

# INVERSE CALIBRATION OF LIDAR FILTERING PARAMETERS: UCODE/GRASS INTEGRATION

M. A. Brovelli, S. Lucca

Politecnico di Milano, DIAR, Piazza L. da Vinci 32, 20133 Milano, Italy  
maria.brovelli@polimi.it, sara.lucca@mail.polimi.it

Commission IV, WG IV/5

**KEY WORDS:** LiDAR, Filtering, Calibration, Accuracy, GRASS, UCODE

## ABSTRACT:

The paper presents a procedure developed to calibrate the value of parameters used in algorithms that perform LiDAR's data filtering.

These cascade commands, developed by the Geomatic Laboratory of the Politecnico di Milano, were created for the software GRASS GIS and were empirically calibrated for LiDAR data with "low" spatial resolution (0.9 point/m<sup>2</sup>).

The final aim of being able to work with every spatial resolution LiDAR dataset drove us to re-calibrate data employing a structured although not empirical procedure.

UCODE\_2005 from USGS was the software employed in the calibration, mainly chosen because of its flexibility and adaptability to every kind of model although its original creation for hydrological modelling purposes.

UCODE\_2005 carries out an inverse model calibration in an iterative way and performs the sensitivity analysis on the parameters.

In UCODE\_2005 the user is required to specify the model to run, the parameters and the observation value that the software will use to verify the accuracy of every iteration result.

After the integration of the two programs, the parameter calibration was performed separately for each command in a sub-region of the whole data-set; this approach allows to obtain faster results, especially when the data-set consists in million of points.

In order to verify the quality of the results, the values obtained were tested in a sub-region independent from the one used in the calibration step.

The paper points out that the calibration modality bring significant results and, once developed, can be easily repeated with different data.

## 1. INTRODUCTION

The LiDAR system (Light detection and ranging) is a recent technology that permits the creation of 3D models (DTM, DSM).

The system allows to obtain a high point density (some point per square meters) and high accuracies (10-15 cm).

Because of it's recent development this system still presents some uncertainties in the data filtering process, that is the automatic distinction, by the use of specific algorithms, between points that belong to bare earth, vegetation or buildings.

In this work filtering algorithms developed by the Laboratory of Geomatic in the Politecnico di Milano – Polo di Como (Brovelli et. al. 2002-2004 ) were used.

The cascade commands are three: v.lidar.edgedetection, v.lidar.growing, v.lidar.correction and for each of them different execution parameters, to ensure flexibility, have to be set.

All the commands were empirically calibrated once using LiDAR dataset with low spatial resolution (0.9 points/m<sup>2</sup>), and the parameter values obtained were proposed as default in the last GRASS release (6.3) available at <http://grass.itc.it/>.

This paper presents the work done to create a structured calibration procedure in order to guarantee the repeatability.

As calibration software UCODE\_2005 from USGS was used since it deals with different kind of model although it was created to deal with hydrological ones.

In the paper the dataset and the procedure used to recalibrate the command parameters are described.

The results obtained with the calibration were verified in a zone independent from the one used for the calibration.

In the first part the paper presents an overview of the available algorithms of LiDAR data filtering and some details about the ones we used.

It is also presented the software used for the calibration (UCODE\_2005) and the integration procedure between UCODE\_2005 and GRASS GIS.

The central part of the paper describes the dataset used and shows how the procedure is organized, both for the calibration and the verification aspect, and the results obtained for every cascade command.

## 2. FILTERING ALGORITHM

In this work some filtering algorithms are used, their purpose is the classification of points in order to differentiate between the ones belonging to bare earth and the ones belonging to objects (buildings, vegetations...).

In general different kind of filtering algorithms exist and it's possible to differentiate them because of their different approaches and characteristics.

A first classification can be done by means of their characteristics, for example filters can use different kind of input (raw cloud data or the result of a rasterization procedure) and can perform the analysis in one step or in an iterative way.

Another difference between algorithms consist in the number of points classified each time because the classification can be performed in three modality: one point can be a function of another point or a function of many points and many points can be function of a group of neighbouring points.

