Hydrol. Earth Syst. Sci., 14, 1153–1165, 2010
www.hydrol-earth-syst-sci.net/14/1153/2010/
doi:10.5194/hess-14-1153-2010


Hydrology and Earth System Sciences

# On the uncertainty of stream networks derived from elevation data: the error propagation approach

**T. Hengl**[1], **G. B. M. Heuvelink**[2], **and E. E. van Loon**[1]

[1]Institute for Biodiversity and Ecosystem Dynamics, University of Amsterdam, Nieuwe Achtergracht 166, 1018 WV Amsterdam, The Netherlands
[2]Wageningen University and Research, P.O. Box 47, 6700 AA Wageningen, The Netherlands

Received: 22 December 2009 – Published in Hydrol. Earth Syst. Sci. Discuss.: 29 January 2010
Revised: 18 June 2010 – Accepted: 18 June 2010 – Published: 2 July 2010

**Abstract.** DEM error propagation methodology is extended to the derivation of vector-based objects (stream networks) using geostatistical simulations. First, point sampled elevations are used to fit a variogram model. Next 100 DEM realizations are generated using conditional sequential Gaussian simulation; the stream network map is extracted for each of these realizations, and the collection of stream networks is analyzed to quantify the error propagation. At each grid cell, the probability of the occurrence of a stream and the propagated error are estimated. The method is illustrated using two small data sets: Baranja hill (30 m grid cell size; 16 512 pixels; 6367 sampled elevations), and Zlatibor (30 m grid cell size; 15 000 pixels; 2051 sampled elevations). All computations are run in the open source software for statistical computing R: package geoR is used to fit variogram; package gstat is used to run sequential Gaussian simulation; streams are extracted using the open source GIS SAGA via the RSAGA library. The resulting stream error map (Information entropy of a Bernoulli trial) clearly depicts areas where the extracted stream network is least precise – usually areas of low local relief and slightly convex (0–10 difference from the mean value). In both cases, significant parts of the study area (17.3% for Baranja Hill; 6.2% for Zlatibor) show high error ($H > 0.5$) of locating streams. By correlating the propagated uncertainty of the derived stream network with various land surface parameters sampling of height measurements can be optimized so that delineated streams satisfy the required accuracy level. Such error propagation tool should become a standard functionality in any modern GIS. Remaining issue to be tackled is the computational burden of geostatistical simulations: this framework is at the moment limited to small data sets with several hundreds of points. Scripts and data sets used in this article are available on-line via the www.geomorphometry.org website and can be easily adopted/adjusted to any similar case study.

*Correspondence to:* T. Hengl
(t.hengl@uva.nl)

## 1 Introduction

In geomorphometry, Digital Elevation Models (DEM) are routinely used to extract various continuous (gridded) land surface parameters, and/or discrete (vector) land surface objects. Assuming that DEMs are perfectly accurate, extraction of land surface parameters and objects is a simple one iteration operation (Fig. 1a). However, in reality, DEMs are not perfect representations of reality – DEMs suffer from systematic and random errors and DEM elevations differ from what we measure on the field. In fact, errors are inevitable, even if elevation models are produced using highly accurate and dense sampling techniques such as LiDAR (Evans and Hudak, 2007; Bater and Coops, 2009). Errors are inherent both in measurements of elevations, and in the DEM analysis algorithms, and can possibly have a significant influence on the reliability of final products. By ignoring errors in the input layers, analysts often get disappointed when their products are evaluated versus ground truth data. This is true especially for hydrological applications (Wise, 2000; Wechsler, 2007).

The approach to GIS analysis that takes into account that GIS input layers are of limited accuracy, and that provides

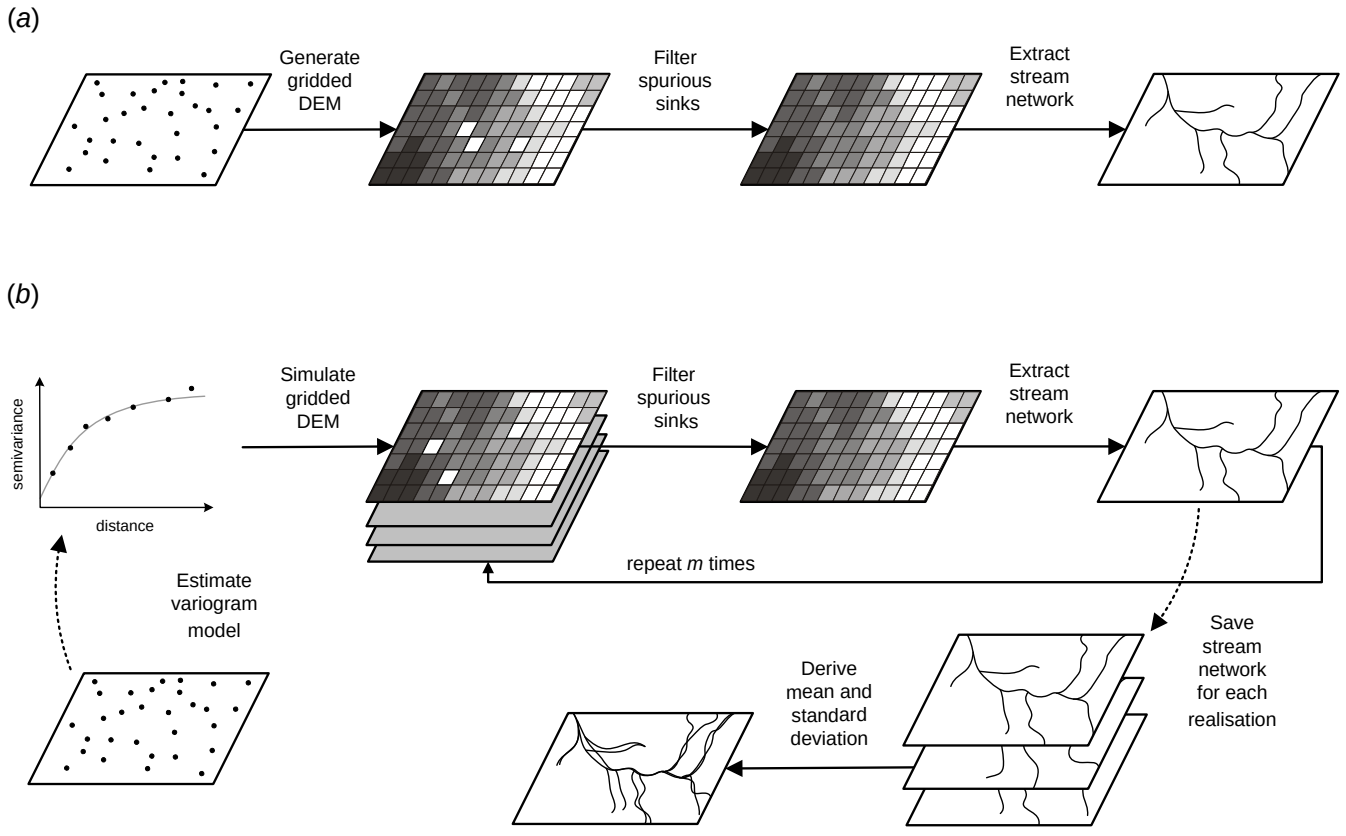


**Fig. 1.** Workflow scheme for stream extraction from elevation data: **(a)** assuming that elevations carry no uncertainty; **(b)** the Monte Carlo error propagation approach with $m$ realizations. In this case, filtering of spurious sinks is specific to the case studies and not a general operation.

a way to assess the propagated uncertainty associated with the output of the analysis, is known as "error propagation" (Heuvelink, 1998). The potential of using error propagation has been first recognized by Burrough (1986) and Englund (1993). At that time, it seemed unlikely that stochastic simulations would become routinely available in a GIS environment. Since then, the world has evolved: computers are more powerful, statistical tools are more accessible and more sophisticated. We are slowly reaching a point when error propagation will become a standard toolbox of any GIS software (Wechsler, 2007). Examples of using error propagation methods to assess the accuracy of various scalar-type land surface parameters derived from DEMs can be found in the work of Fisher (1992); Heuvelink (1998); Dutta and Herath (2001); Raaflaub and Collins (2006) and Oksanen and Sarjakoski (2005). Brown and Heuvelink (2007) recently produced a generic library for uncertainty modeling called "Data Uncertainty Engine" (DUE). A group at Aston University has been developing the Uncertainty Markup Language (UncertML, http://www.uncertml.org) that could become a standard for writing metadata for error propagation applications. However, there are still technical and conceptual issues that need to be solved before an uncertainty engine becomes a standard part of any GIS (Heuvelink, 2002; Temme et al., 2008). One such open issue is that the problem of assessing accuracy of vector-type features (watershed boundaries, stream networks, break lines, escarpments and similar) has been under-represented in the literature, and theory to support applications is in general missing (with few exceptions e.g. Poggio and Soille, 2008). Most of papers that suggest ways to model uncertainty of vector-based objects in a GIS do not specify how to actually compute these using real data.

