III - Modelling Week UCM

Efficient interpolation of LiDAR Altimeter datasets in the obtention of Digital Surface Models (DSM)

Problem proposed by StereoCarto, S.L.

Participants:

Francisco Aguilera Orihuel (Universidad Complutense de Madrid)
Silvia Castellani (Università degli Studi di Firenze)
Diego Gómez Sanz (Universidad Complutense de Madrid)
Juan Carlos Luengo López (Universidad Complutense de Madrid)
Xavier Santallusia (Universitat Autònoma de Barcelona)
Ana Valeiras Jurado (Universidad Complutense de Madrid)

Coordinators:

Dra. Pilar Romero
Dr. Roberto Antolín

Madrid, June 22-30, 2009

Contents

1 Introduction

This study has been proposed by the firm *Stereocarto*, to get the representation of a digital model in order to minimize costs to obtain planimetric points.

The *Airborne Laser Scanning* (ALS) technology is based on the ground survey from an airborne laser telemeter. The telemeter measures the distance between the emission point, A, and the echoing point, B, which is a generic ground point hit by the laser ray. Thus, the laser telemeter measures the distance between the instrument and the echoing surface. However, the ground point coordinates are actually wanted.

ALS technology is built into an airplane which flies over the area they wish to explore. As described above, the points of land that are included in the area of influence of the laser are collected. Since lasers are fired from the plane, the higher it flies, the more ground it covers.

Once the company has the coordinates of the scanned points of the study area, it starts our work of interpolation to get the digital model as well as its subsequent improvement by reducing the density of data.

Reducing the number of points required for the construction of the digital model will result in an increase in the height of the flight carrying the integrated laser which brings out a reduction of the cost to the company.

2 Methodology

2.1 Previous analysis of data

Originally the data are collected by geodetic coordinates aircraft equipped with a *LIDAR* system that flies at an altitude of 800 m. The data are presented in a text file that contains three columns, the first two are the Cartesian position and the third is the height.

The domain of our problem is an area of 300x300 meters belonging to an urban area (a village called Villalba) whose average altitude is 875 meters.

To start our study of the problem, it is necessary to know how data were collected in order to get an initial idea of the distribution throughout our study area.

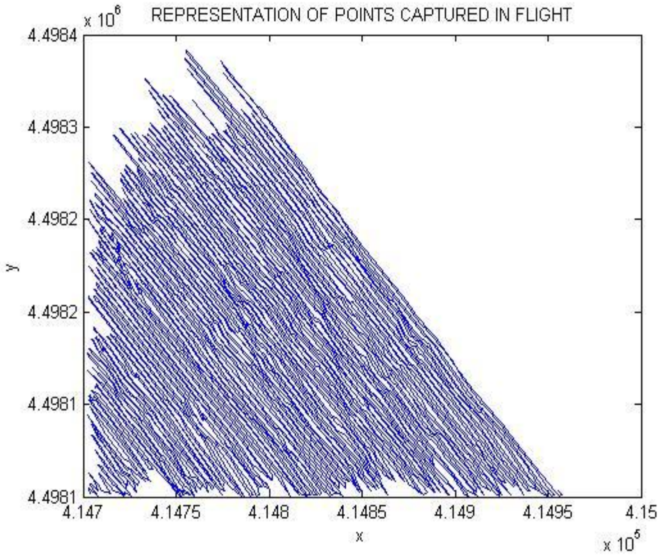


Figure 1: Representation of some points captured by the laser

We may observe in the figure 1 that the plane overflew the area in a diagonal way shooting the laser from the left to the right.

2.2 Random reduction of points

The objective of our problem is to reduce the density of points needed for the digital model, so we can minimize the costs of flight, as if we reduce the number of points, we can make a flight to higher altitudes.

First we extract from data provided by the company a set of points to validate our digital model. We removed 50 points at random, of which we left 32 to verify our criteria for error to avoid the selection of outliers in the data control and they must be distributed over the field. This step is necessary because the company has not given us these points, that's why the zone we are studing here is very small ($300m \times 300m$) and, actually, there's only one control point.

Since our goal is to see how we can reduce the number of points without worsening the resolution, each time we reduce the number of points we will do it in a random manner, according to the algorithm we have developed in this project. The ignorance of the variation in the distribution of points in the domain that results increasing the height of the flight justifies the use of randomness with a uniform distribution.

In the figure 2, there is an example to show how works the data reduction algorithm.

