Master Thesis - GEO 511

Analyzing the Impact of High Resolution DEM Uncertainty on Hydrological Models Using a Parallel Computing Approach

Christian Venzin

Matrikelnr: 07-714-694

Email: christian.venzin@uzh.ch

January 31, 2013

Faculty Representative: Prof. Dr. Ross Purves
Advisers: Prof. Dr. Ross Purves and Prof. Dr. Jan Seibert

University of Zurich[UZH]

GIScience: Geocomputation
Department of Geography
University of Zurich

Acknowledgments

I would like to thank my advisers of this thesis, Prof. Dr. Ross Purves and Prof. Dr. Jan Seibert for their valuable guidance and advice. In addition I would like to thank the PhD students of the H2K unit, Benjamin Fischer and Michael Rinderer for providing me with hydrological data and additional information about the study area Alpthal. Special thanks go to Benjamin who showed me the Zwackentobel and provided me with recent hydrological research findings of the study area. I would also like to thank Curdin Derungs for his fast and corresponding helpful answers to GIS related questions. I would also like to convey thanks to Felix Morsdorf and Philip Joerg from the RSL unit for the interesting and useful discussions about LiDAR data. Also, I would like to thank the proofreaders Rino Darms, Martin Reich and Fabian Schneider. Finally, an honorable mention goes to my family and friends for their understanding and support. Without their assistance this study would not have been successful.

Abstract

In geographic information science, it is common to use gridded digital elevation models (DEMs) to derive various topographic attributes, hydrological parameters and compound indices. Usually, DEMs are assumed to be accurate representations of the real-world terrain surfaces. But, as the name "model" indicates, a DEM is only an approximation of the real terrain surface. The quality of DEMs suffers from systematic and random errors, caused by the tasks commonly involved in digital terrain modeling. When DEMs are used in digital terrain analysis, the errors of the model influence the analysis results in a negative manner. The investigation of this phenomena is known as error propagation analysis. The results of this analysis have a direct influence on any decision-making process based on applications and interpretations of digital terrain analysis.

Since the mid 1990's, light detection and ranging (LiDAR), also known as laser scanning, has allowed to collect highly accurate elevation data in a rapid and cost effective way. This thesis focuses on such micro-scale DEMs derived from LiDAR data, but even with highly accurate measurement techniques such as LiDAR, the resulting DEMs still contain uncertainty. The LiDAR derived DEM used in this thesis has a spatial resolution of 2 m and an average error (RMSE) in horizontal and vertical direction of 0.5 m.

This thesis provides an uncertainty analysis approach for investigating error propagation in high resolution DEM based terrain analysis. All algorithms are implemented with C++ using a multiprocessing approach based on MPI allowing an efficient error propagation analysis of high resolution DEMs. The implemented algorithms cover common hydrological models. The proposed approach includes four different flow routing algorithms (D8, D∞, MD8 and MD∞) to compute the upslope area and the compound indices TWI, SPI and STI, and supports watershed and stream network extraction. Appropriate methods were implemented to investigate the statistical and spatial characteristics of uncertainty. Uncertainty is visualized by error maps showing the relative or absolute standard deviations, probabilities or the error H. In addition, various results can be written on textfiles allowing further analysis in statistical software suits. Sample points in the DEMs extent which could then be analyzed more detailed in the uncertainty analysis can be defined . This can be used for example to investigate the convergence of different hydrological models in an error propagation analysis.

The uncertainty model is based on the Monte Carlo simulation approach and utilizes the process convolution method to generate the error surfaces. Additional knowledge about the spatial structure of error in LiDAR derived DEMs has made it possible to improve the uncertainty model accordingly.

Besides the programming task and the proposed frame-work for investigating error propagation, this thesis also provides various uncertainty assessments and an analysis of the spatial pattern of uncertainty for common hydrological models. Furthermore, the impact of uncertainty model parameters and coarser resolution DEMs on the uncertainty analysis results are investigated. In addition, con-

vergence tests for different hydrological models are run. Finally, this thesis provides an evaluation of the sensitivity of the four different flow routing algorithms and evaluates the performance of the provided uncertainty analysis approach.

The results show that the applied parallelization approach can speed up an error propagation analysis considerably depending on the number of processes used. The execution time could be reduced in most cases by up to a factor of three by using four processes instead of one. The uncertainty assessment shows that multiple flow direction algorithms are less sensitive than the single flow direction algorithm D8. Errors observed are up to three times higher when using the D8 instead of the MD∞ flow routing algorithm. This conclusion is also valid for the convergence tests. The D8 based TWI appears to converge substantially later (400 to 200 simulation runs). In general, all flow routing algorithms show very unstable analysis results below 100 simulation runs. In contrast, the extraction of watersheds appears to be more stable. In our case, the area of the watershed already converges after 150 simulation runs. Regarding the spatial pattern of uncertainty, the results show that high errors can be observed in all cases mostly in very flat and rough terrain. The results also show that the spatial resolution of the DEM has a minor impact on the analysis results. These depend largely on the uncertainty model parameters. For example, as shown in this thesis, a high spatial autocorrelation range applied to the error surface can reduce the observed error substantially and vice versa. This shows that the process convolution method is reaching its limits. Better methods must be implemented to produce more accurate error surfaces.

The methodology and its application presented in this thesis to assess uncertainty in hydrological models demonstrate that it is also usable and worth investigating in error propagation analysis when working with high resolution DEMs. Especially anthropogenic features such as roads and bridges cause a high amount of uncertainty in such high resolution DEMs.

Zusammenfassung

In den geographischen Informationswissenschaften ist es üblich digitale Höhenmodelle (DHMs) zur Gewinnung von verschiedenen topografischen Attributen, hydrologischen Parametern und Indizes zu verwenden. Normalerweise werden DHMs als korrekte Abbilder der Geländeoberfläche wahrgenommen. Aber wie die Bezeichnung "Modell" impliziert, ist ein DHM nur eine Annäherung an die reale Geländeoberfläche. Die Qualität der DHMs leidet unter systematischen und zufälligen Fehlern, welche durch in der digitalen Geländemodellierung häufig vorkommenden Anwendungen verursacht werden können. Wenn DHMs in der digitalen Geländeanalyse verwendet werden, wirken sich die Fehler des Models negativ auf die daraus abgeleiteten Resultate aus. Die Untersuchung dieses Phänomens wird als Fehlerfortpflanzungsanalyse bezeichnet. Die Resultate dieser Analyse haben einen direkten Einfluss auf jeglichen Entscheidungsprozess der auf Anwendungen und Interpretationen von digitalen Geländeanalysen beruht.

Ab Mitte der 1990er, Light Detection And Ranging (LiDAR), auch bekannt als Laserscanning, hat das Erfassen von hochpräzisen Höheninformationen auf eine schnelle und gleichzeitig kosteneffiziente Weise ermöglicht. Diese Arbeit fokussiert auf solche micro-scale DHMs, welche aus LiDAR Daten erstellt wurden, weil auch mit hochpräzisen Messverfahren wie LiDAR, die daraus erstellten DHMs noch Unsicherheiten enthalten. Das in dieser Arbeit verwendete LiDAR basierte DHM weist eine räumliche Auflösung von 2 m und einen durchschnittlichen Fehler in horizontaler und vertikaler Richtung von 0.5 m auf.

Diese Arbeit stellt eine Unsicherheitsanalysenmethodik zur Untersuchung der Fehlerfortpflanzung in Geländeanalysen basierend auf hochauflösenden DHMs vor. Sämtliche Algorithmen sind mit C++ unter der Nutzung eines Multiprozessierungsansatzes basierend auf MPI implementiert, um eine effiziente Fehlerfortpflanzungsanalyse von hochauflösenden DHMs zu ermöglichen. Die implementierten Algorithmen decken die gebräuchlichsten hydrologischen Modelle ab. Die vorgeschlagene Methode beinhaltet vier verschiedene Fliessrichtungsalgorithmen (D8, D∞, MD8 und MD∞) zur Berechnung der akkumulierten Fläche und der topografischen Indizes TWI, SPI und STI, und ermöglicht die Extraktion von Einzugsgebieten und Flussnetzwerken. Zur Untersuchung der statistischen und räumlichen Eigenschaften der Unsicherheiten wurden entsprechende Methoden implementiert. Unsicherheiten werden mit Fehlerkarten visualisiert, welche die absolute oder relative Standardabweichung, Wahrscheinlichkeiten oder den Fehler H aufzeigen. Zusätzlich können verschiedenste Resultate in Textfiles geschrieben werden, welche für weitere Analysen in entsprechend Statistik-Programmen genutzt werden können. Stichproben können innerhalb des DHMs definiert werden, welche dann in der Unsicherheitsanalyse genauer untersucht werden können. Dies kommt zum Beispiel bei der Analyse der Konvergenz verschiedener hydrologischer Modelle in einer Fehlerfortpflanzungsanalyse zum Einsatz.

Das Unsicherheitsmodell basiert auf der Monte-Carlo-Simulation und nutzt den Process Convolution Ansatz zur Erzeugen von sogenannten Fehleroberflächen. Zusätzliches Wissen über die räumliche

Struktur des Fehlers in LiDAR basierten DHMs ermöglichte eine entsprechende Verbesserung des Unsicherheitsmodells.

Neben der Programmierung und des vorgeschlagenen Rahmenkonzeptes zur Untersuchung der Fehlerfortpflanzung, liefert diese Arbeit auch verschiedene Unsicherheitsassessments und Analysen zur räumlichen Struktur der Unsicherheiten von gebräuchlichen hydrologischen Modellen. Des Weiteren werden die Einflüsse verschiedener Parameter des Unsicherheitsmodells und schlechter aufgelösten DHMs auf die Resultate der Unsicherheitsanalyse ausgewertet. Zusätzlich werden Konvergenztests für verschiedene hydrologische Modelle durchgeführt. Schliesslich präsentiert diese Arbeit eine Evaluation der Sensitivität der vier verschiedenen Fliessrichtungsalgorithmen und evaluiert die Leistung der vorgeschlagenen Unsicherheitsanalysenmethode.

Die Resultate zeigen, dass der angewandte Parallelisierungsansatz Fehlerfortpflanzungsanalysen in Abhängigkeit der Anzahl an verwendeten Prozessen deutlich beschleunigen kann. Die Ausführungszeit kann durch die Nutzung von vier statt einem Prozess um bis zu einen Faktor von drei reduziert werden. Die Unsicherheitsassessments zeigen, dass Multiple Flow Direction Algorithmen weniger sensitiv sind im Vergleich zum single flow direction Algorithmus D8. Die beobachteten Fehler liegen bis zu dreimal höher, wenn der D8 an Stelle des MD∞ Fliessrichtungsalgorithmus benutzt wird. Diese Erkenntnis trifft ebenfalls auf die Konvergenztests zu. Der D8 basierte TWI konvergiert deutlich später (400 zu 200 Simulationsdurchläufe). Generell zeigen alle Fliessrichtungsalgorithmen sehr instabile Werte unter 100 Simulationsdurchläufen. Im Gegenzug erscheint die Extraktion von Einzugsgebieten stabiler. In diesem Fall konvergierte die Fläche des Einzugsgebietes schon nach 150 Simulationsdurchgängen. Bezüglich der räumlichen Struktur der Unsicherheiten zeigen die Resultate, dass hohe Fehler hauptsächlich in flachen und rauen Geländeoberflächen vorkommen. Die Resultate zeigen ebenfalls, dass die räumliche Auflösung des DHMs einen weniger bedeutenden Einfluss auf die Analyseergebnisse hat. Diese hängen jedoch stark von den gewählten Parametern des Unsicherheitsmodells ab. Zum Beispiel wird in dieser Arbeit gezeigt, dass eine hohe räumliche Autokorrelationsdistanz die erhaltenen Fehler deutlich reduzieren kann und umgekehrt. Das zeigt, dass der Process Convolution Ansatz an seine Grenzen stösst. Bessere Methoden müssen implementiert werden um genauere Fehleroberflächen erzeugen zu können.

Die in dieser Arbeit präsentierte Methodik und ihre Anwendung zur Bewertung der Unsicherheiten in hydrologischen Modellen zeigen auf, dass es möglich und sinnvoll ist die Fehlerfortpflanzung zu untersuchen, wenn man mit hochaufgelösten DHMs arbeitet. Gerade anthropogene Erscheinungen wie Strassen und Brücken führen zu hohen Unsicherheiten in hochaufgelösten DHMs.

Contents

List of Figures

List of Tables

List of Codes

List of Abbreviations

ANS	Average Neighborhood Slope
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DEM	Digital Elevation Model
DHS	Downhill Slope
DSM	Digital Surface Model
DTM	Digital Terrain Model
GIS	Geographic Information System
GPGPU	General-Purpose Computing On Graphics Processing Units
GPS	Global Positioning System
GPU	Graphics Processing Unit
HPC	High Performance Computing
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LiDAR	Light Detection And Ranging
m a.s.l.	meters above sea level
MFD	Multi Flow Direction
MPI	Message Passing Interface
RAM	Random-Access Memory
RMSE	Root Mean Square Error
rstd	relative standard deviation
SCA	Specific Catchment Area
SFD	Single Flow Direction

SGS	Sequential Gaussian Simulation
SPI	Stream Power Index
std	standard deviation
STI	Sediment Transport Index
TauDEM	Terrain analysis using Digital Elevation Models
TWI	Topographic Wetness Index
USLE	Universal Soil Loss Equation
WSL	Swiss Federal Institute for Forest, Snow and Landscape Research

1 Introduction

1.1 Context

Gridded digital elevation models (DEMs) play an important role in geographic information science due to their wide range of uses and applications. DEMs are often assumed to be accurate representations of the real-world terrain surfaces. But, as the name "model" already indicates, a DEM is only an approximation of the true terrain surface. The accuracy of DEMs suffers from systematic and random errors, caused by the tasks commonly involved in digital terrain modeling. These tasks can range from elevation data acquisition to geospatial applications (Oksanen, 2006). The sources of uncertainty are defined by Li et al. (2005) as follows:

- Gross, systematic and random errors
- Accuracy loss due to lattice representation of terrain

Oksanen (2006) described the sources of errors in DEMs more extensively (Figure 1.1). Errors in phase 1 (Figure 1.1) cover random and human error in data acquisition including errors caused by measurement equipment. There could also be errors due to cloud and forest cover, Earth's curvature, instable remote sensing equipment, atmospheric refraction, image processing, global positioning system (GPS) and inertial navigation system (INS), cartographic generalizations, and geometric transformations (Florinsky, 1998b; Oksanen, 2006). The selected method for digital surface representation, the method of DEM approximation, and the level of detail are sources of error in phase 2 (Figure 1.1) (Oksanen, 2006). In phase 3 (Figure 1.1) uncertainty is caused by the algorithms used to derive topographic attributes (Zhou and Liu, 2004; Oksanen, 2006). Finally, in phase 4 (Figure 1.1), uncertainty may be caused by misinterpretations (Oksanen, 2006).

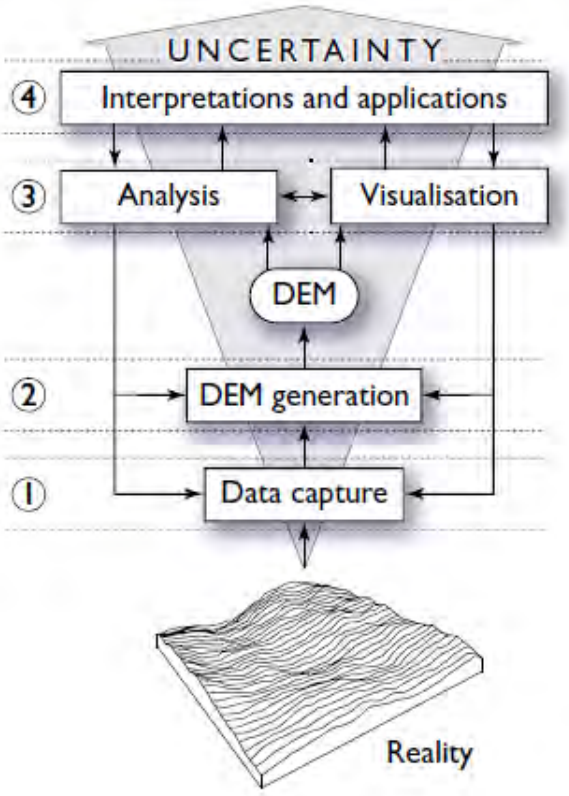


Figure 1.1: The 4 phases of digital terrain modeling which may cause uncertainty (Oksanen, 2006).

Since the mid 1990's, light detection and ranging (LiDAR), also known as laser scanning, has allowed to collect highly accurate elevation data in a rapid and still very cost effective way (Fowler, 2001). This thesis focuses on such micro-scale DEMs retrieved from LiDAR data. But even with highly accurate measurement techniques such as LiDAR, the resulting DEMs still contain a certain amount of uncertainty (Evans and Hudak, 2007; Bater and Coops, 2009). This uncertainty is caused by the unknown spatial distribution and magnitude of errors, inherent in any DEM. These errors describe the deviation of measurements from the true values (Hebeler and Purves, 2008). In this context, uncertainty is used for values which are expected to deviate from its real value with an unknown deviation extent and error is considered as a measurable part of uncertainty. When DEMs are used in terrain analysis, the errors inherent in any DEM will affect the results and the extent of deviations can only be approximated by using uncertainty models (Holmes et al., 2000). The investigation of this phenomena is known as error propagation analysis (Heuvelink, 1998). The results of these analyses have a direct influence on any decision-making process based on interpretations and applications of terrain analysis (Oksanen, 2006).

Error propagation analysis is computationally very demanding and time intensive even on coarse resolution DEMs (e.g. Oksanen, 2006; Hengl et al., 2010). Therefore, error propagation analysis using high resolution DEMs is a challenge. New methods are needed to facilitate such analysis when working with LiDAR based DEMs. Tesfa et al. (2011) proposed highly efficient and parallelized algorithms to compute various hydrological models. This parallelization approach may also be applied to an error propagation analysis approach allowing such analysis to be completed in appropriate time frames even when working with LiDAR data.

1.2 Review

Uncertainty has been an important topic in geographical information science for over two decades. As a result of this, researchers recommend two outputs for spatial analysis: a map representing the result of the analysis and some assessment of uncertainty for this map (Atkinson and Foody, 2006). The error propagation approach was first used by Burrough (1986). At that time, computers were too slow to run stochastic simulations in an appropriate time frame. Since then, algorithms have improved significantly and computers are more powerful. A goal of the research in DEM uncertainty analysis has been to develop a general purpose "error button" for geographic information systems (GIS) (Openshaw et al., 1991). The functionality of the "error button" is seen as user-dependent which offers the possibility to adjust the error model according to the level of knowledge of the user (Heuvelink, 2006). But as of today, we are still far away from an uncertainty button in any GIS. Researchers are questioning themselves why users have overlooked the importance of error propagation analysis, making such tools an exception rather than a rule (Oksanen, 2006). Too many limitations still remain in recently proposed uncertainty analysis approaches (e.g. Hengl et al., 2010; Achour et al., 2012). An often mentioned problem is the performance, even though computers are getting faster, these tools are often limited to DEMs containing a few thousand cells due to inefficient algorithms. But there are other critical issues, like open questions about the uncertainty model itself. Besides the selection of an appropriate uncertainty model approach, many parameters of the uncertainty model can vary in a certain range and thus have a direct influence on the result of the uncertainty analysis. Defining these parameters for an uncertainty analysis is a very difficult task and one of the main reasons why uncertainty analysis still attracts little attention (Heuvelink,

2006).

DEM error propagation analysis was introduced in the early 1990's. Heuvelink et al. (1989) used Monte Carlo simulations and Taylor series approximations in their pioneering work for error propagation analysis of slope and aspect calculations. The standard deviations (std) were higher than expected and the sensitivity of run-off and erosion models was recognized. Since then, the assumption has raised that a spatially uncorrelated error can be used for a worst case scenario (e.g. Heuvelink, 1998; Wechsler, 2007). Wechsler (2007) proposed a practical implementation of error propagation analysis for commercial GIS software using the Monte Carlo approach. Besides the programming task, the focus was on the effects of uncertainty on topographic parameters and compound indices. Uncalibrated moving spatial averages were used, leading to a lack of control on the standard deviation and spatial autocorrelation range for the generation of the error surfaces. Only a few Monte Carlo based studies have paid attention to the convergence of error propagation analysis results (Veregin, 1997; Nackaerts et al., 1999; Aerts et al., 2003; Oksanen, 2006). Oksanen (2006) showed that the extraction of drainage basins appeared to converge after 500 simulation runs. Such studies using high numbers of simulation runs are rare. The most studies performed Monte Carlo simulations with 100 or less runs (e.g. Temme et al., 2009; Hengl et al., 2010). The problem is that convergence analysis is always case-specific. The results depend on the terrain, uncertainty model parameters, and the terrain analysis task. Hebeler and Purves (2009) provided a methodology for calculating an error surface using the process convolution approach with calibrated parameters and applying such error surfaces in developing uncertainty models. They showed that for two larger watersheds, hypsometric curves, elevation, catchment area, and slope have considerable uncertainty as a function of uncertainty in elevation. Hebeler and Purves (2008) used the same uncertainty model approach for ice sheet model uncertainty assessments. They came to the conclusion that DEM uncertainty has some impact on the ice sheet modeling processes, but that additional uncertain parameters such as the temperature are needed to relate the impact of DEM uncertainty. Hengl et al. (2010) proposed a Monte Carlo based error propagation analysis approach to evaluate the sensitivity of stream network extraction. They showed that streams are difficult to map in low relief areas and on slightly convex terrain shapes. As of today, error propagation in hydrological modeling using high resolution DEMs such as LiDAR derived DEMs was rarely investigated (e.g. Ulmer, 2007).

1.3 Motivation and Objectives

This thesis covers three larger research fields: GIScience, hydrology and computer science. Each field provides solutions and articulates demands. In general, the sensitivity of hydrological algorithms is of interest (e.g. Kenward et al., 2000; Wechsler, 2007; Hengl et al., 2010; Achour et al., 2012). In hydrology, multiple algorithms exist to calculate the upslope area and it is of interest to know if there are any algorithms more robust than others in uncertainty analysis. It is also possible that the uncertainty of hydrological models depends on the spatial structure of the terrain and changes its behavior on different scales. This additional information can help to improve hydrological applications which use results of geomorphometric analysis as input data. In GIScience, questions are raised about the parameters of the uncertainty model itself (Heuvelink, 2006; Hengl et al., 2010). If a Monte Carlo simulation approach is used, users have to know how many simulation runs are sufficient for an adequate and valid result. As of today, such information is scantily available in literature (e.g. Oksanen, 2006). If control points (true elevations) are available, correlograms or

(semi)variograms can be fitted and a sequential Gaussian simulation (SGS) can be used to generate appropriate error surfaces (e.g. Hengl et al., 2010). Typically control points are not available, but an uncertainty assessment is still required. The process convolution approach is a possible solution for this case (e.g. Oksanen and Sarjakoski, 2005a; Hebeler and Purves, 2008). Users are commonly provided only by a global uncertainty estimator such as the RMSE (Root Mean Square Error) and the spatial autocorrelation of the error must be estimated. However, a slightly different spatial autocorrelation range can already affect the derived results. Analyzing the impact of different uncertainty model parameters in process convolution approaches could also help to improve such uncertainty analysis models. As already mentioned, the performance is a huge limitation in uncertainty analysis. Recently, computer science researchers provided new computation approaches for established geomorphometric algorithms (e.g. Tesfa et al., 2011; Zhan and Qin, 2011). A famous word in this context is "parallelization". By exploiting the computation power of multi-processing systems, algorithms can be accelerated significantly. These approaches can be very useful in compute intensive tasks such as Monte Carlo simulations. Shortening the time needed for such analyses would probably make this optional step of an uncertainty assessment more attractive for hydrological users. Very large LiDAR data DEMs may be processed in an appropriate time frame or more simulation runs may be completed in the same time leading to more stable and thus more reliable results.

Based on the above described issues concerning the analysis of uncertainty and its impact on hydrological models, I thought about a way to analyze the impact of DEM uncertainty on hydrological models efficiently with as few limitations as possible and to analyze the influence of different uncertainty model parameters. The first and main objective of this study is to provide a solution to how an uncertainty model could be implemented to fit the requirements of a hydrological user working with high resolution DEMs. Secondly, the implementation of the uncertainty model should consider the spatial pattern of uncertainty in LiDAR data and make use of recently provided computing approaches. Thirdly, the study should provide some uncertainty assessments of common hydrological models and investigate the impact of different uncertainty model parameters, coarser spatial resolution DEMs, and the terrain characteristics on the uncertainty analysis.

1.4 Research Questions and Hypotheses

Regarding the above mentioned objectives, the following key questions will be answered by the end of this thesis:

Research question 1: Can the time needed for uncertainty analysis be reduced considerably with parallelized algorithms?

Working hypothesis 1: Considerably lower execution times with an increasing number of parallel processes will be expected.

Research question 2: How many realizations in a Monte Carlo simulation are sufficient for common hydrological algorithms? How well do they converge with an increasing number of runs?

Working hypothesis 2: The results will vary spatially depending on the terrain characteristics.

Research question 3: How sensitively do the different hydrological algorithms behave in uncertainty analysis?

Working hypothesis 3: Multiple flow direction algorithms will be less sensitive than the single flow direction algorithm and the hydrological algorithms will show the highest errors in flat terrain.

Research question 4: What are the impacts of the input parameters of the uncertainty model on the uncertainty analysis results?

Working hypothesis 4: Input parameters changing the error surface in the expected sensitive flat areas will have an influence on the derived uncertainty assessments.

Research question 5: Do coarser spatial resolutions of DEMs change the findings of the above asked research questions?

Working hypothesis 5: Major differences will not be expected.

1.5 Thesis Structure

The structure of the thesis reflects the conducted work. After the introduction, Chapter 2 describes the related work and provides the necessary theoretical background for this thesis. Chapter 3 presents the methodology. Besides the study area and related data, the integration of the uncertainty analysis is presented. A short performance evaluation of the proposed uncertainty analysis approach is presented in Chapter 4. The results of the uncertainty analysis are evaluated and discussed in Chapter 5. The following Chapter 6 discusses the acquired results in the context of recent research and answers the research questions. Finally, the concluding remarks are given in Chapter 7, describing what this study has achieved, providing the insights and giving an outlook for future work.

2 Background

This chapter describes the relevant areas of research and explains the most important terminology for this study. It includes eight sections starting with a general introduction to digital elevation models and a more detailed explanation of the elevation measurement technique LiDAR. Section 2.2 describes the current state of the art for DEM uncertainty analysis and provides detailed information of the Monte Carlo simulation approach and some insights to geostatistics and random fields. The next three sections are about various models derived from DEMs. Section 2.3 describes the topographic attributes slope and convexity, whereas Section 2.4 introduces the hydrological algorithms to compute flow directions, watersheds and stream networks. Section 4.2 characterizes the compound indices and Section 2.6 gives an introduction to parallel computation focusing on the multi-processing approach. The following Section 2.7 outlines some common applications in hydrology and finally, Section 2.8 introduces and illustrates the study area used for the analysis in this thesis.

2.1 Digital Elevation Models

In geographic information science, land-surfaces are commonly presented as DEMs, which are representations of the continuous surface (Hengl and Evans, 2009). This means that the elevation can be calculated at any point of interest in the representation. DEMs can be classified into two types: raster-based or vector-based DEMs (Weibel and Heller, 1991). In this thesis gridded DEMs are used, because they have a very simple structure and thus are more suited for implementing geomorphometric algorithms. A raster-based DEM consists of a rectangular matrix of heights, which means that each cell can be represented by a x, y, z-triple. The x and y variables define the coordinate, whereas the z variable contains the height information.

In general, there are three sources of DEM data: Ground survey techniques, topographic maps, and remote sensing (Nelson et al., 2009). Three different remote sensing methods exist to derive DEMs: stereo photogrammetry, radar interferometry, and laser altimetry. Traditionally, stereo pairs of aerial photographs are used to derive DEMs at spatial resolutions between 15 and 100 m (Creed and Sass, 2011). The vertical error of this type of DEMs is in the range of 1-8 m. Radar interferometric techniques using satellite platforms are able to provide global coverage of DEMs at 25-100 m spatial resolution with vertical errors varying between 10 and 20 m (Creed and Sass, 2011). The latest approach to collect elevation data is laser altimetry, also known as light detection and ranging. LiDAR provides sub-meter spatial resolution with a vertical accuracy of usually less than 50 cm, even in dense vegetation areas (Andersen et al., 2005). The vertical error of LiDAR-derived DEMs under various conditions has been studied extensively in literature (e.g. Andersen et al., 2005; Hodgson et al., 2005; Joerg et al., 2012).

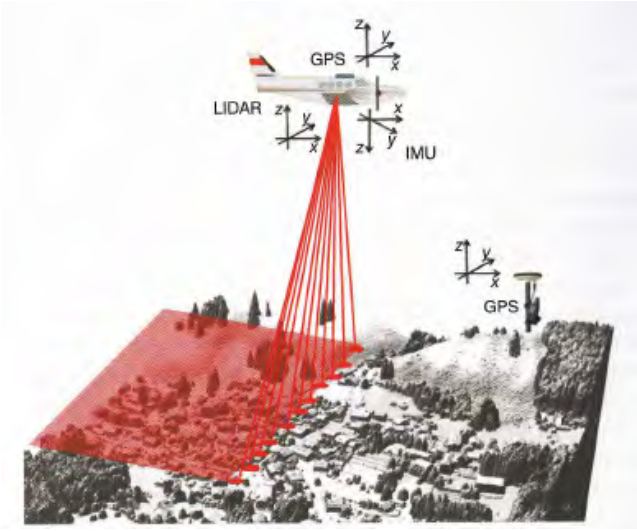


Figure 2.1: The airborne laser scanning principle (Beraldin et al., 2010).

2.1.1 LiDAR

In the early 1990's, the first commercial topographic mapping systems using LiDAR appeared (Nelson et al., 2009). According to Beraldin et al. (2010), an airborne laser scanner is based on six principle components:

- Scanner assembly
- Airborne GPS antenna
- Inertial measurement unit (IMU)
- Control and data recording unit
- Operator laptop
- Flight management system

The LiDAR system is completed by a GPS ground station (Figure 2.1). The active sensor of the LiDAR system transmits a signal towards the ground and records its reflection. The time delay between the sending and recording of the signal can be used to measure the distance from the sensor to the reflection point on the ground. A signal can be first reflected by the canopy, but still continue downwards reflecting all objects it hits until it may finally reflect on the ground (Figure 2.2). It is possible to record these multiple reflections of a signal, thus allowing to extract a digital terrain model (DTM) and a digital surface model (DSM) from the collected point cloud of a LiDAR mission using various interpolation methods. The last echoes are used for the DTM extraction, whereas the first echoes are used for the DSM extraction (Beraldin et al., 2010).

2.1.2 Uncertainty in DEMs

As already mentioned, DEMs can be produced in different ways. All these methods are subject to uncertainties. Measurement errors, interpolations of point data and digitization errors will affect the

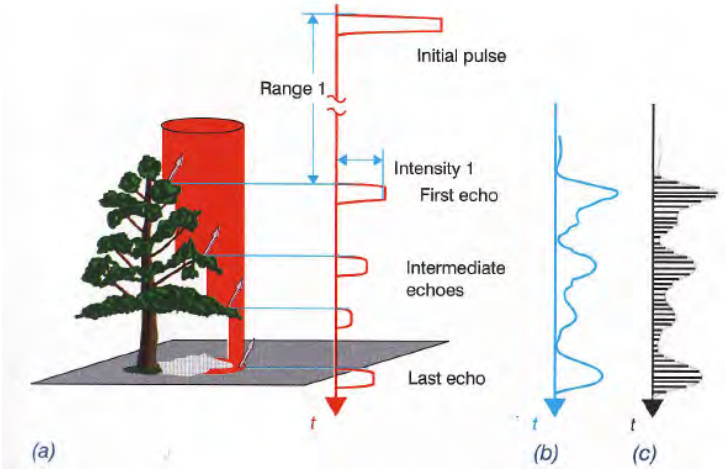


Figure 2.2: LiDAR based elevation measurements can produce multiple echoes: (a) discrete pulse; (b) waveform; (c) digitized waveform (Beraldin et al., 2010).

accuracy of the DEM (Temme et al., 2009). According to Wise (2000) and Reuter et al. (2009), the errors in DEMs can be classified into three groups:

- Artifacts, blunders or gross errors
- Systematic errors
- Random errors or noise

Artifacts can have a huge impact on the derived land-surface parameters. They can often be seen in the data as unrealistic or missing values. It is important to be aware of them to avoid misinterpretation of the derived results.
Systematic errors are caused by the collection method, e.g. LiDAR. They are almost not detectable in a single DEM, but they can often be shown by comparing different DEM data of the same area (e.g. from two different LiDAR missions). We have to rely on the people responsible for the generation of DEMs correcting most of the systematic errors.
Random errors are inherent in any DEM and are caused by the measurement itself, i.e. signal noise. The difference between the estimated and true value is often available for final DEM products and indicated by the root mean square error of elevation (RMSE) (Wise, 2000):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}[z(s_i) - z_{REF}(s_i)]^2}{n}} \tag{2.1}$$

where $z(s_i)$ refers to the i^{th} interpolated elevation, $z_{REF}(s_i)$ refers to the i^{th} known or measured elevation and n is the number of samples.

