DEVELOPMENT OF A METHOD NETWORK FOR OBJECT RECOGNITION USING DIGITAL SURFACE MODELS

G. Bohmann, J. Schiewe

University of Vechta, Research Center for Geoinformatics and Remote Sensing, PO Box 1553, 49364 Vechta, Germany, {gbohmann, jschiewe}@fzg.uni-vechta.de

Commission IV, WG IV/6

KEY WORDS: method integration, system architecture, object recognition, DEM/DTM/DSM, blunder analysis, normalization

ABSTRACT:

While the integration of data has been performed on various levels for quite a time, the logical consequence - the integration of methods - has been rather neglected especially in the context of the analysis of remotely sensed data. On the other hand the present way of combining methods is partially responsible for unsatisfying results which have to be noticed for instance for object recognition processes. Hence, goal of the paper is to demonstrate the respective drawbacks of currently applied linear sequence approaches, to present general design concepts for alternative method networks, and to describe a corresponding implementation for the task of object recognition based on data of Digital Surface Models.

1. MOTIVATION

It is well known that progresses in the fields of data acquisition and processing are acting as catalysts for the development of integrated evaluation approaches between but also within disciplines (e.g., Ehlers, 1993). With respect to the remote sensing domain we can observe not only the development of single sensors showing better geometrical, spectral and radiometrical properties, but in particular the trend to *data integration* which is driven by multi-sensor systems that acquire not only spectral but also elevation data (e.g., by laserscanning) and orientation information (e.g, by GPS/IMU) in a sequential or simultaneous mode.

While the integration of data has been performed on various levels for quite a time, the logical consequence - the *integration of methods* - has been rather neglected especially in the context of the analysis of remotely sensed data. Conventionally this task is performed by means of a linear sequence of the single, special-purpose processes. Beside the facts that a couple of these processes are far away from maturity and intermediate errors are propagated from one step to the other, especially the way of method integration is responsible for unsatisfying results which have to be noticed in the field of object recognition.

General goal of this paper is to verify the mentioned drawbacks of such a linear approach on one hand (chapter 2), and to present the concept (chapter 3) as well as an implementation example (chapter 4) of an alternative *method network*.

For these purposes we will concentrate on an object recognition based on information from Digital Surface Models, which have become a very important source for this task due to their improved operational features (e.g., availability) and technical characteristics (i.e, horizontal resolutions and vertical accuracies). The specific goal will be to demonstrate that an intelligent integration of the involved key processing steps - blunder analysis, terrain surface estimation and object recognition - leads to more reliable classification results in a network configuration instead of using a sequential approach.

2. CURRENT METHOD INTEGRATION APPROACHES

2.1 Application description

The status as well as the drawbacks of current method integration approaches will be demonstrated with the concrete application of an object recognition based on elevation data. Conventionally, this evaluation is performed by means of a linear sequence of the following key steps (see also figure 1).

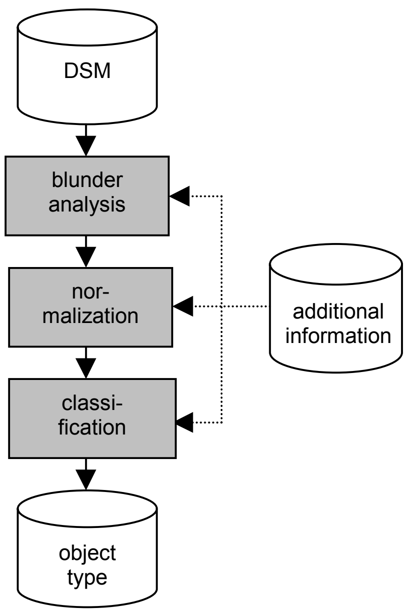

Figure 1. Linear sequence architecture for given application

Firstly, a *blunder analysis* takes places which eliminates extreme height values from the given Digital Surface Model (DSM) by user or statistically defined thresholds. Secondly, the derivation of object heights takes place by subtracting a given or an estimated Digital Terrain Model (DTM) from the DSM (*normalization*). In the case of the non-trivial estimation process, morphological filtering (e.g., see Vosselman, 2000), stochastical procedures (e.g., see Kraus, 1997) or region-based approaches (Schiewe, 2001) can be applied. Finally, these object heights (and eventually other parameters) are introduced into the *classification* step which is generally based upon probabilistic or fuzzy logic approaches. For a more detailed description of the algorithms which are used within our study we refer to the implementation example in section 4.2.