This article proposes a methodology to assess errors of stream networks extracted from digital elevation models. It uses two small case studies to demonstrate how to implement geostatistical simulations and assess the propagated uncertainty and map the error of locating streams. Our secondary objective is to promote the geostatistical tools implemented in the open source environment for computing (R), and geographical analysis tools implemented in the open source GIS (SAGA). Scripts and data sets used in this article are available on-line via the www.geomorphometry.org website. Users and developers are encouraged to adopt, extend and improve.

## 2 Methods and materials

### 2.1 Error propagation

GIS error propagation can be defined as a set of statistical procedures that model uncertainties in the input maps, and for a given GIS operation, estimate the (propagated) error of mapping a feature of interest. In mathematical terms, the output map is a result of an operation applied to multiple spatial layers (Heuvelink, 1998):

$$U(\boldsymbol{s}) = g\left\{A_1(\boldsymbol{s}), \ldots, A_p(\boldsymbol{s})\right\} \quad (1)$$

where $A_1(\boldsymbol{s}), \ldots, A_p(\boldsymbol{s})$ are the GIS inputs (spatial layers), $U(\boldsymbol{s})$ is the output map, $p$ is the number of inputs, $\boldsymbol{s}$ is the vector of coordinates (spatial location $x, y$), and $g$ is the GIS operation. The main focus of error propagation is determination of the mean value ($\bar{U}(\boldsymbol{s})$) and its standard deviation ($\sigma^2_U(\boldsymbol{s})$), or ideally the entire probability distribution of the output map $U$ for any location $\boldsymbol{s}$ in the area of interest $\mathbb{A}$. Note that the probability distribution of the output map is quite involved because it must also capture the spatial statistical dependencies. In case of GIS output that is a spatial object such as a streamline, the probability distribution is even more complex. Possibly the easiest way to characterize uncertainty of discrete spatial objects is by generating a number of those objects (especially for objects that cannot easily be specified): for example, river network is the output from numerical algorithm that operates on the terrain data; although the flow modeling formulas are deterministic, the consequent uncertainty can not be specified separately from the terrain on which it was generated. In fact, Tarboton and Baker (2009) argue that it is close to impossible to integrate uncertainty in the flow-algebra.

The benefit of running an error propagation analysis is, first and foremost, that it quantifies the uncertainty in the GIS result. If the probability distribution of the input $A$ is narrow, then we might expect that the propagated uncertainty will be narrow as well, but this need not always be the case. The sensitivity of model output to small changes in the input is also important. Also, when there are multiple uncertain inputs it becomes difficult to predict the impact of error in input maps on derived products. The situation is even more difficult if errors in inputs are spatially variable – in some parts of the study area they can be high, in others low – so that it becomes difficult to predict where in the study area the uncertainty of the derived map becomes critical. By ignoring the fact that errors in input maps exist and that they are significant, we create a wrong idea about the precision of the derived land surface objects. Hence the primary benefit of running error propagation is visual and statistical assessment of errors in the output maps.

In principle, there are two main approaches to error propagation: (a) the analytical, and (b) the Monte Carlo approach (Heuvelink, 2002). In the first case, the propagated error is derived using some mathematical technique such as via a Taylor series expansion; in the second case, stochastic simulation is used to sample $m$ times from the input probability distribution and the operation is repeated $m$ times. The Monte Carlo approach is more suited for cases where the GIS operation $g$ is so complex that it is practically impossible to mathematically derive the propagated distribution model. Since this is the case for many GIS applications, the Monte Carlo approach has become the dominant approach to error propagation (Wechsler, 2007; Poggio and Soille, 2008).

In the case of Monte Carlo simulation, the mean value ($\bar{U}(\boldsymbol{s})$) and the standard deviation ($\sigma^2_U(\boldsymbol{s})$) of the output feature is simply:

$$\bar{U}(\boldsymbol{s}) = \frac{\sum_{j=1}^{m} U_j^{\mathrm{SIM}}(\boldsymbol{s})}{m} \quad (2)$$

$$\sigma_U(\boldsymbol{s}) = \sqrt{\frac{\sum_{j=1}^{m}\left(U_j^{\mathrm{SIM}}(\boldsymbol{s}) - \bar{U}(\boldsymbol{s})\right)^2}{m-1}} \quad (3)$$

In the case of stream network extraction from DEMs, the error propagation model (Eq. 1) is:

$$U^{\mathrm{SIM}} = g\left\{z^{\mathrm{SIM}}, b_1, \ldots, b_p\right\} \quad (4)$$

where $z^{\mathrm{SIM}}$ is the simulated elevation map, $U^{\mathrm{SIM}}(\boldsymbol{s})$ is the output value of stream (either 1 or 0, depending on whether the location is part of the stream or not), and $b_1, \ldots, b_p$ are the user-defined, constant, hydrological model parameters, for example: minimum segment length, initiation grid, initiation threshold etc. These parameters can be uncertain too. Although this looks like a trivial model, the function $g$ involves a spatial analysis with respect to flow direction on the input elevation map, so that small differences in elevation at some locations can result in completely different stream patterns while large differences at other locations can have no effect.

Streams have several specific properties that distinguish them for other land surface parameters and objects. Streams are discrete objects – a stream is composed of a set of interconnected points (represented as grid cells). These objects have attributes such as length and curviness, Horton or Strahler ordering. A grid cell can be part of a stream (value 1) or not (value 0) i.e. it becomes a Bernoulli variable with probability $p$ being part of the stream. The majority of cells will have a small value for $p$ simply because streams are by definition rare events. The mean of the Bernoulli variable at some location is simply $p$; its variance is given by $p \cdot (1-p)$. The uncertainty of detecting streams can be alternatively characterized by the Shannon entropy (Shannon and Weaver, 1949):

$$H(\boldsymbol{s}) = -p(\boldsymbol{s}) \cdot \log(p(\boldsymbol{s})) - [1 - p(\boldsymbol{s})] \cdot \log(1 - p(\boldsymbol{s})) \quad (5)$$

where $p$ is the probability of a grid cell being part of the stream estimated by the number of times the model puts a

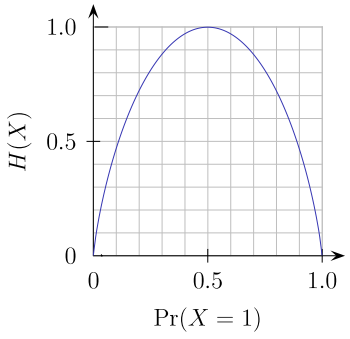


**Fig. 2.** The uncertainty of deriving stream can be best described using the information entropy ($H$) of a Bernoulli trial. This plot is courtesy of Brona Brejova, Comenius University in Bratislava, Slovakia.

stream at the cell, divided by the total number of Monte Carlo realizations. The precision of estimating the propagated uncertainty is inversely related to the Monte Carlo sample size. This means that if we run 100 simulations, and then at some location detect stream 99/100 times ($p$=0.99), the estimated error will be 0.056, and we can not map uncertainty with a finer precision. If the model detects streams with equal probability of stream and not-stream ($p$=0.5), this will produce the highest error of 1 (Fig. 2).