Another characteristic of filter consist in the information used, in fact some filters use only the last pulse of LiDAR data while others use also the first one (Brovelli et al., 2002; Tarsha-Kurdi et al., 2006). Some filters can use other kind of information like

aerial images, existing DEMs (Bretar and Chehata, 2007; Matikainen et al., 2007; Secord and Zakhor, 2007), land use maps, cadastral maps and a wide variety of multispectral data (Steinle and Vogtle, 2001).

As seen filtering algorithms can be discriminated on the basis of different operative characteristics but a more interesting distinction can be done upon the assumptions on which the different filters work.

Tovari and Pfeifer (2005) point out four distinct algorithm types summarized below.

**Morphological or slope based algorithms:** these algorithms use height differences to recognize points belonging to objects. The classification is performed using a threshold that determines the admissible height difference between two neighbour points such that one can be considered as bare earth. This approach can use also a fixed slope or gradient threshold values instead of height differences.

**Progressive densification algorithms:** these filters at first identifies some points belonging to the ground and then, depending on those, classify as ground more points. Usually the points used as seed are the ones with lower height. Additional ground points are determined by investigating their neighbour in the reference surface.

**Surface based algorithms:** these filters use a parametric surface that iteratively approaches to hypothetic bare earth. The surface is modified depending on the influence of the individual input points.

**Clustering-Segmentation algorithms:** these filters are based on the idea that a cluster of points belong to an object if it has height values greater than its neighbours. In these cases the classification is performed in two steps, at first a segmentation is carried out and then the segment are divided in different classes depending on the differences in height between segments.

### 3. GRASS FILTERING ALGORITHM

The filtering algorithms used in this work were developed for the GIS GRASS.

GRASS (Geographic Resources Analysis Support System) is a GIS software used for geospatial data management and analysis, image processing, graphics and maps production, spatial modeling, and visualization (Mitasova, H., Neteler, M., 2004.).

It is a Open Source and Free Software GIS with GNU GPL licence and this means users can run the program for any purpose and can redistribute copies but also, and more important, user can study how the program works and can improve the program and release the improvements.

The cascade commands are three (`v.lidar.edgedetection`, `v.lidar.growing`, `v.lidar.correction`) and are present in the last release of GRASS (Brovelli et al., 2002 - 2004).

These algorithms can be classified as morphological or slope based and for each of them many execution parameters, ensuring algorithms flexibility, have to be set.

`v.lidar.edgedetection` allows the detection of the edges of the surface objects; where an edge can be thought as a significant change in the height value corresponding to a small shift of the horizontal position. The algorithm performs at first an approximation of the DSM by means of bilinear and bicubic spline functions with Tikhonov regularisation in a least squares approach. The two surfaces are regularised minimising the gradient and the curvature. In the former case a low regularisation parameter ( $\lambda_g$ ) brings the interpolating functions as close as possible to the observations, whereas in the latter one the choice of a high value for  $\lambda_r$  gives a rough and loose-fitting surface.

The imposition of a unique threshold to the gradient magnitude is not suitable because if a low value was chosen it would be

impossible to discriminate between an actual edge and possible measurement noise and if a high value was chosen only very sharp height changes would be detected. The basic hypothesis is that noise corresponds mostly to an isolated observation or, at least, adjacent noises are generally not organised in a regular shape. In contrast, an edge shows a regular, chain-like behaviour.

For these reasons two thresholds for the magnitude gradient, the high ( $T_g$ ) and low ( $t_g$ ) thresholds, are set. Every point P where the magnitude gradient exceeds  $T_g$  is classified as a possible edge point. For every point where the magnitude gradient is lower than  $T_g$  but exceeds  $t_g$  we find, along the direction of the maximum direction of gradient rise (perpendicular to the direction of the edge vector), the two neighbouring and opposite points P1 and P2. If these points have the same edge direction of P and if the magnitude gradient for the eight nearest neighbouring points exceeds  $T_g$  in at least two instances, the point will be classified as a possible edge point. In other cases it is classified as non-edge point.

The output contains the classification of the nature of the measurement points (edge, terrain).

