theFileName, #FILE_PERM_READ )

'Create a New Point FTab
theOutputName = FileDialog.Put("lidar.shp".AsFileName,"*.shp","Choose
a name for the LIDAR shapefile")
theFTab = FTab.MakeNew(theOutputName,Point)
theFTab.SetEditable(TRUE)

theIDField = Field.Make("ID", #FIELD_DECIMAL, 13, 6)
theLatitudeField = Field.Make("East1", #FIELD_DECIMAL, 10, 2)
theLongitudeField = Field.Make("North1", #FIELD_DECIMAL, 10, 2)
theHeightField = Field.Make("NHeight1", #FIELD_DECIMAL, 6, 2)
theAmplitudeField = Field.Make("Amp1", #FIELD_SHORT, 4, 0)
theScanLineField  = Field.Make("Nr_scan", #FIELD_SHORT, 4, 0)

theFTab.AddFields({theIDField,theLatitudeField,theLongitudeField,theH
eightField,theAmplitudeField,theScanLineField})

theShapeField = theFTab.Findfield("Shape")

'Read the coordinates from the text file
theSize = theLIDARFile.GetSize
theLIDARFile.SetPos(0)
i=0

for each elt in 0..(theSize-1)
  aNewString = theLIDARFile.ReadElt
```

```
  if (aNewString=NIL) then
    av.ShowMsg("Cannot read string...")
  else
    theIDString = aNewString.Extract(0)
    if (theIDString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theID = theIDString.BasicTrim (" "," ")
    end

    theLonString = aNewString.Extract(1)
    if (theLonString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theLon = theLonString.BasicTrim (" "," ")
    end

    theLatString = aNewString.Extract(2)
    if (theLatString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theLat = theLatString.BasicTrim (" "," ")
    end

    theNormHeightString = aNewString.Extract(3)
    if (theNormHeightString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theNormHeight = theNormHeightString.BasicTrim (" "," ")
    end

    theAmplitudeString = aNewString.Extract(4)
    if (theAmplitudeString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theAmplitude = theAmplitudeString.BasicTrim (" "," ")
    end

    theScanLineString = aNewString.Extract(5)
    if (theScanLineString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theScanLine = theScanLineString.BasicTrim (" "," ")
    end

    if ((theIDString=NIL) or (theLatString=NIL) or (theLonString=NIL)
or (theNormHeightString=NIL) or (theAmplitudeString=NIL) or
(theScanLineString=NIL)) then
      av.ShowMsg("One or more empty strings...")
    else
      'Add the points to the new created theme
      thePoint = Point.Make (theLat.AsNumber,theLon.AsNumber)

      rec = theFTab.AddRecord

      theFTab.SetValue(theShapeField, rec, thePoint)
      theFTab.SetValue(theIDField, rec, theID.AsNumber)
      theFTab.SetValue(theLatitudeField, rec, theLat.AsNumber)
      theFTab.SetValue(theLongitudeField, rec, theLon.AsNumber)
      theFTab.SetValue(theHeightField, rec, theNormHeight.AsNumber)
      theFTab.SetValue(theAmplitudeField, rec, theAmplitude.AsNumber)
```

```
      theFTab.SetValue(theScanLineField, rec, theScanLine.AsNumber)
    end

    'Status line
    av.ShowStopButton
    progress = (i/theSize) * 100
    doMore = av.SetStatus(progress)
    if (not doMore) then
      break
    end
    i=i+1
    av.ShowMsg("Reading file...")
  end
end

theFTab.SetEditable(False)

'Create a Theme from the FTab
theTheme = FTheme.Make(theFTab)
theView.AddTheme(theTheme)
theTheme.SetActive(TRUE)
theTheme.SetVisible(TRUE)
av.GetProject.SetModified(TRUE)

MsgBox.Info("Ready!","Ready!")
```

A.2 Read the Fotonor laserdata

```
'=====================================================================
' Name:         ReadLIDAR_Fotonor
'
' Headline:     Generate Theme from Text File (Fotonor data)
'
' Description:  Reads the coordinates plus amplitude and other data
                from a text file adds them to a Point Theme
'=====================================================================
' Created by:   Calin Arens
' Date:         Mar-21-2002
'=====================================================================

'Get the Active View
theView = av.FindDoc ("View1")

'Locate the Points Coordinates File
theFileName = FileDialog.Show("*.txt","Text File (*.txt)","Locate the
LIDAR file")
theLIDARFile = LineFile.Make(theFileName, #FILE_PERM_READ )