### 2.2 Geostatistical simulations

Monte Carlo analysis of spatial error propagation requires the generation of realistic simulations of elevation values. The most common technique in geostatistics used to generate equiprobable realizations of a spatial feature is the Sequential Gaussian Simulation (Goovaerts, 1997). To simplify matters, it is assumed that elevation can be modeled as a stationary random function (Goovaerts, 1997; Kyriakidis et al., 1999) with a constant mean:

$$\mu = E\{Z(\boldsymbol{s})\} \tag{6}$$

and a variogram model that only depends on distance between points:

$$\begin{aligned} 2\cdot\gamma(\boldsymbol{h}) &= \mathrm{Var}\{Z(\boldsymbol{s}) - Z(\boldsymbol{s}+\boldsymbol{h})\} \\ &= E\left\{[Z(\boldsymbol{s}) - Z(\boldsymbol{s}+\boldsymbol{h})]^2\right\} \end{aligned} \tag{7}$$

where $\boldsymbol{h}$ is the separation vector between two locations, and $\gamma(\boldsymbol{h})$ is the semivariance. A capital letter $Z$ is used because we assume that the model is probabilistic, i.e. there is a range of equiprobable realizations of the same model. If the variable of interest (elevation) has been sampled at a set of point locations ($z(\boldsymbol{s}_1)$, $z(\boldsymbol{s}_2)$,..., $z(\boldsymbol{s}_n)$, where $\boldsymbol{s}_i$=($x_i$, $y_i$)), then these can be used to fit a variogram model. Once we have estimated the variogram model parameters, we can use this model to produce simulations of $Z$ that have the same spatial structure:

$$z^{\mathrm{SIM}}(\boldsymbol{s}_0) = E\{Z(\boldsymbol{s}_i)|z(\boldsymbol{s}_i), i = 1,\ldots,n\} \tag{8}$$

where $z^{\mathrm{SIM}}$ is the simulated value at location $\boldsymbol{s}_0$. In this case, simulations will be conditioned on the observations at sampling locations $z(\boldsymbol{s}_i)$. Under the assumption of second-order stationarity, we can use for example a global exponential variogram with three parameters to produce a simulated DEM. A slightly more sophisticated variogram is the Matérn variogram model, which has an additional parameter to describe the smoothness (Stein, 1999; Minasny and McBratney, 2005):

$$\gamma(\boldsymbol{h}) = C_0\cdot\delta(\boldsymbol{h}) + C_1\cdot\left[\frac{1}{2^{\upsilon-1}\cdot\Gamma(\upsilon)}\cdot\left(\frac{\boldsymbol{h}}{R}\right)^{\upsilon}\cdot K_{\upsilon}\cdot\left(\frac{\boldsymbol{h}}{R}\right)\right] \tag{9}$$

where $C_0$, is the nugget parameter, $C_1$ the sill parameter, $R$ the range parameter, $\delta(\boldsymbol{h})$ is the Kronecker delta, $K_{\upsilon}$ is the modified Bessel function, $\Gamma$ is the gamma function and $\upsilon$ is the smoothness parameter. The Matérn variogram model is especially suited for elevation data because the smoothness, common for topographic features, can be nicely represented with the $\upsilon$-parameter. Note, however, that using the Matérn variogram is only sensible when the nugget variance is insignificant i.e. close to zero.

When additional auxiliary maps are available that can be used to explain the deterministic component in the spatial distribution of elevation values, more accurate simulations of topography can be produced using the regression-kriging model (Hengl et al., 2008). For the purpose of this article, we will follow a simple case and assume: (a) that the elevation values are realizations of a second-order stationary random function with a constant trend; and (b) that the spatial autocorrelation can be modeled using a Matérn variogram.

In summary, the error propagation approach to extraction of streams from elevation data can be summarized in five steps (Fig. 1b):

1. calculate an experimental variogram from the data and fit a Matérn variogram model (with parameters: $C_0$, $C_1$, $R$ and $\upsilon$) to represent the variability of the input DEM;
2. generate multiple realizations of the DEM using conditional simulation and the variogram model fitted previously (Eq. 8);
3. filter spurious sinks; derive stream network for each realization, and save the temporary result (Eq. 4);
4. aggregate the derived maps to estimate stream occurrence frequency and error of mapping streams (Eq. 5);
5. evaluate how the propagated error relates to various topographic parameters; then consider improving quality of input DEM or filtering elevations where necessary.

A disadvantage of the Monte Carlo approach is that it requires a significantly large number of realizations to produce a reliable estimate of the distribution function. The number of realizations $m$ must be sufficiently large to obtain stable results, but exactly how large $m$ should be depends on how accurate the results of the uncertainty analysis should be. Theoretically speaking, the accuracy of the Monte-Carlo method is proportional to the square root of the number of runs $m$ (Temme et al., 2008). Therefore, to double the accuracy one must quadruple the number of runs. This means that although many runs may be needed to reach stable and accurate results, any degree of precision can be reached by taking a large enough sample $m$. As a rule of thumb, we can take 100 simulations as being large enough, and everything below 20 as insufficient (Heuvelink, 1998). Consequently, the Monte-Carlo method is computationally demanding, particularly when the GIS operation takes much computing time (Heuvelink, 2002).

### 2.3 Software tools

In this article we use a combination of statistical and geographical computing software to assess propagated error of detecting streams: SAGA GIS for geographical computing, and R for statistical computing; all operations are in fact combined in the same script. In this case, R is used to control both internal add-on packages, but also external GIS SAGA (R "on top") via a special link library RSAGA. A detailed description of R+SAGA integration can be can be found in Brenning (2008).

Because most of the packages used in this article are not common to majority of GIS users and hydrologists (especially to users of ESRI-products), we consider worth introducing SAGA, gstat and geoR, and reviewing its main functionality. A small guide on how to install, set and make first steps in the two packages, is also given in the Appendix A. This should help you reproduce the analysis shown in this article with your own data. Even more detailed instructions on how to combine R and SAGA using the same data sets can be found in Hengl (2009).

#### 2.3.1 SAGA GIS

SAGA[1] (System for Automated Geoscientific Analyses) is an open source GIS that has been developed since 2001 at the University of Göttingen (the group recently collectively moved to the Institut für Geographie, University of Hamburg), Germany, with the aim to simplify the implementation of new algorithms for spatial data analysis (Conrad, 2006, 2007). A point data set of measured elevations can be used in SAGA to generate a Digital Elevation Model (DEM), that can then be used to extract a stream network (see scheme in Fig. 1a). For example, you can open the point layer in SAGA GIS, then use the module *Grid* $\mapsto$ *Gridding* $\mapsto$ *Spline interpolation* $\mapsto$ *Thin Plate Splines (local)* and generate a smooth DEM. Then, you can preprocess the DEM to remove spurious sinks using the method of Planchon and Darboux (2001). Select *Terrain Analysis* $\mapsto$ *Preprocessing* $\mapsto$ *Fill sinks*, and then set the minimum slope parameter to 0.1.

Once you have prepared a DEM, you can derive stream networks using the *Channel Network* function which is available in SAGA under *Terrain Analysis* $\mapsto$ *Channels*. This implements the original algorithm described in Conrad (2007) and which is based on the FD8 multiple flow direction algorithm by Quinn et al. (1995). As a result, you should get a map shown in Fig. 3. Assuming that the DEM and the stream extraction model are absolutely accurate, i.e. that they perfectly fit the reality, this would then be the end product of the analysis (which corresponds to the scheme in Fig. 1a).