The parameters to be set are the interpolation step in east and north direction (See, Sen), the Tikhonov regularising parameter with bilinear splines ( $\lambda_g$ ), the two thresholds  $t_g$  and  $T_g$  for the gradient magnitude, the threshold  $\theta_g$  for the edge direction, the Tikhonov regularising parameter with bicubic splines ( $\lambda_r$ ).

`v.lidar.growing`: this command fills-in the edge previously obtained using the idea that the inner part of an object has generally a greater height than its edges. Using the mean height value of the points within each cell the data are rasterized with a resolution  $t_d$  equal to the minimum data raw density. For each cell the presence of points with double pulse is evaluated (difference between first and last pulse greater than  $t_j$ ). Starting from the cells classified as 'edge' and with only one pulse, all the linked cells are found and a convex hull algorithm is applied on them computing at the same time the mean of the corresponding heights (mean edge height). The points inside the convex hull are classified as objects in case their height is greater or equal to the previously mean computed edge height.

The output contains the classification in four classes (terrain, terrain with double pulse, object with double pulse, object).

The parameters to be set are a rasterizing grid resolution to be used in the region growing algorithm ( $t_d$ ) and the minimum difference to assume a double pulse for each cell ( $t_j$ ).

`v.lidar.correction`: this command corrects imprecision that the `v.lidar.growing` procedure can generate. A bilinear interpolation with Tikhonov regularising parameter ( $\lambda_c$ ) only on the ground points is performed.

The analysis of the residuals between the observations and the interpolated values compared with two thresholds  $t_c$ ,  $T_c$  consents the reclassification of some points to obtain the four output classes (terrain, terrain with double pulse, object with double pulse, object).

The parameters to be set in the last command are the spline steps in east and north direction (See, Sen), the Tikhonov regularising parameter with bilinear splines ( $\lambda_c$ ), the two thresholds  $t_c$  and  $T_c$  for residuals analysis.

All the parameters were once calibrated in an empirical way for LiDAR dataset with low spatial resolution ( $0.9 \text{ point/m}^2$ ), so there was the necessity to create a structured calibration procedure to obtain calibrated parameter values for different spatial resolution dataset.

### 4. INTRODUCTION TO UCODE\_2005

UCODE\_2005 is a software developed by United States Geological Survey (USGS) that performs model calibration with nonlinear regression methods and sensitivity analysis.

It can be applied with any application model or set of models; the only requirement is that they have numerical (ASCII or text only) input and output files and that numbers in these files have sufficient significant digits (Poeter, E., P., Hills, M., C., Banta, E., R., Mehl, S., Christensen, S., 2008.).

UCODE\_2005 is built using the JUPITER API capabilities and conventions, which facilitates inter program communications. The estimated parameters can be defined with user-specified functions: for example prior, or direct, information on estimated parameters also can be included in the regression.

UCODE\_2005 needs different files to work, and the most important one is the configuration file (.in).

The .in file specifies the command to run the model, the parameters to be evaluated and their information (starting value, lower reasonable value, largest reasonable value, scale parameter values, adjustability, fractional amount of perturbation for sensitivity, applied log-transformation, maximum fractional change between iterations, etc.), and the other files needed for the calibration procedure: the model input file (.if), the model input file template (.tpl), the model output file (.est), the model output file structure (.ins).

The program to run requires the definition of the parameters to be estimated and the definition of the observations that it use to verify every iteration results.

In the present work all the parameters had been defined in the configuration file while for the observations an ad hoc file (.flo) was created.

It contains all the observation used by the software to verify the accuracy of the simulated equivalent values.

Observations to be matched in the regression can be any quantity for which a simulated equivalent value can be produced, thus simulated equivalent values are calculated using values that appear in the application model output files and can be manipulated with additive and multiplicative functions, if necessary.

The file .if (model input file) and .tpl (input file template) are used by UCODE\_2005 to update the parameter values every iteration.

In the configuration file the last pair of files to be defined are the model output file (.est) and the model instruction file (.ins). The former is the file in which the results of the model at every iteration are saved.