'Create a New Point FTab
theOutputName = FileDialog.Put("lidar.shp".AsFileName,"*.shp","Choose
a name for the LIDAR shapefile")
theFTab = FTab.MakeNew(theOutputName,Point)
theFTab.SetEditable(TRUE)

theIDField = Field.Make("ID", #FIELD_DECIMAL, 13, 6)
theLatitudeField = Field.Make("East1", #FIELD_DECIMAL, 10, 2)
theLongitudeField = Field.Make("North1", #FIELD_DECIMAL, 10, 2)
theEllipseHeightField = Field.Make("EHeight1", #FIELD_DECIMAL, 6, 2)
theHeightField = Field.Make("NHeight1", #FIELD_DECIMAL, 6, 2)
```

```
theAmplitudeField = Field.Make("Amp1", #FIELD_SHORT, 4, 0)
theLatitudeField2 = Field.Make("East2", #FIELD_DECIMAL, 10, 2)
theLongitudeField2 = Field.Make("North2", #FIELD_DECIMAL, 10, 2)
theEllipseHeightField2 = Field.Make("EHeight2", #FIELD_DECIMAL, 6, 2)
theAmplitudeField2 = Field.Make("Amp2", #FIELD_SHORT, 4, 0)

theFTab.AddFields({theIDField,theLatitudeField,theLongitudeField,theE
llipseHeightField,theHeightField,theAmplitudeField,theLatitudeField2,
theLongitudeField2,theEllipseHeightField2,theAmplitudeField2})

theShapeField = theFTab.Findfield("Shape")

'Read the coordinates from the text file
theSize = theLIDARFile.GetSize
theLIDARFile.SetPos(0)
i=0

for each elt in 0..(theSize-1)
  aNewString = theLIDARFile.ReadElt
  if (aNewString=NIL) then
    av.ShowMsg("Cannot read string...")

  else
    theIDString = aNewString.Extract(0)
    if (theIDString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theID = theIDString.BasicTrim (" "," ")
    end

    theLonString = aNewString.Extract(1)
    if (theLonString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theLon = theLonString.BasicTrim (" "," ")
    end

     theLatString = aNewString.Extract(2)
    if (theLatString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theLat = theLatString.BasicTrim (" "," ")
    end

    theHeightString = aNewString.Extract(3)
    if (theHeightString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theHeight = theHeightString.BasicTrim (" "," ")
    end

    theNormHeightString = aNewString.Extract(4)
    if (theNormHeightString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theNormHeight = theNormHeightString.BasicTrim (" "," ")
    end

    theLatString2 = aNewString.Extract(5)
    if (theLatString2=NIL) then
      av.ShowMsg("One or more empty strings...")
```

```
    else
      theLat2 = theLatString2.BasicTrim (" "," ")
    end

    theLonString2 = aNewString.Extract(6)
    if (theLonString2=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theLon2 = theLonString2.BasicTrim (" "," ")
    end

    theHeightString2 = aNewString.Extract(7)
    if (theHeightString2=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theHeight2 = theHeightString2.BasicTrim (" "," ")
    end

    theAmplitudeString = aNewString.Extract(8)
    if (theAmplitudeString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theAmplitude = theAmplitudeString.BasicTrim (" "," ")
    end

    theAmplitudeString2 = aNewString.Extract(9)
    if (theAmplitudeString2=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theAmplitude2 = theAmplitudeString2.BasicTrim (" "," ")
    end

    if ((theIDString=NIL) or (theLatString=NIL) or (theLonString=NIL)
or (theHeightString=NIL) or (theNormHeightString=NIL) or
(theAmplitudeString=NIL) or (theLatString2=NIL) or
(theLonString2=NIL) or (theHeightString2=NIL) or
(theAmplitudeString2=NIL)) then
      av.ShowMsg("One or more empty strings...")
    else
      'Add the points to the new created 3D theme

      thePoint = Point.Make (theLat.AsNumber,theLon.AsNumber)

      rec = theFTab.AddRecord

      theFTab.SetValue(theShapeField, rec, thePoint)
      theFTab.SetValue(theIDField, rec, theID.AsNumber)
      theFTab.SetValue(theLatitudeField, rec, theLat.AsNumber)
      theFTab.SetValue(theLongitudeField, rec, theLon.AsNumber)
      theFTab.SetValue(theEllipseHeightField, rec,
theHeight.AsNumber)
      theFTab.SetValue(theHeightField, rec, theNormHeight.AsNumber)
      theFTab.SetValue(theLatitudeField2, rec, theLat2.AsNumber)
      theFTab.SetValue(theLongitudeField2, rec, theLon2.AsNumber)
      theFTab.SetValue(theEllipseHeightField2, rec,
theHeight2.AsNumber)
      theFTab.SetValue(theAmplitudeField, rec, theAmplitude.AsNumber)
      theFTab.SetValue(theAmplitudeField2, rec,
theAmplitude2.AsNumber)

    end
```

```
    'Status line
    av.ShowStopButton
    progress = (i/theSize) * 100
    doMore = av.SetStatus(progress)
    if (not doMore) then
      break
    end
    i=i+1
    av.ShowMsg("Reading file...")
  end
end

theFTab.SetEditable(False)