These errors will propagate to the derived topographic parameters and hydrological models. The impact is not easily predictable, small errors can have a huge impact, whereas large errors may cancel out in other cases. DEMs with a high spatial resolution are often interpreted as less uncertain. This assumption is wrong, even though DEMs with higher spatial resolution are better able to describe small-scale landscape variations than low resolution DEMs, the potential to describe the landscape mainly depends on the accuracy of the attribute value (Temme et al., 2009). There are numerous studies which have analyzed the uncertainty of hydrological models derived from DEMs at different

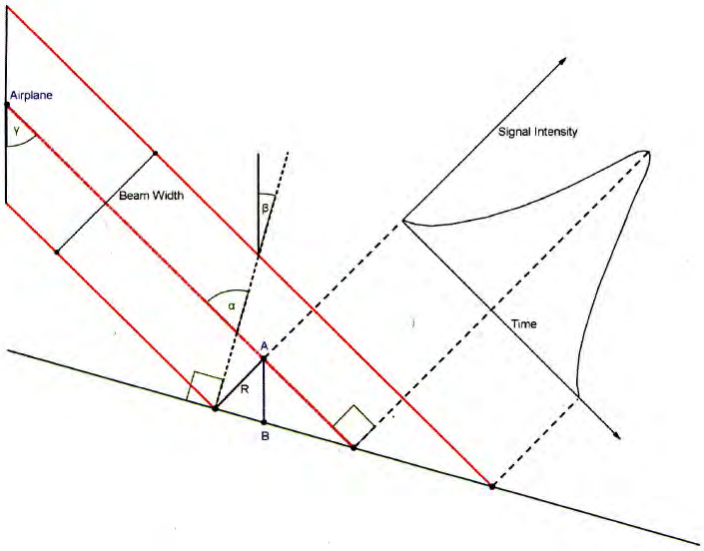


Figure 2.3: Geometrical measurement situation. In this case the elevation is systematically overestimated and the error depends on the angle of incidence (α) and the beam width (Joerg et al., 2012).

scales, but only a few used LiDAR data. For example, Vaze et al. (2010) have resampled a 1 m LiDAR based DEM to different coarser spatial resolutions and have compared the results. In their conclusion, they propose to use high resolution DEMs or resampled DEMs over contour derived DEMs to achieve more accurate results.

Spatial Structure of Error in DEMs derived from LiDAR Data

Schaer et al. (2007) showed that the scanning geometry has a large impact on the vertical accuracy. Figure 2.3 illustrates the error occurring in steeper slopes. The larger the incident angle and the higher the slope, the stronger the vertical accuracy degradation (Lichti and Skaloud, 2010). In addition to the slope, the canopy also has an impact on the accuracy of the derived DEM. At large incident angles or in dense canopy areas, the signal reflects at the canopy without hitting the ground. The penetration ability of a signal also directly depends on the footprint size; smaller footprints are able to hit the ground more often in dense vegetation areas. This results in a higher uncertainty in dense vegetation areas and steep slopes compared to flat and vegetation free terrain (Joerg et al., 2012).

2.2 Modeling Uncertainty

In geomorphometry, geostatistical modeling is used to analyze the propagation of DEM errors. The following theoretical background has been discussed by Oksanen (2006); Temme et al. (2009). Considering a DEM with the spatial domain of interest D:

$$z^* = (z^*(s) \mid s \in D) \tag{2.2}$$

The true elevation Z is uncertain. The regionalized variable theory can be applied to express the error (Burrough and McDonnell, 1998). The true elevation Z is defined as a sum of the structural components with a constant trend $z^*(s)$, a random but spatially correlated component $\epsilon'(s)$ and a spatially uncorrelated random noise term ϵ'':

$$Z(s) = z^*(s) + \epsilon'(s) + \epsilon'' \tag{2.3}$$

The true error ϵ is unknown and can only be quantified by a probability distribution. This is the reason why a probabilistic approach is used. We assume that ϵ is normally distributed and that the DEM is free of systematic errors or that the systematic error is constant over the spatial domain. Thus, the normal distribution parameter mean μ_ϵ is defined as zero. The parameter standard deviation δ_ϵ represents the random error in $z^*(s)$. As discussed in Section 2.1, the error can vary spatially. Therefore, the spatial domain can be divided into subregions with different values for δ_ϵ (Temme et al., 2009).

The errors in DEMs are spatially autocorrelated. If an elevation at some location is underestimated, all of its neighbors are usually underestimated too. This is a result of how DEMs are produced. Correlograms $\rho_\epsilon(|\mathbf{h}|)$ or (semi)variograms $\gamma_\epsilon(|\mathbf{h}|)$ can be used to characterize the degree of spatial correlation between $\epsilon(s)$ and $\epsilon(s + \mathbf{h})$ if it only depends on the distance of vector $\mathbf{h}$ between the locations. This means that the correlation is independent of the direction of vector $\mathbf{h}$ (i.e. isotropy) (Temme et al., 2009). There are many methods in geostatistics for the estimation of correlograms or (semi)variograms for point datasets (Goovaerts, 1997). One method for obtaining the spatial autocorrelation range D is through analysis of the sill of a (semi)variogram of DEM error (Temme et al., 2009). The problem is that data about the spatial structure of error is often not available. Without extensive ground truth, the spatial autocorrelation of the error can be subjective and cannot be determined according to Wechsler (2007).

In GIS operations, the errors in the input data will propagate to the output of the analysis. Users may rarely pay attention to this problem, although they may be aware of it. To assess the quality of the analysis, uncertainties have to be defined and explored through an uncertainty propagation analysis. The common practice in uncertainty analysis has been to use the Monte Carlo method or the analytical approach (e.g. Heuvelink et al., 1989; Florinsky, 1998a; Kyriakidis et al., 1999; Wise, 2000). The Monte Carlo approach is seen as more versatile than the analytical approach for solving real world problems of uncertainty (Zhang and Goodchild, 2002). For example, Raaflaub and Collins (2006) have stated that the analytical method is unsuitable for local derivatives such as the D8 flow routing algorithm. For more information about the analytical method, readers are referred to Schneider (2000). The next section will describe how error propagation in spatial modeling can be computed using the Monte Carlo simulation approach.

2.2.1 Monte Carlo Simulation

The Monte Carlo method is the most often used error propagation method because it is flexible, generic and intuitively appealing (Temme et al., 2009). In a Monte Carlo simulation, the original DEM is recognized as only one possible realization of the real-world topography. With a Monte Carlo simulation multiple realizations are generated and used to quantify DEM uncertainty with

statistical evaluations (e.g. Oksanen and Sarjakoski, 2005b,a; Oksanen, 2006; Wechsler, 2007). After Heuvelink (1998), the error propagation problem can be formulated as follows:

$$U = g(A_1, ..., A_n) \tag{2.4}$$

where U is the output of a GIS operation g on n different inputs A_i. The objective of the uncertainty analysis is to determine the uncertainty in the output U, given the operation g and the uncertain inputs A_i. The operation g could be a slope algorithm, whereas the input A_i could be a DEM. In error propagation analysis of digital terrain analysis usually only the DEM is taken as an uncertain input (e.g. Oksanen, 2006; Wechsler, 2007; Hengl et al., 2010). The Monte Carlo method computes U multiple times with inputs a_i which are randomly sampled from their joint probability distribution of A_i. According to Temme et al. (2009), the Monte Carlo simulation consists of the following steps:

(1) repeat N times:

  (a) generate a random realization a_i from the joint probability distribution of each A_i

  (b) for each a_i, compute and store the output $u = g(a_i, ..., a_n)$

(2) compute and store sample statistics from the N outputs u

To generate a random sample a_i of A_i, a pseudo random number generator is used (e.g. Van Niel and Laffan, 2003). Because of the joint probability distribution, higher numbers of realizations N will lead to more stable results. How large the number of realizations should be depends on the accuracy requirements of the obtained result. According to Brown and Heuvelink (2006), the accuracy of the Monte Carlo approach is proportional to the square root of the number of runs N. This means that any degree of accuracy can be reached with a large enough number of realizations N, but also that the computation time of an uncertainty analysis quadruples if we want to double the accuracy. Heuvelink (1998) came to the conclusion that in most cases, N should be at least 100 and that less then 20 simulations are insufficient. These numbers were proposed a long time ago on limiting computers and coarse resolution DEMs. A more recent study by Oksanen (2006) investigated the convergence of watersheds, coming to the conclusion that in these cases, the analysis appeared to converge after 500 simulation runs.

2.2.2 Geostatistics and Random Fields

The operations g usually involve spatial interactions. This requires that the spatially distributed uncertain inputs are generated spatially autocorrelated. Two common methods exist for stochastic spatial simulation. One is the sequential Gaussian simulation which is based on simple kriging (Goovaerts, 1997). The other one, known as process convolution, is based on spatial moving averages. The idea is to increase the amount of spatial autocorrelation in the random field of an uncorrelated noise by applying a low-pass filter (Cressie and Pavlicová, 2002). A random field is illustrated in Figure 2.4. A selection of various moving window average approaches has been discussed by Wechsler (2007). SGS is more complex and computationally more demanding and needs control points to fit a correlogram or (semi)variogram. Recently, Hengl et al. (2010) proposed a solution for stream network uncertainty analysis based on SGS. Figure 2.5 illustrates their approach.

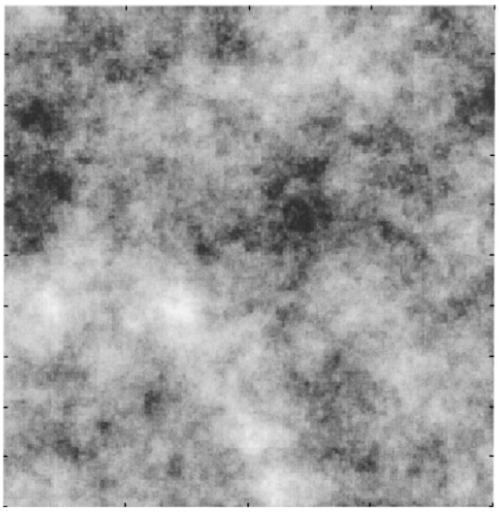


Figure 2.4: A 200x200 random field generated with an exponential correlogram (mean = 0, variance = 1, correlation range = 50 cells) (Ravalec et al., 2000).

Because of the lack of control points in most cases, an appropriate (semi)variogram cannot be fitted and a process convolution approach is the only way to produce appropriate error surfaces. Process convolution is a special case of moving window averages, where the spatial structure of the random field is controlled by a correlogram. The process convolution's objective is to filter an uncorrelated Gaussian noise using a unimodal and symmetric kernel k that produces a match between the a priori defined correlogram and the correlogram of the resulting correlated random field (Kern, 2000). Random fields have been discussed extensively in literature (e.g. Abrahamsen, 1997; Ravalec et al., 2000; Higdon, 2002) and were used successfully in DEM uncertainty studies (e.g. Oksanen and Sarjakoski, 2005a; Oksanen, 2006; Wechsler, 2007; Hebeler and Purves, 2008).

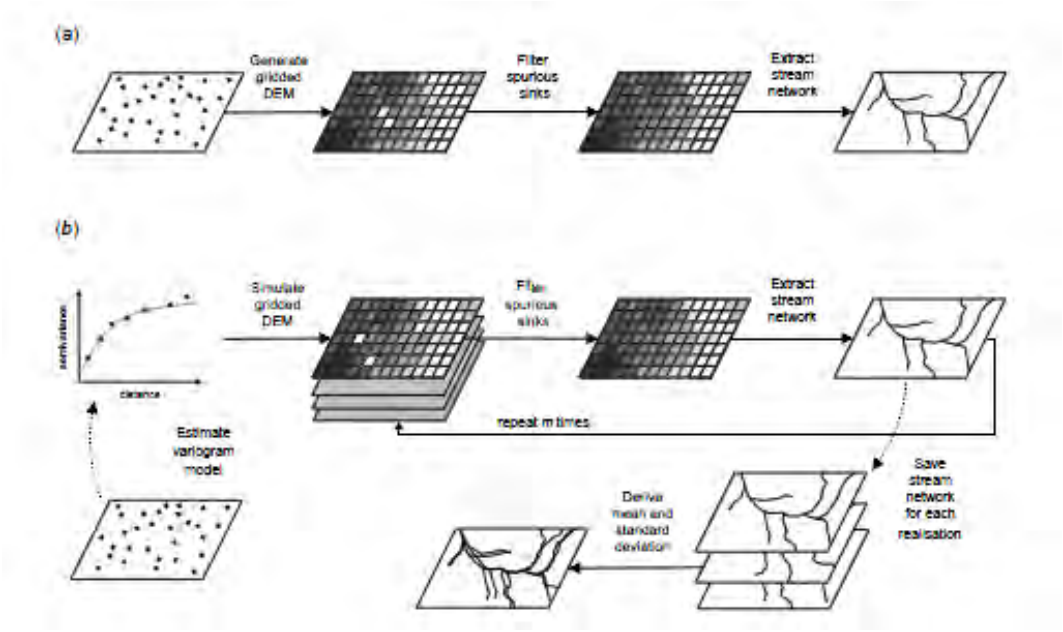


Figure 2.5: Analyzing the uncertainty of stream networks with SGS: (a) the DEM uncertainty is not considered, (b) considering the DEM uncertainty with a Monte Carlo simulation using the SGS approach (Hengl et al., 2010).

2.3 Topographic Attributes

This section gives a short overview over the basic land-surface parameters used in this thesis. These parameters can be derived directly from a DEM without further knowledge of the represented area. Various topographic attributes are required for hydrological modeling (e.g. Callow et al., 2007).

2.3.1 Slope

The slope is derived from the first partial derivatives of its surface and thus belongs to the functions of first derivatives. Usually GIS software provides a slope function based on the average neighborhood approach (ANS). With the ANS method the slope is estimated by calculating the rate of change in elevation over the distance from the central cell to its eight neighbors using an average maximum approach (Burrough and McDonnell, 1998). A 3x3 window is created to determine the average rate of change in elevation along lateral directions $(\delta Z/\delta Y)$ and the horizontal direction $(\delta Z/\delta X)$ from the central cell (Irfan et al., 2012):

$$ANS(e) = arctan\left(\sqrt{\left(\frac{\delta Z}{\delta X}\right)^2 + \left(\frac{\delta Z}{\delta Y}\right)^2}\right) \tag{2.5}$$

with:

$$\frac{\delta Z}{\delta X} = \frac{(Z_c + 2\Delta Z_f + Z_i) - (Z_a + 2\Delta Z_d + Z_g)}{8\Delta x} \tag{2.6}$$

$$\frac{\delta Z}{\delta Y} = \frac{(Z_g + 2\Delta Z_h + Z_i) - (Z_a + 2\Delta Z_b + Z_c)}{8\Delta y} \tag{2.7}$$

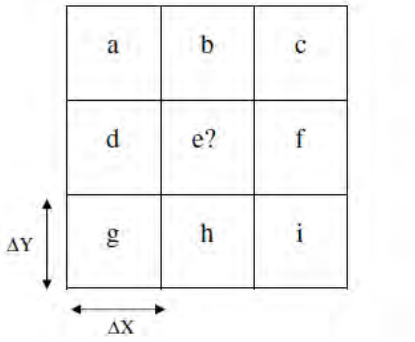


Figure 2.6: A 3x3 window used to calculate the slope for the central cell e (Irfan et al., 2012).

$Z_{j=a...i,\neq e}$ represents the eight neighboring cells and Δx and Δy specify the cell dimensions and are usually equal (Figure 2.6).

According to Irfan et al. (2012), the downhill slope method (DHS) should be preferred over the ANS method if a hydrological analysis is performed on high-resolution LiDAR data. The ANS method generally overestimates the slope in areas with low steepness and underestimates areas with high steepness. In addition, the differences between the two methods increase especially in high-resolution DEMs. Hence, the DHS is usually the standard slope method used by flow routing algorithms such as the D8 to determine the flow directions. The DHS method calculates the slope in one of eight directions from the central cell with the maximum elevation drop (Zhang et al., 1999):

$$DHS(e) = arctan\left[max\left(\frac{Z_e - Z_j}{L_j}\right)\right] \tag{2.8}$$

Z_e ist the central cell while Z_j specifies the neighboring cell with the maximum drop. L_j is Δx in b, d, f and h direction and $\sqrt{2}\Delta x$ in a, c, g and i direction (Figure 2.6).

All hydrological models extracted from a DEM depend directly on the local or the incorporated influence of the upslope slopes (e.g. in flow accumulation calculations). Hence uncertainty in slope will affect the derived hydrological models (e.g. Oksanen and Sarjakoski, 2005b; Wechsler, 2007). Unless otherwise noted, the DHS method was used for the calculation of slope in our study.

2.3.2 Concavity and Convexity

The parameter describing the concavity and convexity of a surface is also often called curvature. To calculate the curvature, usually second derivatives are involved. Olaya (2009) provides a discussion of various curvature algorithms. In this thesis the convexity is not used for hydrological modeling, instead it is only needed for statistical evaluations of uncertainty. Thus, a simple approximation of the convexity as it was used by Hengl et al. (2010) is sufficient. This approach is based on a 5x5 neighborhood, where the mean value of the 5x5 neighborhood is subtracted from the value of the center cell e. DEM cells located in a convex relief have values larger than zero and DEM cells located in a concave relief have values smaller than zero. The convexity can also have an influence on the accuracy of hydrological parameters derived from DEMs. For example, Hengl et al. (2010) showed that stream extraction is slightly more uncertain in convex areas.

2.3.3 Roughness

Roughness, which is defined by both relief (Z) and spacing (X, Y), can be represented by such measures of spatial intricacy as drainage density and changes in sign of slope aspect or curvature per unit area (Iwahashi and Pike, 2007). It is also known as texture to emphasize its fine versus coarse expression of topographic spacing. There are different ways to calculate the roughness of a surface. The formula used to calculate the roughness in this thesis is the same as used by Iwahashi and Pike (2007). In this approach, roughness is computed by extracting grid cells that outline the distribution of valleys and ridges in the DEM. Differences between an original DEM and a second DEM derived by filtering the original DEM with a median filter can be used to identify these cells. More details can be found in the publication of Iwahashi and Pike (2007). So far, the impact of terrain roughness on DEM derived parameters has not yet been extensively analyzed in literature.

2.4 Hydrological Algorithms

In hydrology, the movement, quality and distribution of water are the main objectives of studies. Primarily, the flow of water is driven by the gravity and to some degree by the material it flows through. The effect of gravity can easily be modeled with DEMs, the effect of the materials is rather difficult to include in hydrological models (Gruber and Peckham, 2009). This section introduces the hydrological algorithms used for the uncertainty analysis in this thesis, starting with the flow routing algorithms, followed by a description of watersheds and stream networks.

2.4.1 Flow Routing Algorithms

To compute common hydrological models on a gridded DEM, the following two steps are usually necessary:

- Determination of flow direction for each cell in the gridded DEM
- Computation of the flow distribution from each cell to its neighboring cells

The flow routing algorithms can be classified into two generic approaches (Figure 2.7):

- *Single Flow Direction* (SFD) algorithms, where the total amount of accumulated area, also called upslope area of a cell, is distributed to only one neighboring cell (O'Callaghan and Mark, 1984).
- *Multi Flow Direction* (MFD) algorithms, where the total amount of accumulated area of a cell can be distributed to multiple neighbors according to a predefined rule (Quinn et al., 1991; Freeman, 1991; Tarboton, 1997).

A critical issue is the assignment of flow directions over flat areas and depressions in the DEMs. Many methods exist to handle this issue. Most of them are removing depressions by increasing the elevation of the cells located in a depression until the sink is filled and then they are routing the flow over the flat area to the next lower cell (e.g. O'Callaghan and Mark, 1984; Tarboton, 1997; Planchon and Darboux, 2002; Wang and Liu, 2006). Other methods, after filling the sinks, create a gradient in the flat areas to allow flow routing (e.g. Grimaldi et al., 2007; Santini et al., 2009). Recently,

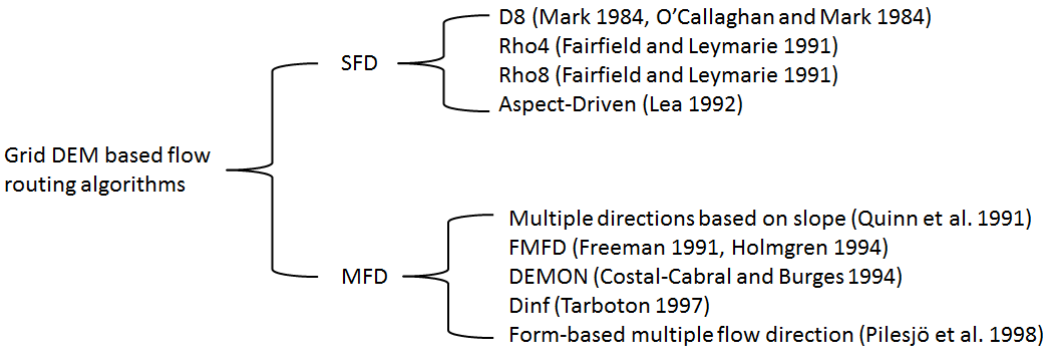


Figure 2.7: Classification of the most common flow routing algorithms. Reprinted from Zhou and Liu (2002).

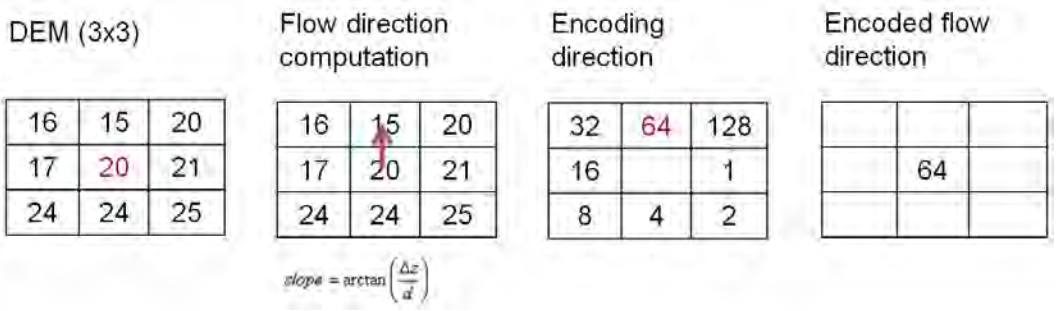


Figure 2.8: Computation steps of the D8 flow routing algorithm.

Magalhães et al. (2012) proposed a new approach which surrounds the terrain with water and then raises the water step by step, filling the depressions when the water reaches their boundary. This approach is significantly faster than previously provided methods.

D8

The D8 flow routing algorithm was introduced in the late 1980's by O'Callaghan and Mark (1984). Since then, the development and use of hydrological land-surface parameters has gained importance. The D8 flow routing algorithm is the most basic one (Gruber and Peckham, 2009). The flow is always passed to the one of the eight neighboring cells with the steepest downslope gradient (Figure 2.8). Hence, the D8 algorithm provides one flow in one of the eight directions and belongs to the single flow direction algorithms. The D8 algorithm is suitable to model convergent flow but will fail to model divergent flow. The D8 algorithm also often tends to produce parallel flow (e.g. Freeman, 1991). However, despite its limitations, the D8 algorithm is useful for many hydrological applications such as extracting stream networks and is implemented in most GIS software packages (Gruber and Peckham, 2009). Because the D8 algorithm assigns only one direction, it should behave very sensitive to DEM uncertainties.

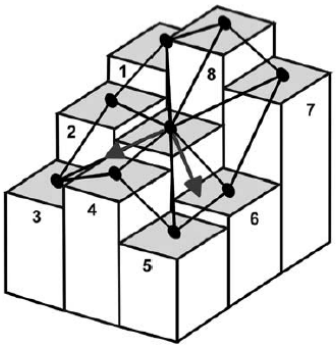


Figure 2.9: 3x3 neighborhood with the middle point m and the triangular facets to its neighboring cells midpoints p_i (Seibert and McGlynn, 2007).

D∞

Tarboton (1997) proposed a multiple flow direction approach where one flow direction is assigned to the cell of interest. In a 3x3 neighborhood, an infinite number of directions varying between 0 and 2π are possible. Eight planar triangular facets are constructed around the middle point M of the center cell (Figure 2.9). The flow direction is defined as the steepest downward slope on planar triangular facets. For each triangular facet the direction and steepest gradient is computed. The normal vector of a triangular facet is defined as (Seibert and McGlynn, 2007):

$$n = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} (z_1 y_2 - z_2 y_1) \\ (z_1 x_2 - z_2 x_1) \\ (y_1 x_2 - y_2 x_1) \end{bmatrix} \tag{2.9}$$

where z_i defines the elevation difference between the midpoint m of the 3x3 neighborhood and the neighboring cells midpoints p_i, and x_i respectively y_i denote the difference in x and y distance. Based on these directions, the divergent flow proportions d_1 and d_2 can be computed as follows (Gruber and Peckham, 2009):

$$d_1 = \frac{4 \cdot \alpha_2}{\pi} \tag{2.10}$$

$$d_2 = \frac{4 \cdot \alpha_1}{\pi} \tag{2.11}$$

The angles α are measured on a horizontal planar surface between the flow direction vector and the vectors from the center cell to the two neighboring cells of the flow direction vector (Figure 2.10). This approach allows to model disperse flow in some way (up to two cells which are neighbors) and should be more robust according to DEM uncertainties compared to the D8 flow routing algorithm, but it cannot model disperse flow of complete different directions (e.g. to cell 6 and 3, see Figure 2.10).

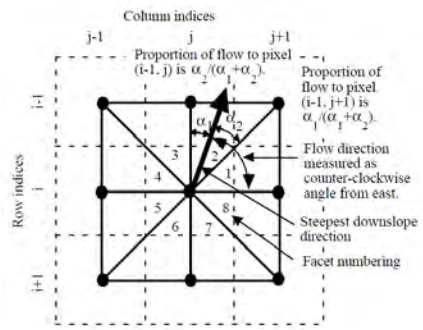


Figure 2.10: The concept of D∞. A 3x3 neighborhood is given and numbered from 1 to 8. The centers are black points and connected. The arrow specifies the flow direction along with its apportioned angles α_1 and α_2. In this case, the flow is distributed between the two cells 2 and 3 (Tarboton, 1997).

MD8

In multiple flow direction algorithms, the cell with the steepest drop, among all neighboring cells, will receive more flow. Quinn et al. (1991) proposed the following equation to model the flow partition:

$$d_i = \frac{(tan\beta_i)^p \cdot L_i}{\sum_{j=1}^{8} (tan\beta_j)^p \cdot L_j} \tag{2.12}$$

The fraction of flow into the i^{th} neighboring cell is defined by d_i. The slope gradient is specified by $tan\beta_i$. The flow partition exponent is p and L_i is the effective contour length. Quinn et al. (1991) proposed that L_i is 0.5 for cells in cardinal directions and 0.354 for cells in diagonal directions. The parameter p determines the amount of divergence. Thus the larger p is, the more similar the MD8 is to a single flow direction approach. This partition exponent p was fixed in analysis over an entire study area. But a fixed partition exponent approach cannot effectively model the impact of local terrain conditions on the dispersion of local flow. Qin et al. (2007) improved this method, by allowing the exponent to adapt to the local terrain conditions. The proposed equation to calculate the partition exponent is as follows:

$$f(e) = 8.9 \cdot min(e, 1) + 1.1 \tag{2.13}$$

where e defines the maximum slope between all neighbors and the center cell. The new MD8 algorithm equation is then expressed as:

$$d_i = \frac{(tan\beta_i)^{f(e)} \cdot L_i}{\sum_{j=1}^{8} (tan\beta_j)^{f(e)} \cdot L_j} \tag{2.14}$$

where $f(e)$ is computed from Equation 2.13.

MD∞

Seibert and McGlynn (2007) proposed a new triangular multiple flow direction algorithm which combines the advantages of the multiple flow direction algorithm MD8 (Quinn et al., 1991) with the triangular facets approach as used in the D∞ algorithm (Tarboton, 1997). First, the determination of the flow directions is needed to compute the portion of accumulated area that is distributed to each of the neighboring cells. This is done with the triangular facets approach proposed by Tarboton (1997). Eight planar triangular facets are constructed and for each triangular facet the direction and steepest gradient is computed (see Tarboton, 1997). The direction d and slope s of the triangular facet can then be computed as follows (Seibert and McGlynn, 2007):

$$d = \begin{cases} 0, & n_x = 0, n_y \geq 0 \\ \pi, & n_x = 0, n_y < 0 \\ \frac{\pi}{2} - arctan\left(\frac{n_y}{n_x}\right), & n_x > 0 \\ \frac{3\pi}{2} - arctan\left(\frac{n_y}{n_x}\right), & n_x < 0 \end{cases} \tag{2.15}$$

$$s = -tan\left(arccos\left(\frac{n_z}{\sqrt{n_x^2 + n_y^2 + n_z^2}}\right)\right) \tag{2.16}$$

The accumulated area of the center cell can now be distributed downslope on the basis of the gradients. As Quinn et al. (1995) have proposed, a flow partition exponent can be defined to adjust the weights of steeper slopes.

The MD∞ is an extension of the D∞ approach proposed by Tarboton (1997). Results often only differ in situations where there is more than one locally steepest downslope direction (e.g. on divergent hillslopes). However, the advantage of the MD∞ algorithm is that unrealistic dispersion on planar or concave hillslopes is avoided, whereas multiple flow directions are allowed on convex hillslopes (Seibert and McGlynn, 2007). Thus, Seibert and McGlynn (2007) have come to the conclusion that the MD∞ algorithm is more appropriate for a range of hydrological applications. Due to its calculation it should also behave more robust according to DEM uncertainties compared to the D∞ method.

2.4.2 Watersheds and Stream Networks

Watersheds are also known as catchment areas or drainage basins and describe an extent of land-surface where water from rain, melting snow or other sources converges to a single point, often called outlet or pour point. The watershed contains many streams that convey the water within the area covered by the basin and channels them to the pour point. Watersheds are separated from adjacent basins by a geographical barrier such as a ridge, hill or mountain. A watershed can be further divided into multiple sub-watersheds. Each link of a stream network channels the water of a specific area to its pour point (the end of the link) and thus defines its sub-watershed (DeBarry, 2004). Stream networks and watersheds can be extracted from a DEM after the D8 flow routing algorithm has been performed. If a cell reaches a certain threshold of accumulated area, it is defined as a part of the stream network and all cells draining to the same pour point belong to the same watershed. Stream networks can also be divided into segments which are hierarchically ordered (Strahler order)

(Strahler, 1952). Each time two streams come together, the order is increased by one. Recently, some Monte Carlo based methods were proposed to analyze the uncertainty of stream network extraction (e.g. Hengl et al., 2010; Achour et al., 2012). They showed that stream networks can vary spatially due to uncertainties in DEMs. Errors were mainly observed in convex areas and for low Strahler orders. The uncertainty of watersheds has also been investigated; Hebeler and Purves (2009) showed that the derivation of watersheds and related statistics vary as a result of uncertainty, whereas Oksanen and Sarjakoski (2005b) stated that the variation of the derived watersheds depends less on the uncertainty model parameters such as the degree of spatial autocorrelation than on constrained derivatives such as slope and aspect.

2.5 Compound Indices

Compound indices use elements derived directly from the DEM (e.g. slope) to generate some physically meaningful property which varies in space (Goodchild et al., 1996). The compound indices discussed in this thesis are the topographic wetness index (TWI), the stream power index (SPI) and the sediment transport index (STI). Many other indices using accumulated flow and related to soil erosion and landslide initiation are proposed and discussed in literature (e.g. Moore and Burch, 1986; Montgomery and W.E., 1994). Moore et al. (1991) and Wilson and Gallant (2000) further provide an overview and a more detailed discussion. All compound indices depend on the accumulated flow and the local slope. The impact of DEM uncertainty on compound indices was investigated. Wechsler (2007) and Temme et al. (2009), for example, provided uncertainty assessments for TWI computations.

2.5.1 Topographic Wetness Index

The TWI was first introduced by Beven and Kirkby (1979). It is a widely used compound index applied in precision agriculture (Qin et al., 2011). The TWI is designed for modeling the spatial distribution of soil moisture and surface saturation and to quantify the effect of local topography on hydrological processes. This index describes the tendency of a cell to accumulate water (Gruber and Peckham, 2009). Areas having similar TWI values are assumed to have a similar hydrological response to rainfall if other environmental conditions are the same (Qin et al., 2011). The TWI is defined as follows:

$$TWI = ln\left(\frac{A_c}{tan\beta}\right) \tag{2.17}$$

where A_c defines the upslope area per unit contour length, also called specific catchment area (SCA) and $tan\beta$ specifies the local slope angle. The TWI is based on a mass-balance consideration where the two parameters *local slope angle* and *drainage contour length* describe the tendency to evacuate water, whereas the total flow accumulation is a parameter of the tendency to receive water. The index is scaled to a more condensed and linear range by the natural algorithm (Gruber and Peckham, 2009).

2.5.2 Stream Power Index

The SPI is a measure for the erosive power of overland flow and related landscape processes. As specific catchment area and slope steepness increase, the amount of water contributed by upslope areas and the velocity of water flow increase. Hence stream power index and erosion risk increase. Stream power index controls potential erosive power of overland flows, thickness of soil horizons, organic matter, pH, silt and sand content, and plant cover distribution (Moore et al., 1993). The SPI is defined as follows:

$$SPI = A_c \cdot tan\beta \tag{2.18}$$

where A_c defines the upslope area per unit contour length and $tan\beta$ specifies the local slope gradient.

2.5.3 Sediment Transport Index

The sediment transport index characterizes the process of erosion and deposition (see Burrough and McDonnell, 1998). The two-dimensional catchment area is used instead of the one-dimensional slope length factor as in the Universal Soil Loss Equation (USLE). The STI equation is given by Burrough and McDonnell (1998):

$$STI = \left(\frac{A_c}{22.13}\right)^{0.6} \cdot \left(\frac{sin\beta}{0.0896}\right)^{1.3} \tag{2.19}$$

where A_c defines the upslope area per unit contour length and $sin\beta$ specifies the local slope gradient.