## 5. GRASS GIS/UCODE\_2005 INTEGRATION

In order to perform the calibration procedure on the algorithm parameters it was necessary to integrate the two software packages (UCODE\_2005 and GRASS GIS).

The integration is due to the need of commands to work within GRASS environment.

A schema of the file structure behind the calibration procedure is shown in figure 1.

The UCODE\_2005 files can be presented in the configuration file divided in different section (command line, parameters, observations, model input file and model output file).

The integration between the two software is performed thanks to an intermediate script that read the current value of the parameters and pass it to the "GRASS script" in order to perform the filtering step.

At every iteration UCODE\_2005 executes the intermediate script that reads the actual parameter values in the UCODE\_2005 model input file and executes the "GRASS script" which in turn executes some commands and creates the numerical output file.

UCODE\_2005 finally reads the output file accordingly to the Model output file component and compares the observation simulated equivalent with the observations value.

Depending on their difference UCODE\_2005 updates the parameters values in the model input file and iterates the process.

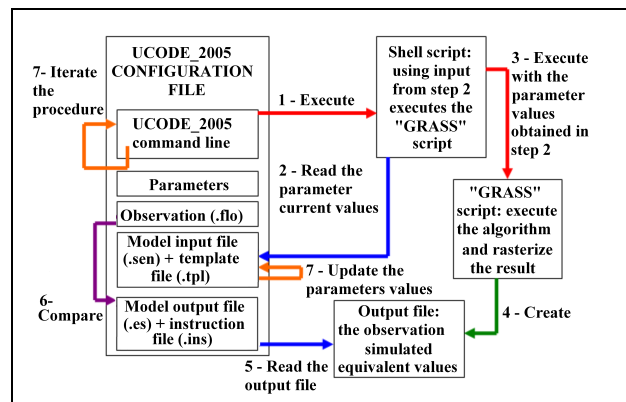


Figure 1. UCODE\_2005/GRASS GIS integration scheme

## 6. EXAMPLES

The data used covers part of the city of Sistiana (Friuli – Italy); it consists in LiDAR raw data and an orthophoto of part of the area.

The LiDAR dataset contains about 7 million points with spatial resolution of 5 points per square meter.

It was decided to use only a part of the dataset to perform the calibration procedure (figure 2.).



Figure 2. Area used for the calibration procedure

The calibration was structured in this way: each cascade command is calibrated separately, the result of the algorithm is rasterized and each cell has a simulated equivalent value.

The observations to be compared with the output were created rasterizing the result of a building map of the area digitized with the support of the orthophoto.

After the calibration procedure a validation in an independent zone (figure 3.) in order to evaluate the accuracy of the results was performed.

In this second zone the algorithms with the calibrated parameter values were executed and the result, after rasterization was compared with the raster map manually obtained from the building digitizing.

All pixels belonging to building edges or building polygons are given value 1 while the remaining cells have value 0.

Therefore the raster difference layer is characterized by three values: 1 (pixel in the digitized layer but not in the command result), 0 (pixel present in both layers), -1 (pixel in the command result but not in the digitized layer).



Figure 3. Area used for the verification procedure

The accuracy of the commands was evaluated counting the number of pixels present in both the maps, therefore the 0 value pixels, with respect the total number. For each command the performance with the default values, the calibration results and the performance with the calibrated values are presented.

### 6.1 v.lidar.edgedetection

The first command consents the extraction of the object edges. Table 1. reports the default values of parameters default values and the calibrated ones (the convergence of the values was obtained after nine iterations).

Parameter	Default value	Calibrated value (after 9 iterations)	Approximated value
See	4	1.749	1.7
Sen	4	1.766	1.7
Tg	6	4.01	4
tg	3	3.51	3.5
$\lambda_g$	0.01	0.00113	0.001
$\theta_g$	0.26	0.1167	0.12
$\lambda_r$	1	1.557	1.6

Table 1. v.lidar.edgedetection - Parameter values

The calibrated spline step values (See, Sen) are lower than the default values because the dataset used has a higher spatial resolution (5 points/m<sup>2</sup>) than the one used for the empirical calibration of the values used as default (0.7 points/m<sup>2</sup>). The result obtained with the default values in the calibration area is presented in figure 4: the edges are not extracted precisely, in fact some points belonging to roofs are classified as edges.

