Remote Sensing for Science, Education, and Natural and Cultural Heritage
Rainer Reuter (Editor)
EARSeL, 2010

Kemmelberg (Belgium) case study: comparison of DTM analysis methods for the detection of relicts from the First World War

Cornelis STAL[1], Jean BOURGEOIS[2], Philippe DE MAEYER[1], Guy DE MULDER[2], Alain DE WULF[1], Rudi GOOSSENS[1], Timothy NUTTENS[1], Birger STICHELBAUT[2]
[1] *Ghent University, Department of Geography*
[2] *Ghent University, Department of Archaeology*

Abstract. Laserscanning is a well known technology in a various range of scientific disciplines. This paper focuses on the manipulation of high resolution DTMs (Digital Terrain Models), to detect relicts in the landscape, dating from the First World War. Possible objects to detect are fire- and communication trenches, artillery emplacements and other features related to the material remains of the First World War. A common method to detect these objects is the use of historical aerial photographs taken during this war (1914-1918). A geodatabase, containing relicts can be made, based on the interpretation of these images (remote sensing), possibly complimented by on-site validation. The interpretation of historical photographs requires thorough and specialized knowledge of the elements on these pictures. Stichelbaut [1] has made a extensive research on the archaeological application of First World War aerial photographs in Flanders, Belgium. During the First World War the existing landscape around the Mount Kemmel (Kemmelberg: municipality of Heuvelland, Belgium) was totally destroyed and replaced by a war landscape with its own characteristic constructions. When the reconstruction started after the war, most war-related elements were removed. However, in some cases, elements are conserved and still visible in the landscape. Some of these features still exist but the height difference with the surrounding surface is very small and therefore they are hardly detectable with conventional techniques such as fieldwalking and archaeological aerial photography. This paper focuses on different filter techniques to detect these objects, using a convolution matrix. The resulting data is an important research tool for the cultural management of this fragile heritage as it will show where elements of the conflict landscape of the First World War are still preserved in the modern day landscape.

Keywords. archaeology, LiDAR, feature detection

Introduction

On behalf of this study, several sources are used. An efficient and linked use of these sources will result in the detection of archaeological features dating from the First World War. The base models consist of the DTM and the DSM (Digital Surface Model) of the region around Mount Kemmel, being the research area of this case study. A DTM is defined as a DEM (Digital Elevation Model) of the bare earth, while a DSM is a model of the earth, including vegetation, buildings and all other surface objects [2]. Hereby, airborne laserscanning or LiDAR (Light Detection And Ranging) is used. This acquisition technique has proven to be very useful for archaeology [3,4]. With these digital models, different filters are executed. By filtering a DTM or DSM, it is possible to highlight small height difference in the landscape. In a more mathematical point of view, the local topographic difference are subtracted from the structural topography of an area. Another important sources to detect the war heritage in the landscape are the First World War aerial photographs. These historical remote sensing images give an unparalleled record of the various entrenchments constructed in relation to battles that took place on and around Mount Kemmel. Besides these his-

toric images also modern aerial photographs were used in an attempt to locate the conflict archaeology in the modern landscape. An overview of the study area is shown in Figure 1.

Studying the different filtered models will show a number of topographic structures that are not immediately visible on the digital modern day orthophotos, provided by the Belgian National Geographic Institute with a resolution of 1 meter. The produced rasters show various results, and it is up to the archaeologist to decide, which visualization technique can usefully be applied, and how the archaeologist will use this material to detect relicts.

In section 1, an overview is given of digital elevation models, based on a LiDAR point set. Section 2 and 3 explain the used concepts for the filters, respectively a general remark about the convolution filter and the way they are programmed in *Java*. Section 4 explains the used filters that are programmed.