2.6 Parallel Computation

As a result of improved DEM production techniques such as LiDAR, availability of large high resolution DEMs is increasing rapidly. Vaze et al. (2010) have analyzed the impact of different resolution DEMs on hydrological models and the loss of accuracy and reliability of results as one moves to coarser resolution DEMs, suggesting the use of high resolution DEMs for hydrological modeling. But high resolution DEMs increase the computation time of hydrological models dramatically compared to coarse resolution DEMs. In addition, the Monte Carlo approach used for uncertainty analysis is also very computationally intensive.

The recent development in computer technology and the wide availability of multi-core computers allow to use parallel processing to analyze large DEMs and Monte Carlo based uncertainty analysis in an appropriate time frame. Tesfa et al. (2011) proposed a multi-processing approach to extract hydrological parameters from DEMs using the Messaging Passing Interface (MPI) for the parallelization of the algorithms. MPI is a common approach in scientific applications to distribute the execution of a program over many processors and even computers in high performance computing (HPC) environments. A program's execution task is divided into different parts and each part is

assigned to a separate processor or processor core. Messages are periodically sent between the processes to coordinate the processing task. This allows an efficient and fast completion of the task (Kumar, 2001).

Besides the multi processing approach, the recently emerged application of the graphics processing units (GPUs) for geocomputation algorithms provides a possibility to parallelize the demanding algorithms. The proposed methods (e.g. Ortega and Rueda, 2010; Zhan and Qin, 2011) are based on compute unified device architecture (CUDA) developed by NVIDIA[1] and supported by their graphic cards. In general-purpose computing on graphics processing units (GPGPU) based applications, the GPU computes the task instead of the CPU. In this thesis we stick with the multi-processing approach. For more details about GPGPU readers are referred to Hwu (2011).

2.7 Applications in Hydrology

Many hydrological applications using digital terrain analysis exist. A review of some applications of DEMs in hydrology was provided by Hengl and MacMillan (2009). Unfortunately, it would go beyond the scope of this section to cover every application. Thus, one particular example of a hydropedological application is discussed in more detail.

Seibert et al. (2007) investigated the topographic influence on soil properties in boreal forests. According to them, topography is a major factor controlling hydrological and soil processes at the landscape scale. The relationship between topography and soil characteristics can be measured quantitatively using DEMs and topographic indices computed from these DEMs. To facilitate this, Seibert et al. (2007) used 4000 sample plots distributed over their study site and for each sample plot, they computed the TWI. They found several significant correlations between the TWI and some soil properties. After Seibert et al. (2007), these correlations confirmed the importance of topographic influences on soil properties although they observed a considerable scatter in their data which they attributed to the heterogeneity in the large dataset and the coarse resolution of the DEM (50 m). However, this scatter could also be a result of DEM uncertainty which was not noted in this study.

2.8 Alpthal - The Study Area

The WSL institute[2] (Swiss Federal Institute for Forest, Snow and Landscape Research) has been studying the forest hydrology in Alpthal (Switzerland, Canton Schwyz) since 1968, when Hans Keller initiated the research with the construction of the first hydrological and climatic measurement station. Since then, hydrological input and output data has been continuously collected in this relatively small catchment area. In the beginning, researchers tried to determine the balance between rainfall and runoff in forested watersheds. Later on, the water quality of groundwater, runoff and rainfall were also measured. The study of the transport of sediments in mountain torrents started in the Alpthal in the 1980's. As of today, such long-term studies are very important for the study of qualitative, quantitative and biochemical changes in hydrological systems (Burch, 1994). For more details about the WSL institute, readers are referred to Hegg et al. (2006).

[1] http://www.nvidia.com/object/cuda_home_new.html
[2] http://www.wsl.ch/

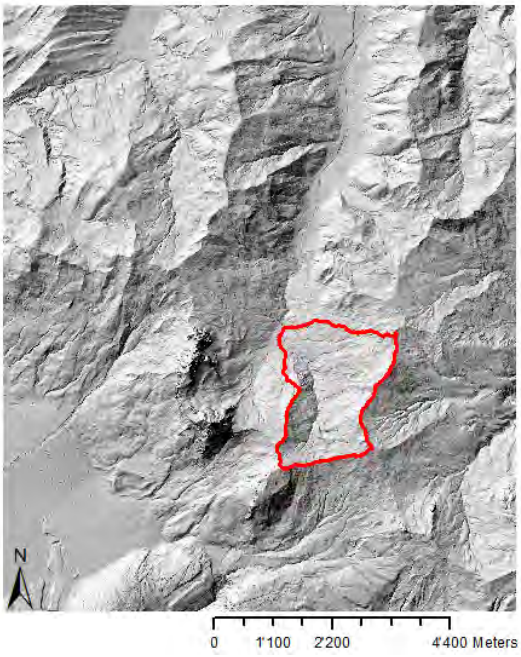


Figure 2.11: LiDAR hillshade of the Alpthal (2 m resolution). The focus of this thesis lies on the study area Zwackentobel marked with a red boundary.

2.8.1 Topography and Climate

The hydrological study areas are located in the southern part of the Alpthal (Figure 2.11). The valley bottom of the averaging 3 to 4 km wide valley lies between 900 and 1100 m a.s.l.. While the valley is well-defined by ridges in the western and eastern part, the Grosse Mythen and the Kleine Mythen build a barrier in the southern and southwestern part and the valley is open to the north. In general the valley is very steep and the ground very wet, leading to strong erosion processes (Burch, 1994). The annual precipitation of 2000 mm is significantly higher than the average observed in Switzerland. This is a result of the location in a basin with a high tendency of heavy rainfall events at the bottom of the Mythen.

2.8.2 The Specific Study Area

The physical geography H2K[3] unit of the University of Zurich has some ongoing monitoring projects in the Zwackentobel of the Alpthal (Figures 2.12 and 2.13). They measure the water throughput and biochemical properties of the water. These measurements are used for research of catchment hydrology and climate impacts on the water cycle.

[3] http://www.geo.uzh.ch/en/units/h2k

Figure 2.12: Orthophoto of the study area Zwackentobel of the Alpthal (2 m resolution).

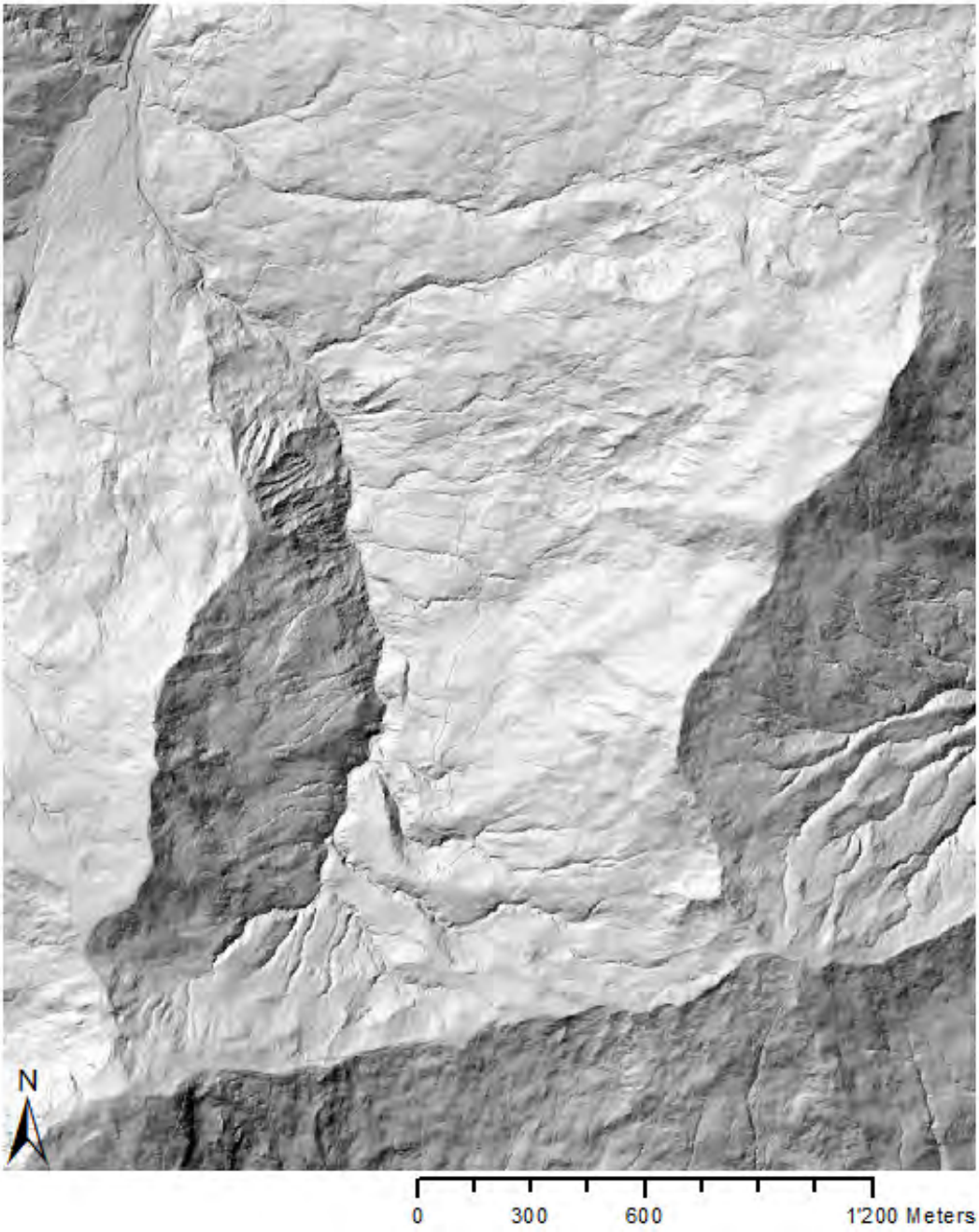


Figure 2.13: Hillshade of the study area Zwackentobel of the Alpthal (2 m resolution).

3 Integrating Uncertainty Analysis in TauDEM

This chapter describes the methods to analyze the uncertainty propagation in digital terrain analysis. Section 3.1 presents the software tools used to achieve the results of this thesis. Section 3.2 introduces TauDEM (terrain analysis using digital elevation models), which provides the basis for the proposed uncertainty analysis approach. TauDEM is a suite of digital terrain analysis tools for the extraction and analysis of hydrological information from topography as represented by a DEM. Section 3.3 provides a detailed report about the DEMs used in this study. Section 3.4 summarizes the requirements which the proposed approach should meet. Section 3.5 presents the approach used to visualize and evaluate the results of the error propagation analysis. The following Section 3.6 provides detailed information about the uncertainty analysis approach with TauDEM. The focus lies on the description of the uncertainty model matching the particular characteristics of LiDAR derived DEMs and the implementation of the most important algorithms.

The method of analysis error propagation is based on the flow chart shown in Figure 3.1. First, the input data is generated in a classical GIS. Then, the uncertainty model produces appropriate error surfaces which are applied to the original input DEM in each Monte Carlo simulation run. On the produced simulated DEMs, various hydrological models can be applied. Each Monte Carlo simulation run results in one realization representing a hydrological product (e.g. TWI). The sum of the realizations can be used to derive several statistical estimators such as the standard deviation describing the uncertainty of the computed product.

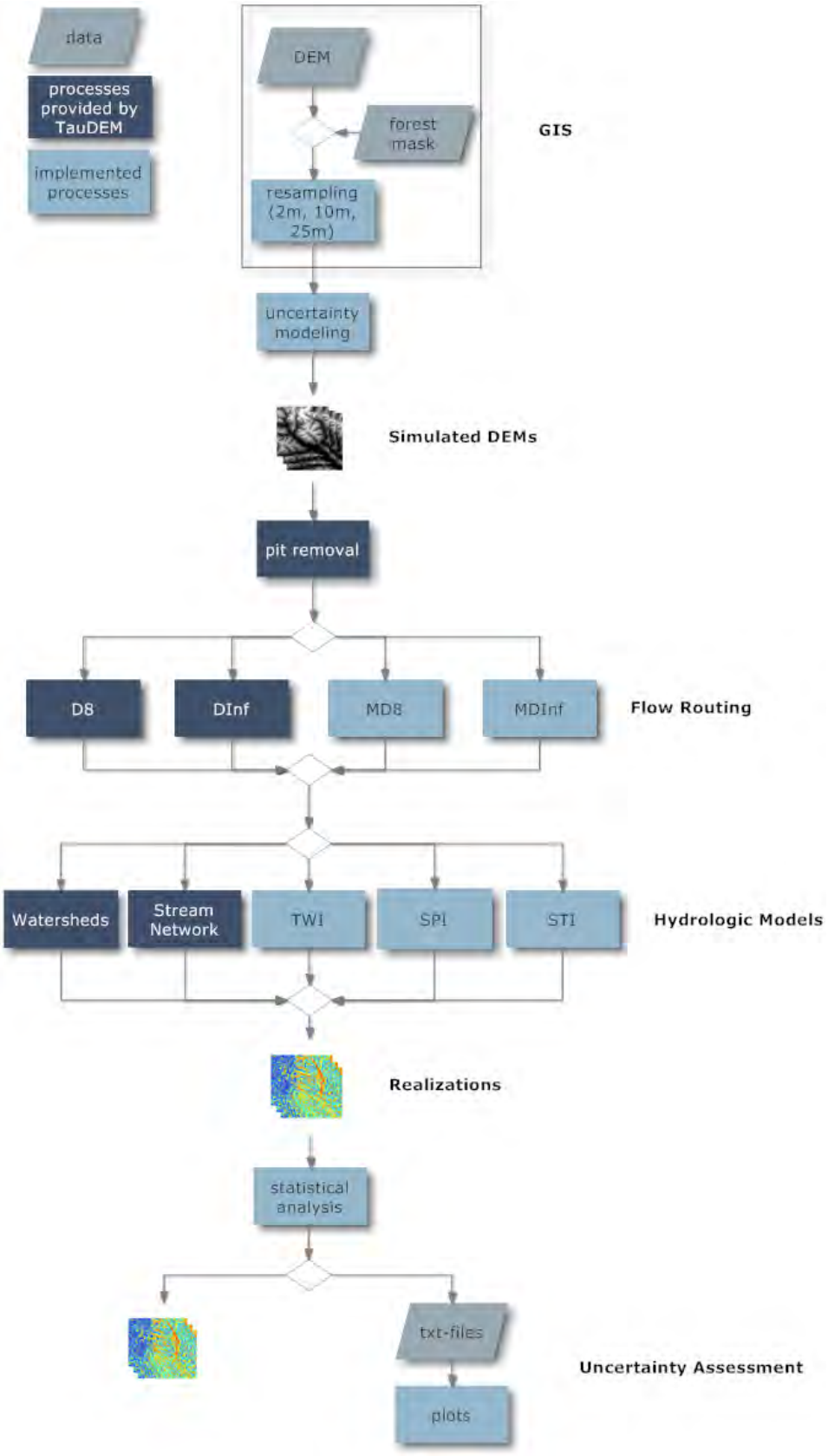


Figure 3.1: Design-overview of the error propagation analysis using the Monte Carlo simulation approach.

3.1 Software Tools

In this thesis a combination of programming, statistical computing, remote sensing, and GIS software is used to assess the DEM uncertainty and its impact on the derived hydrological models. The implementation is based on the programming language C++, because it is a very powerful language when it comes to the implementation of efficient and parallelized applications. Moreover, TauDEM is already written in C++. The programming software used is Microsofts Visual Studio 2010 Premium[1]. As a starting point, the hydrological algorithm suite TauDEM 5.0 with the multi processing interface MPI was used, which are both described in more detail in the following Section 3.2. **R** is the command-based environment for statistical computing and freely available (Team, 2009). **R** is used for plots and histograms of the data and the results of the study. ArcGIS[2] is a commercial GIS software by ESRI. ArcGIS 10 is used to manage and visualize the raster data. Also, the commercial remote sensing software package ERDAS IMAGINE[3] from Intergraph is used to extract a canopy mask for the study site from a multispectral dataset.

3.2 TauDEM - The Starting Point

Developing an uncertainty analysis method for hydrological products from scratch would have consumed too much time. There already are very efficient algorithms related to this topic (e.g. Tesfa et al., 2011). Realizing the uncertainty analysis with common GIS software seemed to be too inefficient. Many provided approaches suffer from performance issues or do not provide the functionality to answer the research questions. The uncertainty analysis approach based on SAGA[4]/RSAGA[5] proposed by Hengl et al. (2010), for example, is limited to very small DEMs and thus less suitable for error propagation with LiDAR data. Although the performance issues may also be related to the SGS approach used for geostatistical simulations. Hence, TauDEM was selected as the basis for the uncertainty analysis approach, because it provides efficient algorithms combined with a powerful parallelization approach. The source code of TauDEM was freely available[6].

TauDEM exploits the power of multi-core processors and multi-processing systems with a MPI based parallelization approach proposed by Tesfa et al. (2011). According to Tesfa et al. (2011), hydrological parameters are initially extracted from DEMs using recursive serial algorithms which are computationally demanding. Especially with large DEM datasets, recursive algorithms can be very memory intensive. Stack overflows may occur rendering the computation of large datasets impossible, because the entire DEM is held in RAM (random-access memory) and the function state is saved on the stack at each recursive iteration. To overcome this limitation, Tesfa et al. (2011) proposed MPI parallel algorithms which are available in the TauDEM suite. This approach divides the spatial domain of the DEM into separate partitions running in separate processes on potentially separate processors. To avoid recursion, a queue based approach is used to compute the grid cells in the appropriate order. TauDEM provides the following features[7]:

[1] http://www.microsoft.com/visualstudio/en-gb/products/2010-editions/premium
[2] http://www.esri.com/software/arcgis
[3] http://geospatial.intergraph.com/products/ERDASIMAGINE/ERDASIMAGINE/Details.aspx
[4] http://www.saga-gis.org/en/index.html
[5] http://cran.r-project.org/web/packages/RSAGA/index.html
[6] http://hydrology.usu.edu/taudem/taudem5.0/downloads.html
[7] http://hydrology.usu.edu/taudem/taudem5.0/

- Development of hydrologically correct (pit removed) DEMs using the flooding approach
- Calculation of flow paths (directions) and slopes
- Calculation of contributing area using single and multiple flow direction methods
- Multiple methods for the delineation of stream networks including topographic form-based methods sensitive to spatially variable drainage density
- Objective methods for determination of the channel network delineation threshold based on stream drops
- Delineation of watersheds and subwatersheds draining to each stream segment and association between watershed and segment attributes to set up hydrological models
- Some specialized functions for terrain analysis

The TauDEM suite runs under Windows and Unix systems (32/64 bit) and is command line based. In order for TauDEM to work properly, the Argonne National Laboratory's MPICH2 library[8] must be installed on the system. This is needed for the MPI based algorithms and is also freely available. In addition, TauDEM works only with the uncompressed GeoTIFF[9] raster format.

3.3 Data

The study area Alpthal has already been introduced in Section 2.8. This section explores the DEM data itself of the study area used in this thesis (Zwackentobel, Figure 2.13). The DEM is generated from LiDAR data and available from Swisstopo[10]. The so called swissALTI3D dataset was last updated in March 2012. It has a 2 m spatial resolution and covers the whole of Switzerland (Swisstopo, 2012). Below 2000 m a.s.l., the vertical accuracy is $\pm$ 0.5 m 1σ and above 2000 m a.s.l. $\pm$ 3 to 8 m 1σ, because above 2000 m a.s.l. the swissALTI3D dataset uses the DHM25 DEM (25 m spatial resolution) which was resampled to a 2 m spatial resolution. The DHM25 was derived from topographic maps in the 1990's. The perimeter for this thesis covers the Zwackentobel, which is in the southern part of the Alpthal. The raster information is given in Table 3.1 and various topographic attributes of the Zwackentobel are shown in Figure B.1.

The elevation of the study site ranges between 1076 and 1656 m with an average and standard deviation of 1380 respectively 130 m. The study area covers an area of approximately 7.9 km^2. Because the elevation does not exceed 2000 m a.s.l., the vertical accuracy indicated by Swisstopo (2012) remains the same for all grid cells ($\pm$ 0.5 m 1σ).

To improve the uncertainty model according to Section 2.1, a canopy mask of the study site is needed. There are multiple ways to get such a mask. Either such data is available, we manually digitize it from an orthophoto in a GIS or we classify the canopy automatically with a remote sensing software from a multispectral dataset. The last approach was used, because a 25 m resolution multispectral Landsat[11] dataset is freely available for the whole of Switzerland from Swisstopo[12]. A quick unsupervised classification of the dataset is adequate enough for a sufficient classification

[8] http://www.mcs.anl.gov/research/projects/mpich2/
[9] http://trac.osgeo.org/geotiff/
[10] http://www.swisstopo.admin.ch/
[11] http://www.landsat.org/
[12] http://www.swisstopo.admin.ch/internet/swisstopo/de/home/products/images/ortho/landsat.html

Property	Value
Columns and Rows	1282, 1540
Cellsize (X, Y)	2, 2
Compression	None
Uncompressed Size	7.53 MB
Format	TIFF
Pixel Type	floating point
Spatial Reference	CH1903+_LV95

Table 3.1: Raster information of the Zwackentobel terrain (Figure 2.13).

of forested areas in the study site (Figure 3.2). Approximately 55% of the study site is covered by forest.

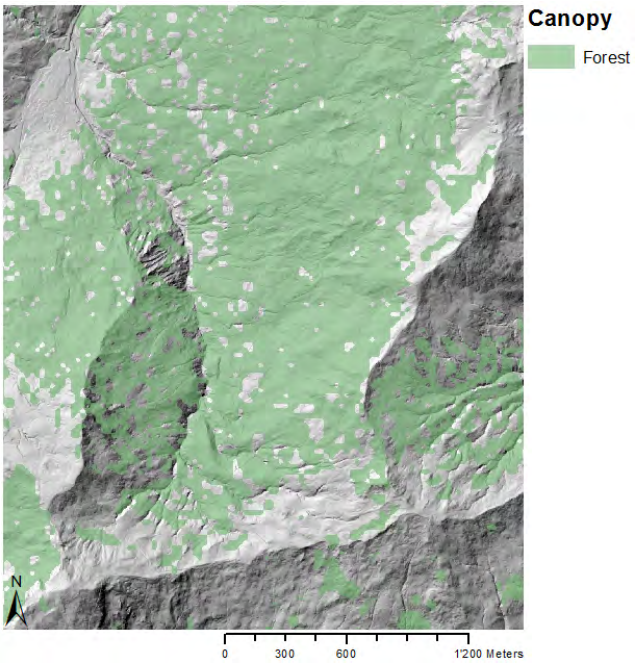


Figure 3.2: Forested areas of the Zwackentobel terrain achieved by a unsupervised classification of a multispectral dataset.

Some histograms and plots of the study site's topographic characteristics are outlined in Figures 3.3 and 3.4. These may help to analyze and interpret the achieved results of this thesis. Figure 3.3(a) shows that the Zwackentobel is relatively steep; approximately 50% of the cells are steeper than 20° and some outliers almost reach the 80° mark. The convexity is centered around zero which means that most grid cells are neither convex nor concave and are located in planar reliefs (Figure 3.3(b)). The roughness of the DEM is approximately normally distributed (Figure 3.3(c)), whereas the elevation shows a typical distribution of an alpine region (Figure 3.3(d)). There are the mountains in the southern, western and eastern part, which are high and steep and the outgoing of the valley in the north, where the terrain is low and flat. This is also illustrated in Figure 3.4(a). Figure 3.4(c) shows that lower areas are rougher, whereas the convexity (Figure 3.4(b)) does not correlate with the elevation. Because roughness and slope seem to be related to the elevation, they could also be related to each other. Figure 3.4(f) shows that flat areas are rougher than steep regions. The forest seems not to be correlated with slope (Figure 3.4(e)), but forest can be found more often in higher terrains (Figure 3.4(d)).

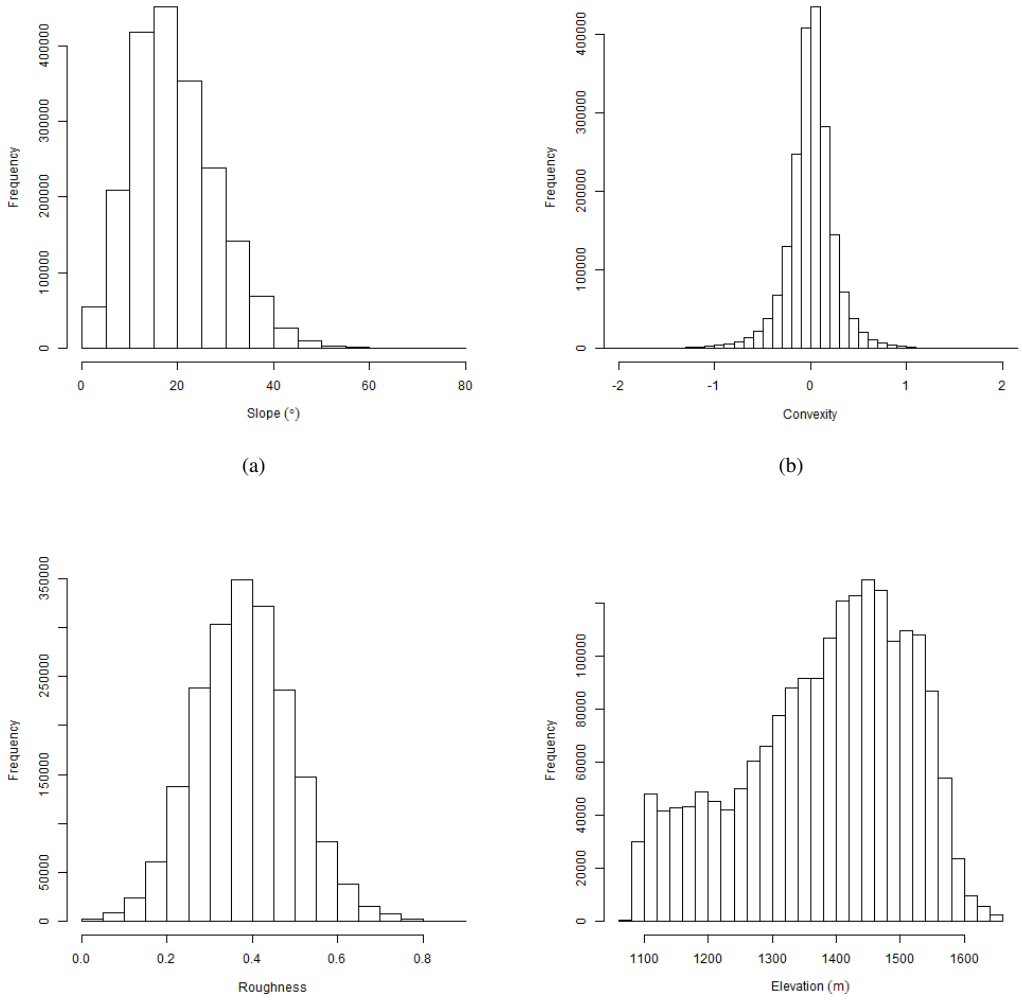


Figure 3.3: Data analysis of the study area Zwackentobel (Figure 2.13) with histograms. Distribution of (a) slope, (b) convexity, (c) roughness and (d) elevation of the Zwackentobel DEM. Statistical measurements are given in Table 3.2.

Topographic Attribute	Mean	STD	Min	Max
Slope (°)	19.4	8.9	0	77.5
Convexity	0	0.2	-5.6	6.3
Roughness	0.4	0.1	0	0.9
Elevation (m)	1380	130	1076	1657

Table 3.2: Statistical measurements of the topographic attribute histograms (Figure 3.3).

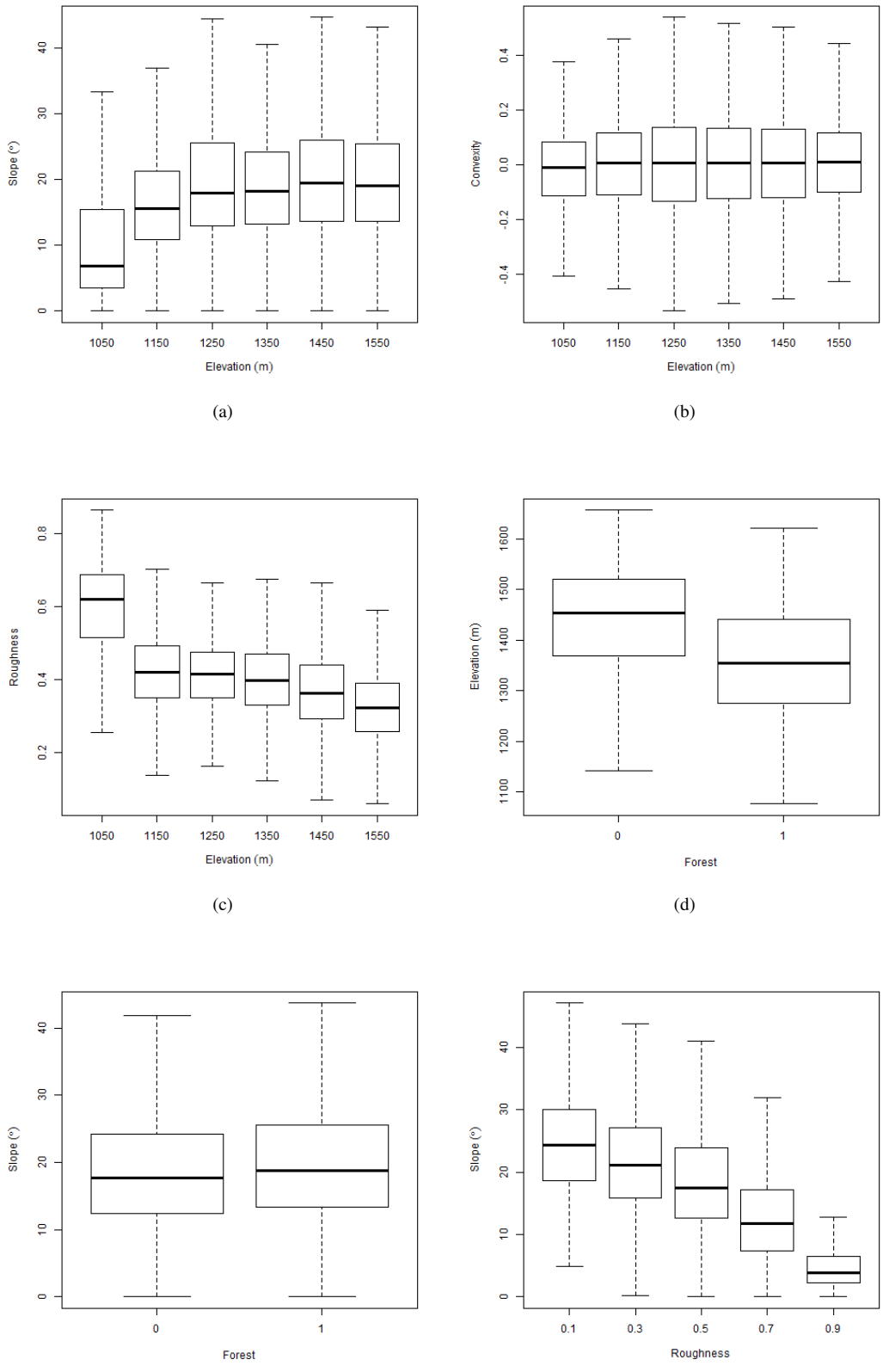


Figure 3.4: Data analysis of the study area Zwackentobel (Figure 2.13) with box plots: (a) Elevation plotted against slope, (b) elevation plotted against convexity, (c) elevation plotted against roughness, (d) forest plotted against elevation, (e) forest plotted against slope and (f) roughness plotted against slope.

3.4 Requirements

The requirements were evaluated in cooperation with the hydrologists of the H2K unit of the University of Zurich. First of all, we decided that the following hydrological models are needed for the proposed uncertainty analysis approach:

- D8 flow routing
- D∞ flow routing
- MD8 flow routing
- MD∞ flow routing
- Watershed
- Stream network
- TWI
- SPI
- STI

Like TauDEM, the proposed method should also be able to be run easily from the console of the operating system. But unlike TauDEM, all the separate algorithms are stringed together to facilitate Monte Carlo simulations. TauDEM makes use of many arguments for its algorithms. Because that would extend the parameter list to an unpractical size, they are trimmed down to the most important arguments. For example, TauDEM provides the possibility to use weighting raster (e.g. for the flow routing algorithms), but the proposed uncertainty analysis approach does not support weighting rasters anymore. But if such functionality should be necessary in the future, it could easily be reintegrated.

As already mentioned, the Monte Carlo simulation was preferred over an analytical approach for the proposed uncertainty analysis approach (see Section 2.2). Thus, an argument defining the numbers of realizations is needed, which should be computed in a Monte Carlo simulation. To model the error surfaces, the process convolution approach is used. The SGS approach needs control points to fit an appropriate (semi)variogram, is more complex to implement, and is computationally more demanding (see Section 2.2). The process convolution approach in uncertainty analysis of DEM derived parameters was also successfully used in literature (e.g. Oksanen, 2006; Wechsler, 2007; Hebeler and Purves, 2009). To model the spatially distributed error of DEMs and especially LiDAR-derived DEMs, additional inputs are necessary. The measurement of DEM error, usually the RMSE or standard deviation is required. In addition, a spatial autocorrelation range to model the spatial autocorrelation of the DEM error and an optional forest mask, which can be used to improve the spatial structure of error according to LiDAR data error characteristics, are needed (see Section 2.1). The individual results of the Monte Carlo simulations are saved in a separate folder and numbered accordingly. These results can be used for example for animations or other analyses. Due to the fact that saving all intermediate results of a Monte Carlo simulation for the statistical evaluation would use way too much memory, the uncertainty estimator statistics are computed iteratively. To analyze the convergence of the hydrological models in the uncertainty analysis, the tool can be used to load text files containing the coordinates of specific grid cells. These coordinates can be read in

the spatial reference format. The method also provides an option to generate exactly the same error surfaces in the same order in different Monte Carlo simulations to allow reasonable comparisons between the different Monte Carlo simulation results. The study analyses the performance of the implementation by measuring the execution time of the algorithms and the Monte Carlo simulation. To enable such evaluations, the implementation is complemented by timestamps allowing to measure execution times for each separate algorithm. Finally, the method provides the user with statistical grids, representing different uncertainty estimator statistics. In addition to the grids, statistics are provided by text files, allowing users to easily analyze the uncertainty assessment from a different perspective. The following Table 3.3 lists all the arguments and their description of the proposed uncertainty assessment approach.