2.2 General problems

Applying such a typical evaluation process as outlined above some typical and commonly known problems occur. First of all the practical realization is done not only by one but by several software packages. These partially monolithic systems are heterogeneous with respect to their data structures and import and export functionalities so that a couple of data conversion processes have to take place (e.g., the elevation model is needed not only in the original point-wise ASCII-, but also in one or two raster image formats).

Furthermore, all methods including the transformations have to be invoked interactively by the user. Very often a batch processing is not possible due to the limited or not available batch functionality of one single component. A couple of information which are needed for the call of one function have to be repeated for another call.

The rather high efforts to invoke a component are one major reason for their single use within a linear sequence. The resulting drawbacks will be elaborated within the next section.

2.3 Problems related to sequential approach

In the following we will demonstrate that the quality of object recognition can be improved significantly if in contrast to the traditional linear sequence of processing steps (figure 1) a network configuration (figure 2) is used. The general idea is to backtrace hypotheses from later into previous processes. In the following some examples are presented.

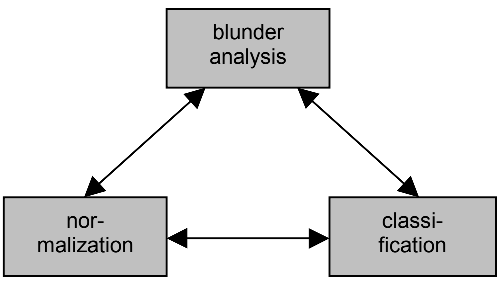

Figure 2. Network architecture for given application

It is obvious, that the **blunder analysis** can be improved by introducing information derived from the **classification** process: Based on a object type hypothesis one can predict its relative height behaviour and detect blunders by comparing this with actual data. For some regions one can assume constant height values (e.g., for waters), while for others height gradients will be very low in all directions (e.g., for airport runways, greenland) or at least in one direction (e.g., for roads).

After detecting blunders or data gaps their meaningful removal becomes necessary: Considering the object type associated to such points or areas one can optimize the method and window size for a reasonable interpolation of surrounding values. For instance, in figure 3 data gaps with the laserscanning data set occured due weak laser beam reflections. With the knowledge of the associated object being a building, the interpolation will take place only within the limits of the building in order to get a sharp transition to the surrounded terrain surface.

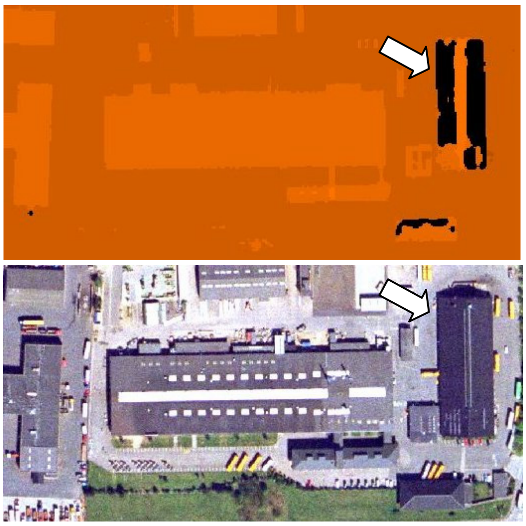

Figure 3. Extraction of interpolation information for data gaps in DSM (top) from semantical information (bottom)
- data courtesy of TopoSys GmbH -

But also the **normalization** process can be improved by introducing **classification** results. If for example morphological filter algorithms are applied for the detection and removal of regions within the DSM that do not belong to the terrain surface (in particular buildings and wooded areas), the critical filter window size can be derived from the actual object extent, or the filtering can be avoided at all if no such region was detected.

Some normalization algorithms separate the detection and the removal of objects, that stand clearly above the terrain surface, from their substitution (i.e., interpolation) which ends up with the so-called estimated Digital Terrain Model (eDTM). For some applications (e.g., hydrological modeling) it is necessary that only some of these regions under consideration will be interpolated (e.g., wooded areas) while others (e.g., buildings) have to be marked as blocking area because no water will actual flow here. Schiewe (2001) describes a respective region-based methodology for the separation of such draining and blocking areas.

Another important example for a meaningful DTM estimation is given in the case of removed buildings where the assumption of a horizontal plane instead of an interpolation within the surrounded, eventually inclined terrain represents a more suitable substitution. As figure 4 points out, the latter approach may lead to inconsistent and wrong object heights.