'Create a Theme from the FTab
theTheme = FTheme.Make(theFTab)
theView.AddTheme(theTheme)
theTheme.SetActive(TRUE)
theTheme.SetVisible(TRUE)
av.GetProject.SetModified(TRUE)

MsgBox.Info("Ready!","Ready!")
```

A.3 Read the GPS data

```
'=====================================================================
' Name:               ReadGPS
'
' Headline:           Generate Theme from Text File
'
' Description:        Reads the coordinates plus amplitude and other
                      data from a text file adds them to a Point
                      Theme
'=====================================================================
' Created by:         Calin Arens
' Date:               Apr-17-2002
'=====================================================================

'Get the Active View
theView = av.FindDoc ("View1")

'Locate the Points Coordinates File
theFileName = FileDialog.Show("*.txt","Text File (*.txt)","Locate the
GPS file")
theLIDARFile = LineFile.Make(theFileName, #FILE_PERM_READ )

'Create a New Point FTab
theOutputName = FileDialog.Put("gps.shp".AsFileName,"*.shp","Choose a
name for the GPS shapefile")
theFTab = FTab.MakeNew(theOutputName,PointZ)
theFTab.SetEditable(TRUE)

theIDField = Field.Make("ID", #FIELD_DECIMAL, 13, 6)
theLatitudeField = Field.Make("East1", #FIELD_DECIMAL, 10, 2)
theLongitudeField = Field.Make("North1", #FIELD_DECIMAL, 10, 2)
theHeightField = Field.Make("NHeight1", #FIELD_DECIMAL, 6, 2)
```

```
theFTab.AddFields({theIDField,theLatitudeField,theLongitudeField,theH
eightField})

theShapeField = theFTab.Findfield("Shape")

'Read the coordinates from the text file
theSize = theLIDARFile.GetSize
theLIDARFile.SetPos(0)
i=0

for each elt in 0..(theSize-1)
  aNewString = theLIDARFile.ReadElt
  if (aNewString=NIL) then
    av.ShowMsg("Cannot read string...")

  else
    theID = elt

    theLonString = aNewString.Extract(1)
    if (theLonString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theLon = theLonString.BasicTrim (" "," ")
    end

     theLatString = aNewString.Extract(2)
    if (theLatString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theLat = theLatString.BasicTrim (" "," ")
    end

    theNormHeightString = aNewString.Extract(3)
    if (theNormHeightString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theNormHeight = theNormHeightString.BasicTrim (" "," ")
    end

    if ((theID=NIL) or (theLatString=NIL) or (theLonString=NIL) or
(theNormHeightString=NIL)) then
      av.ShowMsg("One or more empty strings...")
    else
      'Add the points to the new created 3D theme

      thePoint = PointZ.Make
(theLat.AsNumber,theLon.AsNumber,theNormHeightString.AsNumber)

      rec = theFTab.AddRecord

      theFTab.SetValue(theShapeField, rec, thePoint)
      theFTab.SetValue(theIDField, rec, theID)
      theFTab.SetValue(theLatitudeField, rec, theLat.AsNumber)
      theFTab.SetValue(theLongitudeField, rec, theLon.AsNumber)
      theFTab.SetValue(theHeightField, rec, theNormHeight.AsNumber)

    end

    'Status line
    av.ShowStopButton
    progress = (i/theSize) * 100
```

```
    doMore = av.SetStatus(progress)
    if (not doMore) then
      break
    end
    i=i+1
    av.ShowMsg("Reading file...")
  end
end

theFTab.SetEditable(False)

'Create a Theme from the FTab
theTheme = FTheme.Make(theFTab)
theView.AddTheme(theTheme)
theTheme.SetActive(TRUE)
theTheme.SetVisible(TRUE)
av.GetProject.SetModified(TRUE)

MsgBox.Info("Ready!","Ready!")
```

A.4 Read the old height model data

```
'=====================================================================
' Name:                 ReadOLDDEM
'
' Headline:             Generate Theme from Text File
'
' Description:          Reads the GPS coordinates a text
                        file adds them to a Point Theme
'=====================================================================
' Created by:           Calin Arens
' Date:                 Apr-15-2002
'=====================================================================

'Get the Active View
theView = av.FindDoc ("View1")

'Locate the Points Coordinates File
theFileName = FileDialog.Show("*.txt","Text File (*.txt)","Locate the
OLDDEM file")
theLIDARFile = LineFile.Make(theFileName, #FILE_PERM_READ )