Argument	**Description**	**Optional**
-dem	The name of the DEM input file	
-std	The standard deviation given as float	
-range	The spatial autocorrelation range given in meters	
-vegetationLayer	The name of the canopy mask file	x
-flowDir	ID of the flow routing algorithm	
-product	ID of the product (e.g. of the TWI)	
-runs	The number of runs in the Monte Carlo simulation	
-outlet	The name of the pour point shape file for the products $watershed$ and $stream\ network$	
-writeDems	Boolean if the DEMs of each run should be saved or not (standard: false)	x
-coordFile	The name of the text file containing coordinates	x
-fix	Boolean if the error surfaces should be reproducible (standard: false)	x
-slope	ID of the slope method (ANS/DHS) (standard: ANS)	x
-drop	A fix value for the drop threshold in $watershed$ and $stream\ network$ computation (standard: automatic)	x
-spatial	Boolean if the slope has an impact on the error surface generation or not (standard: true)	x
-writeR	Boolean if the intermediate results of the Monte Carlo simulation should be saved to disk (standard: true)	x

Table 3.3: Argument list of the proposed uncertainty analysis approach.

3.5 Visualizing the Uncertainty

At the end of the Monte Carlo simulation, a map showing the uncertainty for the derived parameters in an adequate way is desired. The uncertainty of the continuous parameters are measured by the relative and absolute standard deviation of each raster cell. The relative standard deviation (percentage difference) is useful to describe the variation of error, especially for products having a high value range (e.g. flow accumulation), so that uncertainties are also identified in areas having low mean values. However, if the mean values are close to zero, then even a small uncertainty may lead to an extremely large relative standard deviation. This is a disadvantage when using the relative standard deviation to visualize uncertainty. Besides the raster representations, all statistics retrieved by the error propagation analysis are written into text files. Although statistics tools like **R**[13] offer the possibility to read raster files, this approach allows further investigations of the data in many statistics tools without any restrictions. The statistics written to text files and rasters are presented in Table 3.4. The input text file containing coordinates provides further options to analyze the error propagation. The results of these grid cells are successively written to a separate text file. In addition, the mean is successively calculated for each grid cell given in the coordinate text file. This information allows to analyze the distribution of the estimator statistics for each of these cells and to investigate how the derived parameters converge with an increasing number of simulation runs.

For the boolean parameters *watershed* and *stream network* (e.g. a cell is a stream or not), the representation of uncertainty is adjusted. To use the standard deviation as an indicator of uncertainty does not make sense any more with boolean parameters. Instead the attribute probability is used to show with what uncertainty a cell belongs to a watershed or a stream. The probability Pr is nothing else than the mean value. To visualize the uncertainty, the Shannon entropy information as proposed by Hengl et al. (2010) is used for stream networks. This approach transforms the probability value Pr into a new dimension H, representing the uncertainty (Figure 3.5). A cell with a probability of 50% (value = 0.5) to represent a stream will, for example, get the new value 1 which is the maximum error in the dimension H. As for the continuous parameters, all statistics are also written to a text file for further analyses.

The further analyses are done in **R**. The text files are read into vectors and these are plotted against each other. To evaluate the impact of the topographic properties on the uncertainty, uncertainty estimator statistics are plotted against the topographic parameters *slope*, *convexity* and *roughness*. In addition to these parameters, the DEM is classified with the terrain classification algorithm proposed by Iwahashi and Pike (2007). This algorithm classifies the topography in up to 16 different classes. These nominal classes are built by the parameters *slope*, *convexity* and *texture (roughness)* and can also be used to discover patterns in uncertainty analysis.

[13] http://www.r-project.org/

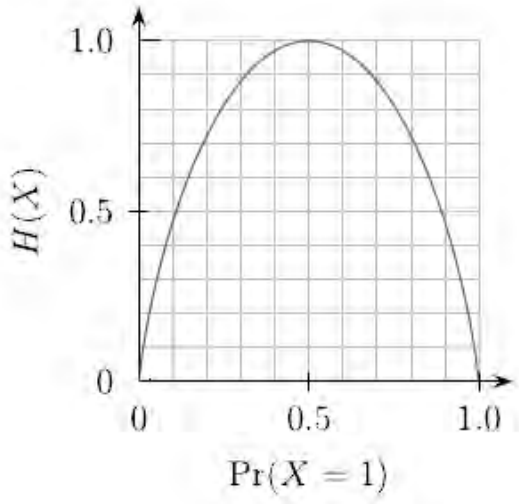


Figure 3.5: Describing the uncertainty of boolean parameters using the information entropy (H) of a Bernoulli trial (Hengl et al., 2010).

Estimator	**Formula**
Mean	$Mean = \frac{\sum_{i=1}^{n} z_i}{n}$
Standard Deviation (std)	$STD = \sqrt{\frac{\sum_{i=1}^{n} [z_i - \bar{z}]^2}{n}}$
Relative Standard Deviation (rstd)	$RSTD = \lvert \frac{STD}{Mean} \rvert$
Shannon Entropy	$H(s) = -p(s) \cdot log(p(s)) - [1 - p(s)] \cdot log(1 - p(s))$

Table 3.4: Uncertainty estimator statistics to quantify the uncertainty of hydrological models, where n is the number of simulation runs, z_i refers to the estimator of parameter z and $\bar{z}$ is the average of the n z_i values. Shannon entropy is used for boolean hydrological models where p is the probability of a grid cell being part of the boolean model estimated by the number of times the model sees the cell as a part of the result, divided by n.

3.6 Implementation

This section presents a detailed insight into the implementation of the uncertainty analysis method proposed in this thesis. The first part of the method is the implementation of a proper uncertainty model. The decisions are based on the fact that the input data is a high resolution LiDAR-derived DEM without any additional information about the error besides the global uncertainty estimator RMSE indicated by the company who has produced the DEM. The uncertainty model is improved by minor tweaks such as the per grid cell adjustment of error depending on the local terrain characteristics. Then, a process convolution approach is used to apply the spatial autocorrelation onto the error surface which is added to the original DEM afterwards. The complete process is illustrated in Figure 3.6. Finally, this section also describes some implemented algorithms in more detail with a closer look at the parallelization approach.

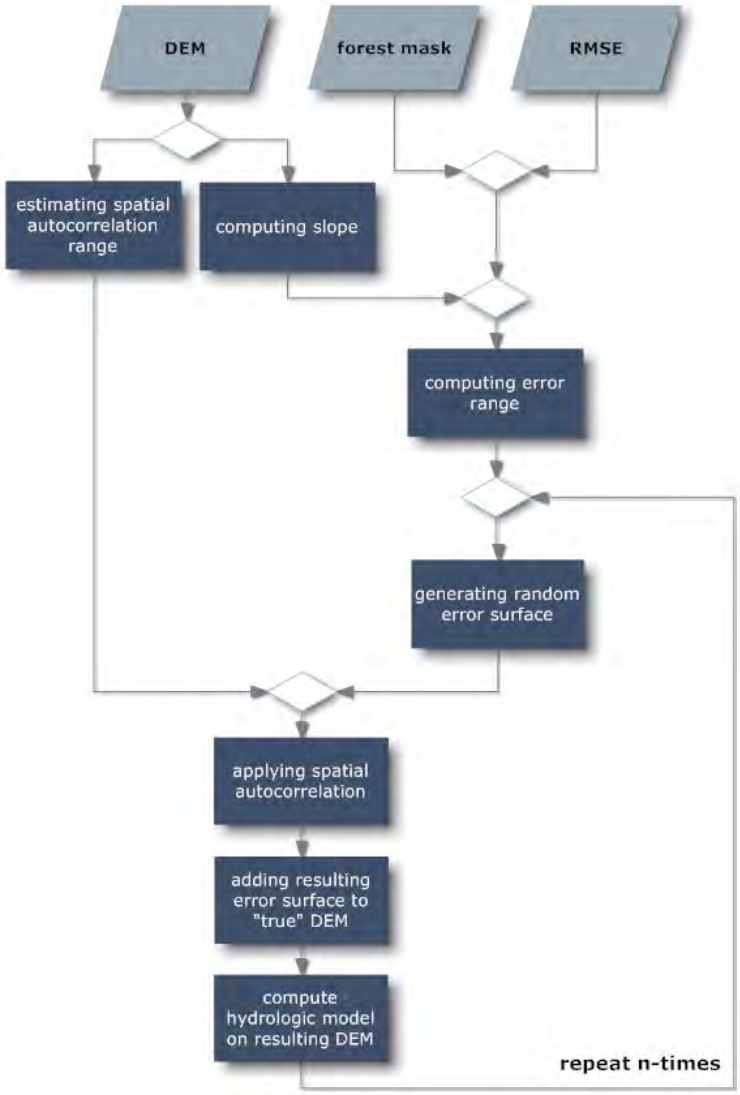


Figure 3.6: Design-overview of the uncertainty model.

3.6.1 Developing an Uncertainty Model

The circumstance that control points are often not available makes a determination of the spatial structure of error very difficult. In this thesis, only one global uncertainty estimator (RMSE) was provided, describing the error of the DEM. Wechsler (2007) even suggested that the RMSE provided to users is not an accurate assessment of how well a cell in the DEM corresponds to its true elevation; it only provides an assessment of how well the DEM represents the data from which it was generated. This error includes artifacts, a random and a systematic error part (see Section 2.1). This approach does not include artifacts, they must be removed before any processing takes place. The same applies to systematic errors; random error fields have a mean of zero. Including a systematic error in the model would change the mean to a positive or negative value and all realizations would show the same trend. But some systematic errors can be transformed into random errors. We know that there are spatially dependent systematic errors in DEMs which may be corrected in the DEM generation process. If these have been corrected or how well the correction has worked is unknown and thus is a form of uncertainty. This information can be used to model some spatial structure of error. To model them, the process convolution process is used. The process convolution method only needs two input parameters:

- RMSE or std - usually available for DEMs
- Spatial autocorrelation range - usually not available for DEMs

Usually information about the spatial autocorrelation of the DEM error is missing. We only know that errors are spatially correlated (see Section 2.2). Unfortunately, this parameter can have significant impact on the results of the uncertainty analysis (see Oksanen and Sarjakoski, 2005b). The spatial autocorrelation range R basically defines the size of the kernel used in the process convolution process. The larger R, the smoother the resulting error surface, which leads to less spatial variation in elevation in the resulting DEM. Due to this, the uncertainty assessed will be lower with an increasing R.

Spatial Structure of Error

In this study, a DEM derived from LiDAR data was used. This allows the assumption of some spatial structures of the error in the DEM. As discussed in Section 2.1, the vertical error in LiDAR based DEMs depends on the slope and the canopy density and height. Figure 3.7 illustrates how the vertical error increases with higher incident angles (θ). The impact of slope (θ) on the vertical error e can be formulated as follows:

$$e = tan(\theta) \cdot Horizontal\ Displacement \tag{3.1}$$

For a constant error in the horizontal location of an observation, the elevation error will increase as the slope of the surface increases. The horizontal displacement is equal to the vertical error in the Swisstopo dataset. In Monte Carlo simulations, errors are assumed to be distributed according to the joint probability distribution of the uncertain inputs. In this case, the horizontal error is assumed to be normally distributed and thus, the vertical error will be calculated with an averaged horizontal error. The mean of the horizontal displacement is $Horizontal\ Displacement/2$. The equation is adjusted as follows:

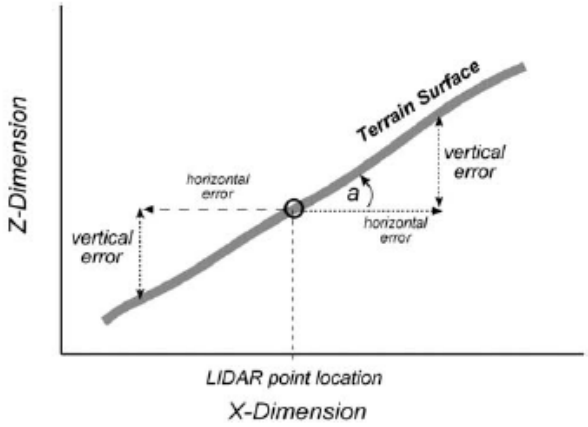


Figure 3.7: Geometrical situation for the slope related source of vertical error in LiDAR data (Hodgson et al., 2005).

$$e = tan(\theta) \cdot Horizontal\ Displacement/2 \tag{3.2}$$

With a horizontal displacement of 0.5 m, e can range from 0 m for θ = 0° to 14 m for θ = 89°. Beyond 85°, the error increases significantly and reaches very high values. When comparing these results to studies investigating the vertical accuracy of LiDAR derived DEMs (e.g. Bater and Coops, 2009; Andersen et al., 2005; Su and Bork, 2006; Tinkham et al., 2012), this formula appears to be appropriate for estimating the vertical error caused by slopes. Usually DEMs have slopes lower than 80° as it is the case for the DEM used in this thesis. Nevertheless, if slopes should be higher than 80°, extreme elevation variations are the result in these steep areas. However, Bater and Coops (2009) showed that errors can indeed reach up to 7.5 m with a RMSE of 0.25 m. The approach proposed in this thesis is similar to the one proposed by Ulmer (2007). He used the footprint size instead of the horizontal displacement. Because the footprint size and the horizontal displacement are related to each other, the results basically remain the same. However, information about the size of the footprint is often missing and very difficult to guess.

To estimate the error in areas covered by dense canopy, control points could be used. But for this thesis, they were not available. Thus, the additional error occurring in areas covered by dense canopy must be estimated. First of all, the surface is classified only into one canopy class, the forest. Otherwise, the classification gets very time-consuming, even with an automatic classification, and on the other hand, information about the uncertainty for every canopy class would be needed. In literature, it was found that the accuracy of elevation in LiDAR based DEMs depends on the land cover type (e.g. Kraus and Pfeifer, 1998; Adams and Chandler, 2002; Andersen et al., 2005; Hodgson and Bresnahan, 2004; Hodgson et al., 2005). It is to be expected that in dense vegetation the laser does often not reach the ground anymore and that therefore elevations are generally overestimated. This would be a systematic error, that cannot be modeled. But studies showed that the elevation in canopy areas can be overestimated (Kraus and Pfeifer, 1998; Andersen et al., 2005) or underestimated (Adams and Chandler, 2002; Hodgson and Bresnahan, 2004; Hodgson et al., 2005). According to these studies, reasons for these biases could have been characteristics of the datasets used or vertical errors in the GPS or INS of the LiDAR system. In general, absolute errors reported remain relatively small even in dense canopy, but significantly higher than in canopy free areas. Figure 3.8 shows the distribution

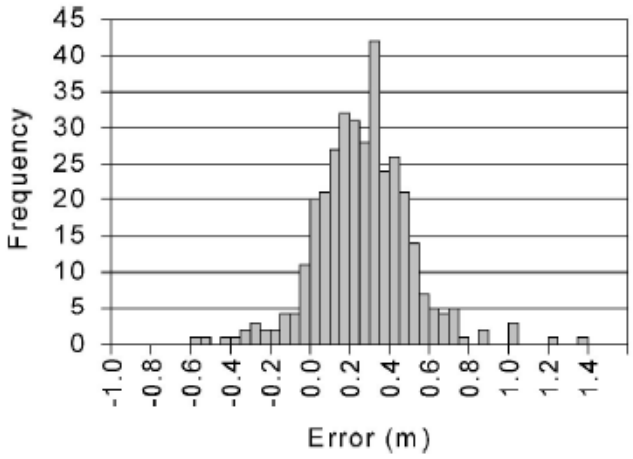


Figure 3.8: Distribution of errors in forested terrain representing the LiDAR accuracy assessment study from Andersen et al. (2005).

of errors obtained in the study of Andersen et al. (2005) using terrain data with various canopy cover types. According to the errors reported in the mentioned literature, the modeled error on grid cells located in forested areas is increased by 50%.

Error Surface

With the informations about the spatial structure of the error an improved uncertainty model can be built. In a first step the random field is generated. This is easily done by visiting each cell of the grid and generating a normally distributed pseudo random number for these cells. Only the random errors are modeled, thus the mean of the normal distribution is zero and the standard deviation is given by the RMSE. The normally distributed pseudo random numbers are generated using the polar method (Marsaglia and Bray, 1964). The value v of a cell for the random field is defined as follows:

$$v = a1 \cdot \sqrt{-2 \cdot log(q)/q} \cdot std \tag{3.3}$$

with:

$$a1 = 2 \cdot randomFloat1 - 1 \tag{3.4}$$

$$a2 = 2 \cdot randomFloat2 - 1 \tag{3.5}$$

$$q = a1 \cdot a1 + a2 \cdot a2 \tag{3.6}$$

where $randomFloat_i$ are pseudo random numbers generated from a typically pseudo random generator and std is the standard deviation (or RMSE). To include the spatial structure of error, the v value is adjusted. In the case of slope, a certain amount of additional vertical error is added to the std depending on the local slope and the horizontal displacement (Equation 3.2):

$$v_{canopy=false} = a1 \cdot \sqrt{-2 \cdot log(q)/q} \cdot (std + tan(\theta) \cdot Horizontal\ Displacement) \tag{3.7}$$

And if the grid cell represents a forested area (this is checked with the canopy mask), the total error is increased by a factor of 1.5, adding the additional error in forested areas:

$$v_{canopy=true} = a1 \cdot \sqrt{-2 \cdot log(q)/q} \cdot (std + tan(\theta) \cdot Horizontal\ Displacement) \cdot 1.5 \quad (3.8)$$

Process Convolution

The spatial autocorrelation could be omitted for a worst case scenario. The error surface is very rough because the error is not spatially autocorrelated, resulting in a very rough topography of the DEM used to compute the hydrological models. But usually errors are spatially autocorrelated (see Section 2.1). The process convolution approach is used to apply the spatial autocorrelation to the error surfaces. Until now, a random field with Gaussian noise was produced. This random field can be spatially autocorrelated with various filters. An overview of different spatial moving average filters was given by Wechsler (2007). An exponential correlation function was chosen to define the weights of the filter according to Abrahamsen (1997):

$$\rho_{exp}(\tau; R, v) = e^{-3(\tau/R)^v} \quad (3.9)$$

where τ is the lag (distance from center cell) and R the correlation range. The scaling factor -3 is chosen so that the correlation at the correlation length is $e^{-3} \approx 0.05$. This equation was also successfully used by Oksanen and Sarjakoski (2005a) in the process convolution based uncertainty analysis of drainage basin delineations. With $v = 2$, the function is a Gaussian correlation function. The spatial moving average window visits every cell and has to compute as many operations as the moving average window contains cells. With a correlation range R of 40 cells, the moving average windows size is 80x80 and thus 1600 operations must be computed to spatially autocorrelate one grid cell. This must be done for all cells of the grid and is computationally intensive. But with a Gaussian correlation function, this two-dimensional filter can be split into two one-dimensional filters, reducing the time needed for the spatial autocorrelation computation significantly. With a correlation range R of 40 cells, the one-dimensional filter only contains 80 cells, thus per grid cell only 80 operations are needed. The error surface has to be filtered two times, one pass with the moving average window in horizontal and one in vertical direction, resulting in a total of 160 operations to spatially autocorrelate one cell. This leads to the exact same result as with a two-dimensional Gaussian filter, but with a runtime of $O(R)$ instead of $O(R^2)$. This is the reason why Gaussian correlation function was chosen.

Now, the error varies over space and because of its positive adjustments in steeper and forested areas, the standard deviation of the error surface is no longer 0.5 m, which was indicated by Swisstopo for the dataset used in this thesis. This can easily be corrected by applying a factor to all error values of the error surface. First, the new standard deviation of the error map is calculated. Second, the old standard deviation is divided by the new one and finally, all error values are multiplied by the resulting factor. With this step, the standard deviation of the error surface is once again 0.5 m. Whereas the error is higher in steeper and forested areas and lower in flat and canopy free areas, the mean always remains at 0 m for the error surfaces. Figure 3.9 shows a histogram of an error surface generated by this approach.

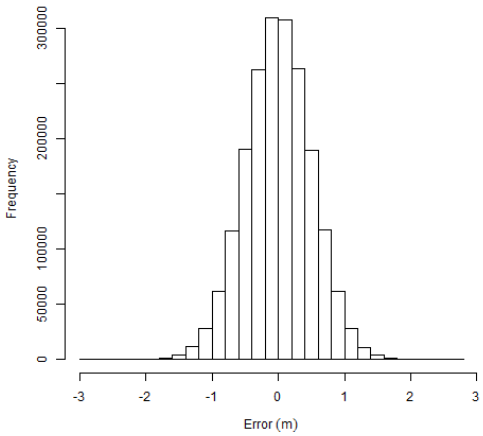


Figure 3.9: Histogram of an error surface (std = 0.5 m, mean = 0 m, spatial autocorrelation range = 40 m).

The spatial autocorrelation range R may have significant influence on the uncertainty analysis (see Oksanen and Sarjakoski, 2005a). Because no information about the spatial autocorrelation range of the error in the dataset is available, it has to be estimated. As mentioned previously, the impact of various parameters on the results of the uncertainty analysis is analyzed too, hence the analysis is performed with different ranges (20 m, 40 m and 80 m) with the 2 m spatial resolution datasets. The chosen ranges are partially based on literature and on own observations; Joerg et al. (2012) found a spatial autocorrelation of the error up to 60 m with a cell size of 1 m (LiDAR), Ulmer (2007) used a guessed range of 80 m with a spatial resolution of 2 m (LiDAR) and Wechsler (2007) used a literature based range of 270 m with a cell size of 30 m (USGS[14] DEM). Oksanen and Sarjakoski (2005a) chose a range parameter of 20 m, which was justified by the assumption that the range should be evidently smaller than the size of the details of the DEM (10 m spatial resolution). They also stated that an increase of the spatial autocorrelation range from 20 to 50 m has only a minor influence on the analysis results. The surface of the DEM gets visually significantly rougher if a range of 20 m and below is used compared to the original DEM (Figure 3.11). Figures 3.10 and 3.11 illustrate some examples of error surfaces and the resulting DEM surfaces with different values for R.

In the final part, the computed error surface is added to the true DEM. The resulting DEM is then used as input data for all following algorithms until the realization is completed in the Monte Carlo simulation.

[14] http://www.usgs.gov/

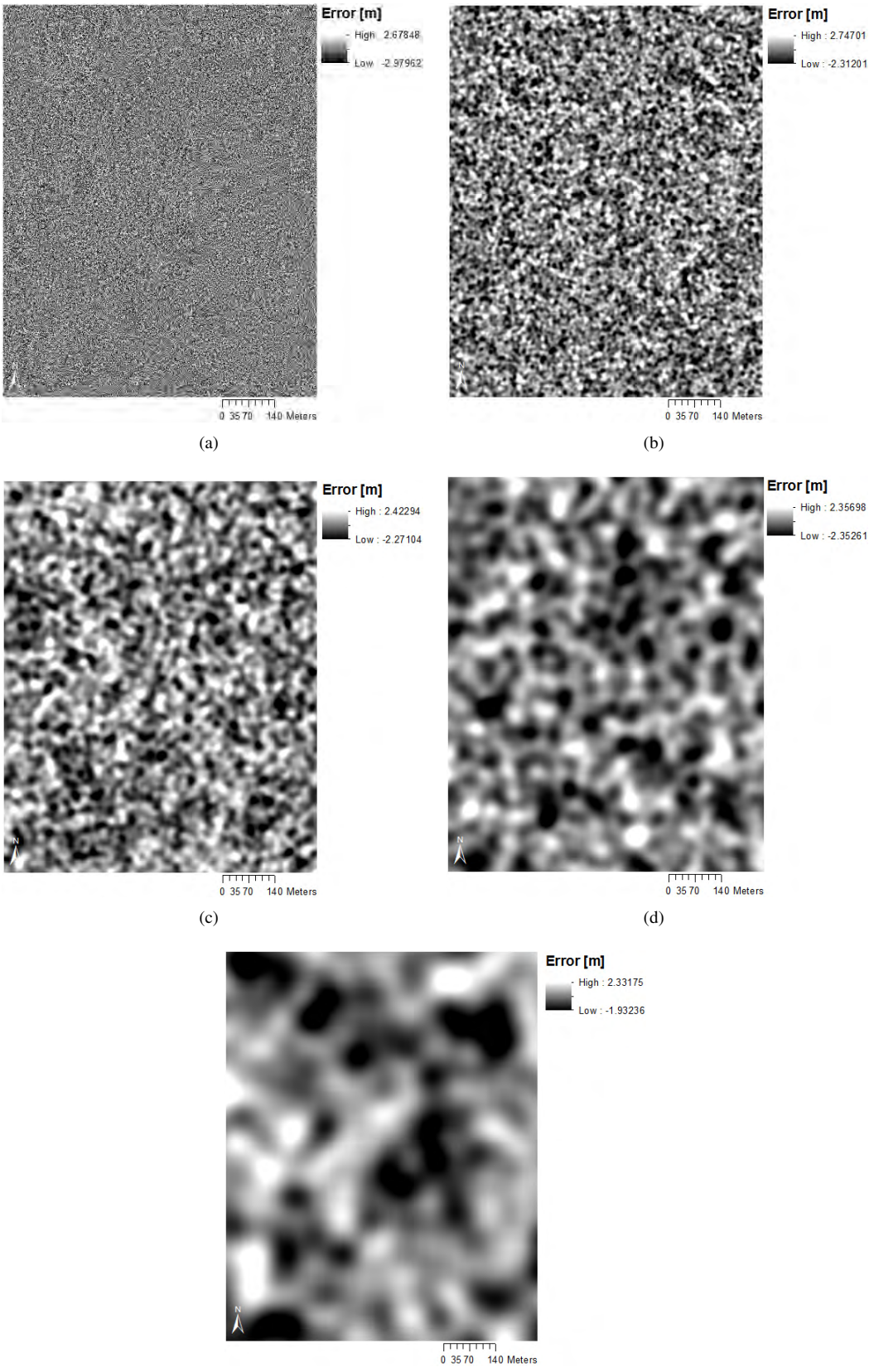
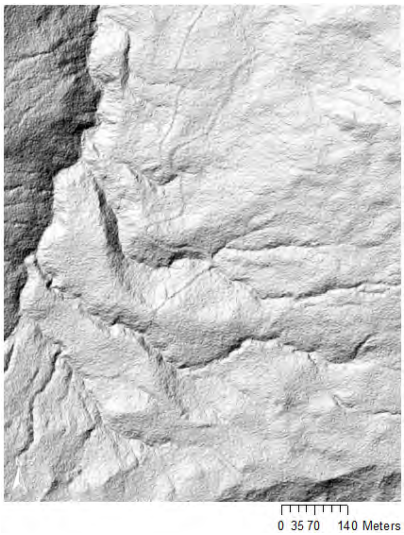


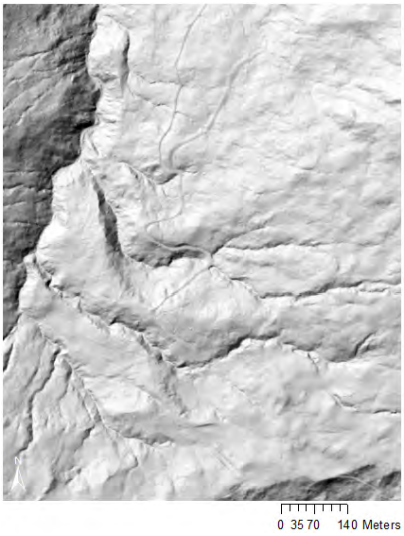
Figure 3.10: Error surfaces: (a) $R = 0$ m, (b) $R = 10$ m, (c) $R = 20$ m, (d) $R = 40$ m, (e) $R = 80$ m. Mean and standard deviation are 0 respectively 0.5 m.

(a)

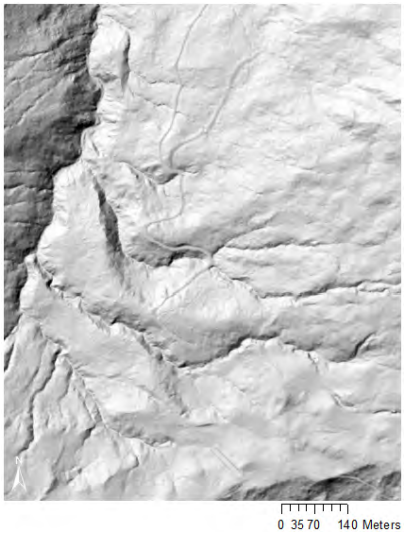


(b)

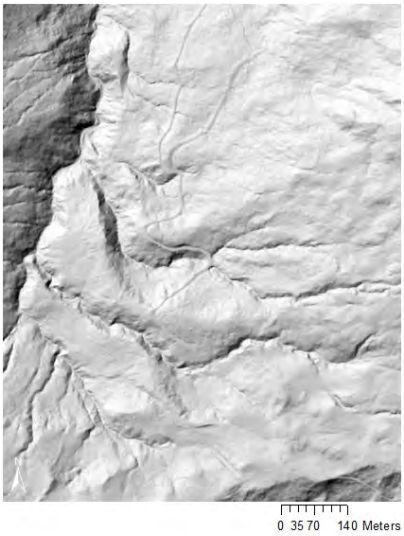
(c)



(d)



(e)



(f)

Figure 3.11: Hillshadings of simulated DEMs after the error surfaces generated with different spatial autocorrelation ranges R have been applied to the original DEM: (a) original DEM, (b) R = 0 m, (c) R = 10 m, (d) R = 20 m, (e) R = 40 m, (f) R = 80 m.

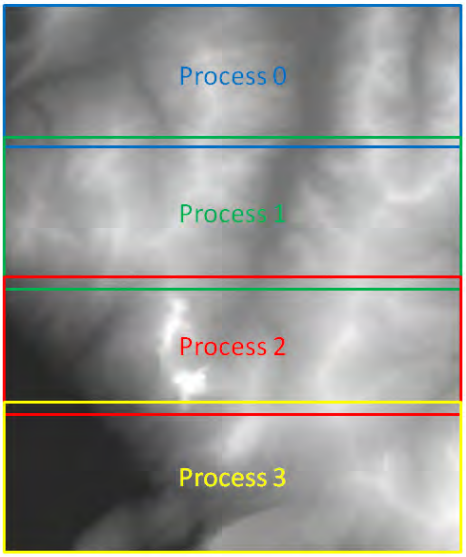


Figure 3.12: Parallelization approach: The DEM is sliced into equal sized horizontal parts and each part is assigned to a separate process. In this example 4 processes are used. The processes share data with row based buffers at the borders of the partitions.

3.6.2 Implemented Algorithms

The basic idea of the proposed uncertainty analysis method is to extend the TauDEM suite by an uncertainty analysis component. All algorithms provided by TauDEM are already parallelized. Due to this fact, all algorithms implemented to enable an uncertainty analysis have to be parallelized too. This approach allows to run a Monte Carlo simulation in parallel with only one dataset in RAM at a time, saving a lot of memory space. Distributing complete realizations along separate processes is less efficient regarding the memory usage. In this case, all processes would have to keep the complete dataset in memory. In addition, this approach also allows to run only one realization in parallel if an uncertainty analysis is not requested.

Like TauDEM, all implemented algorithms use MPI and the distributed memory paradigm for the parallelization. The input DEM is split into n row oriented slices (partitions), where n represents the number of processes (Figure 3.12). Actually, there is no limit on how many processes the algorithms can run. Each process only has to compute the algorithms for the part of the DEM it has been assigned. For many algorithms neighborhood information is required. To facilitate a fast and simple communication basis between neighboring processes, each process includes one buffer row on either side. The buffers include the grid information of the neighboring processes. The spatial autocorrelation algorithm (Gaussian convolution) needs more than only the adjacent neighbors of the center cell. Depending on the spatial autocorrelation range, the buffer can be extended to as many rows as needed.

The generation of the error surface is simple. Each grid cell is visited once and the error is calculated depending on slope, canopy cover and the RMSE value. The parallelization is also very simple; each process calculates the error values for each cell in its partition. The data used is a slope, canopy and error partition ($slope$, $vegPartition$, $errorPartition$). The optional canopy partition is a boolean grid; a grid cell represents forest (value $\neq$ Null) or not (value = Null). For each grid cell, the slope partition holds the slope. The error partition is a empty partition, where the results are saved. The

implementation with C++ (pseudo code) is illustrated in Code 3.1. The final error e of a grid cell can be simplified and formulated as follows:

$$e = error(slope) + error(canopy) + random\ noise\ error \tag{3.10}$$

Code 3.1: Error surface algorithm.

```
sd is defined by the RMSE given by the user
for each second cell c in the partition {
    //generate two normally distributed random numbers
    do{
      randomFloat1=rand()/RAND_MAX
      randomFloat2=rand()/RAND_MAX
      number1=2.0*randomFloat1-1.0
      number2=2.0*randomFloat2-1.0
      q=number1*number1+number2*number2
    } while (q<=0.0 || q>1.0)
    p=sqrt(-2.0*log(q)/q)
    //increase error according to slope
    if error should be increased according to slope {
      slope->getData(c,angle)
      error1=number1*p*(sd+tanf(angle)*sd/2.0)
      slope->getData(c+1,angle)
      error2=number2*p*(sd+tanf(angle)*sd/2.0)
    } else {
      error1=number1*p*sd
      error2=number2*p*sd
    }
    //increase error if cell is covered by forest
    if forest mask is available {
      if(!vegPartition->isNodata(c)){
        error1*=1.5
      }
      if(!vegPartition->isNodata(c+1)){
        error2*=1.5
      }
    }
    errorPartition->setData(c, error1)
    errorPartition->setData(c+1, error2)
  }
}
```

0.05	0.18	0.47	0.82	1	0.82	0.47	0.18	0.05

Table 3.5: Gaussian convolution kernel with a spatial autocorrelation range of 4 grid cells.