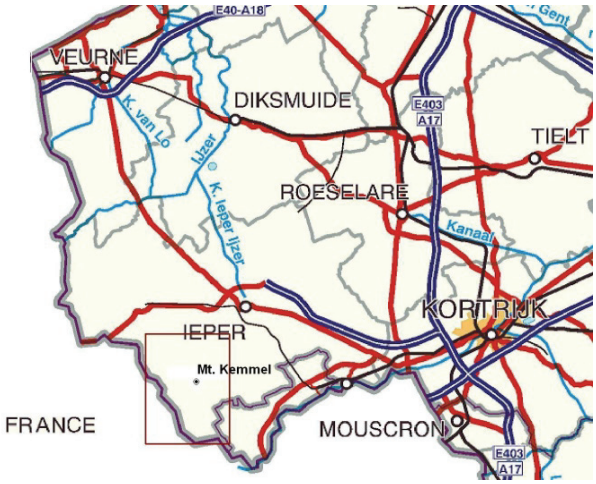

Figure 1: The test area is located in the south of the province of West Flanders, near the French border

1. Airborne laserscanning

An airborne LiDAR measurement has taken place around the Mount Kemmel in 2008 with an average point density of approximate 5 point per square meter. Based on the generated point set, two types of raster files are made. A DSM is constructed, where random divided points are interpolated to a grid with a fixed resolution of 50 cm. Besides, a DTM is constructed, where non-ground points are filtered out and a grid is calculated by interpolating the remaining ground points, also with a resolution of 50 cm. A qualitative analysis or general assessment of the datasets, as mentioned in [5, 6,7], will not be done here. Nevertheless, it is important mention the possibility of quality assessment in post processing, when an error budget is required for the model.

2. Convolution matrix

The discussed filters will all use a so called '*linear convolution operator*'. Here, a particular function is created to make a convolution matrix. A new cell value G_{ij} will be calculated for every height value f_{ij} in cell $i, j = 1,...,n$, by a linear combination of this matrix and the values of the kernel [8]. The function, executed by the convolution matrix, is also called as '*focal function*' [9]. A convolution matrix has got a uneven size of $(2m+1)\times(2m+1)$, with weights $w_{k,l}$ for each cell $k,l = -m,...,m$ in this matrix [10].

The new value G_{ij} is then determined by:

$$G_{ij} = w_{-m} f_{i-m,j-m} + w_{-m+1} f_{i-m,j-m+1} + \ldots + w_m f_{i+m,j+m}$$
$$= \sum_{k=-m}^{m} \sum_{l=-m}^{m} w_{lk} f_{i+k,j+l}$$

Depending on the convolution matrix, a new cell value G_{ij} can describe different quantities. Some examples are either the curvature or the gradient by a *Sobel*-filter. However, it is possible that the new values will calculate an index about the presents or absence of a particular spatial phenomenon, like, for example, the shade by a *Hillshade*-operator [9].

3. Implementing the filters

ArcGIS 9.3 only supports a small number of filters which can be executed on raster datasets. The limitation of this support has probably to do with the fact that *ArcGIS* is, in principle, a vector-oriented GIS. However, the program contains some tools which make it possible to process rasters. The introduction of the *3D Analyst-toolbox* has increased this number for sure. Some of the modules from this toolbox are used in this research, like *hillshade* and *slope*. Other filters have to be executed by another (raster-oriented) GIS or, as in this case, have to be programmed. These filters can be implemented in personalized tools for *ArcGIS*. This means, by programming on demand filters, *ArcGIS* can be extended and the raster-support increases.

3.1. Java

Java is chosen to program the filters, because of its support in *ArcGIS*. Thanks to the availability of rasters in the standard *ESRI*-ASCII-format, many convolution operators can be programmed easily. The simple structure of these rasters makes it easy to read files line by line. The files that have to be filtered, have a standard header, containing 6 lines with information about the geometry of the raster [11]. After storing theses values, an empty matrix is made with a size of $ncols \times nrows$. This matrix will be filled row by row with values from the text file. All elements in a row have a predefined separator, so a simple split and conversion will be enough to store every string in a matrix like numbers.