'Create a New Point FTab
theOutputName =
FileDialog.Put("gpsorolddem.shp".AsFileName,"*.shp","Choose a name
for the OLDDEM shapefile")
theFTab = FTab.MakeNew(theOutputName,PointZ)
theFTab.SetEditable(TRUE)

theIDField = Field.Make("ID", #FIELD_DECIMAL, 13, 6)
theLatitudeField = Field.Make("East1", #FIELD_DECIMAL, 10, 2)
theLongitudeField = Field.Make("North1", #FIELD_DECIMAL, 10, 2)
theHeightField = Field.Make("NHeight1", #FIELD_DECIMAL, 6, 2)

theFTab.AddFields({theIDField,theLatitudeField,theLongitudeField,theH
eightField})

theShapeField = theFTab.Findfield("Shape")
```

```
'Read the coordinates from the text file
theSize = theLIDARFile.GetSize
theLIDARFile.SetPos(0)
i=0

for each elt in 0..(theSize-1)
  aNewString = theLIDARFile.ReadElt
  if (aNewString=NIL) then
    av.ShowMsg("Cannot read string...")

  else
    theID = elt

    theLonString = aNewString.Extract(0)
    if (theLonString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theLon = theLonString.BasicTrim (" "," ")
    end

     theLatString = aNewString.Extract(1)
    if (theLatString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theLat = theLatString.BasicTrim (" "," ")
    end

    theNormHeightString = aNewString.Extract(2)
    if (theNormHeightString=NIL) then
      av.ShowMsg("One or more empty strings...")
    else
      theNormHeight = theNormHeightString.BasicTrim (" "," ")
    end

    if ((theID=NIL) or (theLatString=NIL) or (theLonString=NIL) or
(theNormHeightString=NIL)) then
      av.ShowMsg("One or more empty strings...")
    else
      'Add the points to the new created 3D theme

      thePoint = PointZ.Make
(theLat.AsNumber,theLon.AsNumber,theNormHeightString.AsNumber)

      rec = theFTab.AddRecord

      theFTab.SetValue(theShapeField, rec, thePoint)
      theFTab.SetValue(theIDField, rec, theID)
      theFTab.SetValue(theLatitudeField, rec, theLat.AsNumber)
      theFTab.SetValue(theLongitudeField, rec, theLon.AsNumber)
      theFTab.SetValue(theHeightField, rec, theNormHeight.AsNumber)

    end

    'Status line
    av.ShowStopButton
    progress = (i/theSize) * 100
    doMore = av.SetStatus(progress)
    if (not doMore) then
      break
    end
    i=i+1
```

```
    av.ShowMsg("Reading file...")
  end
end

theFTab.SetEditable(False)

'Create a Theme from the FTab
theTheme = FTheme.Make(theFTab)
theView.AddTheme(theTheme)
theTheme.SetActive(TRUE)
theTheme.SetVisible(TRUE)
av.GetProject.SetModified(TRUE)

MsgBox.Info("Ready!","Ready!")
```

A.5 Slope-based filter

```
' *** Name:           Tin2FilteredPointZ
' *** Created by:     Calin Arens
' *** Date:           Dec-10-2000
' *** Function:       Converts and filters TIN to a PointZ FTheme
' *** Remark:         3D-Analyst extension needs to be available. Tin
                      needs an ID in it.

' ** Get the TIN **
theView = av.GetActiveDoc
theActiveThemes = theView.GetActiveThemes
if (theActiveThemes.Count = 0) then
  MsgBox.Error ("No active theme in View","Tin2FilteredPointZ")
  return NIL
end

fromSTheme = theActiveThemes.Get(0)
if (fromSTheme.Is(STheme).Not) then
MsgBox.Error (fromStheme.getName+ " is not a
TIN","Tin2FilteredPointZ")
  return NIL
end

YesNo = msgBox.YesNo ("Convert and filter "+fromStheme.getName+" to
PointZ FTheme?","Tin2FilteredPointZ",True)
if (YesNo = false) then
  return NIL
end

' ** Make new file for PointZ **
def = av.GetProject.MakeFileName("theme","shp")
theFileName = FileDialog.Put(def, "*.shp", "Convert " +
fromSTheme.GetName)
if (theFileName = NIL) then
  return NIL
end

toFTab = FTab.MakeNew(theFileName,PointZ)
toFTheme = FTheme.Make(toFTab)
theShapeField = toFTab.FindField("Shape")
theXField=Field.Make("X",#FIELD_DECIMAL,10,2)
theYField=Field.Make("Y",#FIELD_DECIMAL,10,2)
theZField=Field.Make("Z",#FIELD_DECIMAL,10,2)
```