The process convolution is realized with a one-dimensional Gaussian kernel. The size of kernel depends on the spatial autocorrelation range ($range$). For each cell in the kernel the lag h (distance to the center cell) is computed to define the resulting weight according to Equation 3.9. By the end, the kernel is a one-dimensional array containing weights which are mirrored at the center cell of the array. An example of a kernel is illustrated in Table 3.5 and the pseudo code for the kernel is given in Code 3.2.

Code 3.2: Gaussian convolution kernel algorithm.

```
range is the spatial autocorrelation range given by the user
int resolution=demPartition->getdx()
int middlePoint=range/resolution
float *kernel=new float[middlePoint*2+1]
float weight, h
float sumWeights=0
for(int i=0;i<middlePoint*2+1;i++){
  h=abs(i-middlePoint)*resolution
  weight=exp(-3.0*(h*h)/(range*range))
  kernel[i]=weight
  sumWeights+=weight
}
```

The error surface is filtered by this kernel. This is done in two passes. The first pass filters the error surface partition ($demPartition$) in horizontal direction. Each cell is visited once and the kernel is applied to this location which means that the values of the neighbors in horizontal direction are multiplied by the weights contained in the kernel and summed up. Dividing this sum (sum) by the sum of all weights ($sumWeights$ or $tempW$) contained in the kernel results in the new and filtered value for the center cell. These values are saved in a new partition ($tempPartition$). This is repeated in the second pass, but with the kernel rotated by 90° and the new partition ($tempPartition$) as input data. In this case information of the error surface of the neighboring processes is needed. This is possible, because the buffers of the processes whose size depends on the autocorrelation range have been filled with this information after the first pass ($tempPartition- > share()$). The spatially autocorrelated error surface is written to the partition ($acPartition$) holding the result, a spatially autocorrelated error surface. This process is outlined as pseudo code in Code 3.3.

Code 3.3: Process convolution algorithm.

```
tempPartition //for saving the results from pass 1
acPartition //for saving the results from pass 2
float demValue, kernelValue
float sum=0
float tempW=sumWeights
int kernelLength=middlePoint*2+1
// pass 1
for each cell c (x,y) in the partition
    if kernel exceeds left/right partition borders {
      for(int k=0;k<kernelLength;k++){
```

```
11            if(demPartition->isNodata(x-middlePoint+k,y)){
12              tempW-=kernel[k]
13            } else {
14              demPartition->getData(x-middlePoint+k,y,demValue)
15              kernelValue=kernel[k]
16              sum+=demValue*kernelValue
17            }
18          }
19          tempPartition->setData(c, sum/tempW)
20          tempW=sumWeights
21        } else {
22          for(int k=0;k<kernelLength;k++){
23            demPartition->getData(x-middlePoint+k,y,demValue)
24            kernelValue=kernel[k]
25            sum+=demValue*kernelValue
26          }
27          tempPartition->setData(c, sum/sumWeights)
28        }
29        sum=0
30      }
31    }
32    //share results from pass 1 with the neighboring processes
33    tempPartition->share();
34    // pass 2
35    for each cell c (x,y) in the partition
36        if kernel exceeds top/bottom partition borders {
37          for(int k=0;k<kernelLength;k++){
38            if(tempPartition->isNodata(x,y-middlePoint+k)){
39              tempW-=kernel[k]
40            } else {
41              tempPartition->getData(i,j-middlePoint+k,demValue)
42              kernelValue=kernel[k]
43              sum+=demValue*kernelValue
44            }
45          }
46          acPartition->setData(c, sum/tempW)
47          tempW=sumWeights
48        } else {
49          for(int k=0;k<kernelLength;k++){
50            tempPartition->getData(x,y-middlePoint+k,demValue)
51            kernelValue=kernel[k]
52            sum+=demValue*kernelValue
53          }
54          acPartition->setData(c, sum/sumWeights)
55        }
56        sum=0
57      }
58    }
```

Besides the uncertainty model, the implementation and parallelization of the two additional flow routing algorithms MD8 and MD∞ are of interest. The MD8 have been implemented on the information provided by Qin et al. (2007). For the implementation of the MD∞ algorithm (Seibert and McGlynn, 2007), the source code was provided by the H2K unit. The MD∞ algorithm is significantly more complex compared to the MD8 due to the inclusion of the triangular facets approach provided by the D∞ algorithm. Both algorithms need a fully processed DEM which means that

flats and pits must be removed. Pits are removed by an algorithm provided by TauDEM. TauDEM resolves flats on the fly with the determination of the flow directions. It was not possible to apply this approach to the two additional flow routing algorithms, because of its complexity and the different computation sequence. A solution has been found with a simple preprocessing algorithm, which increments the elevation of cells located in flats until the flats are removed. However, this approach is not so efficient.

The starting sequence of both algorithms is identical. The algorithms determine the number of neighboring cells draining to the cell for each grid cell of the input DEM (Code A.1). This is easily calculated by comparing the elevation of the center cell with the neighboring cells elevations. The idea is that a cell's accumulated area can only be computed when all cells which drain to the particular cell are already computed. A new grid is created containing the numbers of draining neighboring cells. The grid containing the flow accumulation is initialized to 1 for each cell. All cells which have no draining neighbors are pushed into the queue. The queue always only contains cells which actually can be computed and each process has its own queue.

The general approach to compute the accumulated area is in both cases the same. The first cell in the queue is taken and computed which means that the distribution coefficients for the neighboring cells are calculated and the flow accumulation of this cell is propagated downwards to these particular neighboring cells. All cells receiving flow accumulation have their value in the other partition, which contains the information of draining cells, reduced by one. If such a cell arrives at 0, it is pushed into the queue for processing and the value is set to -1, meaning that this cell has been processed. This is done until the queues of all processes are empty. If this is the case, the borders of the processes are shared by the buffers for both partitions, the one containing the flow accumulation and the one containing the number of draining neighbors. All processes now scan their borders for cells having the number of draining neighbors equal to zero and push them into their queue. The process starts from the beginning. If all processes have empty queues and the borders have no more cells with a value equal to 0, the computation of the upslope area is finished. The part defining the distribution of the flow accumulation to the neighboring cells differs for these two flow routing approaches. The implementation of the MD8 flow routing algorithm is presented in Code A.2 and the implementation of the MD∞ flow routing algorithm is given in Code A.3

4 Experimental Performance Analysis

This chapter presents the evaluation of the proposed uncertainty analysis approach in terms of performance. Section 4.1 provides information about the test system and the used input DEMs. This section further presents the run time of different algorithms in tables and charts. Section 4.2 outlines some issues occurred in the evaluation process and provides ideas how to avoid them if possible or how they could be fixed in future implementation improvements.

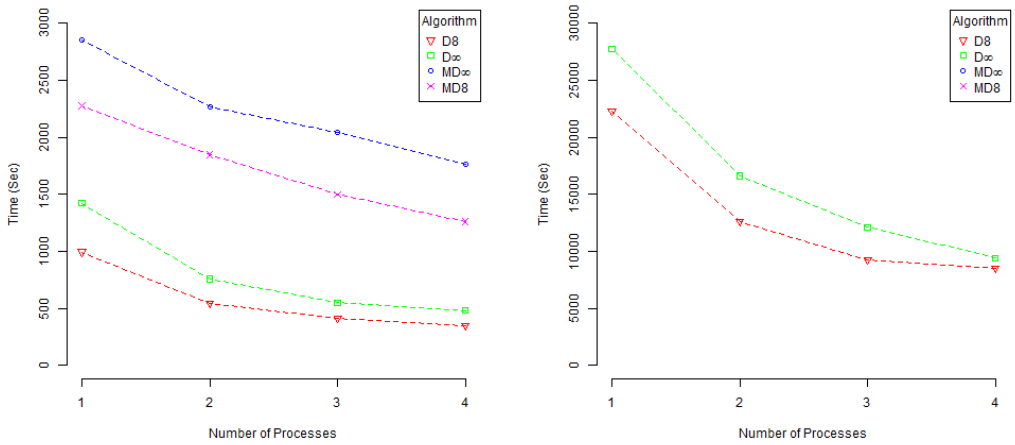
4.1 Performance Tests

The proposed algorithm for uncertainty analysis was evaluated comparing its execution time for different numbers of processes. The algorithm was implemented using C++ and compiled by the Microsoft Visual Studio 2010 compiler. Several tests were done to evaluate the performance increase with increasing number of processes. The tests were executed using a Windows 7 64 bit Operating System based on a Core i7 3770K CPU (4 physical and 8 logical cores) running with 4.2 GHz and 8 GB of RAM. The DEMs used are the one introduced in Section 3.3 with 1282x1540 cells and a different one with larger dimensions (4774x5646) representing the complete Alpthal instead of only the Zwackentobel. The Monte Carlo simulation tests were done with 500 runs and a spatial autocorrelation range of 40 m to get approximately stable results. The error surfaces generated in different simulations were always identical to guarantee comparability. The processing time always included computation time and the disk I/O (input/output) time, but I/O time was reduced to a minimum for this test (no writing of the intermediate results) and the solid state disk (flash based hard disk) of the system used reduced I/O time to a negligible amount.

Table 4.1 and charts in Figure 4.1 illustrate the processing time on the two DEMs. As expected, the more processes are used the lower the execution time is. However, the scaling is not linear, it decreases exponentially with additional processes. Notice that the MD∞ and MD8 algorithm suffer from the inefficient flat removing algorithm. In the Alpthal DEM, the flat removing algorithms processing time explodes, leading to completion times of the uncertainty analysis beyond 100'000 seconds for 500 runs. However, this study approves that parallelization can increase the performance of algorithms significantly and allows uncertainty analysis to be done in a relatively short amount of time or with significantly more runs than before in the same time. Notice that the benefit of this parallelization approach depends on the available CPU cores of the system. The best performance is achieved when the algorithm runs on as many processes as the system provides.

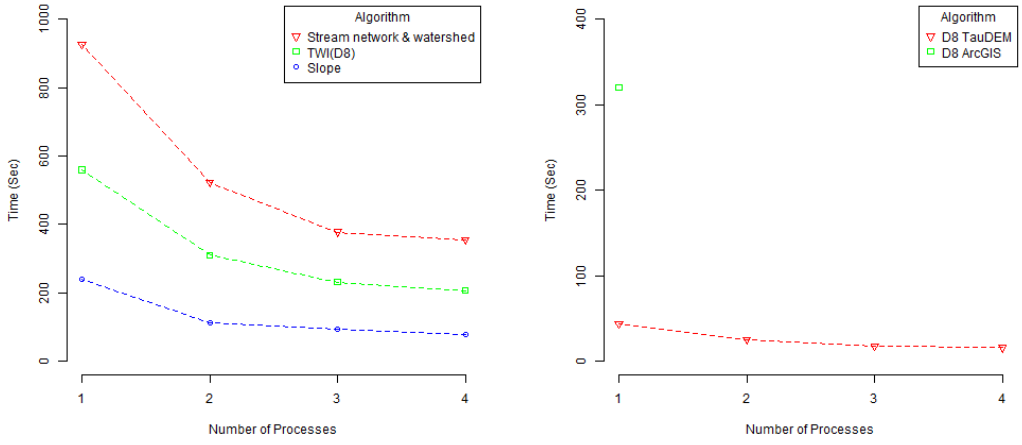
Terrain	Algorithm	Processing time (sec)			
		Number of Processes:			
		1	2	3	4
Zwackentobel (1282x1540)	D8	994	542	410	347
	D∞	1421	757	548	480
	MD∞	2858	2264	2046	1762
	MD8	2276	1849	1499	1259
Alpthal (4774x5646)	D8	22273	12615	9255	8485
	D∞	27781	16599	12106	9411
	MD∞	-	-	-	-
	MD8	-	-	-	-

Table 4.1: Processing time for different upslope area algorithms, terrain sizes and processes. Time represents the complete uncertainty analysis with 500 simulation runs and a spatial autocorrelation range of 40 m.



(a) (b)

Figure 4.1: Processing graphs for different flow accumulation algorithms in (a) Zwackentobel / (b) Alpthal terrain showing the time needed to execute the complete uncertainty analysis with 500 simulation runs using a spatial autocorrelation range of 40 m.



(a) (b)

Figure 4.2: (a) Processing graphs for different topographic algorithms in Zwackentobel terrain showing the time needed to execute the complete uncertainty analysis with 250 runs. TWI uses the D8 flow accumulation algorithm. (b) Processing graphs for the D8 upslope area computation in the Alpthal terrain showing the time needed to execute the algorithm once.

Figure 4.2(a) shows the performance between different topographic algorithms provided by the proposed uncertainty analysis approach in the Zwackentobel terrain. This time, the processing time was measured for 250 Monte Carlo simulation runs. The processing time varies as expected considerably between the different algorithms, but the scaling with the number of processes is in all cases almost the same. In general the algorithms are up to three times faster running on four processes, which is not a bad result. Figure 4.2(b) compares the execution time of the D8 flow accumulation algorithm against the widely used commercial GIS software ArcGIS. As expected, the queue based MPI approach of TauDEM is dramatically faster than the serial recursive algorithm implemented in ArcGIS. Even without the uncertainty model algorithms, the execution time for 500 simulation runs in a Monte Carlo simulation would exceed 140'000 seconds and would be significantly slower than TauDEM running on only one process (22'000 seconds).

Multiple algorithms are needed to compute one realization of the uncertainty analysis. A closer look at the processing time of the various algorithms of one realization computing the MD∞ flow accumulation is given in Table 4.2. These tests were done with four processes and the larger Alpthal DEM. The generation of the error surface without spatial autocorrelation is computationally less demanding. The Gaussian convolution processing time depends on the spatial autocorrelation range, but scales linearly with the range and with the size of the terrain. Overall, the uncertainty modeling part is very fast, even with high values for the spatial autocorrelation range. The algorithms implemented for the uncertainty modeling part scale very well with additional processes. The next step is the DEM filling algorithm using the flooding approach proposed by Planchon and Darboux (2002) which is provided by TauDEM. This approach is also very efficient. The MD8 and MD∞ flow routing algorithms request a fully processed DEM which means that no flats should be existent in the DEM. The implemented algorithm to achieve this is only viable for small DEMs with only a few flats, otherwise the processing time increases dramatically as shown in Table 4.2. The reasons therefore can be found in the brute force approach searching flats and increasing them by a small elevation value until all flats are removed. In addition, this approach can lead to distorted DEMs having a significant impact on the derivation of topographic parameters. The flow routing algorithm MD∞ itself consumes the second most time and is significantly slower than the uncertainty modeling algorithms. However, the flat removing algorithm is definitely the bottleneck in the MD∞ flow routing computation when the DEM is large or contains many flats.

Algorithm	Processing time (sec)	
	Number of Processes:	
	1	4
Error surface	1.1	0.3
Spatial autocorrelation	6.9	1.8
Filling sinks	4.1	2.1
Removing flats	331	147
MD∞ flow accumulation	29.1	13.8
Total time	390	172

Table 4.2: Processing time for the MD∞ flow accumulation computation (1 run) using the Alpthal DEM.

4.2 Computing Issues

Unfortunately, an issue arose in the evaluation: the *stream network & watershed* algorithm crashes reproducibly at run 254 independent of the DEM used. It seems to be a memory based issue which is not surprising because TauDEM has not been optimized regarding to memory management. Dozens of memory leaks had to be fixed, but some must still remain. That is the reason why the results based on this algorithm are computed with a maximum of 250 runs.

Secondly, the drop analysis for the stream network extraction provided by TauDEM can lead to the value of -1.0. This value determines the possible minimum length of a stream link. A very large number of links are generated with this value, slowing down the computation significantly and leading to inappropriate results. This threshold value can be set manually by a parameter in the proposed tool to avoid this issue.

Thirdly, the computation of stream networks and watersheds requires outlets (shape files). TauDEM computes stream networks always in combination with the watershed, thus the stream network also needs an outlet. Usually outlets are placed on a stream. In the proposed uncertainty analysis it can happen that outlets are not located on streams in some realizations. The result of such realizations is basically empty (no watershed or streams). This issue could be fixed by letting outlets snap, but this is not trivial. How do we decide where the outlet should snap? The nearest cell with the highest value must not be the most appropriate in all cases. In addition, it may be necessary that the outlet has to correspond to specific locations in the real-world. If we want to compute the catchment area and its uncertainty of a sampling site, for example, to investigate the relationship between the quality of the water at this specific location and the size of the catchment area, the results would be unsuitable for cases in which the snap distances would have been to far. Because of this issue, snapping outlets were omitted. In return, the outlet should be set on a very certain point, so that in most cases an upslope area can be computed. A preanalysis with the D8 flow accumulation algorithm can help to determine adequate locations for outlets. The quality of the uncertainty analysis is evaluated in a

separate raster, containing the information how often a cell received no value (NULL). If this raster contains too high numbers of NULL values, the uncertainty analysis is most likely not valid and the outlet should be relocated.

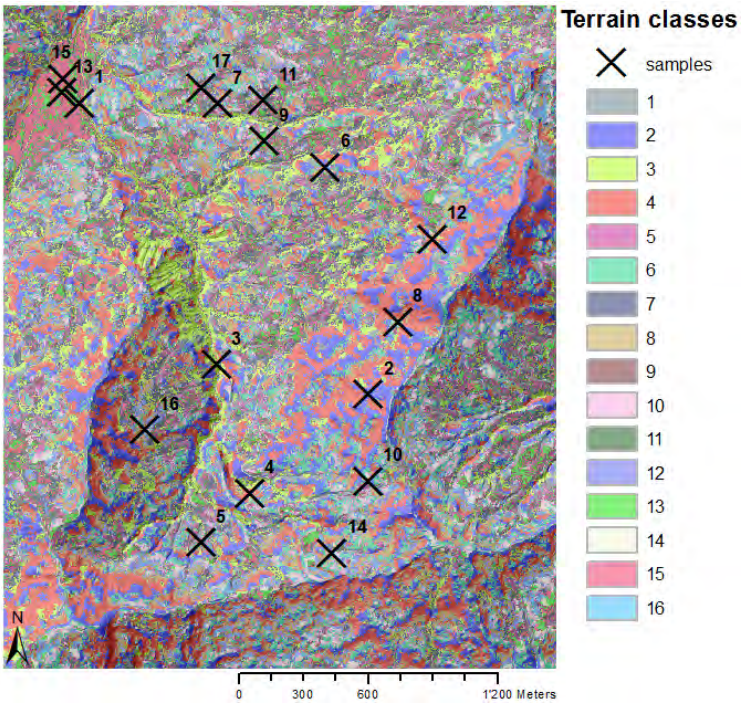
5 Analyzing the Impact of Uncertainty

This chapter presents results of the impact of uncertainty on various hydrological models at different scales and an analysis of multiple input parameters of the proposed uncertainty analysis approach. Section 5.1 analyses the convergence of some hydrological models in a Monte Carlo simulation with up to 1000 runs. The following Section 5.2 presents the uncertainty assessment of various hydrological models. Section 5.3 evaluates the impact of the uncertainty model input parameters on the uncertainty assessment. Section 5.4 analyses how the results of an uncertainty assessment change if coarser resolution DEMs are used. Finally, Section 5.5 demonstrates a possible application of the results in recent hydrological research investigated by the H2K unit of the University of Zurich.

5.1 Convergence Tests

Analyzing the uncertainty with a Monte Carlo simulation approach requires the determination of the number of runs. In principle, the more runs a Monte Carlo simulation incorporates the more stable is the final result. The convergence of the hydrological algorithms may depend on the properties of the algorithm itself, the DEM terrain characteristics and the uncertainty model parameters. However, it is important to do such a convergence test in order that the achieved results of the Monte Carlo simulation are reliable.

Convergence tests can be done in multiple ways. At the end, at least one statistical parameter is required which is tested against a user defined threshold to evaluate if the Monte Carlo simulation delivers sufficiently stable results. In the proposed approach, the DEM is sampled by a few points. First, the DEM was classified into 16 terrain classes according to the terrain classification approach proposed by Iwahashi and Pike (2007) (Figure 5.1(a)). This should guarantee that the sample points are spatially distributed over different terrain characteristics. Then, for each class a random sample was selected. In addition, so that we do not rely completely on randomness, a cell, which was evaluated as very uncertain in a stream network uncertainty analysis, was selected too. This results in a total of 17 sample points (Figure 5.1(a)) and for each sample, the intermediate results of the Monte Carlo simulation are saved and used to calculate the current mean value. This provides us with the statistical parameter needed to evaluate the convergence of the hydrological algorithms. As soon as the mean stabilizes, the result of this sample point has converged.



(a)

surface geometry	Terrain Classes			
fine texture, high convexity	1	5	9	13
fine texture, low convexity	3	7	11	15
coarse texture, high convexity	2	6	10	14
coarse texture, low convexity	4	8	12	16

steep ⟷ gentle

(b)

Figure 5.1: (a) The Zwackentobel classified into 16 terrain classes according to the approach proposed by Iwahashi and Pike (2007) and 17 sampling points. The number of the sample points matches the number of the terrain classes except for sample point 17, which represents an uncertain location evaluated in a stream network uncertainty assessment (belongs to the terrain class 7). (b) Legend for the terrain classification reprinted from Iwahashi and Pike (2007). 16 classes are partitioned according to their terrain characteristics slope, roughness/texture and convexity.

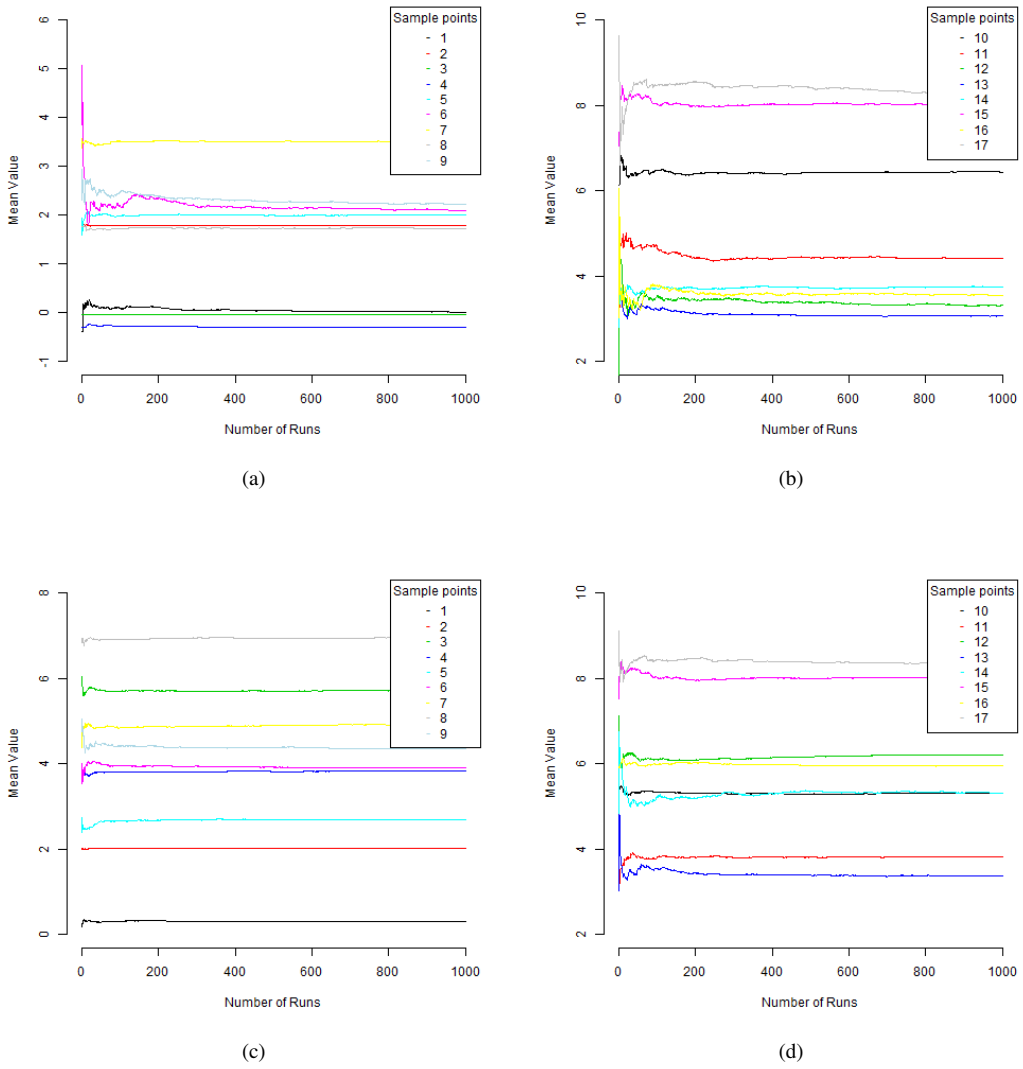
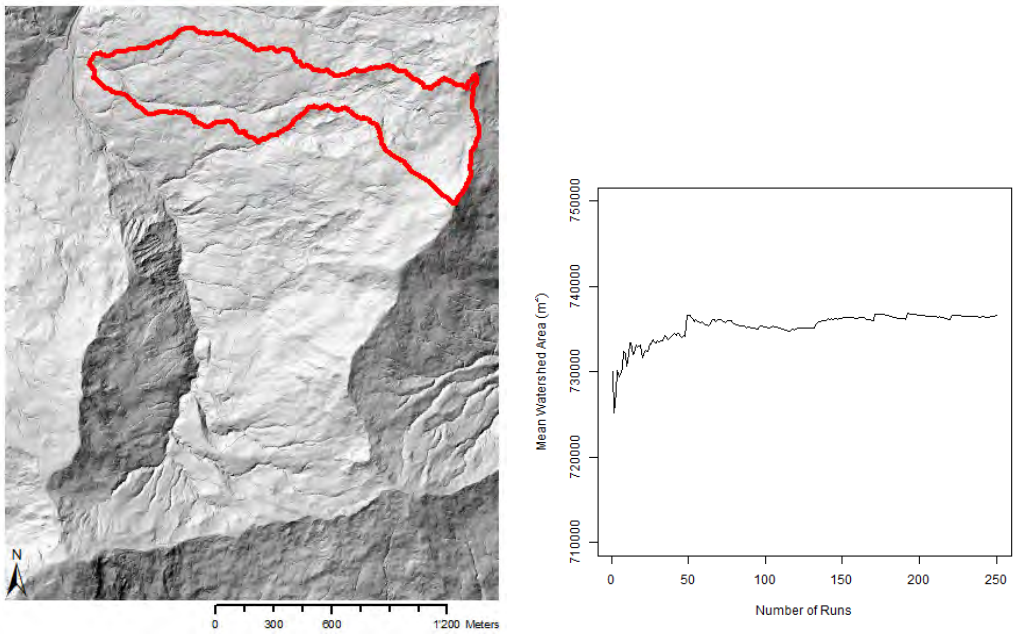


Figure 5.2: Convergence graphs of the TWI based on the D8 (a,b) and MD∞ (c,d) flow routing algorithms for the 17 sample points located in different terrain classes (see Figure 5.1).

The analysis was done with 1000 Monte Carlo simulation runs except for the watershed analysis where only 250 runs were possible due to some implementation issues (see Section 4.2). The parameters of the uncertainty model were 0.5 m for the RMSE and 40 m for the spatial autocorrelation range. In addition, the canopy mask was used to improve the uncertainty model regarding to the spatial structure of error in LiDAR based DEMs. Figure 5.2 illustrates the convergence of the D8 respectively the MD∞ based TWI. The TWI was chosen because it is a very popular index used in hydrological investigations.

The results show that the TWI calculation based on the D8 flow routing algorithm appears to converge after approximately 400 simulation runs. At some locations, e.g. point 3 which is located in a steep, fine texture and concave area, even less than 100 runs already lead to very stable results whereas other sample points, e.g. point 12 (rather gentle, low convexity and coarse texture), show a trend beyond 800 runs. The terrain classes seem to be related to the convergence test results. The



(a) (b)

Figure 5.3: (a) Erlenbach watershed. (b) Convergence graph for the watershed area calculation.

most sample points show a very high variation below 100 runs leading to the conclusion that less than 100 runs are definitely too few for a reliable uncertainty assessment. As expected, the TWI based on the MD∞ flow routing algorithm appears to converge considerably earlier than the D8 based TWI. The MD∞ based TWI seems to be stable after 200 simulation runs. However, below 100 runs the MD∞ based TWI also appears very noisy.

The second hydrological model analyzed was the watershed of the Erlenbach in the Zwackentobel (Figure 5.3(a)). The mean area of the watershed was calculated after each run and these results are illustrated in Figure 5.3(b). The watershed area appears to converge already after 150 simulation runs and the uncertainty analysis seems to be very unstable below 50 simulation runs. The reason for the relatively early convergence could be that watersheds rely on larger terrain forms instead of the local relief. Because the terrain is very steep in the Zwackentobel, the watersheds remain mostly stable in the uncertainty analysis. However, the results may differ in different terrain conditions.

5.2 Sensitivity Analysis

For each flow routing algorithm and compound index, statistical rasters representing the absolute standard deviation, relative standard deviation and mean value were computed for the Zwackentobel DEM. In addition, these errors were plotted against the terrain characteristics slope, roughness, convexity and the terrain classes proposed by Iwahashi and Pike (2007). This allowed analysis of the spatial structure of uncertainty for the different flow routing algorithms and compound indices. However, only the most important and interesting results are outlined to keep the section clear. The uncertainty model parameters are the same as outlined in Section 5.1. The number of simulation runs is defined according to the findings in the previous Section 5.1, where 400 simulation runs led to stable results for all evaluated hydrological models. For the watershed and stream network

analysis 250 simulation runs were used. The statistical grids for the watershed and stream network uncertainty assessment represent the probability of a cell belonging to the watershed respectively to the stream network and the error (H) (Figure 3.5). The spatial structure of error for the stream network computation is analyzed using the error (H) and the probability.

5.2.1 Flow Routing Algorithms

Figure 5.4 illustrates the grid based result of the uncertainty assessment for four different flow accumulation computations. The associated histograms are given in Figure 5.5. The D8 flow routing algorithm provides a very sensitive result (mean error $\pm 119\%$). The grid looks very noisy with a lot of linear features which is typical for the D8 algorithm. The other three flow routing algorithms show a significantly smoother result, with an overall lower average uncertainty. The error also seems to be more spatially autocorrelated. The MD8 and MD∞ flow routing algorithm deliver the most robust results which are almost identical (mean error $\pm 45\%$). The mean error is almost three times lower compared to the D8 flow routing algorithm. However, the maximum error is in all four cases the same with $\pm 1990\%$ (Table 5.1). In general, these very high errors can be observed in the top-left corner where the terrain is extremely flat and rough.

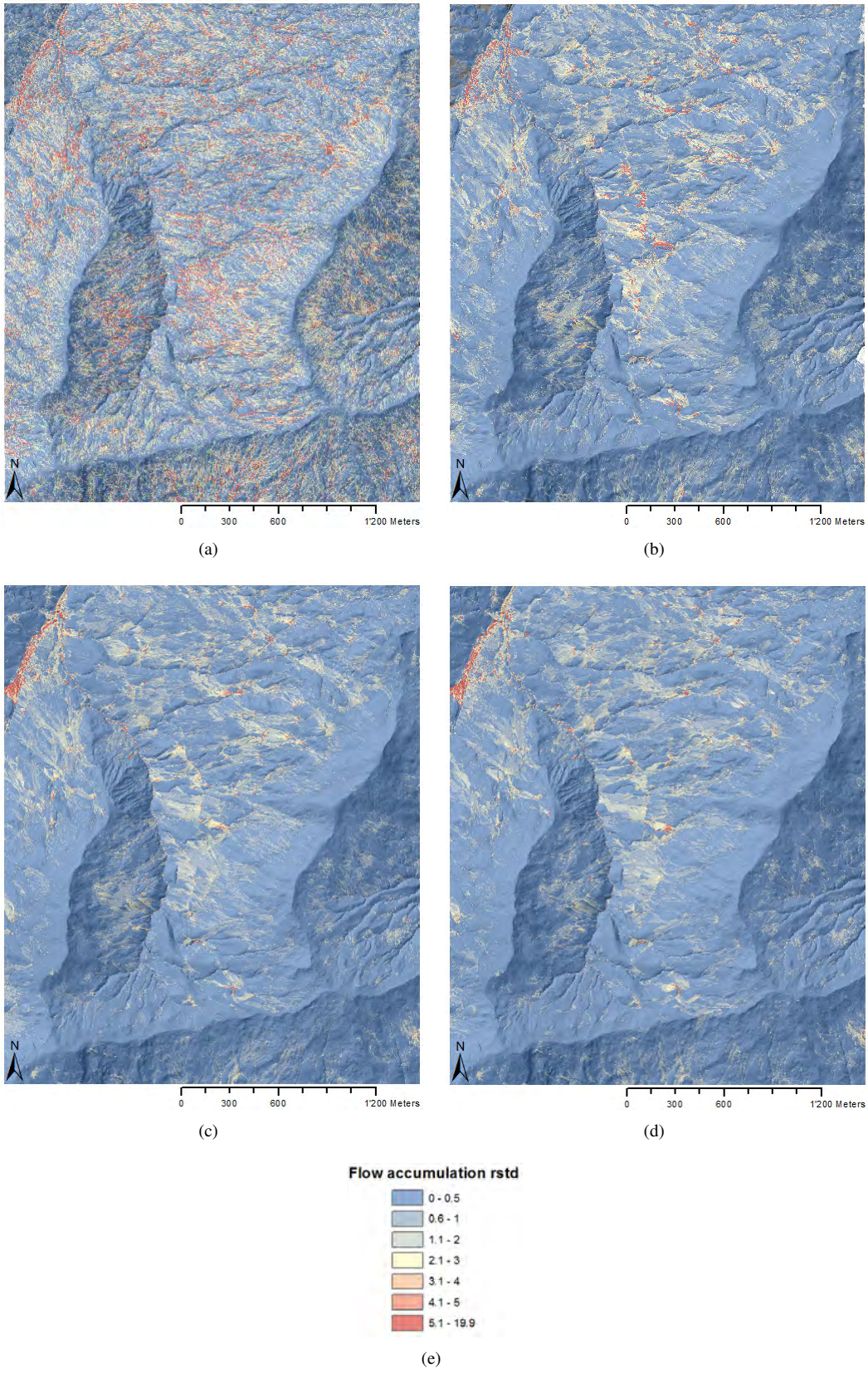


Figure 5.4: Grids (Zwackentobel) illustrating the relative standard deviation of the different flow accumulation algorithms: (a) D8, (b) D∞. (c) MD8 and (d) MD∞ flow routing algorithm.