The filtering is used according to the above conceptual exposition. The kernel contains the same primitives as the base matrix. Difficulties will occur on the edges of the matrix. Whenever a value of the kernel is outside the array of height values, an error will be raised and the central value f_{ij} is used. If all $m \times m$ elements of the kernel are filled, they are sent to the desired filtering module, where the calculations are made. The results of the filtering are put in a new matrix. The entire area will be processed by this method and finally, the procedure at the start will be executed in reversed direction. Hereby, the headers are written at first, followed by each and every corrected value G_{ij}. Again, the values are separated by a space and everything is stored in a new ASCII-file. Besides, it must be noticed, that the above method is useful for linear filters, but for non-linear filters as well. By using this method, non-linear filters can be implemented easily, so per kernel, for example, the minimum or standard deviation can be calculated.

3.2. Python

The filtering is executed, based on an ASCII-file and the result is such a file as well. Although these files can be converted by *ArcGIS* to a real *ESRI*-raster this is a cumbersome procedure to execute one by one. The ASCII-files themselves can be loaded in *ArcGIS* too, but the necessary statistics and image pyramids are not stored for this format. It would be better to create an *ESRI*-raster, directly after the filtering process, so the files can be loaded in *ArcGIS* fast. However, the *ESRI*-raster is a binary type, which means, a conversion module is not easy to program without using the *Ar-*

cGIS-geoprocessor. To use the java-library of *ArcGIS* and its *geoprocessor*, the *ArcEngine* has to be used, which is another program from *ESRI*. An alternative method is using a *Python*-script. This script can be assimilated in the *Java*-code like a string. The *Python command line* has to be called. Then, each command line is prompted directly, like a physical user would work with the tool. It is possible to use an external script as well, which will be processed by *Java*-code.

4. Edge detection

A frequently used technique to detect objects (i.e. First World War trenches) in images is the *edge detection*. An edge can be defined as the boundary between objects, or parts of objects, in an image [10]. Every pixel with a height value h_{ij} can be seen as a function, defined by the value of the cell itself and its neighbouring cells. If the value of a cell is calculated by a new function again, it will be possible to emphasize the difference between the values of neighbouring cells, for example by using the first or second derivative. There are different filters, all with specific mathematical properties, that can achieve this. As part of the edge detection, an overview will be given of the *Sobel*- and *Laplace* filter. There are many other useful filters for this purpose [10] that will not be discussed here. Besides, the *Laplace* filter is called a *high-pass* filter by De Smith *et al.* [9], with only a limited application in the edge detection. *ArcGIS 9.3*, on the other hand, uses a *high-pass* filter with other parameters and without mentioning the *Laplace* filter. Both filters give comparable and useful results, as shown later.

4.1. 1st derivative – Sobel filter

The differential derivative of a function will give the *Prewitt's* filter, where every cell around *i,j* will get the same weight. The *Sobel* filter will assign a higher value to the four direct neighbours and a lower one to the four neighbours in the corner, resulting in the following filters:

$$G_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \qquad G_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

These filters will give a rating for the change in heights in respectively the *x* and *y* direction. The gradient or maximal slope is calculated, based on the first derivative of the function, as the square of the quadratic vectors in the *x* and *y* direction:

$$\nabla f(x,y) = G_{i,j} = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f_{ij}}{\partial x}\right)^2 + \left(\frac{\partial f_{ij}}{\partial y}\right)^2}$$

Results of these calculations are shown in Figure 2.

4.2. 2nd derivative – Laplace filter

The second derivative of f_{ij} is calculated using the *Laplace* filter:

$$\Delta f = \nabla^2 f = \frac{\partial^2 f_{ij}}{\partial x^2} + \frac{\partial^2 f_{ij}}{\partial y^2} \rightarrow \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

It should be noted that, by taking the second derivative, this filter is extremely sensitive for noise. This problem is practically solved by executing a noise reduction, using a *Gaussian* operator. The combination of both the *Laplacian* and *Gaussian* operator is known as the *Laplacian-of-Gauss* filter and is executed by filtering unfiltered and interpolated LiDAR-data [12]. The problem is the existence of a chance, that by noise reduction, the wanted relicts will disappear. That is why the

Laplace filter is implemented, without a *Gaussian* operation. Figure 3 (left) show some interesting structures, which still makes this filter useful, but there is a lot of noise in the images, as expected. The distinction between true noise and wanted relicts is thus difficult.