```
theIDField=Field.Make("ID",#FIELD_DECIMAL,1,0)
toFTab.AddFields({theXField,theYField,theZField,theIDField})
toFTab.SetEditable(true)

fromTIN = fromSTheme.GetSurface
theNumNodes = fromTIN.GetNumNodes

' ** For each point in the TIN **
for each theNodeIndex in 0..(theNumNodes-1)

  theNodeValue=fromTIN.GetNodeValue(theNodeIndex)
  if (theNodeValue <> 0) then

    theNeighborIndexList={}
    theNeighborPointList={}

    thePoint = fromTIN.ReturnNode(theNodeIndex)
    theHeightOfFilteredPoint=thePoint.GetZ

      ' ** Get the neighbours of the active theme **
      theNeighborIndexList = fromTIN.GetNaturalNeighbors(thePoint)

      ' ** Take only points with neighbours

      if (theNeighborIndexList.Count <> 0) then

        ' ** For each neighbour find corresponding point **
        for each theNeighborIndex in theNeighborIndexList
          theNeighborNodeValue=fromTIN.GetNodeValue(theNeighborIndex)
          if (theNeighborNodeValue <> 0) then
            thePoint2=fromTIN.ReturnNode(theNeighborIndex)
            theNeighborPointList.Add(thePoint2)
          end
        end

' ** This value needs to be higher than every other height in the TIN
        theTemporaryPointHeight=100000

        ' ** Search lowest neighbour **
        cnt=theNeighborPointList.Count
        for each punt in 0..(cnt-1)
          theNeighborPoint=theNeighborPointList.Get(punt)
          theHeightOfNeighborPoint=theNeighborPoint.GetZ
          if (theHeightOfNeighborPoint < theTemporaryPointHeight)
then
            theTemporaryPointHeight=theHeightOfNeighborPoint
            tempX=(theNeighborPoint.GetX-
thePoint.GetX)*(theNeighborPoint.GetX-     thePoint.GetX)
                    tempY=(theNeighborPoint.GetY-
thePoint.GetY)*(theNeighborPoint.GetY-thePoint.GetY)
            theDistanceFromPoint=(tempX+tempY).Sqrt
          end
        end
' ** Maximum slope!! Important value to change
' Slope in radials (degrees=radials: 10=0.17, 25=0.44, 30=0.52,
45=0.79) **
      theMaximumSlope=0.52

theMaximumHeightDifference=theDistanceFromPoint*(theMaximumSlope.Tan)
```

```
theErodedPointHeight=theTemporaryPointHeight+theMaximumHeightDifferen
ce

                   ' ** Filtering or not **
                   if (theHeightOfFilteredPoint <=
theErodedPointHeight) then
                     toFTabRec = toFTab.AddRecord
                     theAddedPoint = fromTIN.ReturnNode(theNodeIndex)
                     theX=theAddedPoint.GetX
                     theY=theAddedPoint.GetY
                     theZ=theAddedPoint.GetZ
                     pz=PointZ.Make(theX,theY,theZ)
                     toFTab.SetValue(theShapeField,toFTabRec,pz)
                     toFTab.SetValueNumber(theXField,toFTabRec,theX)
                     toFTab.SetValueNumber(theYField,toFTabRec,theY)
                     toFTab.SetValueNumber(theZField,toFTabRec,theZ)
                     toFTab.SetValueNumber(theIDField,toFTabRec,1)
                   end

                   ' ** Progress meter **
                   progress = (theNodeIndex / theNumNodes) * 100
                   proceed = av.SetStatus(Progress)
                   if (proceed.Not) then
                     av.ClearStatus
                     av.ShowMsg("Stopped")
                     exit
                   end
                end
             end
           end

           toFTab.SetEditable(false)
           theView.AddTheme(toFTheme)

MsgBox.Info("Ready!","TIN2FilteredPointZ")
```

A.6 Compare TOP10DK building heights with laserdata heights

```
'=====================================================================
' Name:          CompareHeights
'
' Headline:      Compares building heights with laserdata heights
'=====================================================================
' Created by       : Calin Arens
' Date             : May-03-2002
'=====================================================================
' The LIDAR theme needs to be the only active theme
'=====================================================================

' Opening the building theme
theView=av.FindDoc("View1")
theBuildingTheme=theView.FindTheme("Bygning.shp")
theBuildingFTab=theBuildingTheme.GetFTab

' Opening the LIDAR theme
theView=av.FindDoc("View1")
theLIDARTheme=theView.GetActiveThemes.Get(0)
theLIDARFTab=theLIDARTheme.GetFTab
```