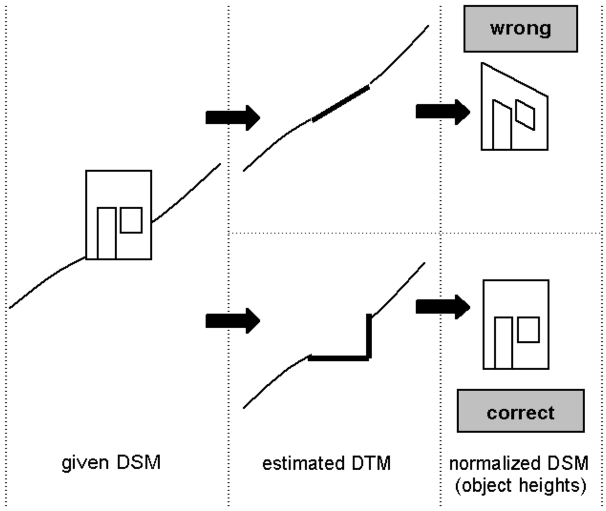

Figure 4. Choice of estimated DTM influences derivation of object heights

Finally, also the **blunder analysis** can be further improved by the results of the **normalization** process by introducing the extent of regions that have been reduced to the terrain surface and that are generally characterized by sharp rather than by ramp height edges.

In summary, the presented examples have pointed out that a significant improvement can be achieved by using a method network instead of a linear sequence architecture enabling the use of all hypotheses and information for all components.

3. CONCEPTUAL DESIGN OF METHOD NETWORKS

Central aim of the conceptual design of a method network is the optimized linkage of the involved components that allows for an objective control of the process with a minimum of user interaction. We will not concentrate on data modeling topics here but will focus on system architecture aspects by testing models coming from the software engineering domain on their applicability to the above mentioned problems. Based on general design criteria (section 3.1) we will present configuration solutions for current as well as for future systems (3.2 and 3.3, resp.), considering the experience that system integration is a evolutionary rather than a revolutionary process (e.g., Ehlers et al., 1989).

3.1 General design criteria

Central aspect of the design of a evaluation architecture is the definition of their connecting elements. With respect to their functionality we have to take into account (Abel et al., 1994)

- *transformation operations* for the exchange of data between the components,
- *constructor operations* for the (automatically or user-driven) generation of control commands, and
- *accessor operations* for the actual execution of these commands.

Designing these interface elements the general principles of continuity and safety have to be considered. Hence, as less information as possible should be exchanged (principle of *loose coupling*) and the number of interfaces should be kept to a minimum. With respect to the latter aspect a complete network between all n components (ending up with a number of interfaces of order n^2) would lead to a too costly and error-prone system.

3.2 Current configurations

For the design of current configurations we have to consider an integration of existing *closed components*. This assumption is based on various experiences that have shown that it is hardly possible to interfere with or to modify existing programs. It has to be noted that with this also an optimization of data modeling and handling will remain a difficult task.

In this context, we see a configuration using a *common interface module* as the best solution. The central component summarizes the user interface but in particular all connecting operations (transformation, constructor, accessor). The number of interfaces is reduced to a minimum (maximum of n interfaces for n linked components). Finally, the desired non-linear processing sequence can be controlled by this central component.

It should be noted that contrast to the field of Geographical Information Systems (GIS) where the general topic has already been discussed for a long time (e.g., within the Open GIS Consortium) and a couple of such architectures have been designed and implemented (e.g., see Abel et al., 1994; Waugh & Healey, 1986), for remote sensing evaluation systems no similar concepts have been presented so far.

3.3 Future configurations

General aim of future developments should be the possibility of an open usage of data and methods for a variety of users from distributed and heterogeneous platforms.
One realization could be the copy of software code (e.g., Java applets) from server to local machines (mirroring). Disadvantages of this approach are rather long downloading times and licensing problems. Alternatively, a standardized communication between software components placed on distributed platforms seems to be possible. The disadvantage of this approach is that the transfer of data to be processed could take too long. As an example for the latter structure the Object Management Group has presented the Common Object Request Broker Architecture (CORBA; OMG, 1998) for the GIS domain.

Finally, it has to be pointed out again that the proposed client-server-architectures for the integration of remote sensing software components are not yet to realize, because we still struggle with heterogeneous, not object-oriented data structures, too large software components and no standards that enable the connection to common interface modules.