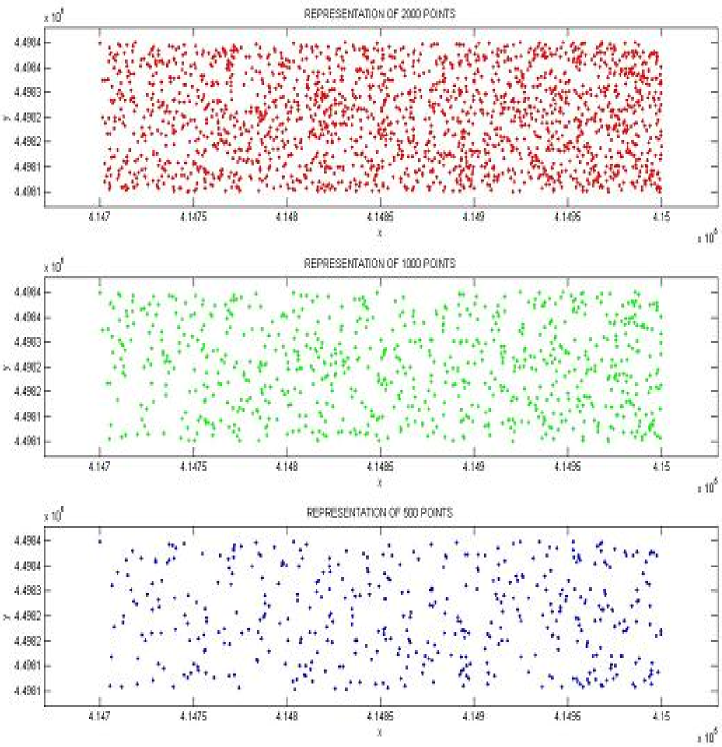


Figure 2: Example of random selection with 2000, 1000 and 500 points

The above example is performed to verify that the reduction algorithm selects random points homogeneously throughout the area. Due to the huge quantity of data, we have selected a reduced number to display this fact. It can be observed that we can not show a detailed image with all prior data as shown below:

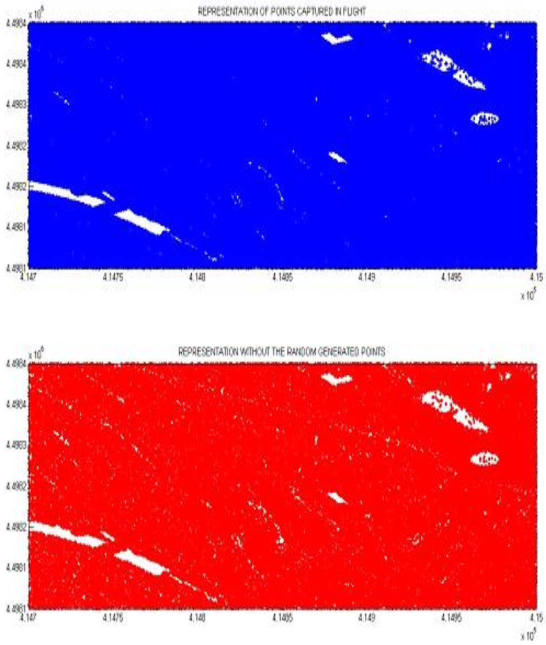


Figure 3: Representation of all points and 100000 points extratec using the algorithm

In the figure 3 we have represented all of the data in blue and data resulting from the extraction of 100 thousand points in red. In them we see the lack of data in certain areas that coincide with the existence of areas that absorb the laser beam and therefore can not be measured. These areas are, for example, a river, a fountain or a pool.

At the bottom there are more voids due to a lower amount of points, however, as we said before, we can not see that the data collection is homogeneous.

3 Interpolation

There are many different methods of interpolation: splines, linear, bi-cubic, polynomial... but in our case, most of them can not be used, since data must be in a regular grid that is not our case.

The algorithms used for the formation of the mesh of irregular triangles are primarily based on *Delaunay Triangulation*, since it is a computational structure, which allows the construction of an optimal triangulation in order to represent the field.

The triangles formed are as regular as possible, also the length of their sides is minimum, and the final triangulation is unique. It results an irregular network of triangles that appear to offer a more accurate picture of the real field, and allows a consistent interpolation of the height of each point or vertice. We need these algorithms to interpolate our data.

3.1 Delaunay Triangulation

A triangulation is a subdivision of an area in triangles. A triangulation of a set of points in the plane is a maximal family of triangles with disjoint interiors whose vertices are points of the set and where there is no point from such set.