```
' Defining height difference list
theHeightDifferenceList={}

' For each building
for each rec in theBuildingFTab

  ' Get the average polygon height
  theBuildingShapeField=theBuildingFTab.FindField("Shape")
  thePolygonZ=theBuildingFTab.ReturnValue(theBuildingShapeField,rec)
  theListOfListofPointZ=thePolygonZ.AsList
  theListofZ={}

  for each listitem in theListOfListofPointZ
    for each thePointZ in listitem
      theZ=thePointZ.GetZ
      theListofZ.Add(theZ)
    end
  end

  theCount=theListofZ.Count
  theCumHeight=0
  for each theHeight in theListofZ
    theCumHeight=theCumHeight+theHeight
  end
  theAvgHeight=theCumHeight/theCount

  ' Select the laser points within the building
  theBuildingBitmap=theBuildingFTab.GetSelection
  theBuildingBitmap.ClearAll
  theBuildingBitmap.Set(rec)


theLIDARFTab.SelectByFTab(theBuildingFTab,#FTAB_RELTYPE_INTERSECTS,0,
#VTAB_SELTYPE_NEW)

  ' Get the average height of the LIDAR points
  theHeightField=theLIDARFTab.FindField("Height") 'CHANGE THIS IN
HEIGHT IF KMS, OR NHEIGHT1 IF FOTONOR
  theCumPointletHeight=0
  theLIDARBitmap=theLIDARFTab.GetSelection
  thePointletCount=theLIDARBitmap.Count
  for each pointlet in theLIDARBitmap

thePointletHeight=theLIDARFTab.ReturnValue(theHeightField,pointlet)
    theCumPointletHeight=theCumPointletHeight+thePointletHeight
  end
  theAvgLIDARHeight=theCumPointletHeight/thePointletCount

  ' Get the height differences and put them in a list
  theHeightDifferenceList.Add(theAvgHeight-theAvgLIDARHeight)
end

' Get the mean
theCumHeightDiff=0
theHeightDiffCount=theHeightDifferenceList.Count
for each theHeightDiff in theHeightDifferenceList
  if (theHeightDiff.IsNull) then
  else
    theCumHeightDiff=theCumHeightDiff+theHeightDiff
  end
end
```

```
theAvgHeightDiff=theCumHeightDiff/theHeightDiffCount
theResultList={}
theNumBuildings=theBuildingFTab.GetNumRecords
theResultList.Add("Number of buildings: "+theNumBuildings.AsString)
theResultList.Add("Mean: "+theAvgHeightDiff.AsString)

' Get the standard deviation
for each theElement in 0..(theHeightDiffCount-1)
  theHtDiff=theHeightDifferenceList.Get(theElement)
  theHtDiff=theHtDiff-theAvgHeightDiff
  theHtDiff=theHtDiff*theHtDiff
  theHeightDifferenceList.Set(theElement,theHtDiff)
end
theCHeightDiff=0
for each theHDiff in theHeightDifferenceList
  if (theHDiff.IsNull) then
  else
    theCHeightDiff=theCHeightDiff+theHDiff
  end
end
theDHtDiff=theCHeightDiff/theHeightDiffCount
theStDev=theDHtDiff.Sqrt
theResultList.Add("St.Dev.: "+theStDev.AsString)

' Show result
MsgBox.ListAsString(theResultList,"Mean height difference between
TOP10DK and laserdata","Compare Heights")
```

A.7 Grid Statistics

```
' *** Name:           GridStatistics
' *** Created by:     Calin Arens
' *** Date:           Apr-22-2002
' *** Function:       Get Grid Statistics

theView=av.FindDoc("View1")
theGrid=theView.GetActiveThemes.Get(0).GetGrid
theList=theGrid.GetStatistics
MsgBox.ListAsString(theList,"Min,    max,    mean,    st.dev.:","Grid
Statistics")
```

A.8 Compare GPS with laserdata heights

```
'======================================================================
' Name:         CompareGPSHeights
'
' Headline:     Compares GPS heights with laserdata heights
'======================================================================
' Created by      : Calin Arens
' Date            : May-16-2002
'======================================================================
' The LIDAR theme and the GPS theme need to be the only active themes
'======================================================================

theDistance=3

theView=av.FindDoc("View1")
theActiveThemesList=theView.GetActiveThemes
```

```
' Opening the GPS theme
theGPSTheme=msgbox.listasstring(theActiveThemesList,"Choose GPS
theme","Compare heights")
theGPSFTab=theGPSTheme.GetFTab

' Opening the LIDAR theme
theLidarTheme=msgbox.listasstring(theActiveThemesList,"Choose LIDAR
theme","Compare heights")
theLIDARFTab=theLidarTheme.GetFTab


' Defining height difference list
theHeightDifferenceList={}

' For each building
for each rec in theGPSFTab

  ' Get the average polygon height
  theGPSShapeField=theGPSFTab.FindField("Shape")
  thePointZ=theGPSFTab.ReturnValue(theGPSShapeField,rec)
  theGPSHeight=thePointZ.GetZ