The performances of the default parameter values are presented in a more quantitative way in the table 2 (statistics obtained in the validation zone from the raster difference layer).

Whole zone	Cell number	N° Pixel value		
		1	0	-1
	17017	299 (2%)	11167 (66%)	5551 (33%)

Table 2. Pixel statistics –results with default values

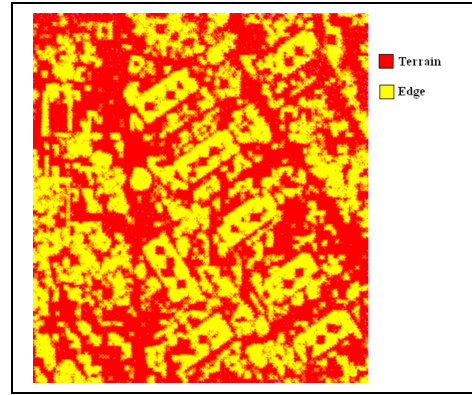


Figure 4. v.lidar.edgedetection result with default values

The accuracy is low, in fact for the whole image the percentage of correct pixels is about 66%. The results obtained using the calibrated values allow to a more precise definition of the edges (figure 5. and table 3. for the validation zone).

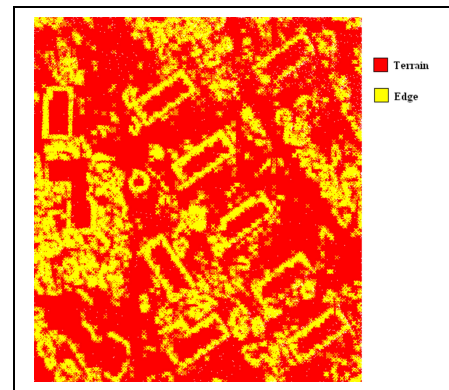


Figure 5. v.lidar.edgedetection result with calibrated values

As visible the accuracy percentage is higher with respect the previous shown, in fact the percentage of correct pixels for the whole image is in this case around 82%.

Whole zone	Cell number	N° Pixel value		
		1	0	-1
	17017	322 (2%)	13984 (82%)	2711 (16%)

Table 3. Pixel statistics –results with calibrated values

### 6.2 v.lidar.growing

This second command aims at filling the areas within the edges previously extracted.

The input file are the result obtained with the v.lidar.edgedetection command, the first pulse map and the parameters presented in table 4.

Parameter	Default value	Calibrated value (after 7 iteration)	Approximated value
tj	0.2	0.2512	0.25
td	0.6	0.1643	0.16

Table 4. v.lidar.growing - Parameter values

In the case of using the default values the result of the command `v.lidar.growing` is presented in figure 6. For this command the values used for calibration and validation are those belonging to the category of objects. The figure shows that the result of the `v.lidar.growing` command is quite correct, apart from a building in the lower part of the image. The statistics presented in table 5. also show a high percentage of correct pixels.

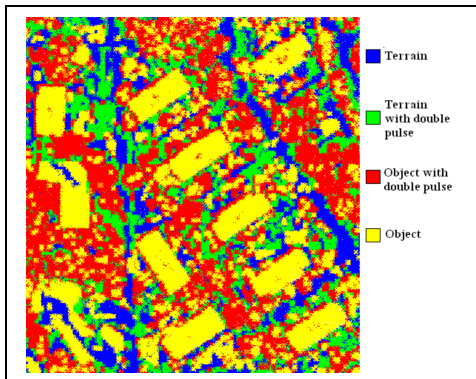


Figure 6. `v.lidar.growing` result with default values

Whole zone	Cell number	N° Pixel value		
		1	0	-1
	17017	511 (3%)	13718 (81%)	2788 (16%)

Table 5. Pixel statistics –results default values

The result obtained with the calibrated values is less precise in the detection of some buildings but allows a better distinction between low (terrain with double pulse) and high vegetation (object with double pulse) (figure 7.).