#### 2.3.2 R and packages gstat and geoR

R is the command-based environment for statistical computing (R Development Core Team, 2009). Many spatial packages have been contributed in the past 3–4 years, which allow R to be also used for spatial analysis. Two important add-on packages that are used in this article are gstat (Pebesma, 2004) and geoR (Diggle and Ribeiro Jr., 2007). In principle, a large part of functionality of gstat and geoR overlap. On the other hand, geoR has many original methods, including an original format for spatial data (called `geodata`). geoR is especially powerful to fit variograms (including interactive visual fitting), and for dealing with non-normal data; gstat is somewhat more fit to run predictions and generate simulations, even with large data sets. gstat also uses spatial classes in R, so that conversion to GIS formats is fairly easy.

Once we have simulated $m$ DEMs using gstat, we can derive stream networks using the "*Channel Network*" function, which is available also via the command line – via the `ta_channels` SAGA library (see further Appendix A). This means that, through scripting in R, one can automate both geographical processing and statistical analysis, and implement the computational scheme shown in Fig. 1b to any similar data set.

### 2.4 Study areas and data sets

We use two previously published examples to demonstrate the method: the "Baranja hill" case study is of mixed low and high relief, and the "Zlatibor" case study is an area of high relief. In principle, the only input for both exercises is a point map showing field-measured elevations (ESRI Shapefile). These maps are used to generate multiple realizations of Digital Elevation Model, and then extract drainage network, as implemented in the SAGA GIS package. Vector maps showing the actual location of streams are also available for both study areas.

[1] http://saga-gis.org

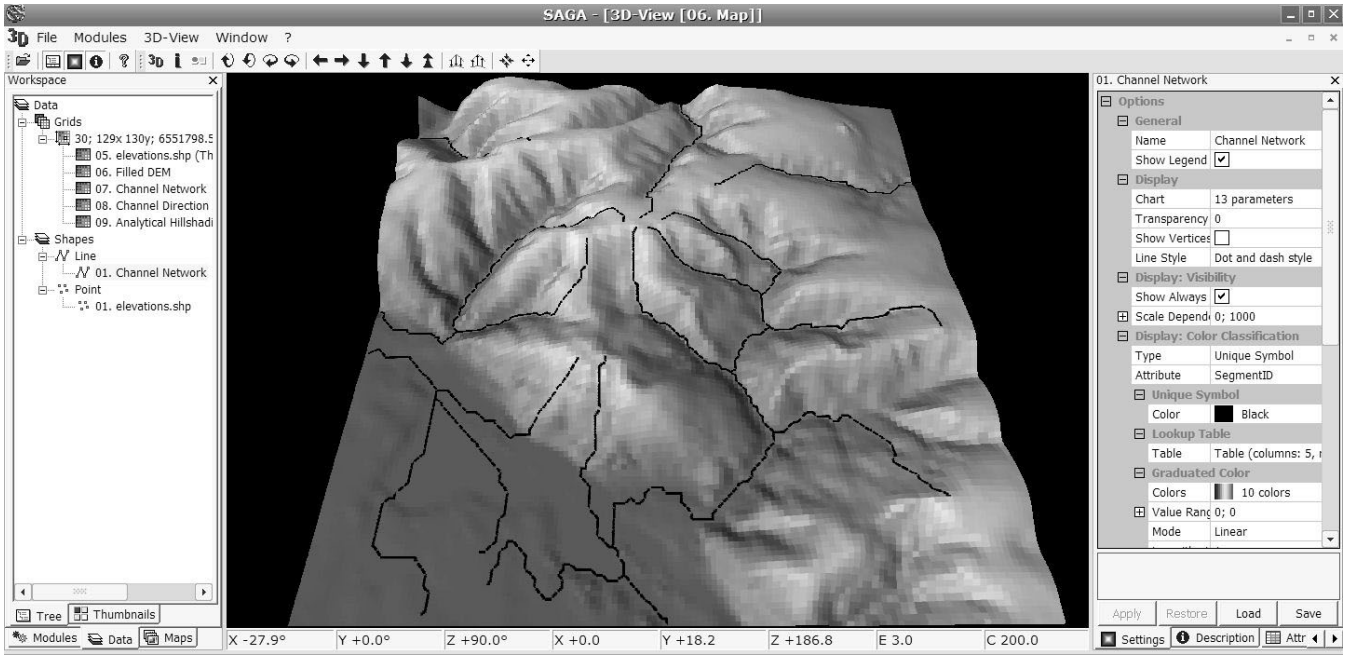


**Fig. 3.** Stream network generated in SAGA GIS using standard settings. In this case we used 40 (pixels) as the minimum length of streams. Case study Baranja Hill; viewed from the West side.

The study area "Baranja hill" is located in eastern Croatia (centered at 45°48′16.4412″ N, and 18°39′54.198″ E); it has been extensively mapped over years and several GIS layers are available at various scales (Hengl and Reuter, 2008). The study area corresponds approximately to the size of a single 1:20 000 aerial photo. Its main geomorphic features include hill summits and shoulders, eroded slopes of small valleys, valley bottoms, a large abandoned river channel, and river terraces (Fig. 3). Elevation of the area ranges from 80 to 240 m with an average of 157.6 m and a standard deviation of 44.3 m. The data set consists of 6367 points of field measured heights. The complete data set is available for download from the geomorphometry dataset repository[2]. A similar error propagation exercise using the same case study can be followed in Temme et al. (2008).

The second case study, "Zlatibor", is located in the South-western part of Serbia (centered at 43°43′44.6″ N and 19°42′37.8″ E). The area is mainly hilly plateau, with the exception of the north-eastern part where the slopes are much steeper (see further Fig. 6b). Elevations range from 850 m to a maximum of 1174 m; the total size of the area is 13.5 square kilometers. The data set consists of 2051 height measurements. An additional set of 1020 very precise spot heights used for error assessment is also available. This data set is described in detail in Hengl et al. (2008) and can be also obtained from the geomorphometry dataset repository[3].

[2]http://geomorphometry.org/content/baranja-hill

[3]http://geomorphometry.org/content/zlatibor

## 3 Results

The first result of analysis are the variogram models fitted in geoR (Fig. 4). These show that the target variable ($z$) in general varies equally in all directions in both study areas. This is especially distinct for shorter distances (<500 m), which allows us to model the variograms using isotropic models. For Baranja Hill study area geoR fits a Matérn variogram model with nugget parameter $C_0$=0, sill parameter $C_1$=1831, and range parameter $R$=1051 m (practical range is 3.1 km); for Zlatibor case study, the elevation values are more variable – nugget parameter is still $C_0$=0, the sill parameter is $C_1$=2173, range parameter is $R$=761 m. In both cases $z$ seems to be a relatively smooth variable – there is no nugget variation and spatial autocorrelation is effective (practical range) up to distance of 2–3 km.

Both are in fact typical variograms for elevation data i.e. representation of a land surface. Note also that, in both cases, the target variable shows close to normal distribution so no transformation was necessary. As expected, the confidence bands (envelopes) are much narrower at smaller distances (Fig. 4). The relatively wide confidence bands at larger distances indicate that it might be worthwhile to consider using local (moving window) geostatistical analysis and adjust the variogram parameters locally.