4. IMPLEMENTATION EXAMPLE

The implementation of such a method network shown in figure 2 has been initially realized based on the concepts we introduced in sections 2.3 to 3.2. The desktop GIS ArcView® was chosen for implementation (section 4.1). For the single processing steps - blunder analysis, normalization and classification - case specific and not general purpose components have been applied (section 4.2) and connected to a desired network (section 4.3). Tests were performed on two different data sets (section 4.4). from which a couple of conclusions could be drawn (sections 4.5, 4.6).

4.1 Choice of ArcView®

Being aware of the variety of problems we decided to implement that network with only one software package - in our case the desktop GIS ArcView®. In contrast to the previous section's final conclusion a homogeneous and object-oriented data structure was supposed to fit best.

Originally, ArcView® is a vector based GIS which can properly model the object representation by continuos areas of elevation points either through its boundary or by characteristic relations (i.e. trends) between these points.

Additionally there are extensions available which process and store raster data, too. Therefore grids can be analyzed and those results can be stored either as raster or as vector data. Furthermore, ArcView® provides the capabilities to implement user-specific functions. This can be done by using Avenue™, an object-oriented script language. In summary, all necessary processing steps can be done within one software environment.

4.2 Description of components

Referring to figure 2 three closed components for the main tasks have to be taken into account. These methods are linked with each other by a common interface module which guarantees for the general design criteria posted in section 3.1 as well as for a free navigation between the components.

4.2.1 Blunder analysis: The blunder analysis has been reduced to gap detecting and gap filling procedures, because there is no guarantee that detected extrema that for instance have been returned by a bias analysis are real blunders and not real objects like flag poles or even wells instead.

Although bias analyses appear to be unsuitable, information about multiple biases and their dispersion can be taken into account for improving the classification of objects. Forest areas or tree groups may not be dense enough to cover all terrain points with its leaf area. Therefore, objects containing widely spread minima with small extents may be interpreted as vegetation.

It has been found that in contrast to figure 1 blunder analyses (i.e. gap filling) performed in a method network are best placed after normalization or classification.

4.2.2 Normalization: The task of normalization is to differentiate between the surfaces of the terrain and of outstanding objects. Considering the approaches mentioned in section 2.1 and based on the experiences that in particular morphological filtering might have negative effects on data quality (like loss of information) we prefer region-based approaches. The algorithm realized in this example is as follows: Depending on the vertical accuracy (s_Z) of the input DSM multiple selections have to be made. Each selection contains the points which heights (h) are greater than

$$h = z_{\min} + (i \cdot s_z) \qquad | \qquad i \in \left\{1, \ldots, \frac{(z_{\max} - z_{\min})}{s_z}\right\} \qquad (1)$$

Continuos areas of points in each layer i are enclosed by its boundary. All boundaries are stored as polygons with a z-coordinate of h in one single data set.

So, all polygons which can be found in two or more layers are declared as objects. The number of detected objects can be increased by accepting a minimum tolerance between two polygons. This might be obligatory while analyzing data sets of lower vertical accuracy or minor reliability.

4.2.3 Classification: The object can now be divided into three parts, i.e. its head, body and base (see figure 5):

1) all points inside the boundary belong to the object's surface and can be considered as its head shape;
2) the minimum height of all DSM points inside the boundary gives an idea of its body height;
3) the maximum height of points outside next to the polygon can be declared as terrain height and as object base.

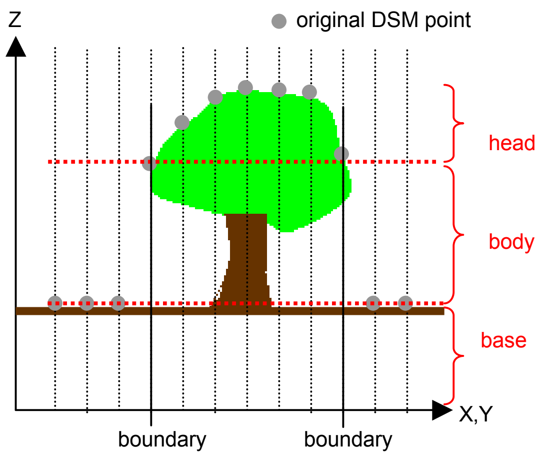

Figure 5. Object represented by DSM heights and its boundary

Furthermore the object can be described by analyzing the corresponding polygon within the detected boundaries considering the following parameters:

1) area, perimeter and volume;
2) compactness (2D, 3D);
3) rectangularity or parallelism of the boundary (after dividing it into line segments);
4) texture (e.g. standard deviation, variance) of the head points.