A triangulation can be obtained by adding, as long as possible, straight segments that join points of the set without crossing the segments previously considered. In the figure 4 there are two triangulations of the same set of points:

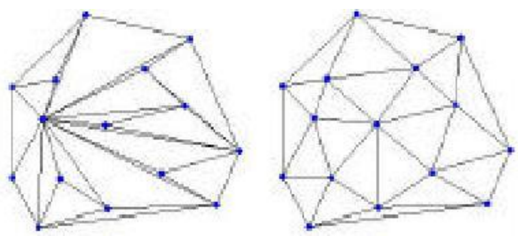


Figure 4: Two different triangulatios of a set of points

Goal: *Given a set of points in the plane, find a triangulation in which next points are connected to each other by an edge. Or, said in another way, where the triangles are as regular as possible.*

A triangulation T_1 is better than another T_2 when the smallest angle of the T_1 triangles is greater than the smallest angle of the T_2 triangles. That is, the optimal triangulation is that one which maximizes the minimum angle of triangles.

The characterization of a Delaunay triangulation is: Let $P = (p_1, p_2, ..., p_n)$ a set of points on the plane; a Delaunay triangulation of P will satisfy the following properties:

Property 3.1. *Three points p_i, p_j and p_k belonging to P are vertices of the same face of the Delaunay triangulation of P if and only if, the circle that passes through the points p_i, p_j and p_k does not contain points of P in its interior.*

Property 3.2. *Two points p_i and p_j from P form one side of Delaunay Triangulation P, if and only if there is a circle that contains p_i and p_j and inside it does not contain a single point of P.*

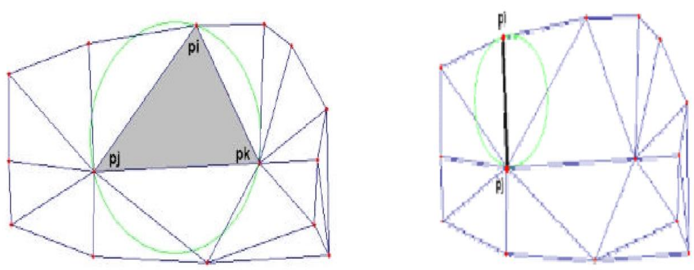


Table 1: Ilustration of property 1 (left side) and property 2 (right side)

With these two properties we can characterize a Delaunay triangulation as follows: Let P a set of points in the plane and T a triangulation of P; T is a Delaunay triangulation of P if and only if the circumscribed circle of any triangle T does not contain points of P.

Now, in the figure 5 we show an example of a Delaunay triangulation with a reduced number of points extracted randomly from the data set:

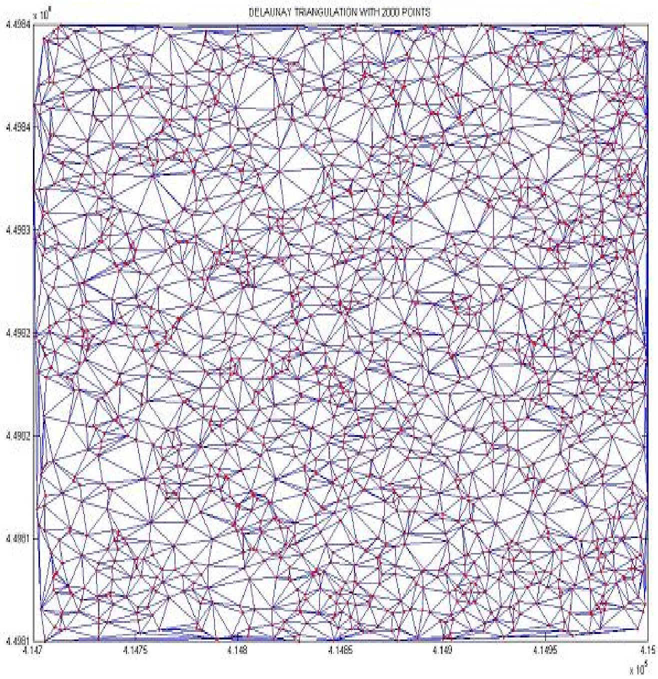


Figure 5: Example of Delaunay triangulation with 2000 points

3.2 Methods of interpolation

The digital model we have built is based on a regular grid (in this case we have used two different grids, one of $1m \times 1m$ and other of $1.5m \times 1.5m$) of our study area. The estimation of the heights of these points is performed using two different methods of interpolation based on Delaunay triangulation:

- **Linear:**The height of the interpolated point will be proportional to the heights of the vertices that form the triangle of the Delaunay triangulation which it is located in.
- **Nearest:** The height of the new point will be the height of the nearest vertex that form the triangle which it belongs to.