In the next step, we look at the dispersion of the stream lines derived for all simulated DEMs. Once the processing is finished, we can visualize all derived streams at top of each

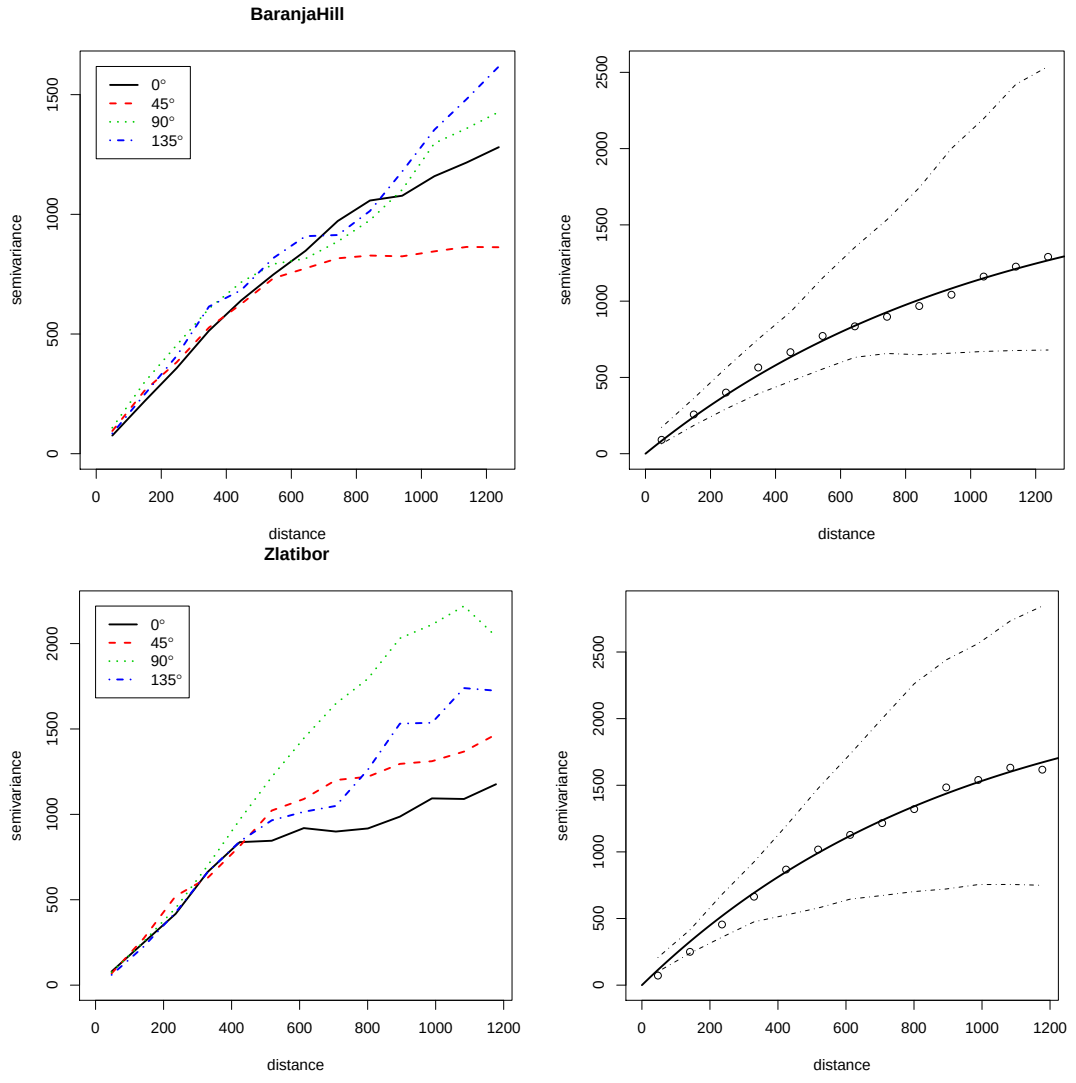


**Fig. 4.** Variograms fitted for Baranja hill (above) and Zlatibor case studies (below); left: anisotropy in four directions; right: isotropic Matérn variogram model fitted using the weighted least squares (WLS) and its confidence bands.

other. The 100 realizations of stream network maps for the two study areas are shown in Fig. 5. The visualization of density of streams clearly illustrates the concept of propagated uncertainty. If you zoom in into this map, you will notice several things. First, in some areas streams are isolated and hence seem to be very improbable; in other areas stream are densely distributed but over a wider area. Note also that the derived streams follow the gridded-structure of the DEMs, which explains some artificial breaks in the lines. Some artifacts in these maps are probably a consequence of the fact that we have used arbitrary input parameters for the minimum length of streams (40) and initial grid. These parameters could have been find-tuned by experts familiar with the study areas, but this is not relevant for this exercise.

In both cases, significant parts of the study area – 17.3% for Baranja Hill; 6.2% for Zlatibor – show relatively high error ($H > 0.5$) of locating streams (Fig. 6). Although high absolute values of error can be observed in both areas of high and low relief, the cumulative propagated variability of detecting a stream is much higher in the terrace region of the study area Baranja Hill (Fig. 7). The errors are, in fact, a bigger problem than we have anticipated. In addition, the course of many streams is dramatically different from where the streamlines are thought to be located on the basis of DEM-streamline analysis. In the case of the Baranja Hill study area, this is because many channels are manmade and hence do not have to follow the topography (Fig. 6a). In the Zlatibor study area only one or two small patches of terrain seem to be problematic: both are at the beginning of the

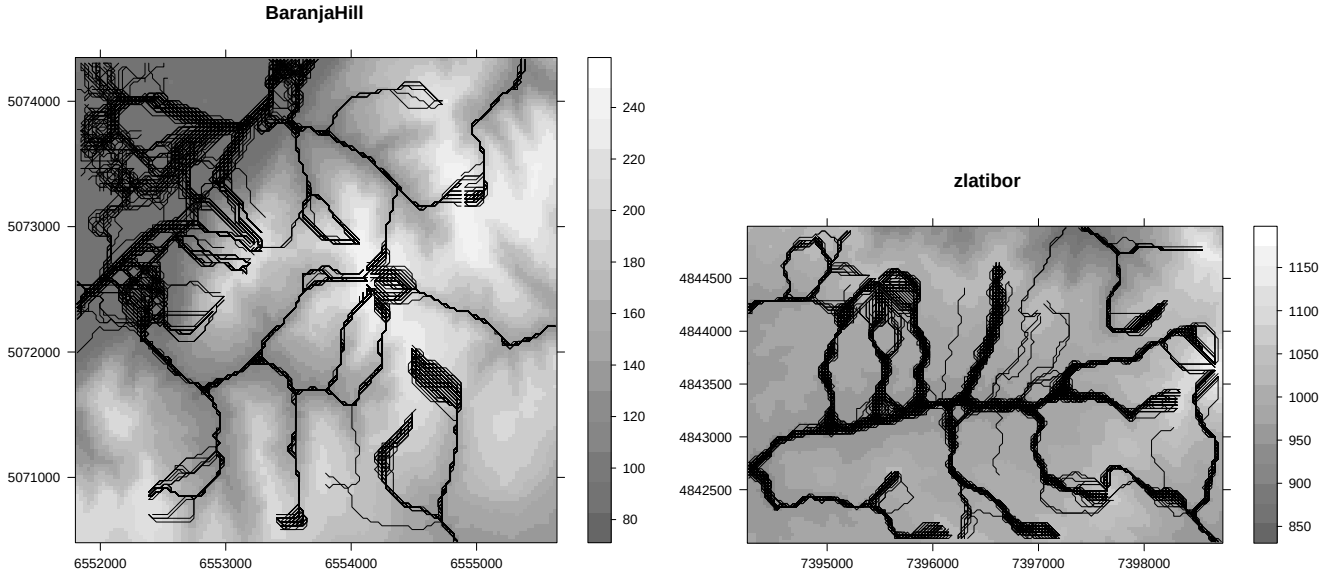


**Fig. 5.** 100 realizations of stream network overlaid on top of each other: left: Baranja hill case study; right: Zlatibor case study. The greyscale legends indicates elevations in meters.

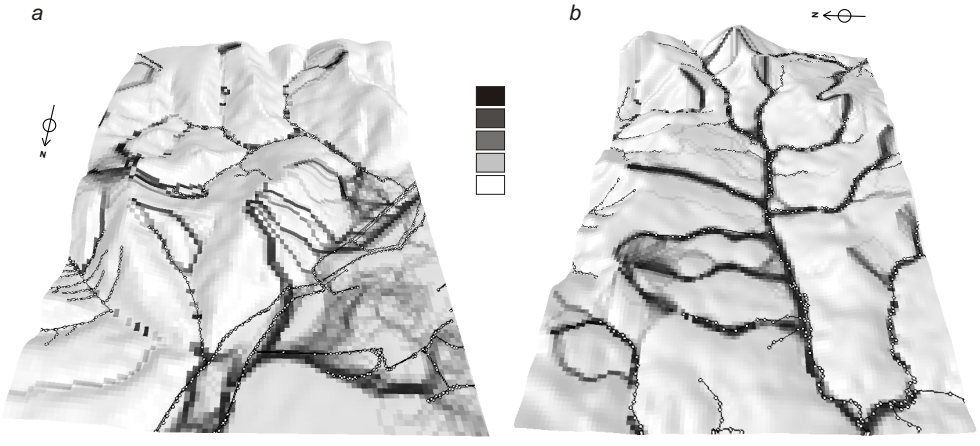


**Fig. 6.** Propagated error of mapping streams estimated using Eq. (5); visualized in SAGA GIS: **(a)** Baranja hill case study; **(b)** Zlatibor case study. The lines indicate the *true* streams – digitized from topo maps.

stream. The results from these two small case studies clearly demonstrates the usefulness of the error propagation analysis – by mapping the propagated error we can delineate the most problematic areas and focus our further efforts.