  ' Select the laser points within the building
  theGPSBitmap=theGPSFTab.GetSelection
  theGPSBitmap.ClearAll
  theGPSBitmap.Set(rec)


theLIDARFTab.SelectByFTab(theGPSFTab,#FTAB_RELTYPE_ISWITHINDISTANCEOF
,theDistance,#VTAB_SELTYPE_NEW)

  ' Get the average height of the LIDAR points
  theHeightField=theLIDARFTab.FindField("Height") 'CHANGE THIS IN
HEIGHT IF KMS, OR NHEIGHT1 IF FOTONOR
  theCumPointletHeight=0
  theLIDARBitmap=theLIDARFTab.GetSelection
  thePointletCount=theLIDARBitmap.Count
  for each pointlet in theLIDARBitmap

thePointletHeight=theLIDARFTab.ReturnValue(theHeightField,pointlet)
    theCumPointletHeight=theCumPointletHeight+thePointletHeight
  end
  theAvgLIDARHeight=theCumPointletHeight/thePointletCount

  ' Get the height differences and put them in a list
  theHeightDifferenceList.Add(theGPSHeight-theAvgLIDARHeight)
end

' Get the mean
theCumHeightDiff=0
theHeightDiffCount=theHeightDifferenceList.Count
for each theHeightDiff in theHeightDifferenceList
  if (theHeightDiff.IsNull) then
  else
    theCumHeightDiff=theCumHeightDiff+theHeightDiff
  end
end
theAvgHeightDiff=theCumHeightDiff/theHeightDiffCount
theResultList={}
theResultList.Add("Mean: "+theAvgHeightDiff.AsString)
```

```
' Get the standard deviation
for each theElement in 0..(theHeightDiffCount-1)
  theHtDiff=theHeightDifferenceList.Get(theElement)
  theHtDiff=theHtDiff-theAvgHeightDiff
  theHtDiff=theHtDiff*theHtDiff
  theHeightDifferenceList.Set(theElement,theHtDiff)
end
theCHeightDiff=0
for each theHDiff in theHeightDifferenceList
  if (theHDiff.IsNull) then
  else
    theCHeightDiff=theCHeightDiff+theHDiff
  end
end
theDHtDiff=theCHeightDiff/theHeightDiffCount
theStDev=theDHtDiff.Sqrt
theResultList.Add("St.Dev.: "+theStDev.AsString)