3.2.1 Process of point cutting off

Initially we create two different meshes, a $1 \times 1m$ and one of 1.5×1.5 meters. We shall make the process of reduction for each mesh.

The steps followed in the process are:

1. We start the process with all the data provided and check they satisfy the validation measures in the data of control.
2. We reduce the number of points using the random algorithm for reducing data, and again we control that the validation measures are fulfilled up in the control data.

The second step must be repeated as long as these measures are satisfied.

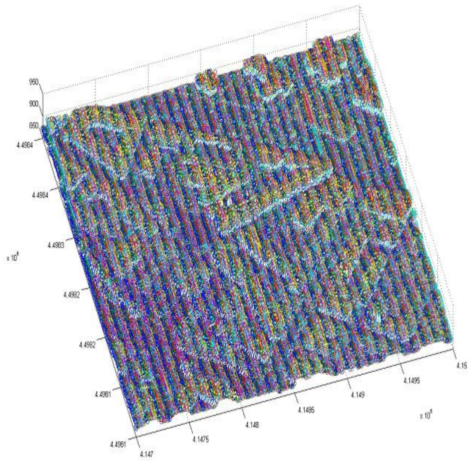
Validation measures:
- Maximum error less than 80 cm
- Mean square error less than 20 cm (only used for the grid of $1.5 \times 1.5m$)

4 Results

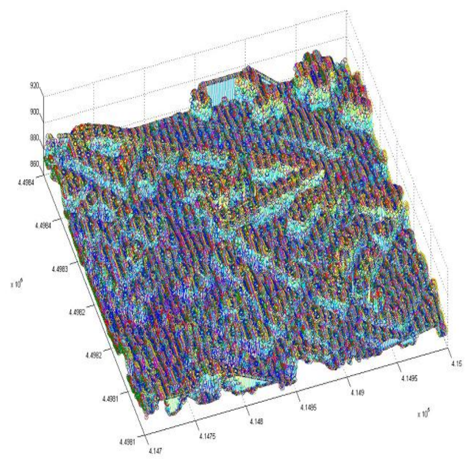
In the table 4 and 5 (at the end of the document) we can see the different reductions made in the process of interpolation with the two grids mentioned before.

Here below we show our digital model obtained trough the algorithm set up in MATLAB.

In the results table for the grid of $1 \times 1m$, we can observe that in spite of violating one of the criteria used to validate the digital model, we continue the reduction process. This is due to the fact that we have found an outlier as a rebound signal in some buildings close to one of the checkpoints. Successive point cuts allow the correction of this outlier. Surely, that's why the point we take the interpolated height from has been eliminated in the reduction. We can see that data of control in the image 6.



$1 \times 1m$ grid



$1.5 \times 1.5m$ grid

Table 2: Representation of the digital model using Matlab)

5 Conclusions

As we have shown in the table 4 and 5, we obtain different results in each grid an each interpolation method. The best result we got is using the nearest interpolation with a grid of 1 meter.

The table 3 shows a summary of the results obtained in the process. Initially, the altitude of the plane was 800m and now, using a grid of 1m, we have increased its altitude until 2700m; an using a grid of 1.5m, we have increased it until 2250m.

Results for a grid of $1 \times 1m$	
Linear Interpolation	Nearest Neighbor Interpolation
70000 points 0.78 points/m^2 2250m altitude of flight	50000 points 0.55 points/m^2 2700m altitude of flight
Results for a grid of $1.5 \times 1.5m$	
Linear Interpolation	Nearest Neighbor Interpolation
70000 points 0.78 points/m^2 2250m altitude of flight	75000 points 0.84 points/m^2 2200m altitude of flight

Table 3: Summary

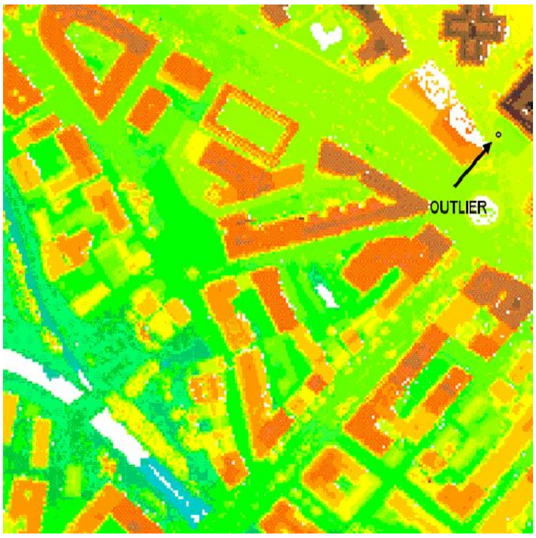


Figure 6: Situation of the outlier

6 References

1. *The Delaunay triangulation applied to digital surface models* Authors: José Enrique Priego de los Santos and María Joaquina Porres de la Haza Universidad Politécnica de Valencia.