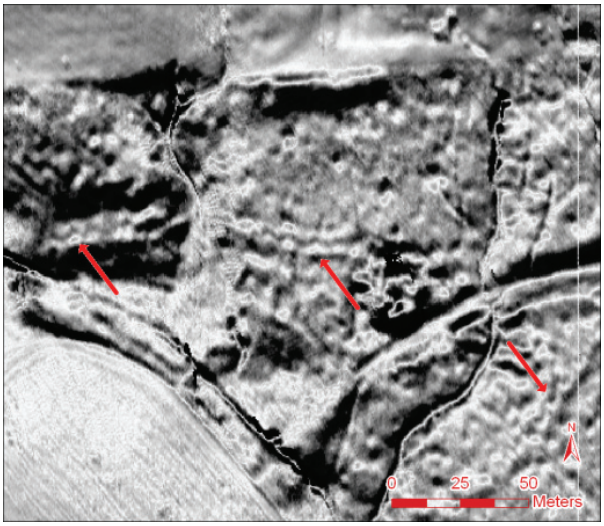

Figure 2: Results of the implementation of the *Sobel* filter

4.3. High-pass filter

The purpose of a *high-pass* filter is to strengthen the difference in contrast of an image. Hereby, the boundaries of objects will be sharpened and logically, this filter is called a sharpener [9]. The convolution matrix of a *high-pass* filter may contain different values, but it is always of the following form:

$$\begin{pmatrix} -c & -a & -c \\ -a & b & -a \\ -c & -a & -c \end{pmatrix}$$

It is possible that all direct and corner neighbours of a cell *i,j* have the same weight (*a*=*c*), but this is not necessary, like the *high-pass* filter used by *ArcGIS* [11]. The sum of all values in the convolution always equals zero, because they are normalized values. Figure 4 (right) shows the results of a *high-pass* filter for the given test area.

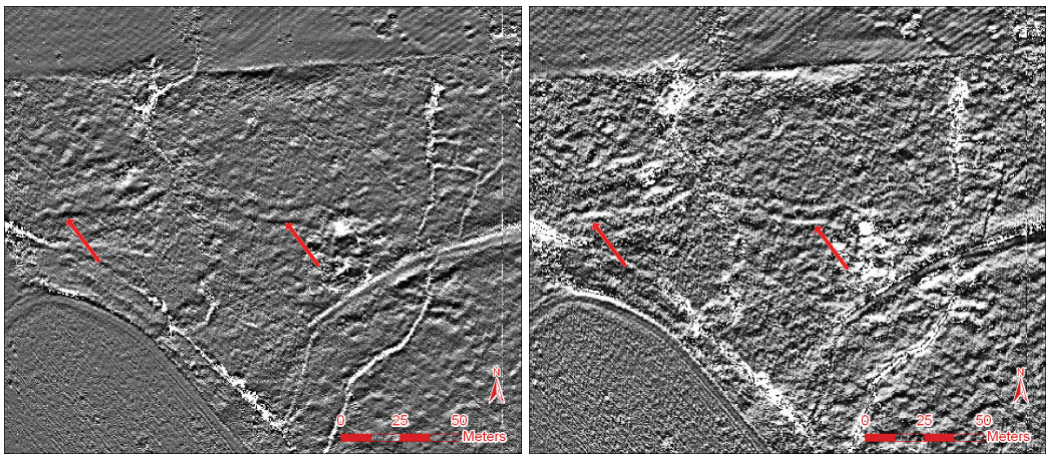

Figure 3: Results of the *Laplace* filter (left) and the *high-pass* filter (right)

4.4. Hillshade

A frequently used technique to distinctly visualize relief is by lighting it by a hypothetic light source. Here, parameters of this light source (the azimuth angle *a* and the zenith angle *z*) are combined with the properties of a cell *ij* and its neighbours to determine the shade at a cell *ij* . These properties exist of the average slope or gradient *G* and the aspect β. The following formula can be used, where the *hillshade* is expressed within the interval [0;255] and the angles are calculated in radians:

$$hillshade_{ij} = 255.0 \cdot (\cos z \cos G) + (\sin z \sin G \sin(a - \beta))$$

The average slope or gradient *G* *G* and the aspect β are calculated using a convolution matrix and equals the *Sobel* filter. To calculate the aspect, the average slope or gradient *G* is used [11].