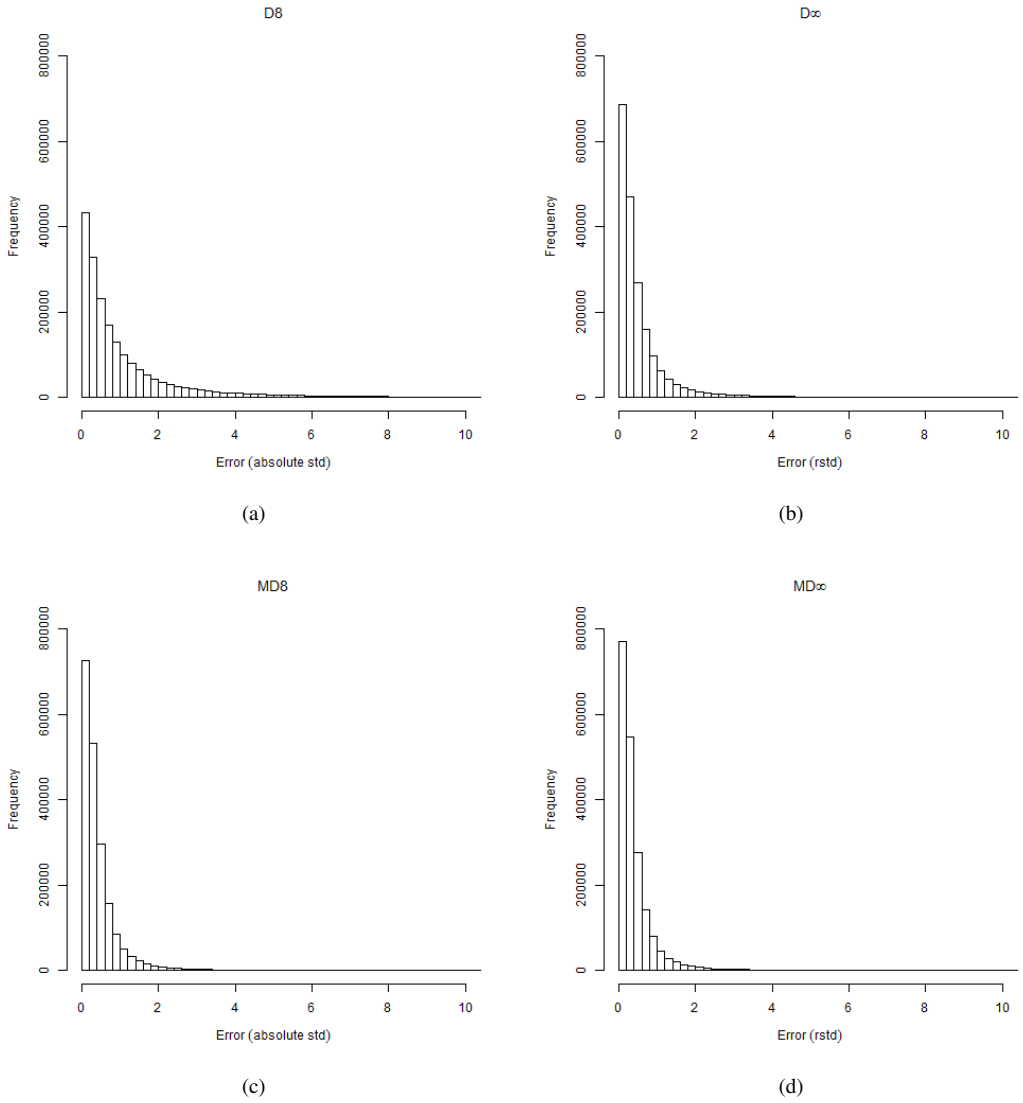


Figure 5.5: Histograms showing the distribution of errors for the grids illustrated in Figure 5.4. Statistical measurements are shown in Table 5.1.

Flow routing algorithm	#Runs	Mean	STD	Min	Max
D8	400	1.19	1.83	0	19.9
D∞	400	0.57	0.95	0	19.9
MD8	400	0.46	0.76	0	19.9
MD∞	400	0.44	0.74	0	19.9

Table 5.1: Statistical measurements of the flow accumulation histograms (Figure 5.5) based on different flow routing algorithms.

Figures 5.6, 5.7, 5.8 and 5.9 present the spatial structure of uncertainty for the four different flow routing algorithms used to compute the flow accumulation. As expected, the single flow direction algorithm D8 shows the highest errors in all cases. The reason can be found in the fact that the D8 algorithm does not weight the flow distributed downwards, instead the determination of the direction highly depends on the elevations of the 3x3 neighborhood. The D∞ already shows a more robust behavior in this uncertainty assessment. Due to the fact that water can be propagated downwards up to two neighboring cells, the algorithm reacts less sensitive to elevation changes in the 3x3 neighborhood. The most robust flow routing algorithms are the MD8 and MD∞, both show mostly equal results. The difference to the D∞ flow routing algorithm is small but existent.

Errors are in all cases higher in flat areas (Figure 5.6). Errors also seem to occur mainly in non convex/concave reliefs with some outliers in highly concave reliefs. The D8 flow routing algorithm is again more sensitive than the multiple flow direction algorithms (Figure 5.7). Errors also increase when the roughness of the terrain increases (Figure 5.8). The D8 surprisingly shows a very stable and high error in less rough terrain. In contrast, the multiple flow direction approaches show an exponential increase in error with increasing roughness. However, in this study, the MD8 and MD∞ show an unexpected high error in highly rough terrain compared to the other two algorithms which cannot be explained. The box plots using the terrain classes as proposed by Iwahashi and Pike (2007) support the findings that slope and roughness of the terrain have the largest impact on the uncertainty in the flow accumulation computation (Figure 5.9).

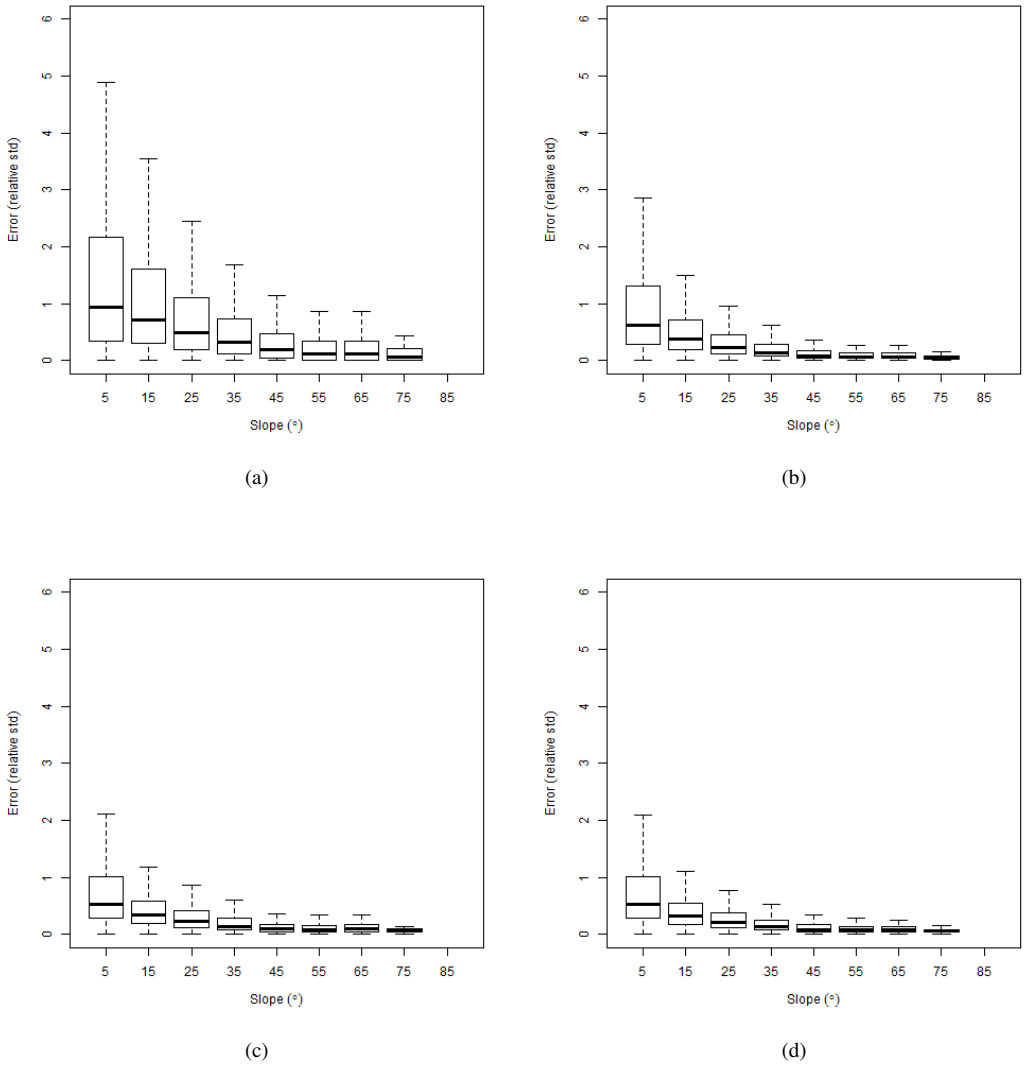


Figure 5.6: Box plots showing the relationship between the relative standard deviation of the flow accumulation and the local slope: (a) D8, (b) D∞, (c) MD8 and (d) MD∞ flow accumulation algorithm.

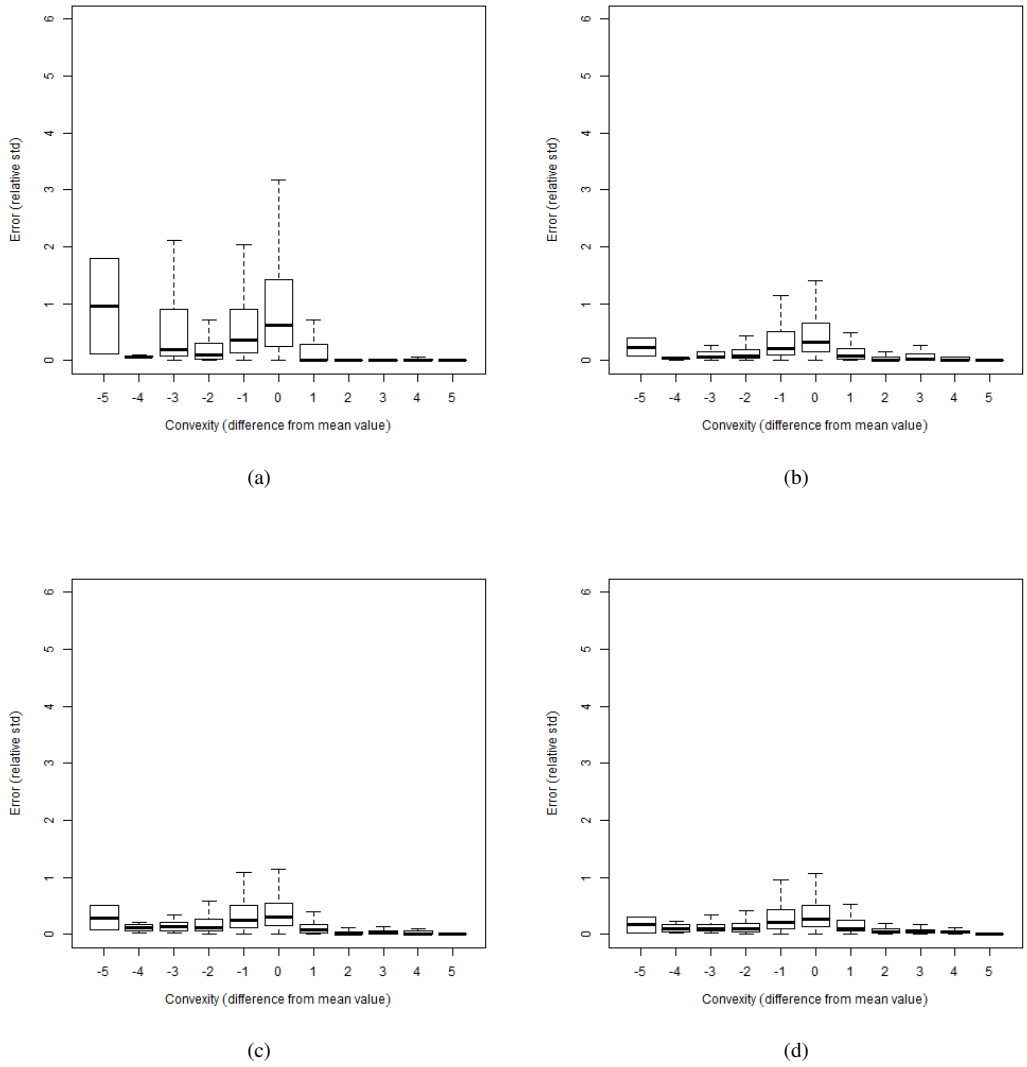


Figure 5.7: Box plots showing the relationship between the relative standard deviation of the flow accumulation and the local convexity: (a) D8, (b) D∞, (c) MD8 and (d) MD∞ flow accumulation algorithm.

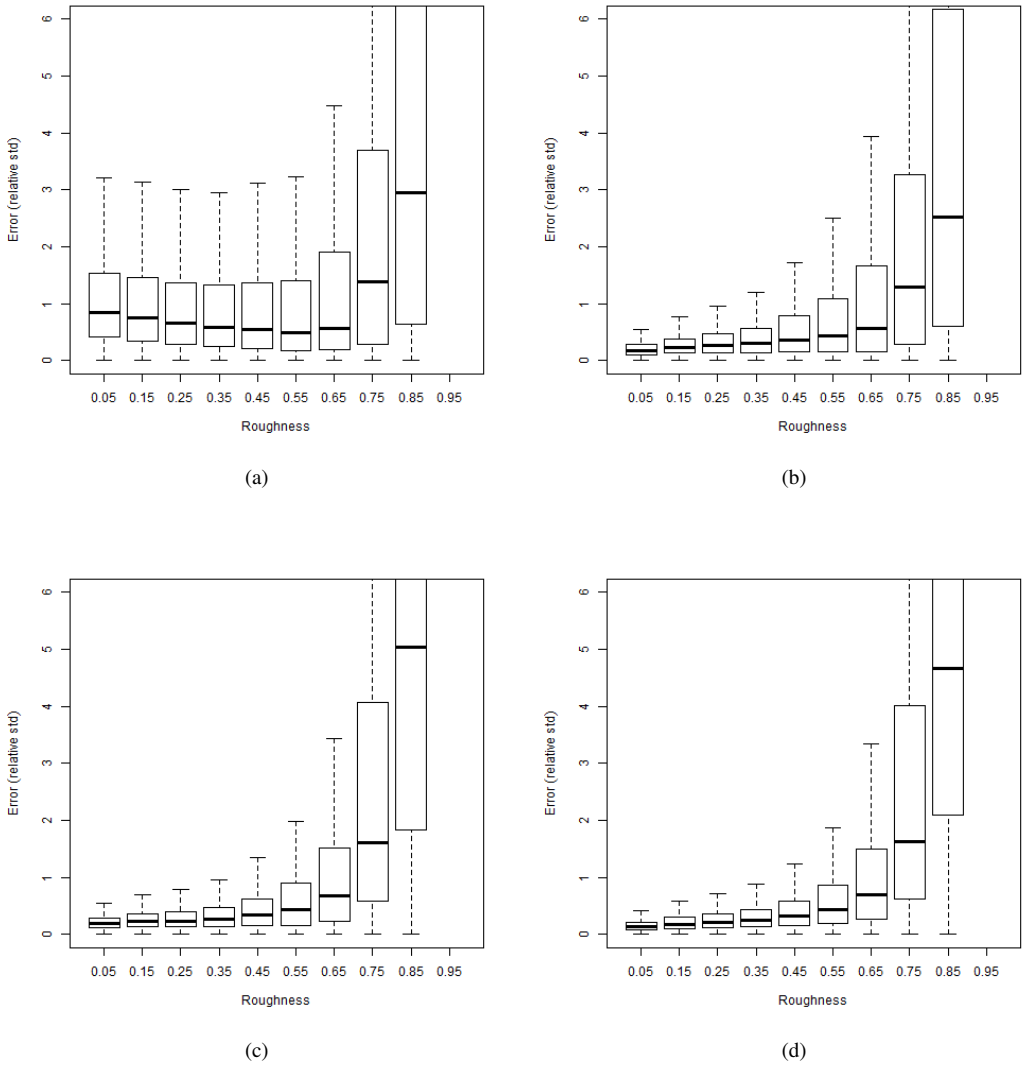


Figure 5.8: Box plots showing the relationship between the relative standard deviation of the flow accumulation and the local roughness: (a) D8, (b) D∞, (c) MD8 and (d) MD∞ flow routing algorithm.

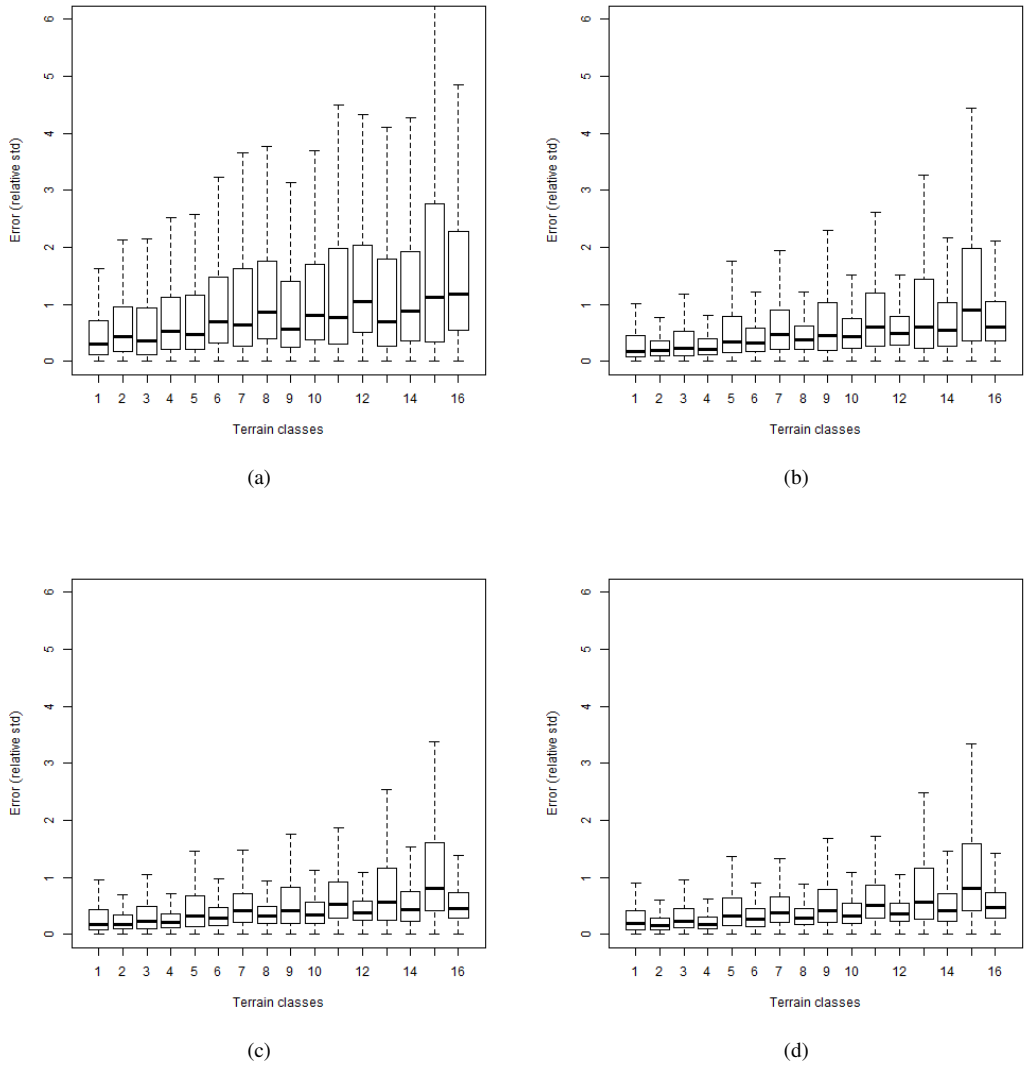


Figure 5.9: Box plots showing the relationship between the relative standard deviation of the flow accumulation and the local terrain classes: (a) D8, (b) D∞, (c) MD8 and (d) MD∞ flow routing algorithm. The terrain classes as proposed by Iwahashi and Pike (2007) incorporate roughness, convexity and slope. The legend is shown in Figure 5.1(b).

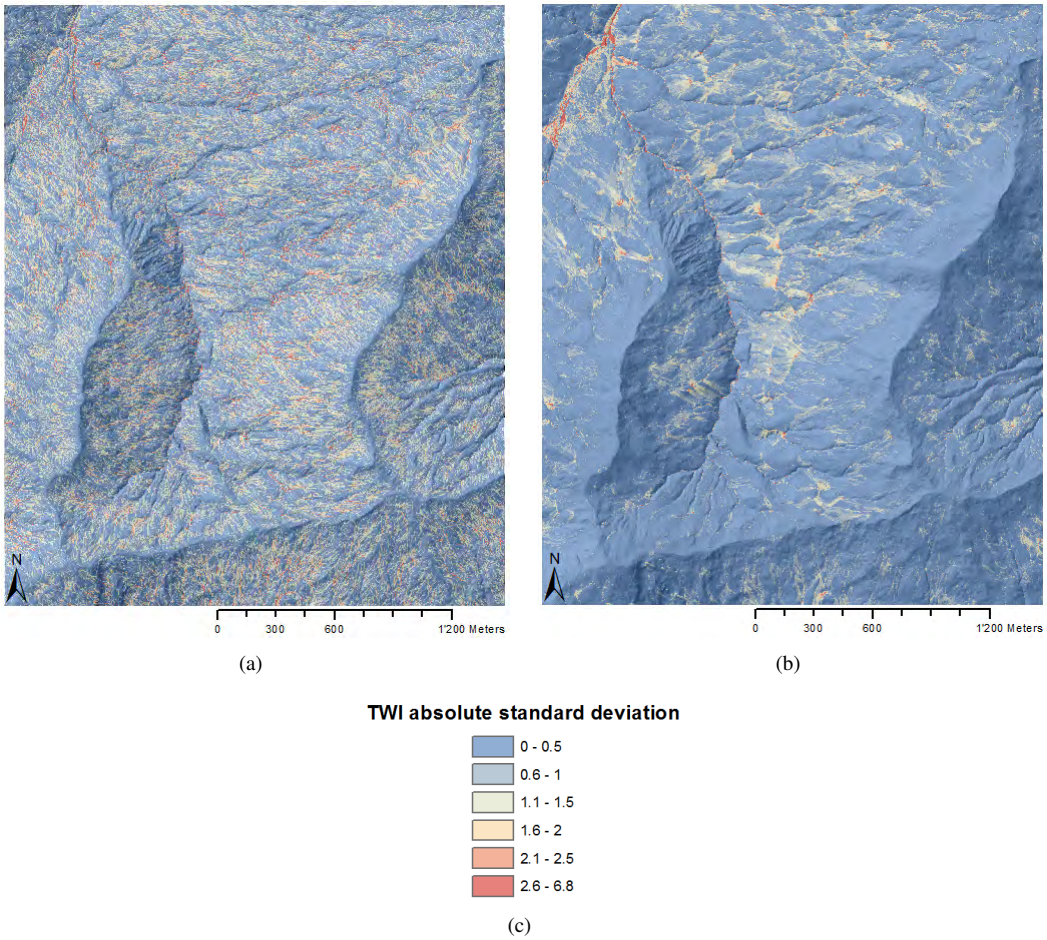
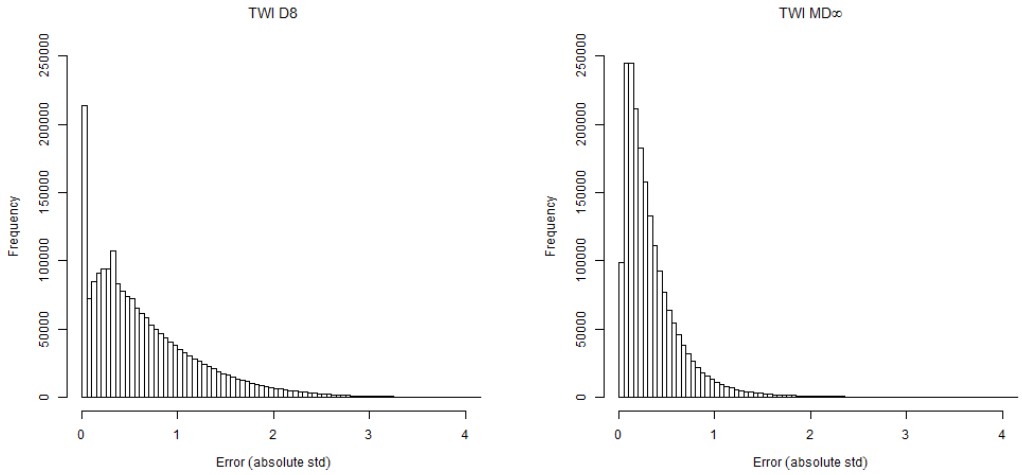


Figure 5.10: Grids (Zwackentobel) illustrating the absolute standard deviation of the two different TWI computations: (a) D8 and (b) MD∞ based TWI maps. The associated histograms are presented in Figure 5.11.

5.2.2 Compound Indices

Previous analysis have shown that the most differences can be found between the D8 and MD∞ flow routing algorithm, thus these two approaches were used to analyze the impact of DEM uncertainty on the TWI calculation. Figure 5.10 illustrates the absolute standard deviation of the TWI calculation with the D8 and the MD∞ flow routing algorithm. Because TWI values are very often near zero or lower than one, the absolute standard deviation was chosen to prevent extreme outliers which occurred when using the relative standard deviation. The results look very similar to the results of the flow accumulation calculation. This is not surprising as the TWI is computed from the flow accumulation area and the local slope. The MD∞ based TWI (mean error $\pm$0.35) is a considerable improvement regarding the robustness compared to the D8 based TWI (mean error $\pm$0.65) algorithm. In this case, the mean error is almost twice as high when using the D8 flow routing algorithm to compute the TWI (Table 5.2). In general, high errors can be found once again in the top-left corner of the DEM where the terrain is low, rough and very flat.

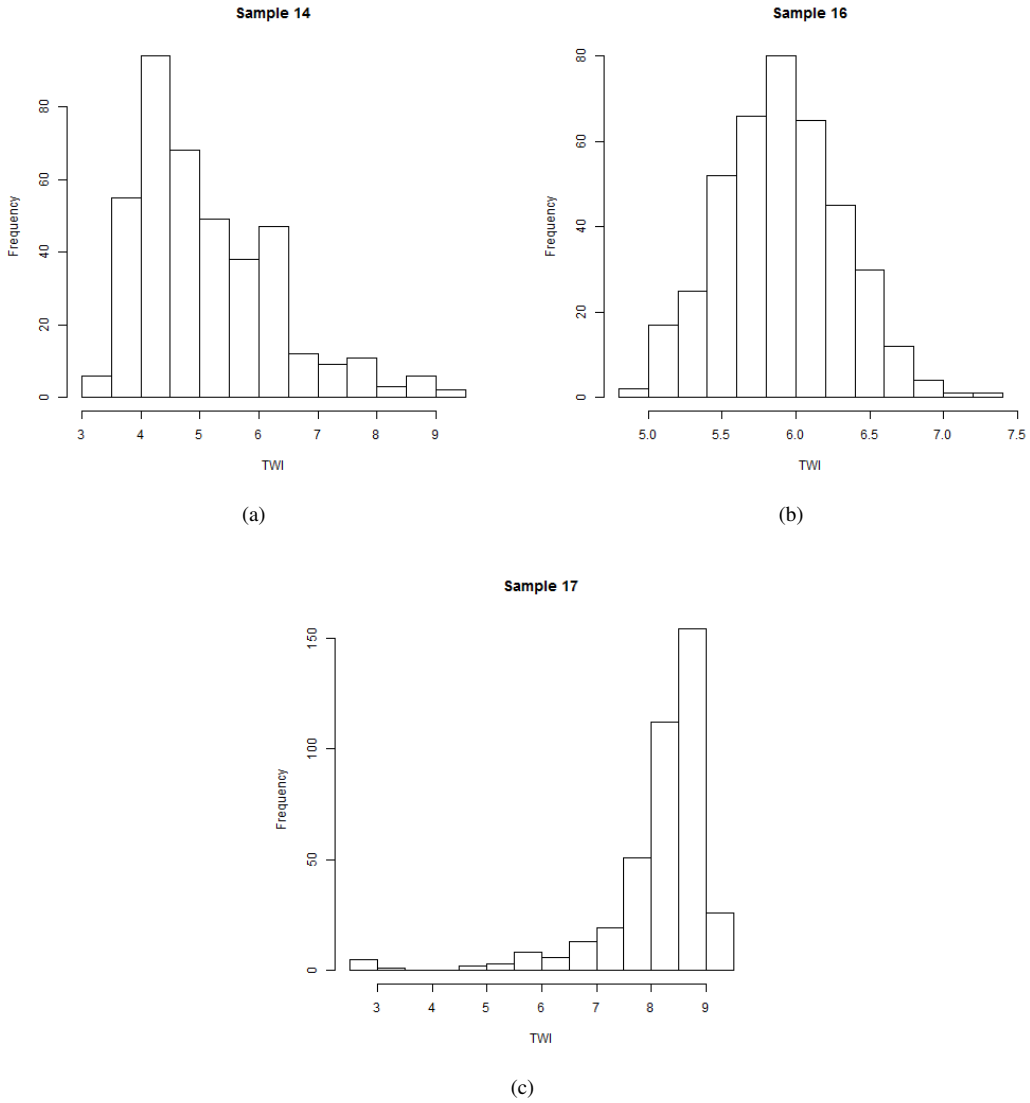


(a) (b)

Figure 5.11: Histograms showing the distribution of errors for the grids illustrated in Figure 5.10. Statistical measurements are shown in Table 5.2.

Flow routing algorithm	#Runs	Mean	STD
D8	400	0.65	0.59
MD∞	400	0.35	0.35

Table 5.2: Statistical measurements of the TWI histograms (Figure 5.11) based on different flow routing algorithms.

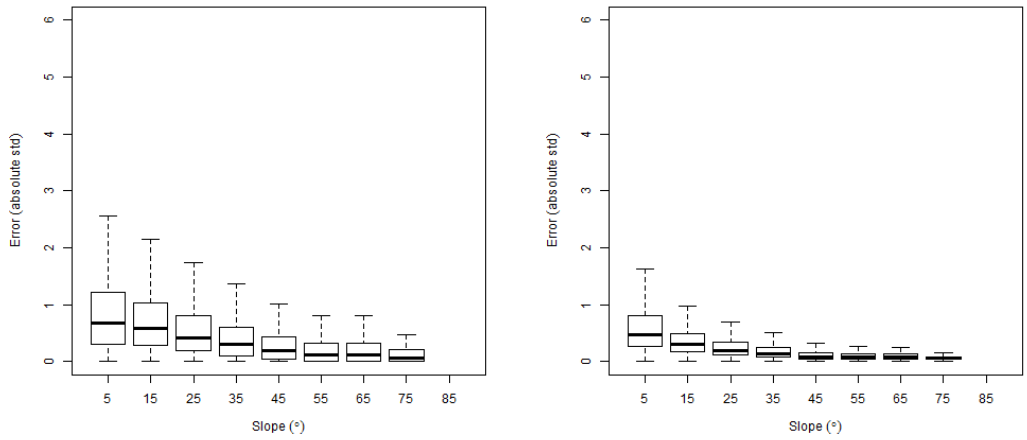


(a) (b)

(c)

Figure 5.12: Histograms showing the distribution of the TWI computations using the D8 flow routing algorithm for different locations in the Zwackentobel DEM (see Figure 5.1(a)). (a) Sample point 14, (b) sample point 16 and (c) sample point 17.

For a more detailed analysis, three samples from the sample points shown in Figure 5.1(a) were chosen and for each sample the distribution of the TWI computation results are illustrated in histograms (Figure 5.12). These histograms show that the computed values must not be normally distributed. In sample 16 (Figure 5.12(b), gentle, coarse texture, low convexity), the TWI calculations are normally distributed, but in sample 14 (Figure 5.12(a), gentle, coarse texture, high convexity) and sample 17 (Figure 5.12(c), rather steep, fine texture, low convexity), the histogram shows that the distribution is skewed to the left respectively to the right. This means that if we ignore uncertainty analysis or we execute too few simulation runs, the results can be heavily over- or underestimated at some points, leading to inappropriate decisions or considerable impacts on models using digital terrain analysis results as input data.