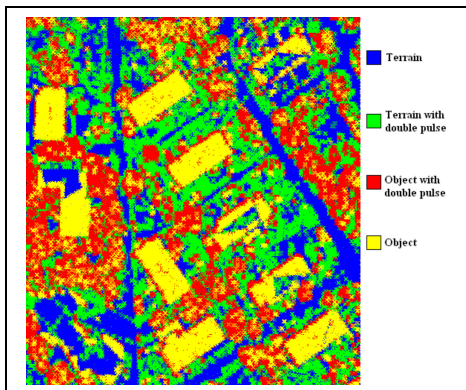


Figure 7. `v.lidar.growing` result with calibrated values

The statistics in the validation zone (table 6.) more or less are equivalent to those of the calibration zone.

Whole zone	N° Cells	N° Pixel value		
		1	0	-1
	17017	1170 (7%)	15024 (88%)	823 (5%)

Table 6. Pixel statistics –results with calibrated values

Moreover it's useful to remember that some errors of the `v.lidar.growing` procedure will be corrected by the `v.lidar.correction` command.

### 6.3 `v.lidar.correction`

As said this passage is useful to correct some residuals (see as example the building in the lower left part of image in figure 6.).

The input in this case is the output of the previous command and the parameters are shown in table 7.

Parameter	Default value	Calibrated value (after 7 iteration)	Approximated value
Scce	25	28.27	28
Scn	25	32.90	33
$\lambda_c$	1	0.8699	0.87
Tc	2	2.188	2.2
tc	1	1.783	1.8

Table 7. `v.lidar.correction` - Parameter values

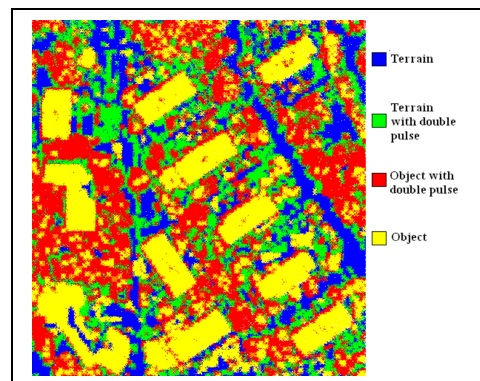


Figure 8. `v.lidar.correction` result with default values

With the `v.lidar.correction` procedure the number of pixel classified as objects and also the percentage of correct pixels (table 8.) increases with respect to the `v.lidar.growing` result (compare table 8. vs table 5. percentages).

Whole zone	Cell number	N° Pixel value		
		1	0	-1
	17017	498 (3%)	14430 (85%)	2089 (12%)

Table 8. Pixel statistics –results with default values

The improvement of the results is present also using the calibrated values; in fact, as figure 9. shows, all the buildings that in the `v.lidar.growing` procedure were not completely filled now are more correct.

The same is visible in table 9. that reports the statistics for the validation zone (compare table 9. vs table 6. percentages).

Whole zone	N° cells	N° Pixel value		
		1	0	-1
	17017	522 (3%)	15577 (92%)	918 (5%)

Table 9. Pixel statistics –results with calibrated values

For this command the better performances of the calibrated values are evident: the percentage of correct pixel for the whole

image is around 92% while that corresponding to that default values is around 85%.

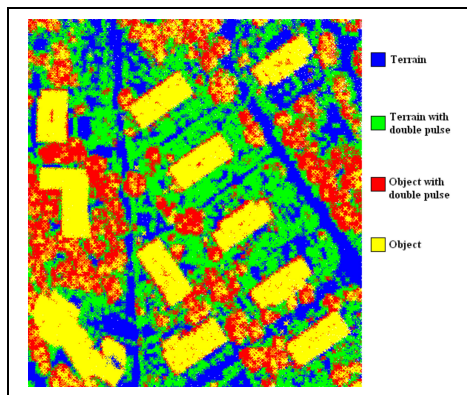


Figure 9. v.lidar.correction result with calibrated values

## 7. CONCLUSIONS

The work presented in this paper aims at solving two problems: the creation of a structured and repeatable calibration procedure and the evaluation of the accuracy of LiDAR data filtering algorithms.