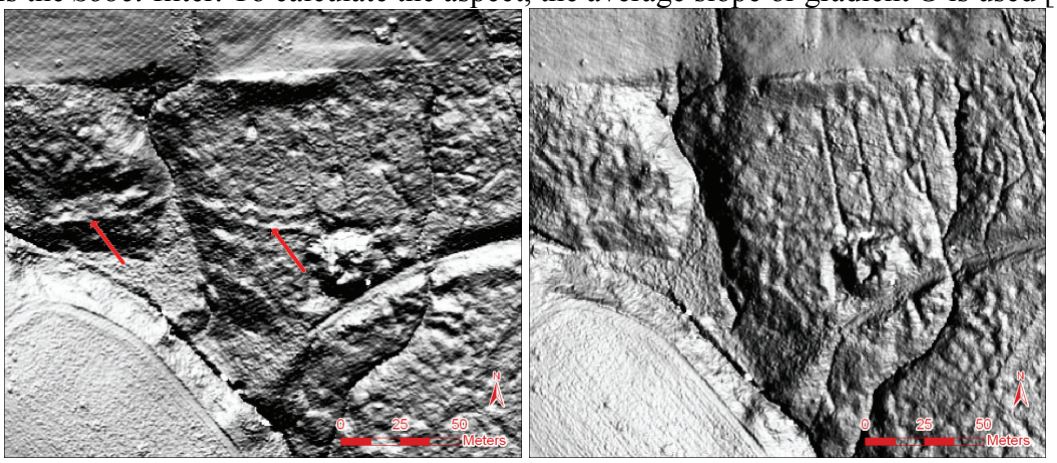

Figure 4: *Hillshade* with the light source in the north (left) and in the east (right)

Single shading is not an optimal method to visualize and detect micro relief [4,13]. It is logical that objects, parallel to the azimuth angle of the hypothetical light source, will not rise a shade. As a result, these objects will not have a difference in contrast with their neighbouring cells and it will not be possible to distinguish them. It is obvious that the trench with a west-east orientation, visible in Figure 4 (left), is not visible in Figure 4 (right). So the detection of objects is dependant of the direction of the hypothetic light source. This problem may be solved by using principal component analysis [4], where the new visualization is formed by a determined percentage of the information about the variability on the original dataset. Combining different *hillshade* results by summoning could be a solution as well, but it is not a perfect one [13]. Here, two results are chosen with perpendicular light sources. The resulting *pseudo-hillshade* is shown in Figure 5, with (left) and without (right) the visualization of the geodatadase, containing locations of war-related constructions.

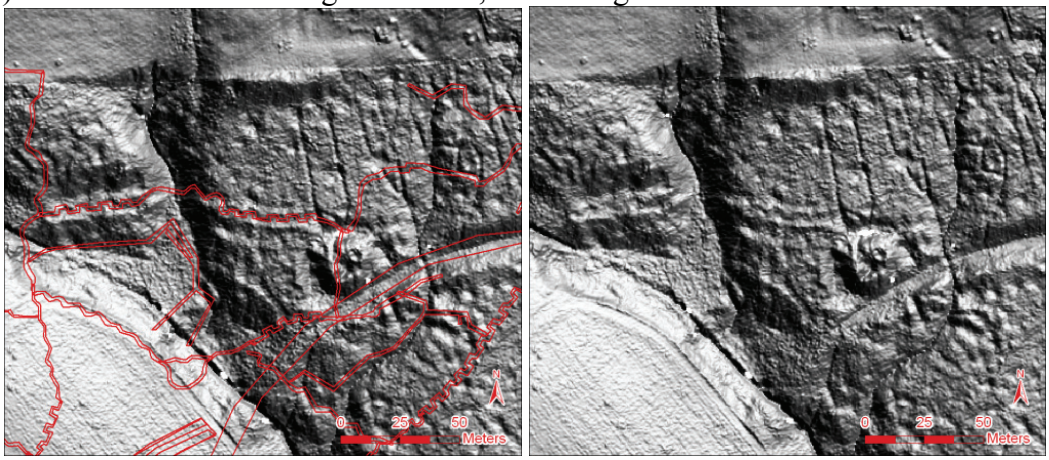

Figure 5: Pseudo *hillshade* with (left) and without (right) Stichelbaut's geodatabase, containing locations of war-related constructions in red