Now that we have estimated the propagated uncertainty of extracting channel networks (streams) from DEMs, we can try to understand how this uncertainty relates to the geomorphology of terrain. It is interesting to derive a map of channel-slope and/or topographic wetness index, as it largely controls the hydrological properties, and the difference from the mean value in $5\times5$ search radius, as it describes local variability of shapes.

The results from the two case studies show that some 30–35% of the variability in the error maps can be explained with the difference from the mean value in the $5\times5$ window (Fig. 7). By knowing this, we could now allocate resources and collect more accurate, more densely sampled elevations in the areas that have similar geomorphological properties (in this case: slightly convex shapes). In fact, one could further optimize elevation sampling and improve the accuracy of extracted streams to reach the required threshold. The alternative is to down-grade the effective scale of the streams derived using this point data. For example, it is obvious from Figs. 5 (below) and 6b that the model has not much problems

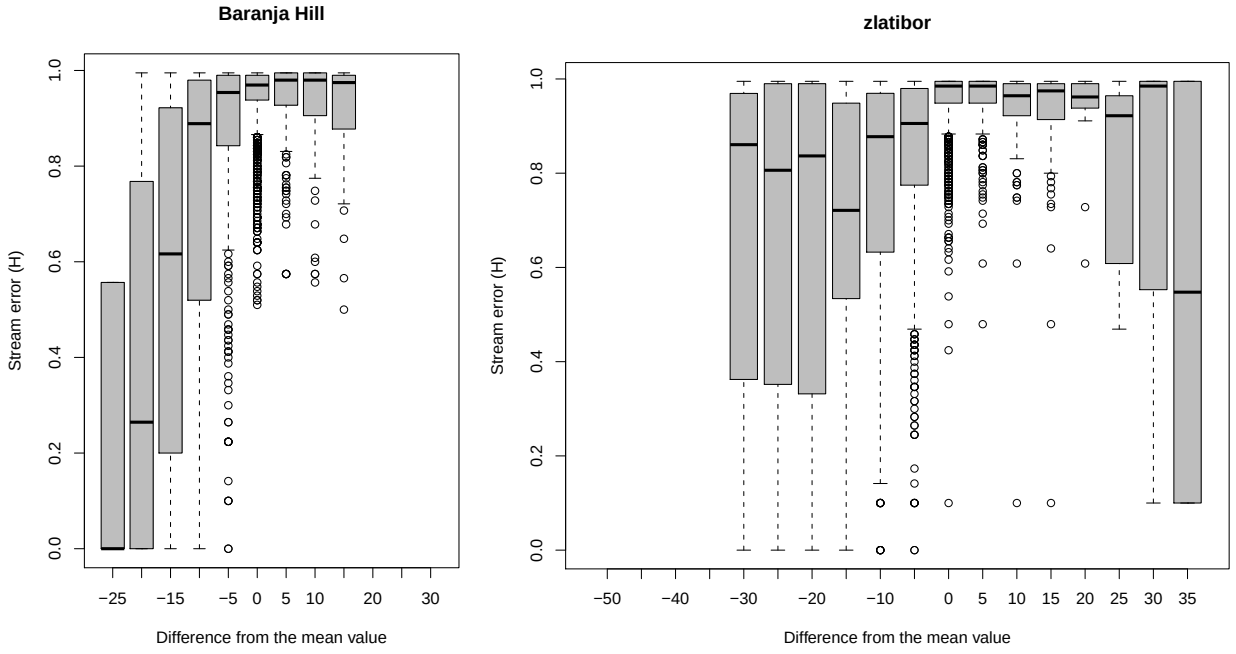


**Fig. 7.** Bar plots showing relationship between the relative relief (difference from the mean value in the 5×5 search radius) and cumulative errors. In both cases the highest errors of mapping streams are in slightly convex areas (positive values in range 0–10).

of locating streams in the area of relatively high relief (Zlatibor), however, the spatial accuracy of derived streams does not get better than ±50 m, so that it is reasonable to consider degrading the scale of the output map e.g. from 1:15 000 to 1:50 000 scale.

The computational burden of this method is also an issue. The most costly operations are geostatistical simulations and extraction of stream networks. Geostatistical simulations, even with a search radius of only 30 closest points, takes 5–10 min to generate 100 simulations for these small study areas (150×100 pixels). This means that this framework is at the moment limited to small data sets with few hundreds of points; it would be probably of limited use for large LiDAR point data sets.

## 4 Discussion and conclusions

The two case studies demonstrate that it is worth investing in error propagation – in both cases we are able to detect some difficult areas where extracted stream networks will be critically imprecise. Figures 5 and 6 show two interesting things: (1) the dispersion of stream networks is in some areas significant; (2) streams are especially difficult to map in low-relief areas where the difference from the mean value is positive – meaning areas with convex shapes. This largely reflects our expectation, but it is rewarding to be able to prove these assumptions using hard data. Our results correspond with the results of Poggio and Soille (2008) who discovered that uncertainty of stream segments is in general significant and especially high for Strahler order one or two. Some remaining issues and ideas for further research are discussed below.

In the two studies, we have have ignored many aspects of data analysis and used model parameters that now deserve justification. For example, for geostatistical simulations, we have set the kappa parameter for the Máthern variogram relatively high at 1.2 (see Diggle and Ribeiro Jr., 2007, for more examples). Following the knowledge about the feature of interest, we assumed that a land surface is inherently smooth – due to the erosional processes and permanent leveling of topography. Hence, we wanted to generate realizations of DEMs that fit our knowledge of the area. Why is the high kappa parameter necessary? If we run DEM simulations with e.g. an exponential model, we noticed that realizations will be much noisier than what we would expect (Hengl et al., 2008). This will happen even if we set the nugget parameters at zero (smooth feature). There are several explanations for this. Having a non-zero grid resolution implies that the correlation between adjacent grid cells is not equal to 1, so that grids may still appear to have noise (Temme et al., 2008). A noisy DEMs leads to completely different drainage networks – the streams will be shorter and more random – which we know does not fit knowledge about the area. The Matérn variogram model (Eq. 9), on the other hand, allow us to produce smoother DEMs, while still using objectively estimated nugget, sill and range parameters. This makes it especially suitable for modeling of land surface.

We have also limited the number of simulations to 100. Perhaps this number should be larger, particularly because we are dealing with a feature that commonly has a small $p$. It should be feasible to evaluate the increase in accuracy with an increasing number $m$, e.g. by evaluating the change in derived probability or attribute property such as estimated stream length or catchment width. If such a parameter or function does not change anymore below a certain threshold, no more simulations seem to be required. An elegant alternative can be to calculate the information content of each additional realization. With an increasing sample size, the change in the ultimate probability field becomes less and less. This is certainly an idea worth further research.

We have also set the grid cell size at 30 m without any real justification. The next step would be to consider some statistically sound approach to select a grid cell size based on the accuracy of the derived stream network. This follows the idea of Hutchinson (1996), who use an iterative DEM cell-size optimization algorithm as implemented in the ANUDEM package. By plotting the error of mapping streams versus the grid spacing index, one can select the grid cell size that shows the maximum information content in the final map. The optimal grid cell size is the one where further refinement does not change the accuracy of derived streams. It would be interesting to see if the optimal detection of the grid cell size for hydrological objects can be operationalized, so that the users only need to provide the point data.