' Show result
MsgBox.ListAsString(theResultList,"Mean height difference between GPS
and laserdata","Compare Heights")
```

Appendix B Software

This appendix describes the tools that are used in this research in short. The first paragraph is about the well-known GIS software ArcView. In paragraph B.2 a short discussion about Surfer will follow. The following paragraphs each contain some in house programs that were mainly used for height transformations.

B.1 ArcView

ArcView GIS is a well-known software package from a company called ESRI, it is used for storing, visualising and querying geographical data. ArcView can be adapted in a number of ways. In ArcView 3.x this happens through the build-in programming language Avenue and in ArcView 8.x any COM-compliant programming language can do this. In this way you have the possibility to change the appearance of ArcView, to adapt standard functions and to make new functions.

The implementation in Avenue is called a script. Because of my knowledge of Avenue, and my lack of knowledge of e.g. Visual Basic, I did most of the programming in ArcView 3.2. The Avenue scripts can be found in Appendix A. The work that just needed standard functions I did in ArcView 8.1. The reason for this is because it seems to work faster and more reliable.

For the 3D visualisation an extension to ArcView was used. This extension is called 3D Analyst. This is a very powerful and easy to use too, just look at the figures in this report. The 3D Analyst was available in both versions. For the grid comparison in Chapter 5, the Spatial Analyst was used. This extension was available for an evaluation period. It was only used to compare two grids. The other grid functions used are already available in 3D Analyst.

For more information about ArcView 3.x and its extensions:
http://www.esri.com/software/arcview/index.html

For more information about ArcView 8.x and its extensions:
http://www.esri.com/software/arcgis/arcview/index.html

B.2 Surfer

Surfer 8 is a program that can do almost everything with grids. In the visualisation part it can do just about everything the 3D Analyst can do too. It is recommended to explore the features of Surfer, because it is a lot cheaper than 3D Analyst. So far it looks like Surfer works a little less intuitive with geographical data, but this can also just be a matter of being used to a certain program.

For more information about Surfer 8:
http://www.goldensoftware.com/products/surfer/surfer.shtml

B.3 Geoip9

I used the Geoip9 program to add the normal heights to the laserdata. This program makes use of a grid file with geoid heights and computes the normal heights with help of the location and ellipsoid heights.

For more information about Geoip (GravSoft):
http://forskning.kms.dk/PK/gravsoft.htm

B.4 KMS Trans

KMS Trans can transform in and out any coordinate system. I mainly used this program to do some datum transformations and to compute some normal heights where Geoip9 did not work. Geoip9 is preferred though, because the output format looks better for my applications.

For more information about KMS Trans:
http://www.kms.dk/geodata/geodaetiske/kmstrans_en.html

Appendix C Test area extents

This section contains the upper left and lower right corner extents of the test areas. The coordinates are in UTM zone 32N WGS84.

Area	Upper left (Northing, Easting)	Lower right (Northing, Easting)
1	6259092,824 540918,722	6256592,839 543418,714
2	6259792,809 536818,761	6257292,822 538718,757
3	6260192,800 534318,787	6258192,814 536818,772
4	6263492,789 534318,768	6260192,808 536818,759
5	6247292,867 544168,781	6244792,883 546668,779
5*	6246410,000 545250,000	6245975,000 545800,000
6	6243542,865 536268,891	6241042,880 538768,887
6*	6242500,000 537900,000	6242000,000 538400,000
Møn	6097421,000 720081,000	6094921,000 722581,000

5* and 6* are subsets of area 5 and 6 respectively to speed up the development of testing scripts.

National Survey and Cadastre – Denmark, Technical Reports

The National Survey and Cadastre – Denmark, Technical Report series is intended as an informal report series, published at irregular intervals. The following reports have so far been published in the series (up to number 3, the reports were named "Geodætisk Institut, Technical Reports", from number 4 through 7, the reports were named "National Survey and Cadastre – Denmark, Geodetic Division, Technical Reports").

1. Jørgen Eeg: *On the Adjustment of Observations in the Presence of Blunders*, 32 pp., 1986.

2. Per Knudsen, C.C. Tscherning and René Forsberg: *Gravity Field Mapping Around the Faeroe Islands and Rockall Bank from Satellite Altimetry and Gravimetry*, 30 pp., 1987.

3. Niels Andersen: *The Structure and Filling of a 19.2 Kilometer Hydrostatic Leveling Tube*, 83 pp., 1988.

4. René Forsberg: *Gravity Measurements in East Greenland 1986-1988*, 32 pp., 1991.

5. Gabriel Strykowski: *Automation Strategy for Repeated Tasks in DOS*, 15 pp., 1992.

6. Simon Ekholm and Kristian Keller: *Gravity and GPS Survey on the Summit of the Greenland Ice Sheet 1991-1992*, 26 pp., 1993.

7. Per Knudsen: *Integrated Inversion of Gravity Data*, 52 pp., 1993.

8. Thomas Knudsen: *Geophysical Use of Geographical Information Systems*, 76 pp., 1996.

9. Simon Ekholm: *Determination of Greenland Surface Topography from Satellite Altimetry and Other Elevation Data*, 23 pp., 1997.

10. René Forsberg, Arne Olesen and Kristian Keller: *Airborne Gravity Survey of the North Greenland Shelf 1998*, 34 pp., 1999.

11. Cecilia S. Nielsen: *Topography and Surface Velocities of an Irregular Ice Cap in Greenland Assessed by the means of GPS, Laser Altimetry and SAR Interferometry*, 81 pp., 2001

12. Cecilia S. Nielsen: *Estimation of Ice Topography and Surface Velocities Using SAR Interferometry*, 37 pp., 2001

13. Thomas Knudsen (ed): *Proceedings of the seminar on remote sensing and image analysis techniques for revision of topographic databases, Copenhagen, Denmark 2000-02-29*, 119 pp., 2000.

14. Lars Brodersen: *Maps as Communication - Theory and Methodology in Cartography*, 88 pp., 2001

15. Claus V. Petersen og Simon Ekholm: *Analyse af digitale terrænmodeller beregnet fra satellitbåren SAR interferometrii, Case studies af udvalgte områder i Grønland og Danmark,* 20 pp., 2001

16. Olwijn Leeuwenburgh: *Combined Analysis of Sea Surface Height and Temperature for Mapping and Climate Studies*, 96 pp., 2001

17. Călin Arens: *Some examples of topographic applications and accuracy of laser scanning, 67 pp, 2002*

18. Rene Forsberg, Arne V. Olesen, Kristian Keller and Mads Møller: *Airborne Gravity Survey of Sea Areas Around Greenland and Svalbard 1999-2001,* 55pp., 2002

Reports may be ordered from the individual authors at the following address: Kort & Matrikelstyrelsen, Rentemestervej 8, DK-2400 Copenhagen NV, Denmark, Internet: www.kms.dk