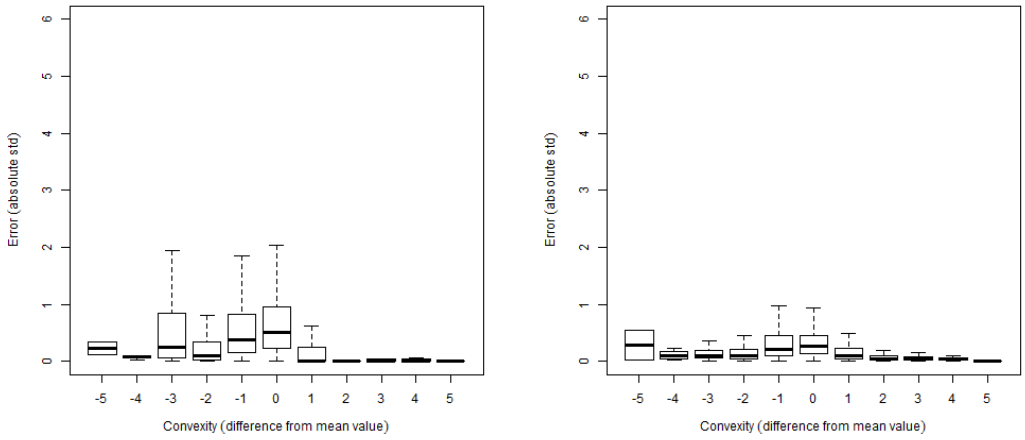


(a) (b)

Figure 5.13: Box plots showing the relationship between the relative standard deviation of the TWI and the local slope: (a) D8 and (b) MD∞ based TWI.

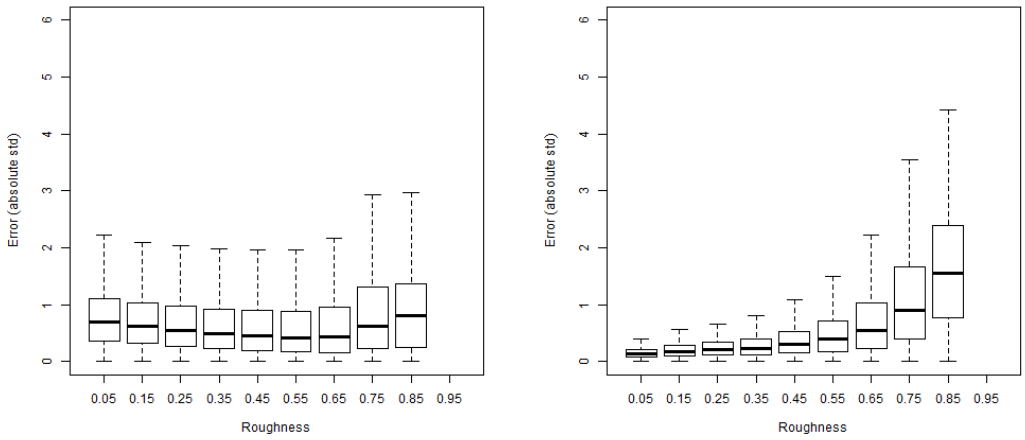
The same applies to the spatial structure of error. Figures 5.13, 5.14, 5.15 and 5.16 present the spatial structure of uncertainty for the two TWI computations using two different flow routing algorithms. In all cases, the results are similar to the one from the flow accumulation computation. Errors are higher in flat areas and the MD∞ based TWI is slightly more robust than the D8 based TWI (Figure 5.13). Errors seem to occur in non convex/concave or slightly concave reliefs (Figure 5.14). The D8 based TWI is again slightly more sensitive. Figure 5.15 shows that errors increase when the roughness of the terrain increases. The D8 based TWI shows a from roughness almost independent error overall. The multiple flow direction approaches show an exponential increase in error with increasing roughness, but have a negotiable error in very flat reliefs. However, the MD∞ shows again a higher error in highly rough terrain than the D8 based TWI. The box plots in Figure 5.16 show that slope and roughness of the terrain have the largest impact on uncertainty in the TWI computation and that the multiple flow direction based TWI shows a considerable smaller error overall.

One might question if the uncertainty of the flow accumulation area or the local slope has a larger impact on the TWI uncertainty. Figure 5.17 presents the result of this analysis by plotting the TWI error against the slope and flow accumulation error. In all cases the error is represented by the standard deviation. The flow accumulation error has a considerably higher impact on the TWI certainty than the local slope. This can be explained by the huge differences in the value ranges. The standard deviation of the flow accumulation can reach up to 550'000, whereas the standard deviation of the local slope only reaches up to 0.04. The local slope must vary significantly more widely to have a larger impact on the TWI error (Equation 2.17).



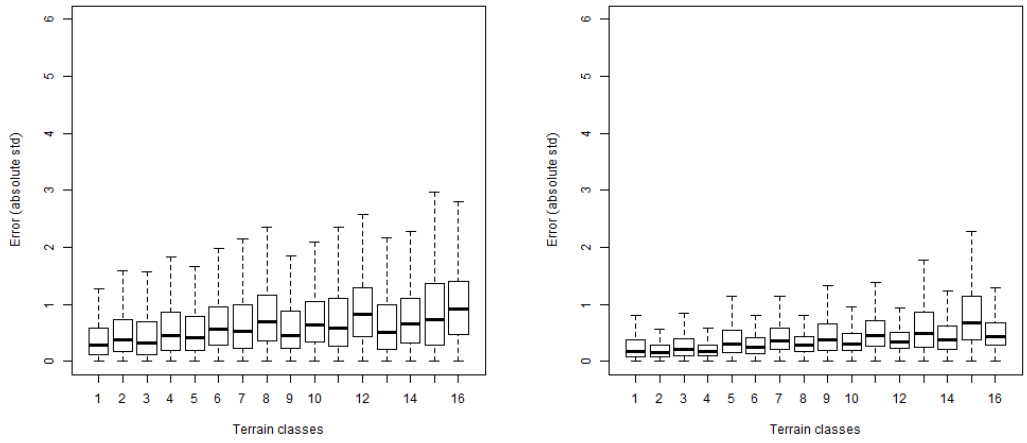
(a) (b)

Figure 5.14: Box plots showing the relationship between the relative standard deviation of the TWI and the local convexity: (a) D8 and (b) MD∞ based TWI.



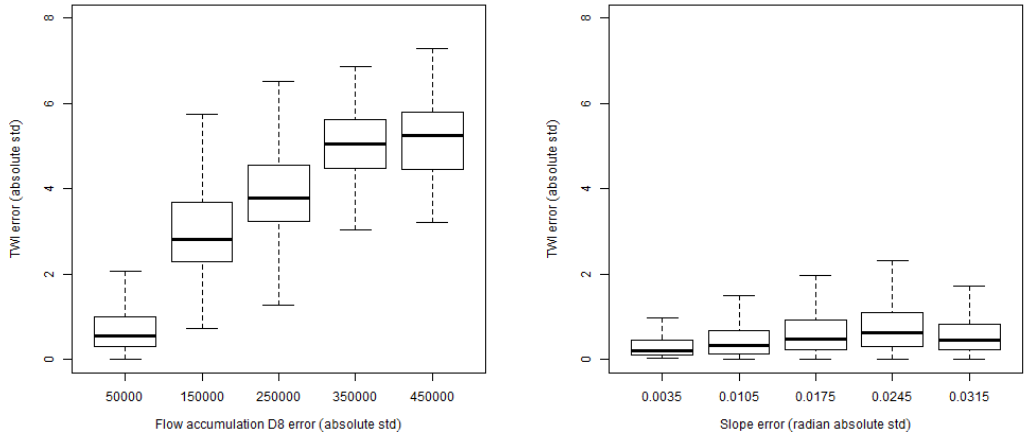
(a) (b)

Figure 5.15: Box plots showing the relationship between the relative standard deviation in TWI and the local roughness: (a) D8 and (b) MD∞ based TWI.



(a) (b)

Figure 5.16: Box plots showing the relationship between the relative standard deviation of the TWI and the local terrain class: (a) D8 and (b) MD∞ based TWI. The terrain classes as proposed by Iwahashi and Pike (2007) incorporate roughness, convexity and slope. The legend is shown in Figure 5.1(b).



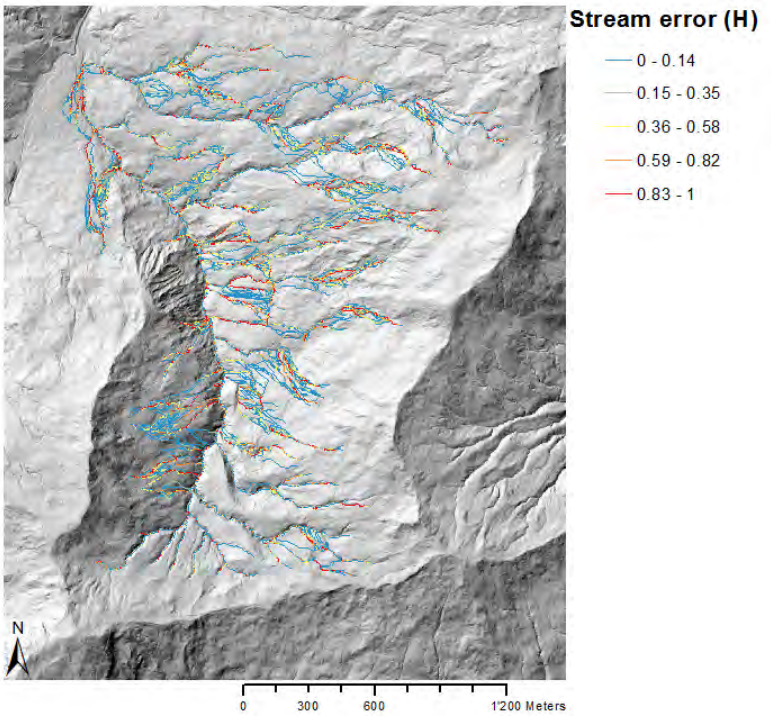
(a) (b)

Figure 5.17: Box plots showing the relationship between the TWI error and (a) the flow accumulation area error and (b) the local slope error.

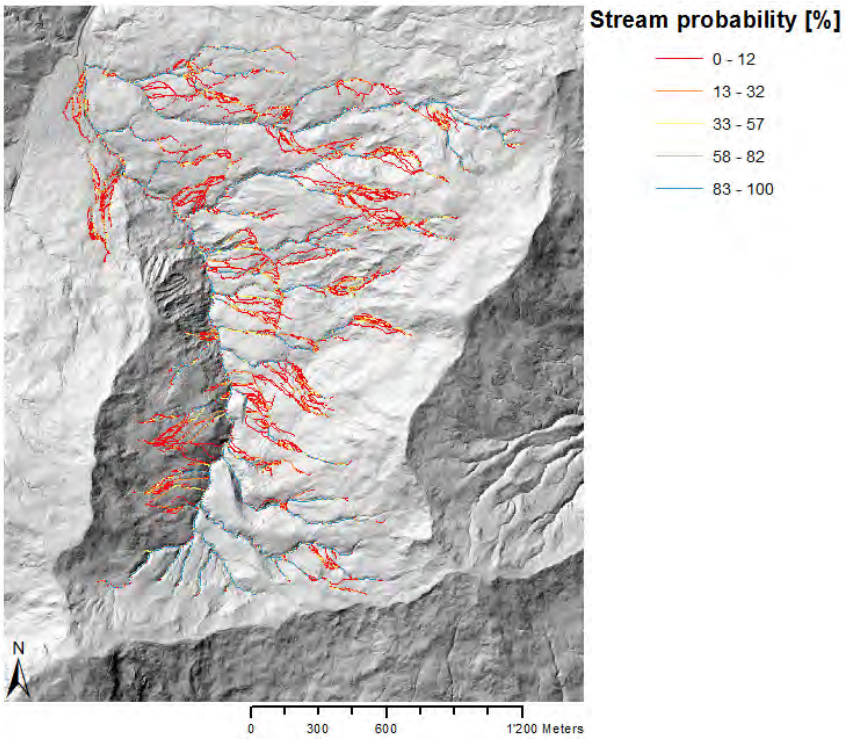
5.2.3 Stream Networks and Watersheds

For the stream network analysis, an outlet near the entrance of the Zwackentobel valley was chosen. The calculation of the stream network was done with a fix minimum stream length parameter of 300. Figures 5.18 and 5.19 illustrate the results of the uncertainty analysis. In some areas, streams are very isolated and have a very high probability (blue color) to be mapped by the uncertainty analysis. This is occurring mainly in channels, where we already expect streams just by a visual evaluation of the DEM. In other areas, streams are widely distributed with a low probability (red color), highlighting areas where stream mapping is very uncertain. These areas do not show significant features just by inspecting the hillshade of the Zwackentobel. After estimating the propagated uncertainty of extracting stream networks from the DEM, we can analyze the uncertainty and its relation to the geomorphology of the terrain.

Figures 5.20 and 5.21 show the relations between the error respectively the probability of the stream network and the terrain characteristics convexity, slope, roughness, the different terrain classes and the MD∞ based TWI. The slope does not seem to have any significant impact on the stream extraction error (H) or probability (Figures 5.20(b) and 5.21(b)). The probability of streams is considerably higher in concave than in convex areas, but the uncertainty is relatively high in concave reliefs (Figures 5.20(a) and 5.21(a)). This can be explained by streams located in channels, but below roads. As seen in Figure 5.18(a), channels where we would expect a high probability of streams are instead marked red meaning that in 50% of the simulations water is flowing down the street instead of passing it. In reality there are bridges, allowing the water to flow underneath it. Such anthropogenic features cause a lot of uncertainty in high resolution DEMs. This is an issue caused by the two- or two-and-a-half-dimensional representation of the terrain surface. The DEM only contains one elevation information per cell. The slope and terrain classes box plots do not show any significant relations (Figures 5.20(d) and 5.21(d)). However, the roughness plot (Figure 5.20(c)) shows that rough terrain increases the uncertainty of the derived stream networks. The TWI also shows a strong correlation (Figure 5.20(e)). Cells having high TWI errors in the uncertainty analysis also have a higher probability to belong to a stream, but are also more uncertain (probability near 50%). By knowing this, users can delineate the most problematic areas and collect more accurate elevation data for these areas or correcting the DEM manually (e.g. removing anthropogenic features such as bridges).

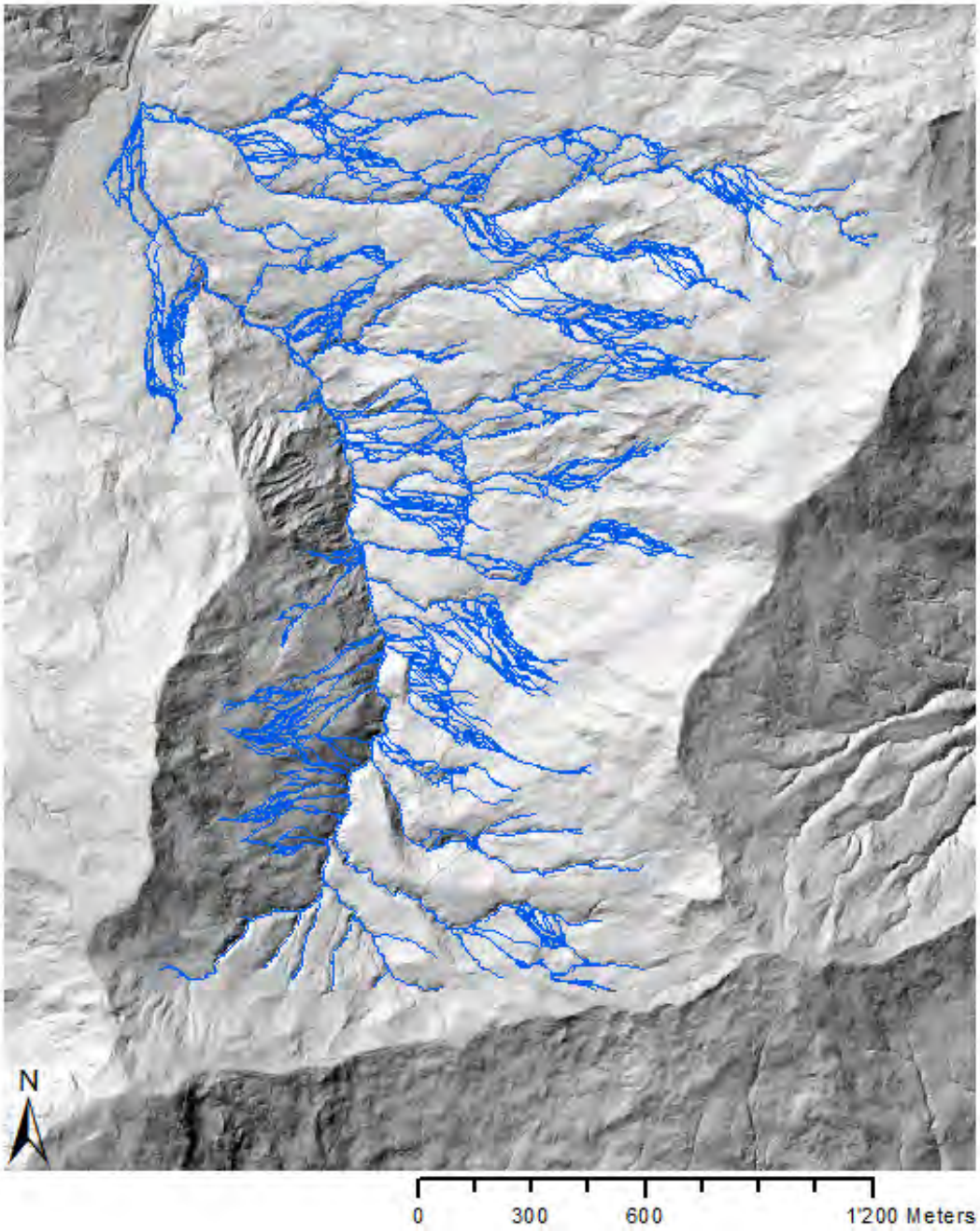


(a)



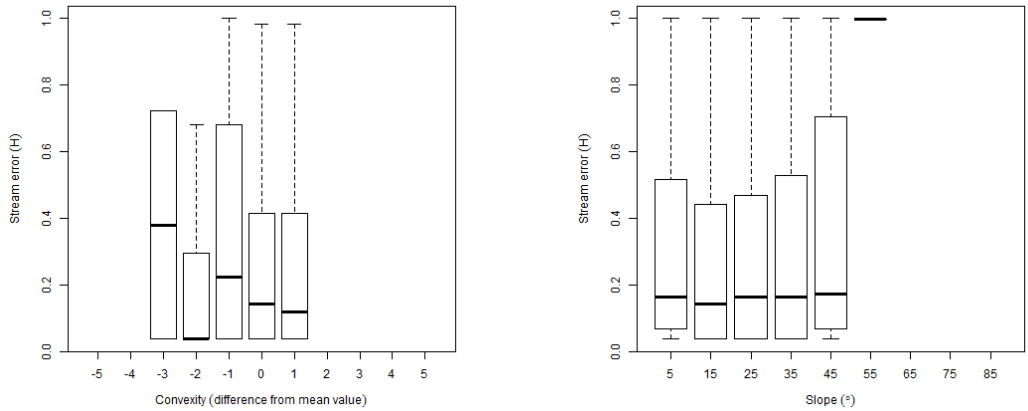
(b)

Figure 5.18: Statistical rasters showing the results of the stream network uncertainty analysis for the Zwackentobel valley. (a) Showing the stream extraction error (H) (see Figure 3.5). (b) Showing the probability of a cell belonging to a stream.



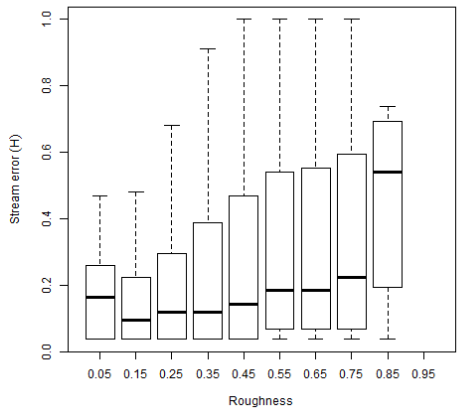
(a)

Figure 5.19: Grid showing all realizations of the stream network uncertainty analysis for the Zwackentobel valley.

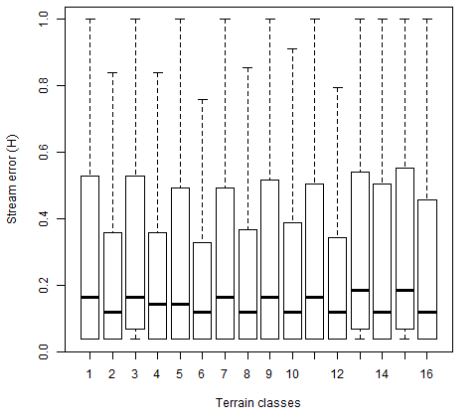


(a)

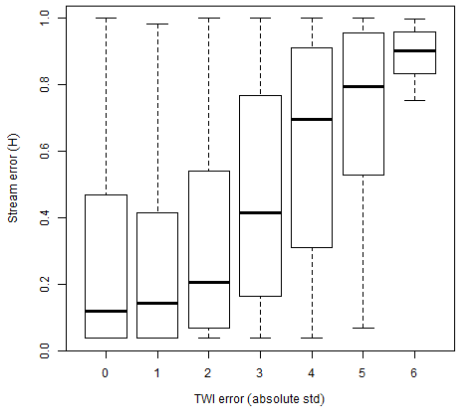
(b)



(c)



(d)



(e)

Figure 5.20: Box plots showing the relationship between the stream error (H) and the terrain characteristics (a) convexity, (b) slope, (c) roughness, (d) the different terrain classes and (e) the MD∞ based TWI.

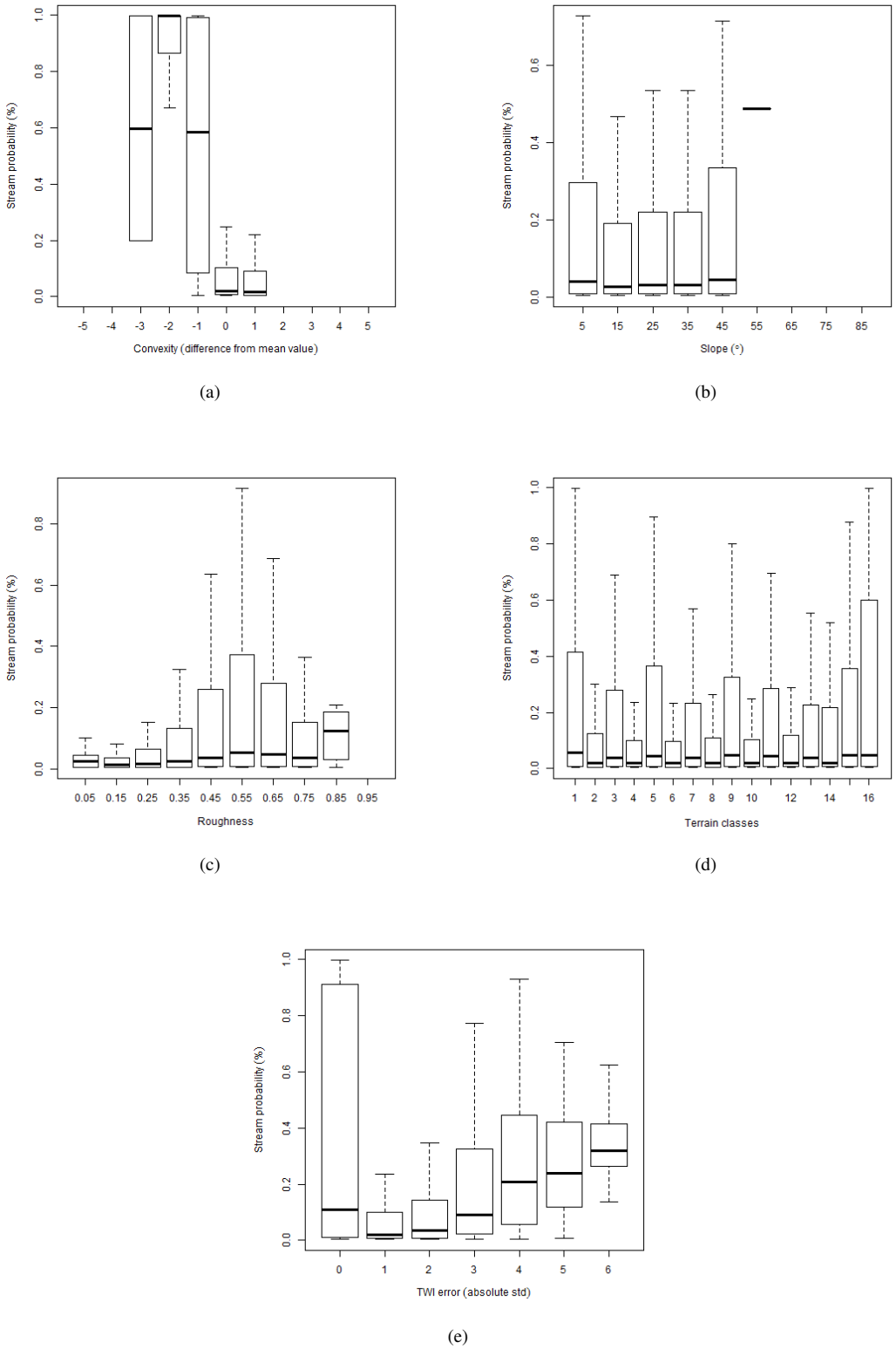


Figure 5.21: Box plots showing the relationship between the probability of a cell belonging to a stream and the terrain characteristics (a) convexity, (b) slope, (c) roughness, (d) the different terrain classes and (e) the MD∞ based TWI.

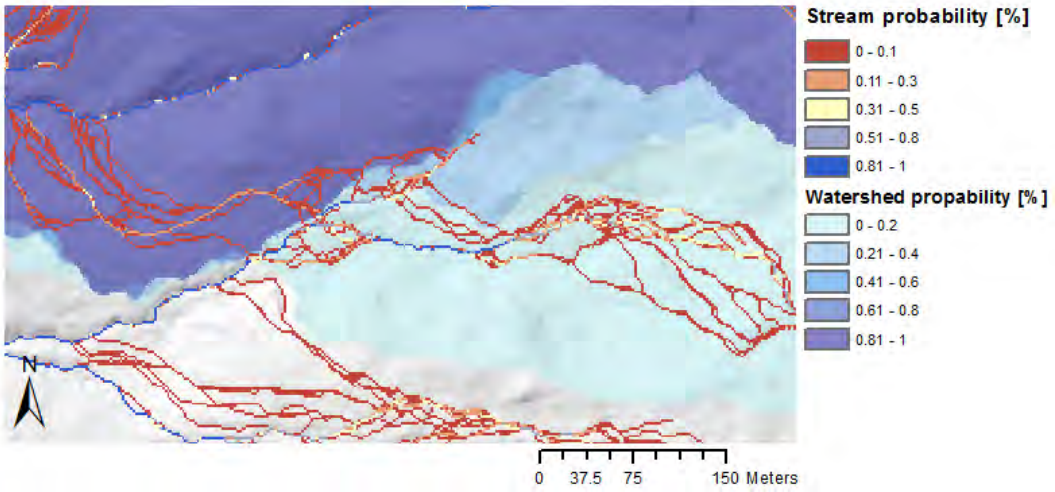


Figure 5.22: This critical region at the $zw04$ watershed border leads to substantial overestimations in some Monte Carlo simulation runs. There are multiple locations where streams enter the catchment area $zw04$ during the simulation, but the probability of these streams entering the catchment area is relatively low.

The uncertainty assessment of watersheds was done for three different catchment areas located in the Zwackentobel. The results are shown in Figure 5.23 and as we can see, the size of the area can vary substantially (up to $\pm 20\%$) depending on the study site. However, larger variations in size have a very low probability. The distribution of the intermediate results of the Monte Carlo simulation are shown in Figure 5.24. The corresponding statistical measurements such as the mean size of the area are outlined in Table 5.3. The study area $zw04$ shows few outliers representing substantially larger area sizes, whereas the study area $zw07$ shows few outliers representing substantial lower area sizes. This shows how small elevation errors may have a significant impact on watershed computations. Typically for watersheds, the uncertain regions can be found at the borders of the watersheds. The uncertainty basically depends on streams near the borders of the watershed. If a stream flows into the watershed and contributes to the flow accumulation of the outlet, the watershed is extended by the upslope area of this stream. If a stream leaves the watershed and does not contribute anymore to the specified outlet, the upslope area of this stream does not belong to the calculated watershed anymore. An example of such an uncertain region in the $zw04$ watershed calculation is illustrated in Figure 5.22.

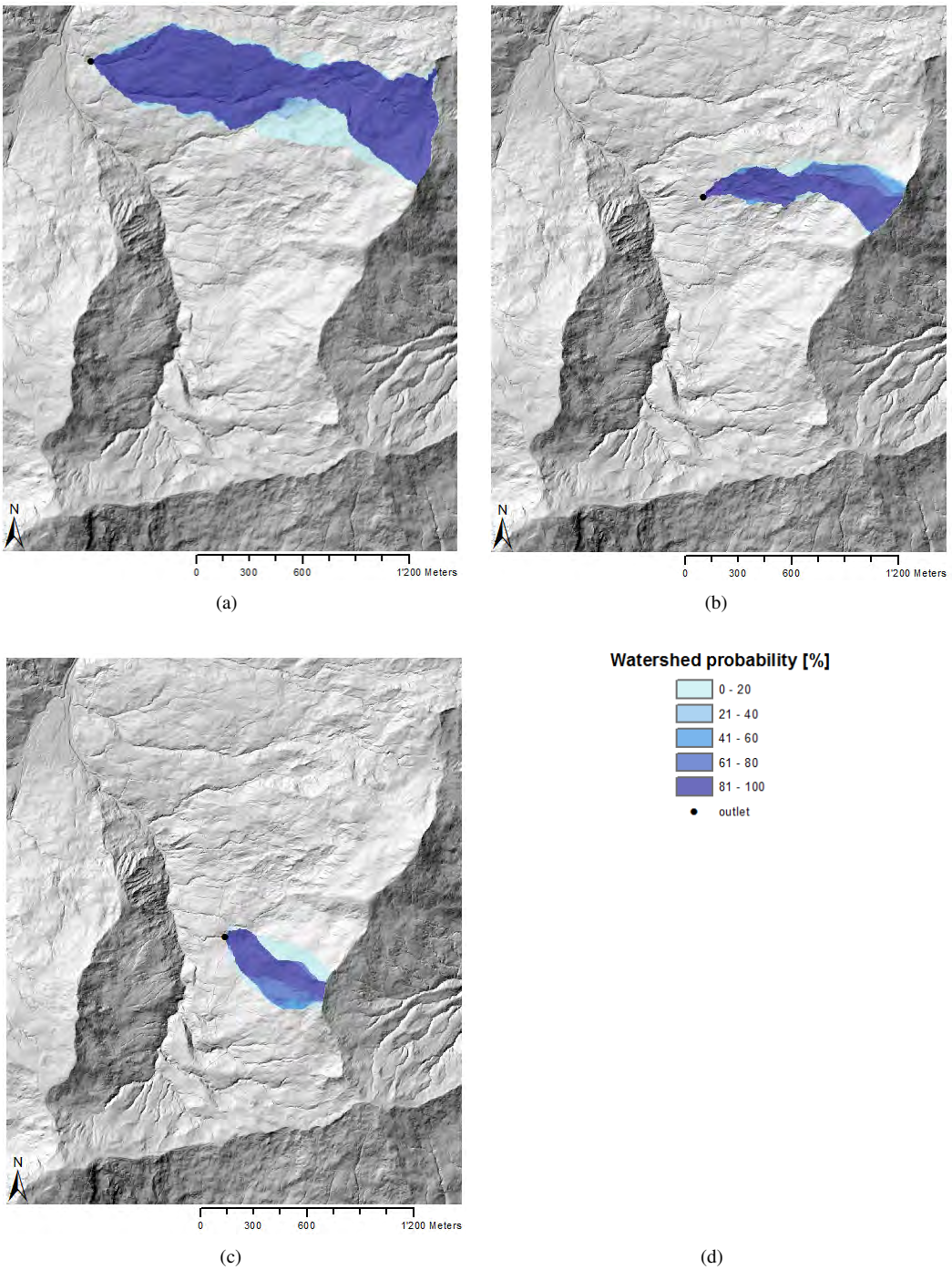


Figure 5.23: Statistical rasters showing the probability of a cell belonging to a watershed. These catchment areas are named as (a) $zw04$, (b) $zw07$ and (c) $zw11$. The outlets are marked as black points.