4.3 Fusion of methods

4.3.1 Concepts: As discussed in section 3.2 a common interface module has been established in order to control the application, to evaluate the current progress and, if necessary, also to stop it. It is also possible that the central module can switch back to a prior state if the classification has become worse.

Neglecting some pre-processing modules generating point data sets to work with, the network might be entered at any step of the process, i.e., the normalization, classification or blunder analysis module.

The three methods can be performed one by one. But it is also possible to repeat or recall a single method or to jump back to a previous method. Additionally, the main methods are further split into more simple sub-modules which can be invoked separately. For example there are several sub-modules established in order to perform the process of normalization (e.g., height filtering, boundary generation, polygon comparison etc.).

Multiple or recurring runs are given not only by the network structure, but also by applying different or by changing parameters like the vertical accuracy (equation 1) or shape constraints (e.g., for determining buildings). Therefore the duration of the object recognition process can vary strongly.

Obviously it is not possible to evaluate all permutations of these methods. Hence, we have tested only one which will be described in the following sections.

4.3.2 Status quo of implementation: Presently the implemented modules perform the operations of data conversion, height selection, boundary generation, polygon comparison, gap detection and filling, bias analyses, texture analyses, calculation and analyses of shape parameters.

Significant modules not realized yet are concerned with the detection and analysis of linear or planar trends. Furthermore the full potential of the control module is not implemented yet so that a couple of its duties are still performed by a human operator.

We prefer the normalization as starting point. Assuming that significant objects like buildings and trees show larger height values we start with an elevation interval of 1 m as s_Z (equation 1).

After the differentiation between objects and terrain an analysis of the object's boundary takes place. Area, perimeter and volume are calculated and compared to predefined values. Furthermore the boundary is simplified and the single line segments are compared with each other in order to search for rectangular or parallel sections. This boundary gives a first representation of the object. Nevertheless a blunder analysis is necessary in order to detect neighbouring gaps which can be adjacent to an object or belong to the object, respectively. Filling and joining the gap's and the object's area might lead to a better classification (compare figure 3).

Finally, a boundary-based classification completes the object recognition process. If the results are less satisfactory, the process is repeated taking a lower elevation interval, higher tolerances or both of them into account.

4.4 First results

Tests with the implemented method network have been performed with DSMs from two sensors: Two sites have been obtained with the TopoSys laser scanner (www.toposys.com) which produces first and last pulse elevation data with a height accuracy of about ± 0.2 m delivered as point data in ASCII. The other two DSMs have been derived by multiple matching from stereo imagery of the High Resolution Stereo Camera – Airborne (HRSC-A, http://solarsystem.dlr.de/FE/) given with a horizontal resolution of 0.5 m and a proposed vertical accuracy of ± 0.2 m.

Defining buildings as test objects the number of detected buildings was compared to the actual existing number (table 1).

Sensor		**TopoSys**				**HRSC-A**			
# Test site		**1**		**2**		**1**		**2**	
# Buildings		38		28		27		33	
Elev.	**Tol.**	**Detected objects**							
		#	**%**	**#**	**%**	**#**	**%**	**#**	**%**
1 m	0 m²	1	3	2	7	0	0	0	0
	2 m²	16	42	14	50	1	4	2	6
0,5 m	0 m²	8	21	5	18	0	0	1	3
	2 m²	29	76	25	89	4	15	11	33
0,25 m	0 m²	15	39	18	64	1	4	5	15
	2 m²	34	89	28	100	25	93	25	76

Table 1. Results of normalization depending on different elevation intervals ("Elev.") and tolerances ("Tol.")

It can be concluded that the minor the elevation interval and the higher the applied tolerance is, the larger the number of detected objects becomes. But the higher the elevation interval and the higher the tolerance is, the less reliable the corresponding results will be.

Exemplary the detected objects with an extent of more than 10 m² were classified. Differing only between buildings and vegetation, and assuming that buildings show certain parameters (area = 75 m², compactness = 0.4, standard deviation of elevation = ± 1.7 m) the tests already led to satisfying results (table 2). Buildings classified as trees show higher elevation standard deviations (i.e. up to ± 2.1 m) compared to the predefined parameters.