2. 2. *Evaluating error associated with lidar-derived DEM interpolation* Authors: Christopher W. Bater and Nicholas C. Coops Computers & Geosciences, Volume 35, Issue 2, February 2009, Pages 289-300.

Validation test with 32 points				
DEM grid of 1m	Error in the linear interpolation		Error in the nearest interpolation	
Original data 210418 Data 2.35 Points/m^2 800 m Height Flight	Max Min Aver Median Std. Dev.	0.295 $1.13 \cdot 10^{-5}$ 0.0869 0.0655 0.076	Max Min Aver Median Std. Dev.	0.32 0 0.072 0.0655 0.081
1^{st} Data Reduction 100000 Data 1.11 Points/m^2 1300 m Height Flight	Max Min Aver Median Std. Dev.	0.327 0.00055 0.0922 0.0654 0.089	Max Min Aver Median Std. Dev.	5.44 0 0.259 0.045 0.951
2^{nd} Data Reduction 75000 Data 0.84 Points/m^2 2200 m Height Flight	Max Min Aver Median Std. Dev.	0.287 0.002 0.081 0.049 0.079	Max Min Aver Median Std. Dev.	0.47 0 0.095 0.035 0.108
3^{th} Data Reduction 70000 Data 0.78 Points/m^2 2250 m Height Flight	Max Min Aver Median Std. Dev.	0.226 0.002 0.072 0.045 0.064	Max Min Aver Median Std. Dev.	0.3 0 0.084 0.035 0.084
4^{th} Data Reduction 60000 Data 0.66 Points/m^2 2500 m Height Flight	Max Min Aver Median Std. Dev.	**0.664** 0.001 0.112 0.073 0.131	Max Min Aver Median Std. Dev.	0.3 0 0.092 0.03 0.091
5^{th} Data Reduction 50000 Data 0.55 Points/m^2 2700 m Height Flight			Max Min Aver Median Std. Dev.	0.26 0 0.078 0.03 0.079
6^{th} Data Reduction 60000 Data 0.44 Points/m^2 3000 m Height Flight			Max Min Aver Median Std. Dev.	**0.484** 0 0.246 0.03 0.846

Table 4: Results of the interpolation with different reductions using a grid of 1m

Validation test with 32 points				
DEM grid of 1.5	Error in the linear interpolation		Error in the nearest interpolation	
Original data 210418 Data 2.35 Points/m^2 800 m Height Flight	Max Min Aver Median Std. Dev. MSE	0.455 0 0.100 0.062 0.111 0.111	Max Min Aver Median Std. Dev. MSE	0.45 0 0.99 0.075 0.097 0.097
1^{st} Data Reduction 100000 Data 1.11 Points/m^2 1300 m Height Flight	Max Min Aver Median Std. Dev. MSE	0.465 0.077 0.118 0.087 0.109 0.373	Max Min Aver Median Std. Dev. MSE	0.78 0 0.135 0.085 0.166 0.856
2^{nd} Data Reduction 75000 Data 0.84 Points/m^2 2200 m Height Flight	Max Min Aver Median Std. Dev. MSE	0.424 0.001 0.107 0.085 0.097 0.296	Max Min Aver Median Std. Dev. MSE	0.47 0 0.108 0.07 0.117 0.421
3^{th} Data Reduction 70000 Data 0.78 Points/m^2 2250 m Height Flight	Max Min Aver Median Std. Dev. MSE	0.423 0.006 0.09 0.076 0.094 0.278	Max Min Aver Median Std. Dev. MSE	**20.5** 0 0.735 0.07 3.608 403.53
4^{th} Data Reduction 60000 Data 0.66 Points/m^2 2500 m Height Flight	Max Min Aver Median Std. Dev. MSE	**2.575** 0.002 0.191 0.089 0.131 6.226		

Table 5: Results of the interpolation with different reductions using a grid of 1.5m