Another question that needs to be addressed is how much of the analysis should be automated? Can and should error propagation be automated so that it becomes a default operation of any DEM analysis? If yes, users will not even have to see the steps behind error propagation (black-box approach), but simply select a land surface object/parameter of interest and the software will decide about the reasonable number of simulations, suitable grid cell size, depict the areas that are critical etc. The case studies shown in this paper are fairly small in size, hence it was not expensive to run 100 simulations. How to deal with the computational complexity of error propagation? These case studies obviously demonstrated that such analysis provides richer picture of the spatial variability of propagated errors, but is this always needed? What if error propagation is useful only for small parts of the study area; is there then still a need to run such analysis globally? How would geostatistical simulation + error propagation techniques perform with LiDAR surveys that consists of millions of points? Are results of error propagation very dependent on the type of data (field survey, LiDAR, SRTM DEM etc.) or will the spatial patterns of uncertainty be different?

The two case studies shown in this article consists of precisely measured elevations over a small and homogenous area with relatively constant variogram parameters. How to generate simulated DEMs when a spatial auto-correlation structure model (variogram) is not available or differs locally? Traditionally, geostatistical techniques are developed to work with point-sampled values. For DEMs generated directly from a scanning device (e.g. SRTM DEM) it is a serious problem to get a reliable estimate of a variogram. In addition, uncertainty of measured elevations is heavily dependent on the type of land use (local spatial auto-correlation structure), hence simulated DEMs should reflect this property also. A solution to generate simulations of e.g. SRTM DEM is the co-kriging framework. Separate estimation of the variogram and cross-variogram parameters for the error surface and the main signal in the DEM is rather inexpensive, but simulations using co-kriging are even more computationally intensive.

There is also an issue of how to represent the outputs of error propagation. Should the land surface object derived using error propagation represented as fuzzy objects? Should we abandon concept of absolutely discrete land surface objects at first place? If yes, which data structure should be used to save and exchange such objects? Or is the spaghetti representation shown in Fig. 5 more informative? Tøssebro and Nygård (2008) provide a probabilistic framework for computing uncertainties for simple geographic objects such as points and unstructured lines, but how could these be combined with geostatistical simulations?

Floor for discussion is open and everybody is welcome to contribute. For the beginning, software developers can try implementing error propagation frameworks as standard toolboxes to extract information from elevation data. The users can further consider testing this framework in areas of variable relief, surface roughness and with elevation measurements from various sources. We anticipate that the mean challenge of the proposed framework will be processing of the LiDAR data that is typically very large and requires localization of analysis. With the further advances of technology (computing power) and geostatistics (local variograms), both operations should become feasible.

## Appendix A

### Installation and first steps with R+SAGA

The following text provides instructions how to obtain and install SAGA and R and implement the analysis described in this article with your own data. R+SAGA can be run on Windows™ and Linux operating systems. Mac OS™ version of SAGA is still under development.

Start with installing R and its spatial packages. Visit the R project homepage[4] and obtain the recent installation from CRAN. After you finish installing R, open the new session and install the contributed packages: select the *Packages* $\mapsto$ *Install package(s)* from the main menu. Note that, if you wish to install a package on the fly, you will need to select a suitable CRAN mirror from where it will download and unpack a package. Another quick way to get all packages used

[4] http://r-project.org

in R to do spatial analysis[5] (as explained in Bivand et al., 2008) is to install the ctv package and then execute the command:

```
> install.packages("ctv")
> library(ctv)
> install.views("Spatial")
```

This will allow most of the *spatial* packages available for R, including maptools, rgdal, gstat, geoR, and RSAGA.

Next, if you are a Windows™ user, obtain the SAGA binaries from a Source Forge repository. SAGA GIS is a full-fledged GIS with support for raster and vector data. It includes a large set of geoscientific algorithms (over 300 modules), being especially powerful for the analysis of DEMs. With the release of version 2.0 in 2005, SAGA works under both Windows and Linux operating systems. In addition, SAGA is an open-source package, which makes it especially attractive to users that would like extend or improve its existing functionality. To install SAGA simply unzip the binaries to your program files directory. Then open SAGA GUI and test its functionality using point-and-click operations. Now you can consider switching to the scripting environment. Go to your R session and load the RSAGA library:

```
> library(RSAGA)
```

First check if R is able to locate SAGA on your machine:

```
> rsaga.env()

$workspace
[1] "."

$cmd
[1] "saga_cmd.exe"

$path
[1] "C:/Progra~1/saga_vc"

$modules
[1] "C:/Progra~1/saga_vc/modules"
```

which means that you can now send operations from R to SAGA. Open the `modules` folder under the SAGA directory and you will notice a large number of DLL libraries. To get an info what can a certain module do, type:

```
> rsaga.get.modules("ta_channels")

$ta_channels
 code             name interactive
1  0       Channel Network    FALSE
2  1       Watershed Basins    FALSE
3  2 Watershed Basins (extended)    FALSE
4  3   Vertical Distance to CN    FALSE
5  4 Overland Flow Distance to CN    FALSE
6  5       D8 Flow Analysis    FALSE
7  6        Strahler Order    FALSE
8  NA             <NA>    FALSE
9  NA             <NA>    FALSE
```

[5] http://cran.r-project.org/web/views/Spatial.html

Next, we need to get the list of parameters needed to extract channel network from a DEM map:

```
> rsaga.get.usage("ta_channels", 0)

SAGA CMD 2.0.4
library path:  C:/Progra~1/saga_vc/modules
library name:  ta_channels
module name :  Channel Network
Usage: 0 -ELEVATION <str> [-SINKROUTE <str>]
-CHNLNTWRK <str> -CHNLROUTE <str>
-SHAPES <str> -INIT_GRID <str>
[-INIT_METHOD <num>] [-INIT_VALUE <str>]
[-DIV_GRID <str>] [-DIV_CELLS <num>]
[-TRACE_WEIGHT <str>] [-MINLEN <num>]
 -ELEVATION:<str>   Elevation
    Grid (input)
 -SINKROUTE:<str>   Flow Direction
    Grid (optional input)
 -CHNLNTWRK:<str>   Channel Network
    Grid (output)
 -CHNLROUTE:<str>   Channel Direction
    Grid (output)
 -SHAPES:<str>     Channel Network
    Shapes (output)
 -INIT_GRID:<str>   Initiation Grid
    Grid (input)
 -INIT_METHOD:<num>  Initiation Type
    Choice
    Available Choices:
    [0] Less than
    [1] Equals
    [2] Greater than
 -INIT_VALUE:<str>   Initiation Threshold
    Floating point
 -DIV_GRID:<str>    Divergence
    Grid (optional input)
 -DIV_CELLS:<num>   Tracing: Max. Divergence
    Integer
    Minimum: 1.000000
 -TRACE_WEIGHT:<str>  Tracing: Weight
    Grid (optional input)
 -MINLEN:<num>     Min. Segment Length
```

Finally, you can generate a stream network shown in Fig. 3 using the `rsaga.geoprocessor`:

```
> rsaga.geoprocessor(lib="ta_channels",
+  module=0, param=list(ELEVATION="DEM.sgrd",
+  CHNLNTWRK="tmp.sgrd",
+  CHNLROUTE="tmp.sgrd",
+  SHAPES="streams.shp",
+  INIT_GRID="DEM.sgrd",
+  DIV_CELLS=3, MINLEN=40))

SAGA CMD 2.0.4
library path:  C:/Progra~1/saga_vc/modules
library name:  ta_channels
module name :  Channel Network
author   :  (c) 2001 by O.Conrad
```

```
Load grid: DEM.sgrd...
ready

Load grid: DEM.sgrd...
ready

Parameters

Grid system: 30; 128x 129y;
    6551817x 5070464y
Elevation: DEM.sgrd
Flow Direction: [not set]
Channel Network: Channel Network
Channel Direction: Channel Direction
Channel Network: Channel Network
Initiation Grid: DEM.sgrd
Initiation Type: Greater than
Initiation Threshold: 0.000000
Divergence: [not set]
Tracing: Max. Divergence: 3
Tracing: Weight: [not set]
Min. Segment Length: 40

Channel Network: Pass 1
Channel Network: Pass 2
Channel Network: Pass 3
Create index: DEM.sgrd
ready
Channel Network: Pass 4
Channel Network: Pass 5
Channel Network: Pass 6
ready
ready

Save grid: tmp.sgrd...
ready

Save grid: tmp.sgrd...
ready

Save shapes: streams.shp...
ready

Save table: streams.dbf...
ready
```

More detail on how to produce results shown can be found in the R script, available via www.geomorphometry.org.