5. Conclusion

Not every filtering technique is useful to detect objects or to emphasize micro relief in a DTM, as shown with the *Laplace* filter and the *high-pass* filter. However, the results of the *Sobel* filter and the *pseudo-hillshade* offer good perspectives. Based on the figures in this article, it can be assumed that there are still many relicts from the First World War in the present landscape on and near the Mount Kemmel. This assumption is not only stated by studying the photogrammetric material from this war: the already mentioned database of [1] confirms this nowadays presence, as shown in Figure 5. Further analysis of the products of this research shows more relicts still exist. It can be concluded that these new documents will be a very useful tool for archaeological research.

Acknowledgements

This work forms a part of an extended archaeological research programme on the Mount Kemmel, executed by the department of Archaeology of the Ghent University, with technical support by the department of Geography of the same university. Our special thanks go to the province of West Flanders and provincial permanent delegate G. Perty in particular.

References

[1] Stichelbaut, B., *World War One Aerial Photography: An Archeological Perspective*, in *Department of Archeology*. 2009, Ghent University: Ghent. p. 416.
[2] Sties, M., et al., *Comparison of Digital Elevation Data from Airborne Laser and Interferometric SAR Systems.* International Archives of Photogrammetry and Remote Sensing, 2000. **33**: p. 866-873.
[3] Humme, A., R. Lindenbergh, and C. Sueur. *Revealing celtic fields from lidar data using kriging based filtering.* in *ISPRS Commission V Symposium*. 2006. Dresden, Duitsland.
[4] Devereux, B.J., G.S. Amable, and P. Crow, *Visualisation of LiDAR Terrain Models for Archeological Feature Detection.* Antiquity, 2008. **82**: p. 470-479.
[5] Kraus, K., et al. *Quality measures for digital terrain models*. in *International Archives of Photogrammetry and Remote Sensing, XXth ISPRS Congress*. 2004. Istanbul, Turkije.
[6] Kraus, K., et al., *Local accuracy measures for digital terrain models.* The Photogrammetric Record, Vol. XXI, part 116, 2006: p. 342-354.
[7] Sithole, G. and G. Vosselman. *Comparison of filtering algorithms*. in *ISPRS Workshop "3D Reconstruction from Airborne Laser-Scanner and InSAR data*. Dresden, Germany, 2003.
[8] Heijmans, H. (2000) *Rekenen met beelden: is een plaatje duizend woorden waard?* , 20.
[9] De Smith, M., M. Goodchild, and P. Longley, *Geospatial Analysis: a Comprehensive Guide To Priciples, Techniques and Software Tools*. 3 ed. 2009, Leicester, UK: Troubadour Publishing.
[10] Glasbey, C.A. and G.W. Horgan, *Filters*, in *Image Analysis for the Biological Sciences*. 1995, John Wiley & Sons, Inc.: New York, NY, USA p. 218.
[11] ESRI, *ArcInfo 9.3*, in *ArcMap*, ESRI, Editor. 2008: Redlands, California, USA.
[12] Wack, R. and A. Wimmer, *Digital terrain models from airborne laser scanner data - a grid based approach.* IAPRS, Vol. XXXIII, part 3, 2002: p. pp. 293-296.
[13] Yokoyama, R., M. Shirasawa, and R.J. Pike, *Visualizing Topography by Openness: A New Application of Image Processing to Digital Elevation Models.* Photogrammetric Engineering & Remote Sensing, 2002. **68**(3): p. 257-265.