All the filtering algorithms parameters were calibrated for low spatial resolution LiDAR data and the use of the same values with different resolution data generated results with low accuracy, as seen in the example presented.

The calibration procedure was implemented with the integration of UCODE\_2005 and GRASS with very good results.

The procedure can be repeated with different LiDAR resolution dataset.

The process can be performed using only a part of the whole dataset in order to make the calibration procedure faster; as described in this work all the results obtained can bring significant results also in the whole area.

A second topic of this work is related to the accuracy of the proposed filtering algorithms; as described in the paper two filtering test were done, the former using the "default" values and the second with calibrated ones.

As visible in the images the algorithms are generally reliable because they can classify objects with a good accuracy, but the results are significantly better if the calibrated values are used.

In conclusion, for a correct LiDAR dataset filtering process it is necessary to perform at first the calibration procedure on a test field (a small portion of the whole area in which external information regarding buildings and vegetation is available) and then to apply the filtering procedure with the calibrated parameters.

## AKNOWLEDGEMENTS

We would like to thank OGS (Dr. F. Coren) (Istituto Nazionale di Oceanografia e di Geofisica Sperimentale) having provided the LiDAR dataset and the aerial imagery.

This research was partially supported by grants of the Italian Ministry for School, University and Scientific Research (MIUR) in the frame of the project MIUR-COFIN 2007 "Free and open source geoservices as interoperable tools for sharing geographic data through the internet".

## 8. REFERENCES

- Bretar, F., Chehata, N., 2007. Digital terrain model on vegetated areas: Joint use of airborne LiDAR data and optical images. *Photogrammetric Image Analysis*. 36 (3/W49A), pp. 19-24.
- Brovelli, M., A., Cannata, M., Longoni, U., M., 2004. LiDAR data filtering and DTM interpolation within GRASS, *Transaction in GIS*. 8(2), pp.155-174.
- Brovelli, M., A., Cannata, M., 2002. Digital terrain model reconstruction in urban areas from airborne laser scanning data: The method and the example of the town of Pavia (Northern Italy). *International Archives of Photogrammetry and Remote Sensing*. 34, pp. 43-48.
- Crosetto, M., Tarantola, S., 2001. Uncertainty and sensitivity analysis: tools for GISbased model implementation. *International Journal of Geographical Information Science*, 5(15), pp. 415-437.
- Hills, M., C., 1998. Methods and guidelines for effective model calibration. *Water-Resources investigations report*, pp. 1-90.
- Matikainen, L., Kaartinen, H., Hyypä, J., 2007. Classification tree based building detection from laser scanner and aerial image data. *ISPRS Workshop on Laser Scanning and SilviLaser*, 36(Part 3/W52) pp. 280-287.
- Mitasova, H., Neteler, M., 2004. GRASS as Open Source Free Software GIS: Accomplishments and Perspectives. *Transaction in GIS*, 8(2), pp. 145-154.
- Poeter, E., P., Hills, M., C., 1998. Documentation of UCODE, a computer code for universal inverse modelling. *Water-Resources investigations report* pp. 1-116.
- Poeter, E., P., Hills, M., C., Banta, E., R., Mehl, S., Christensen, S., 2008. UCODE\_2005 and six other computer codes for universal sensitivity analysis, calibration and uncertainty evaluation. *Book 6 (11) Modeling Techniques, Section A. Ground Water*.
- Secord, J., Zakhor, A., 2007. Tree detection in urban regions using aerial LiDAR and image data. *Geoscience and Remote Sensing Letters*. IEEE 4, pp. 196-200.
- Steinle, E., Vogtle, T., 2001. Automated extraction and reconstruction of buildings in laserscanning data for disaster management. Automatic extraction of man-made objects from aerial and space images (III), Swets and Zeitlinger, Lisse, The Netherlands, pp. 309-318.
- Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., Smigiel, E., 2006. New approach for automatic detection of buildings in airborne laser scanner data using first echo only. *ISPRS Symposium of Commission III - Photogrammetric Computer Vision*, pp. 20-22.