*Acknowledgements.* This article evolved from a two-day workshop entitled "Automated analysis of elevation data in R" that was held at the University of Zürich on 29 and 30 August 2009. The principal author of this article would like to thank the Geomorphometry conference organizers Ross Purves and Stephan Gruber (Department of Geography) for hosting this workshop, and Carlos Grohmman for kindly helping us run this workshop. The authors would also like to thank all the workshop participants for their comments and suggestions. Join the open source initiative by sending your ideas/suggestion via the R-sig-geo mailing list and/or via the geomorphometry-organized meetings.

Edited by: P. Molnar

## References

Bater, C. and Coops, N.: Evaluating error associated with lidar-derived DEM interpolation, Comput. Geosci., 35, 289–300, 2009.

Bivand, R., Pebesma, E., and Rubio, V.: Applied Spatial Data Analysis with R, Use R Series, Springer, Heidelberg, 14–15, 2008.

Brenning, A.: Statistical Geocomputing combining R and SAGA: The Example of Landslide susceptibility Analysis with generalized additive Models, in: SAGA – Seconds Out, edited by: Böhner, J., Blaschke, T., and Montanarella, L., vol. 19, Hamburger Beiträge zur Physischen Geographie und Landschaftsökologie, 23–32, 2008.

Brown, J. and Heuvelink, G.: The Data Uncertainty Engine (DUE): a software tool for assessing and simulating uncertain environmental variables, Comput. Geosci., 33, 172–190, 2007.

Burrough, P.: Principles of Geographical Information Systems for Land Resources Assessment, Oxford University Press, Oxford, 241–264, 1986.

Conrad, O.: SAGA – Program Structure and Current State of Implementation, in: SAGA – Analysis and Modelling Applications, edited by: Böhner, J., McCloy, K. R., and Strobl, J., vol. 115, Verlag Erich Goltze GmbH, 39–52, 2006.

Conrad, O.: SAGA – Entwurf, Funktionsumfang und Anwendung eines Systems für Automatisierte Geowissenschaftliche Analysen, PhD thesis, University of Göttingen, Göttingen, 2007.

Diggle, P. J. and Ribeiro Jr., P. J.: Model-based Geostatistics, Springer Series in Statistics, Springer, 51–53, 2007.

Dutta, D. and Herath, S.: Effect of DEM Accuracy in Flood Inundation Simulation using Distributed Hydrological Models, Seisan Kenkyu, 53, 602–605, 2001.

Englund, E.: Spatial Simulation: Environmental Applications, in: Environmental modeling with GIS, edited by: Goodchild, M., Parks, B., and Steyaert, L., chap. 43, Oxford University Press, New York, 432–446, 1993.

Evans, J. S. and Hudak, A. T.: A multiscale curvature filter for identifying ground returns from discrete return lidar in forested environments, IEEE T. Geosci. Remote., 45, 1029–1038, 2007.

Fisher, P.: First experiments in viewshed uncertainty: Simulating fuzzy viewsheds, Photogramm. Eng. Rem. S., 58, 345–352, 1992.

Goovaerts, P.: Geostatistics for Natural Resources Evaluation (Applied Geostatistics), Oxford University Press, New York, 380–392, 1997.

Hengl, T.: A Practical Guide to Geostatistical Mapping, University of Amsterdam, Amsterdam, 221–239, 2009.

Hengl, T. and Reuter, H.: Geomorphometry: Concepts, Software, Applications, vol. 33 of Developments in Soil Science, Elsevier, Amsterdam, 26–29, 2008.

Hengl, T., Bajat, B., Reuter, H., and Blagojević, D.: Geostatistical modelling of topography using auxiliary maps, Comput. Geosci., 34, 1886–1899, 2008.

Heuvelink, G.: Analysing uncertainty propagation in GIS: why is it not that simple?, in: Uncertainty in Remote Sensing and GIS, edited by: Foody, G. and Atkinson, P., Wiley, Chichester, 155–165, 2002.

Heuvelink, G. B. M.: Error Propagation in Environmental Modelling with GIS, Taylor & Francis, London, UK, 35–46, 1998.

Hutchinson, M. F.: A locally adaptive approach to the interpolation of digital elevation models, in: Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modeling, National Center for Geographic Information and Analysis, Santa Barbara, CA, p. 6, 1996.

Kyriakidis, P. C., Shortridge, A. M., and Goodchild, M. F.: Geostatistics for Conflation and Accuracy Assessment of Digital Elevation Models, Int. J. Geogr. Inf. Sci., 13, 677–708, 1999.

Minasny, B. and McBratney, A. B.: The Matérn function as a general model for soil variograms, Geoderma, 128, 192–207, 2005.

Oksanen, J. and Sarjakoski, T.: Error propagation of DEM-based surface derivatives, Comput. Geosci., 31, 1015–1027, 2005.

Pebesma, E. J.: Multivariable geostatistics in S: the gstat package, Comput. Geosci., 30, 683–691, 2004.

Planchon, O. and Darboux, F.: A fast, simple and versatile algorithm to fill the depressions of digital elevation models, Catena, 46, 159–176, 2001.

Poggio, L. and Soille, P.: Quality assessment of hydro-geomorphological features derived from Digital Terrain Models, EUR 23489 EN, European Commission, DG Joint Research Centre, 2008.

Quinn, P. F., Beven, K. J., and Lamb, R.: The ln(a/tan b) index: how to calcute it and how to use it within in the TOPMODEL framework, Hydrol. Processes, 9, 161–182, 1995.

R Development Core Team: R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, 2009.

Raaflaub, L. D. and Collins, M. J.: The effect of error in gridded digital elevation models on the estimation of topographic parameters, Environ. Modell. Softw., 21, 710–732, 2006.

Shannon, C. and Weaver, W.: The Mathematical Theory of Communication, University of Illinois Press, Urbana, 50–52, 1949.

Stein, M. L.: Interpolation of Spatial Data: Some Theory for Kriging, Series in Statistics, Springer, New York, 218–220, 1999.

Tarboton, D. G. and Baker, M. E.: Towards an Algebra for Terrain-Based Flow Analysis, in: Modelling, and Visualizing the Natural Environment, edited by: Mount, N., Harvey, G., Aplin, P., and Priestnall, G., chap. 12, CRC Press, Boca Raton, 167–194, 2009.

Temme, A., Heuvelink, G., Schoorl, J., and Claessens, L.: Geostatistical simulation and error propagation in geomorphometry, in: Geomorphometry: concepts, software, applications, edited by: Hengl, T. and Reuter, H. I., Developments in Soil Science, Elsevier, 121–140, 2008.

Tøssebro, E. and Nygård, M.: Computing the Probabilities of Operations in Vector Models for Uncertain Spatial Data, in: IEEE International Conference on Signal Image Technology and Internet Based Systems, IEEE Computer Society, 78–85, 2008.

Wechsler, S. P.: Uncertainties associated with digital elevation models for hydrologic applications: a review, Hydrol. Earth Syst. Sci., 11, 1481–1500, doi:10.5194/hess-11-1481-2007, 2007.

Wise, S.: Assessing the quality for hydrological applications of digital elevation models derived from contours, Hydrol. Processes, 14, 1909–1929, 2000.