Objects	Classified as		Correctly classified
	Buildings	Trees	
Buildings	15	3	83 %
Trees	-	46	100 %

Table 2. Exemplary object classification

4.5 Gain of the network approach

Although the results of the methods of normalization (table 1) and classification (table 2) are not satisfactory yet the main advantage of the network approach already becomes obvious: Due to the recursive architecture valuable information can be exploited (in terms of data mining) for all modules while this is not possible using a linear sequence of methods.

As an example, classification parameters can be adapted with respect to the shape parameters of not correctly classified buildings. Referring to table 2 a classification adapting a modified standard deviation of ± 2.1 m led to further improved results (table 3).

Objects	Classified as		Correctly classified
	Buildings	Trees	
Buildings	18	-	100 %
Trees	2	44	95 %

Table 3. Exemplary object classification with adapted parameters

Furthermore the number or percentage of detected objects during the normalization process can determine acceptable tolerances and / or elevation intervals, respectively.

Hence, multiple runs can be evaluated by adapting parameters resulting from previous classifications and by comparing the new results with previous ones. If the classification is getting worse within a single run the currently applied parameters are to be neglected for future runs.

4.6 Problems and limitations

In fact the above described implementation example appears just as another linear sequence of methods. Due to the yet incomplete implementation the gain of the network approach could only be outlined in this chapter.

Beside the incomplete implementation also the single modules have to be further developed. For example, the algorithm failed to detect objects at elevation intervals of 1 m (table 1). This can be explained partially by the lower quality of the HRSC-A data sets derived by stereo matching (Bohmann, 2001). As a consequence, the normalization should be performed with intervals related to the vertical accuracy. However, this leads to longer computation times and requires a lot more disk space to store the data sets and its derivatives.

5. SUMMARY

The unsatisfying quality of object recognition procedures is partially due to the fact that no intelligent integration of the involved processing components is applied. Using various examples it has been shown, that in contrast to a linear sequence of methods a network architecture is able to improve the results of all inherent modules.

To realize this general idea we have presented general design concepts adopted from software engineering. While for current realizations a common interface module seems to be the best approach, for future developments an open usage from distributed platforms based on software code mirroring or on exchanging data and commands between distributed components should be taken into account.

We have presented an implementation example that aims for an object recognition based on information from Digital Surface Models. It is based on a common interface module which has been implemented under the ArcView® software environment. First experiences have proven the general applicability and gain of the network solution (in particular, the advantage of the recursive nature) but also the costs in terms of time and disk space. Further developments within the single modules as well as the networking elements have to be made in order to come a satisfying and operational solution.

REFERENCES

Abel, D.J., Kilby, P.J. & Davis, J.R., 1994. The systems integration problem. *International Journal of Geographical Information Systems*. 8(1): pp. 1-12.

Bohmann, G., 2001. Entwicklung eines Methoden-Netzwerks zur Integration Digitaler Oberflächenmodellen (DOM) in den Prozeß der Objektextraktion. Diploma thesis at the University of Vechta, Institute for Environmental Sciences.

Ehlers, M., Edwards, G. & Bedard, Y., 1989. Integration of remote sensing with geographic information systems: a necessary evolution. *Photogrammetric Engineering and Remote Sensing*. 55(11), pp.1619-1627.

Ehlers, M., 1993. Integration of GIS, remote sensing, photogrammetry and cartography: the geoinformatics approach. *GIS - Geo-Informationssysteme*. 6(5), pp.18-23.

Kraus, K., 1997. Eine neue Methode zur Interpolation und Filterung von Daten mit schiefer Fehlerverteilung. *Österr. Zeitschrift für Vermessung und Geoinformation*, (1), pp. 25-30.

OMG, 1998.: CORBA 2.0/IIOP Specification. Object Management Group formal document 98-07-01.

Schiewe, J., 2001. Ein regionen-basiertes Verfahren zur Extraktion der Geländeoberfläche aus Digitalen Oberflächen-Modellen. Photogrammetrie - Fernerkundung - Geoinformation, 2, pp. 81-90.

Vosselman, G., 2000: Slope based filtering of laser altimetry data. In: *International Archives of Photogrammetry and Remote Sensing,* Amsterdam, NL, 33 (B3), pp. 935-942.

Waugh, T.C. & Healey, R.G., 1986. The Geolink system, interfacing large systems. In: *Proceedings of AutoCarto London*, pp. 76-85.