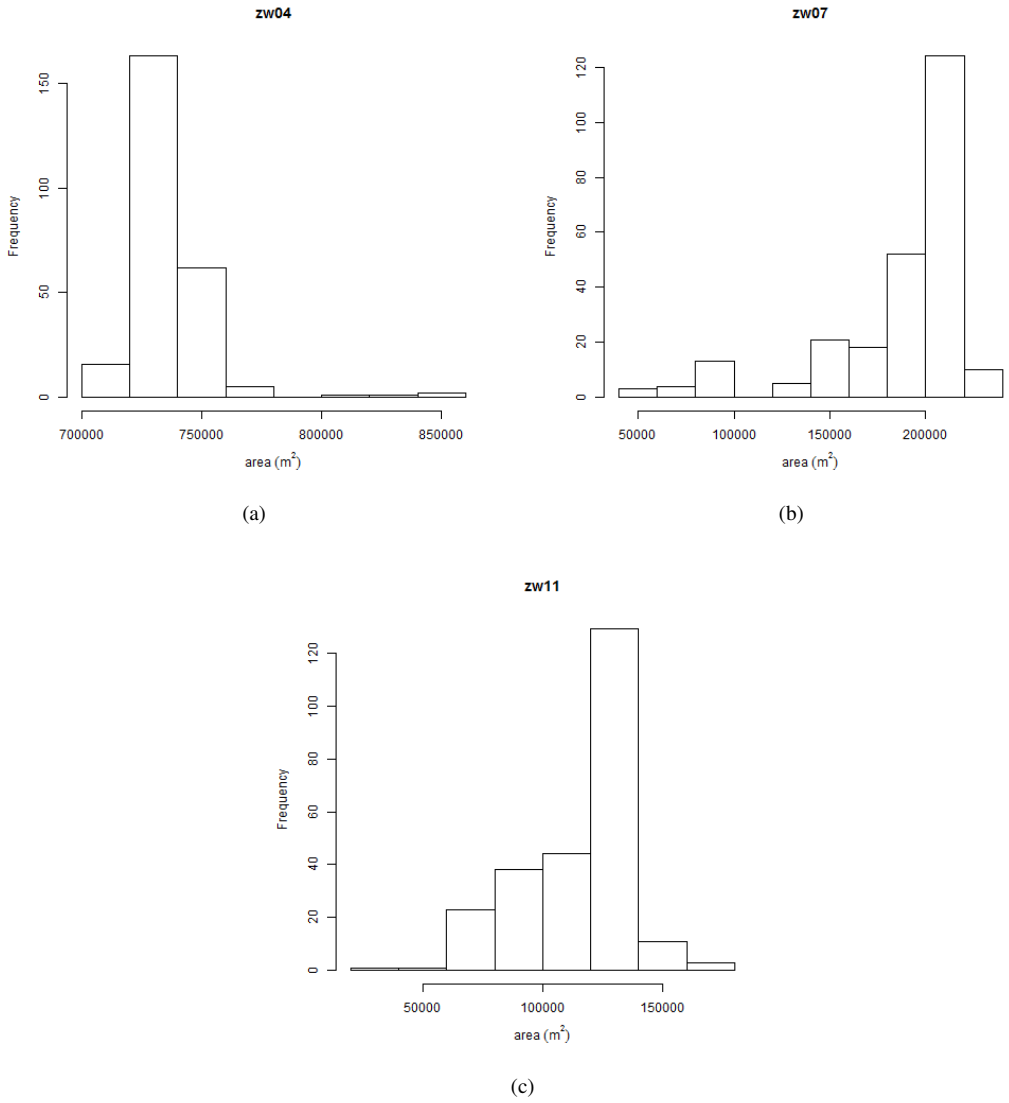


Figure 5.24: Histograms showing the distribution of the Monte Carlo simulation results for 250 simulation runs. (a) $zw04$, (b) $zw07$ and (c) $zw11$. The statistical measurements are shown in Table 5.3.

Watershed	#Runs	Mean [m^2]	STD [m^2]	RSTD	Min [m^2]	Max [m^2]
zw04	250	736989	17735	0.02	709800	858700
zw07	250	186236	38208	0.20	53770	227700
zw11	250	114291	21810	0.19	31820	163100

Table 5.3: Statistical measurements of the watershed histograms (Figure 5.24).

5.3 Impact of Different Uncertainty Model Parameters

The adjustments regarding the LiDAR measurements errors in steep terrain and dense canopy areas and the spatial autocorrelation range used to apply the autocorrelated part of error to the error surfaces have been analyzed. All these parameters control the spatial structure of error and modify the error surface applied to the original DEM in each simulation run. It is of interest to know if and how strong the influence of these parameters is on the derived results of an uncertainty analysis. Especially, the impact of the chosen spatial autocorrelation range which had to be estimated should be analyzed in more detail because it may directly determine the degree of uncertainty in an uncertainty assessment.

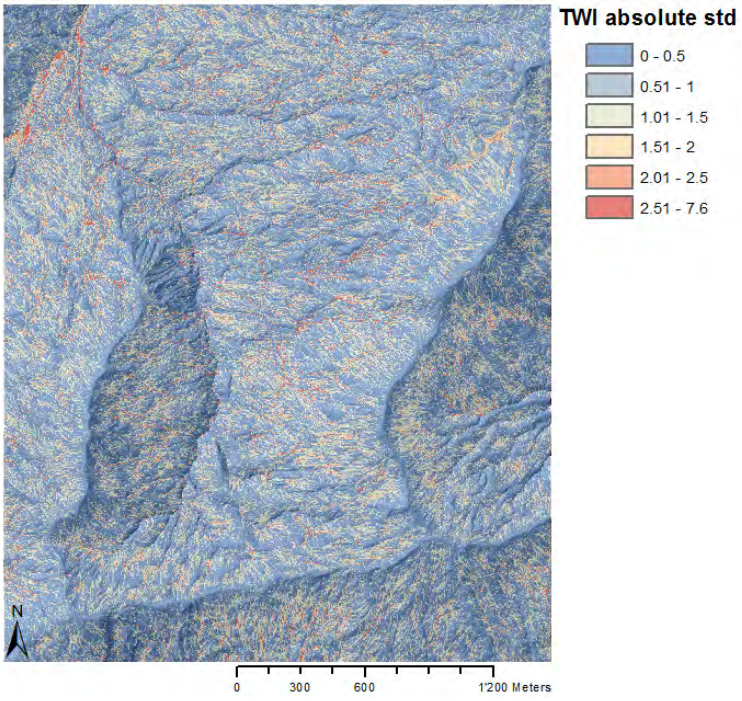
5.3.1 LiDAR based Uncertainty Model Parameters Slope and Canopy

The analysis was done calculating the TWI based on the D8 flow routing algorithm with 400 simulation runs. The spatial autocorrelation range was set to 40 m. The TWI was chosen because it is commonly used in hydrological applications and incorporates the uncertainty of the local slope and the flow accumulation area. The D8 flow routing algorithm was the most sensitive in previous analysis and thus was chosen for this analysis.

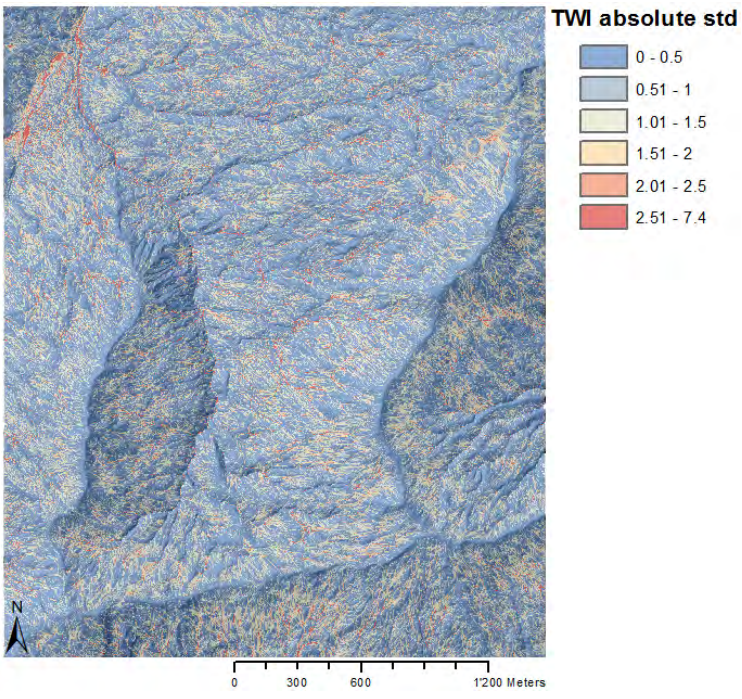
The results of the uncertainty analysis using no error surface adjustments in steep terrain respectively in dense canopy areas are illustrated in Figure 5.25. Differences can almost not be seen in both cases. In order to better visualize the differences, the differences to the original analysis shown in Figure 5.10(a) were computed (original minus adjusted result) for both cases and are presented in Figure 5.26. The results show that if the uncertainty model does not adjust the error according to the steepness, flat areas show a higher uncertainty in the TWI calculations, whereas steep terrains show a smaller uncertainty. Similar behavior can be seen in the case where the error was not increased in forested areas. Here, the uncertainty increased in non-forested areas and decreased in forested areas. The changes can indeed be significant. In the uncertainty analysis without the error surface adjustments according to steepness, the TWI error can differ up to 3.6 (std). In the case where the error is not increased in forested terrain, the differences can reach up to 4.6 which is huge compared to the maximum TWI calculated in this study area which is just below 8.

This behavior can be explained by the way the uncertainty model works. Because the standard deviation of the error surface is always 0.5 m in this study, the error surface is rescaled after the error increments in steep and forested terrain due to the fact that these adjustments lead to a higher standard deviation than 0.5 m. This rescaling process also reduces the error for cells located in non-forested and flat terrain. Previous analyses have shown that especially flat areas are subject to pronounced uncertainties. If these spatial structures of error are not applied, the standard deviation of the error in such terrain remains at 0.5 m and hence increases uncertainty. The terrain in the top-left corner of the study area is flat and not forested, thus in both cases the most significant changes in the TWI uncertainty analysis can be observed in this area. However, no changes can be observed globally due to fact that the standard deviation remains at 0.5 m. Figures 5.25(a), 5.10(a) and 5.25(b) show very similar results. The mean error in all three cases is around 0.66. Plots neither show any significant changes. Hence, one might conclude that these parameters have indeed an impact on the DEM derived results, but the effect is spatially depending; some areas show higher, whereas others

show lower uncertainty in the TWI calculation. The results would probably be slightly different showing lower differences when using multiple flow direction algorithms. However this assumption must still be verified.

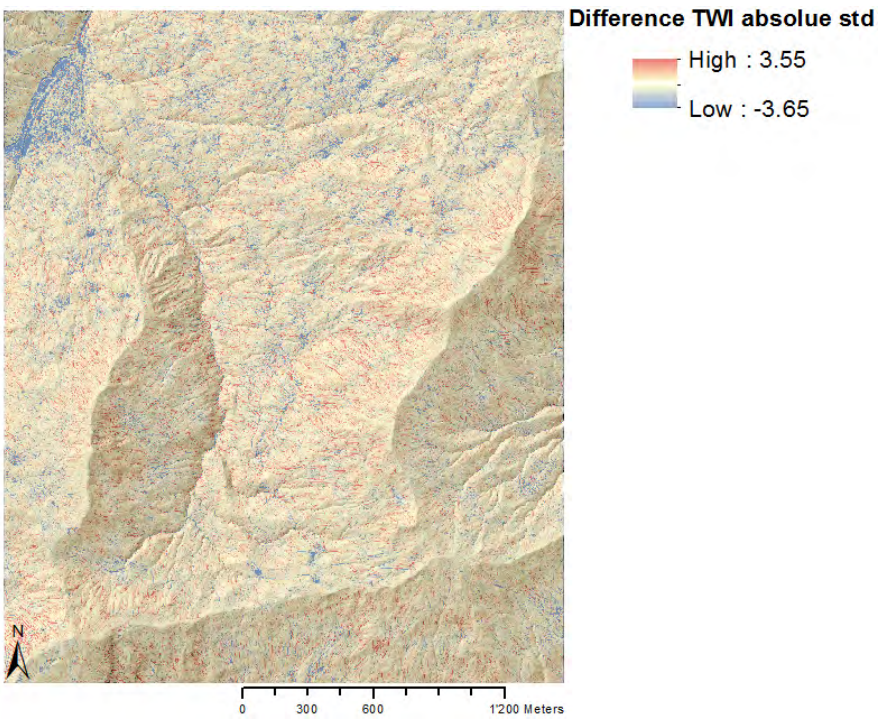


(a)

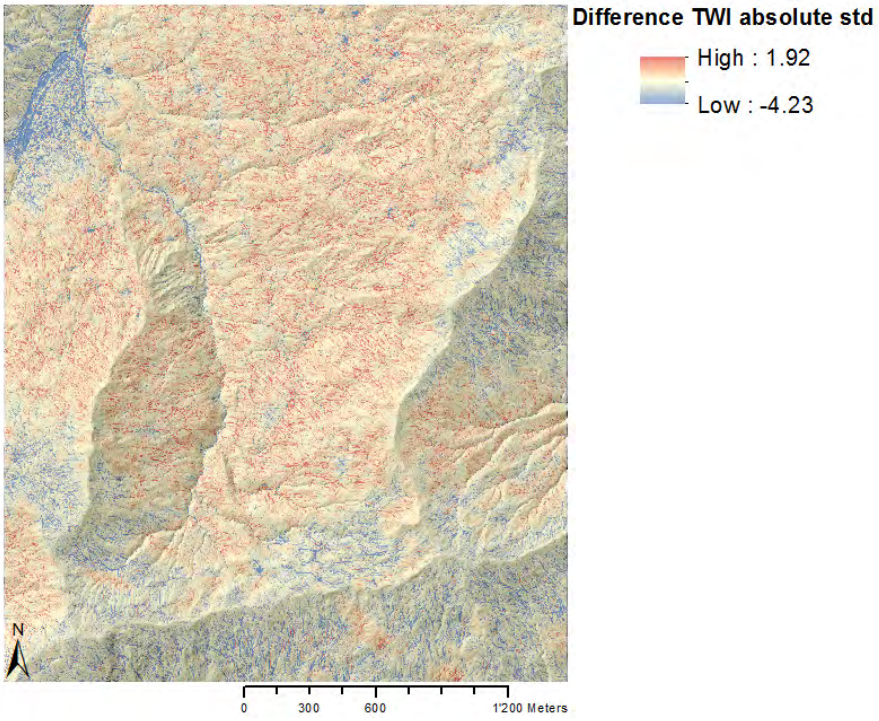


(b)

Figure 5.25: Statistical rasters showing the error (standard deviation) of the TWI using the D8 flow routing algorithm. The uncertainty model parameters regarding the slope of the terrain and the canopy for the error surface generation were changed: (a) The error was not increased according to the slope. (b) The error was not increased in forested areas.



(a)

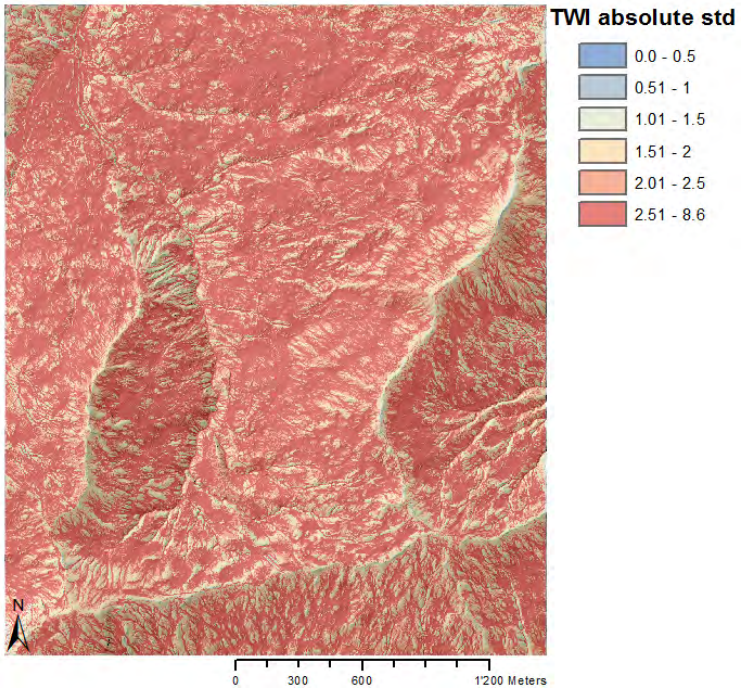


(b)

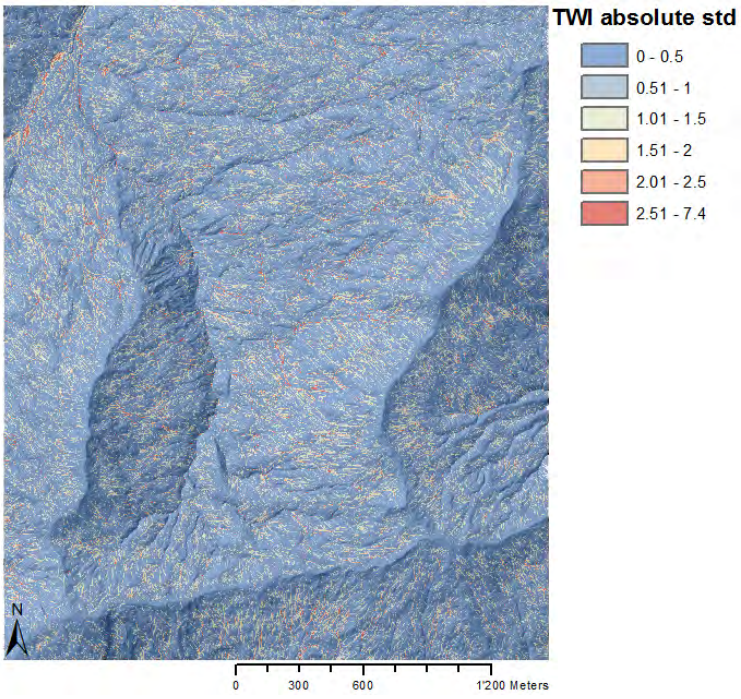
Figure 5.26: Statistical rasters showing the differences from the original uncertainty analysis of the D8 based TWI calculation and the D8 based TWI calculation representing the absolute standard deviation. The uncertainty model parameters regarding the slope of the terrain and the canopy for the error surface generation were changed: (a) The error was not increased according to the slope. (b) The error was not increased in forested areas.

5.3.2 Spatial Autocorrelation Range

These analyses were done calculating the TWI based on the D8 flow routing algorithm with 400 simulation runs. A worst case scenario, in which the spatial autocorrelation range was set to 0 m and a more conservative scenario with a range of 80 m were chosen. Figure 5.27 shows the result of the D8 based TWI uncertainty analysis (absolute standard deviation). As expected the uncertainty increases dramatically with a spatial autocorrelation range of 0 m (mean error $\pm$2.7), whereas a range of 80 m reduces the derived uncertainty (mean error $\pm$0.44) only slightly compared to the results using a range of 40 m (mean error $\pm$0.65). The histograms of these two results are illustrated in Figure 5.28 with the statistical parameters outlined in Table 5.4.

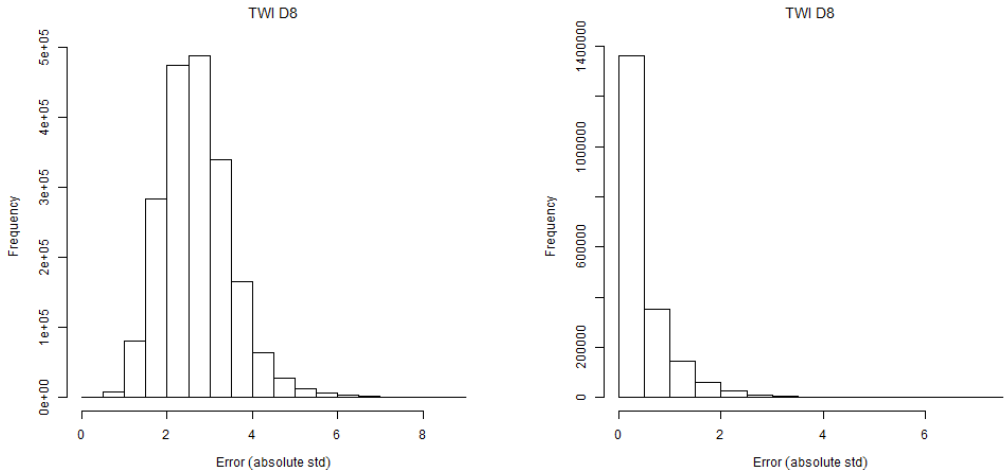


(a)



(b)

Figure 5.27: Statistical rasters showing the error (standard deviation) of the TWI using the D8 flow routing algorithm. The spatial autocorrelation used in the uncertainty model was changed to (a) 0 m and (b) 80 m.

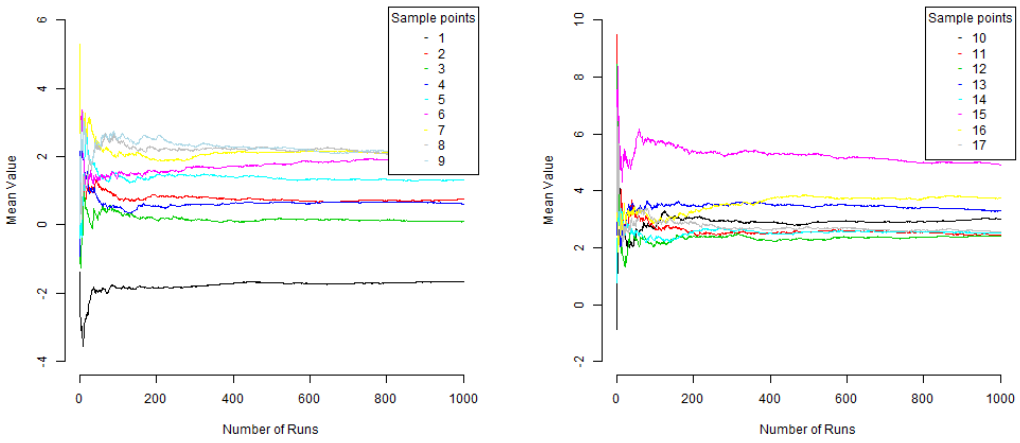


(a) (b)

Figure 5.28: Histograms showing the error (standard deviation) of the TWI using the D8 flow routing algorithm. The spatial autocorrelation used in the uncertainty model was changed to (a) 0 m. (b) 80 m. Statistical measurements are given in Table 5.4.

Range [m]	#Runs	Mean	STD
0	400	2.7	0.8
40	400	0.65	0.59
80	400	0.44	0.52

Table 5.4: Statistical measurements of the TWI histograms (Figures 5.28 and 5.11(a)) with different spatial autocorrelation ranges.



(a) (b)

Figure 5.29: Graphs showing the convergence of different sample locations (Figure 5.1(a)) on the DEM in the D8 based TWI calculation uncertainty analysis. The spatial autocorrelation used in the uncertainty model was changed to 0 m.

In Section 5.1, the convergence for the TWI using the D8 flow routing algorithm was tested with the result that the TWI computation converged after approximately 400 simulation runs. It is more than obvious that this result depends on the uncertainty of the DEM. So, how does the TWI calculation converge with a spatial autocorrelation range of 0 m where the DEM's surfaces are very rough? The results are illustrated in Figure 5.29. The computation of the TWI converges considerably later, even after 1000 simulation runs, some sample points still do not provide a 100% stable result. Out of these findings, it can be expected that higher spatial autocorrelation ranges will decrease the number of runs needed to get stable results.

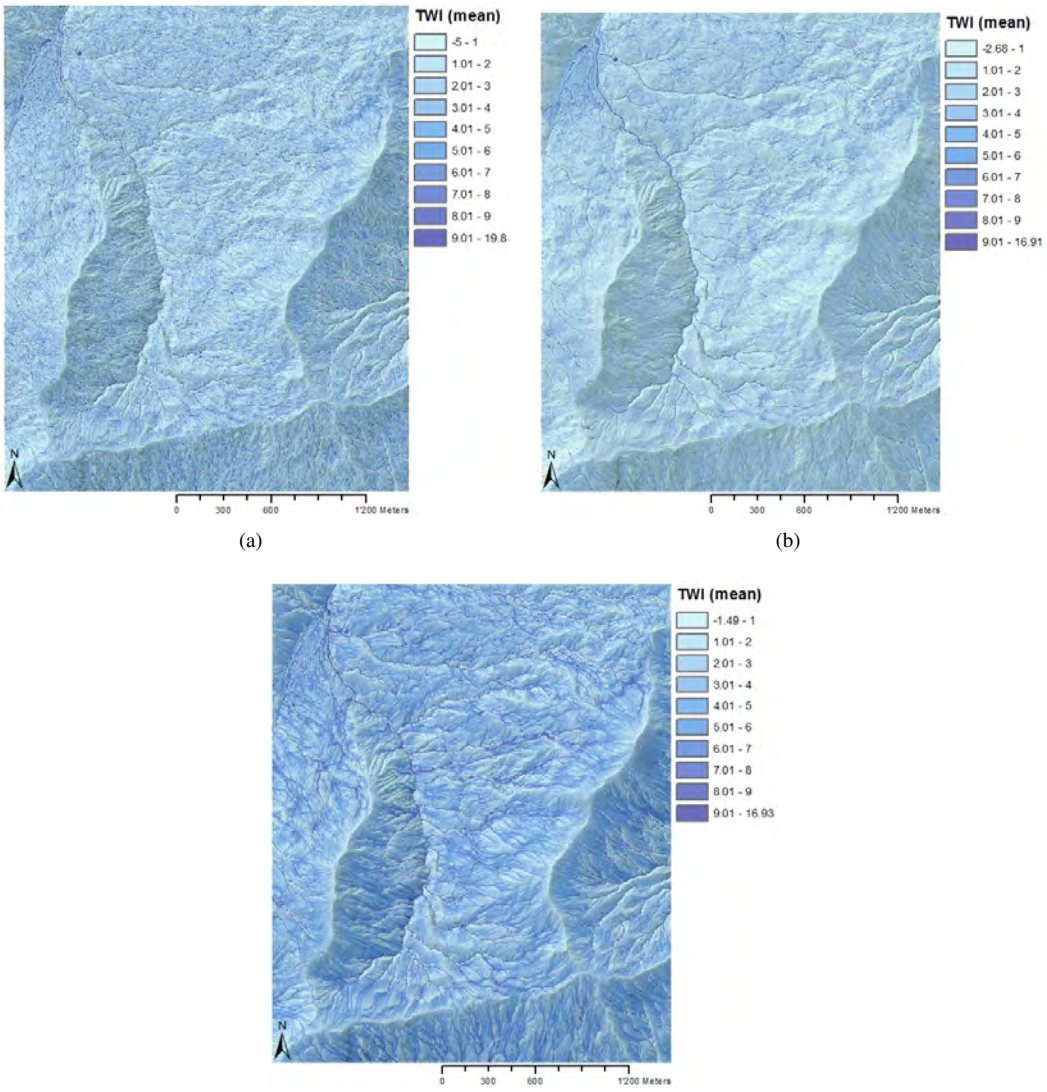


Figure 5.30: Grids showing the mean TWI calculation of 400 simulation runs with different flow routing algorithms and spatial autocorrelation ranges: (a) D8 with a range of 40 m, (b) D8 with a range of 0 m and (c) MD∞ with a range of 40 m.

The final result of the TWI calculation was achieved by computing the mean of all realizations. The TWI results based on the D8 flow routing algorithm with a spatial autocorrelation range of 0 and 40 m and based on the MD∞ flow routing algorithm with a spatial autocorrelation range of 40 m are given in Figure 5.30. The D8 based TWI with a spatial autocorrelation range of 40 m shows the classical linear features of D8 flow routing based computations. But, with an autocorrelation range of 0 m, the D8 flow directions start to vary so considerably in the uncertainty analysis that the D8 flow routing algorithm mutates to a multiple flow direction algorithm showing a very smooth TWI surface. However, the TWI values reported are lower compared to the MD∞ based flow routing algorithm. However, the determination of the most realistic TWI is not the scope of this thesis.

5.4 Using Coarser DEM Resolutions

This section evaluates if a coarser spatial resolution DEM will change the results of an uncertainty analysis compared to the high resolution DEM used previously. Several possibilities exist to produce a DEM with a coarser spatial resolution. A DEM with a native coarser resolution could be used or the LiDAR derived DEM could be resampled. For the resample approach, ArcGIS provides different methods: e.g. nearest neighbor or bilinear interpolation. A native DEM provided by Swisstopo (DHM25[1]) with a 25 m spatial resolution and resampled LiDAR derived DEMs (25 m) using the nearest neighbor and bilinear interpolation methods were used. Both resampled DEMs only differ slightly, differences in elevation up to 1.4 m could be observed which can be mainly found in steep areas. The native DEM provided by Swisstopo differs considerably from both resampled DEMs. Differences up to 19 m could be observed in steep areas, which is not surprisingly due to the fact that this dataset uses a completely different elevation data source (contour derived).

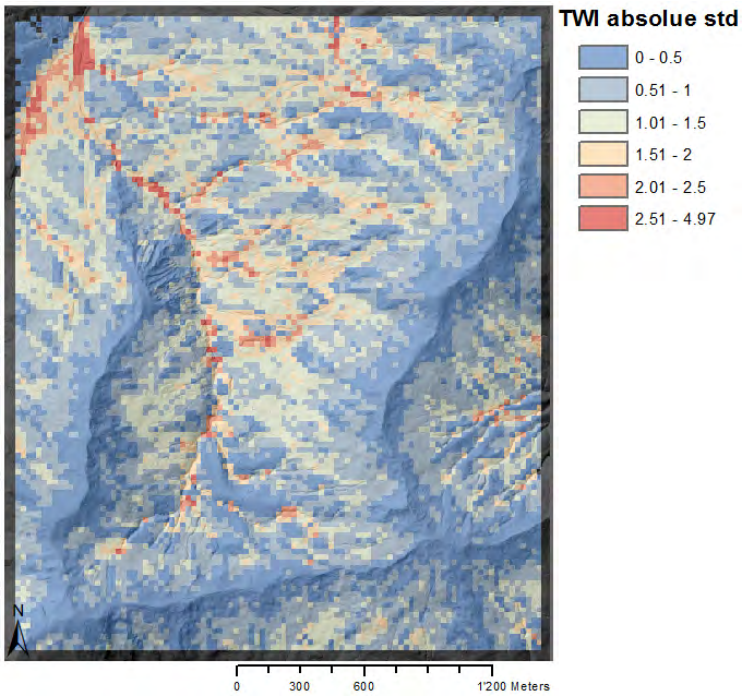
The RMSE of the LiDAR derived DEM is 0.5 m according to Swisstopo. The accuracy of the elevation data is lower in coarser resolution DEMs. For the native 25 m DEM, Swisstopo indicates a RMSE value of up to 8 m in alpine terrain. The resampled DEMs altered the original data and therefore caused additional uncertainty (Montgomery, 2001). An estimation of the elevation error can be done by comparing coarse resolution DEMs to DEMs with a higher spatial resolution. Computing the differences between the resampled DEMs and the LiDAR DEM results in error surfaces which can be used to extract the mean errors. Both resampled DEMs have a mean error of 4.3 m. The native DEM (DHM25) has a higher mean error of 6.5 m which is near the 8 m RMSE stated by Swisstopo. The acquired mean error can be used as new RMSE input value for the uncertainty model. In addition to the RMSE, the spatial autocorrelation range for the uncertainty analysis method is needed.

The DEM resampled with the nearest neighbor method was selected for the analysis in this section. The spatial autocorrelation range was estimated at up to 500 m or 20 cells. The simulated DEMs look realistic with that range and a range of 20 cells was also used for the analysis of the 2 m DEM. However, this range was chosen without any further justifications. Because previous analysis (Section 5.3) showed that the spatial autocorrelation range has an impact on the derived results, also a spatial autocorrelation range of 125 m was used in order to justify this finding.

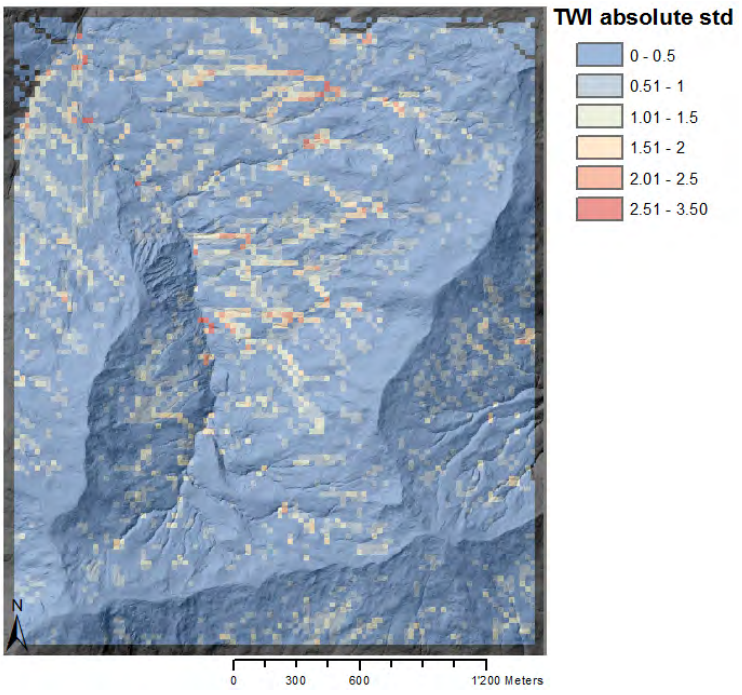
Due to the resampling, the DEM only has an extent of 103x123. This makes computations dramatically faster compared to the 2 m resolution DEM covering the same area. 1000 simulation runs computing the TWI based on the D8 flow routing algorithm were done in 20 seconds using four processes and the same system as described in Chapter 4. With the 2 m DEM, already 250 simulation runs need more than 200 seconds to complete (see Figure 4.2(a)). The results of this uncertainty assessment are illustrated in Figure 5.31 with the corresponding histograms and statistical measurements shown in Figure 5.32 respectively Table 5.5. The spatial structure of error is given in Figure 5.33. The uncertainty in this case seems to be relatively independent from the spatial resolution of the DEM. The mean error (std) is slightly higher with 0.8 (125 m range) and substantially lower with 0.33 (500 m range) compared to the 0.65 of the original LiDAR derived DEM (Table 5.2). The histograms (Figure 5.32 and 5.11(a)) also show a very similar distribution of error. Differences can be observed in the spatial structure of error. Due to the resampling the terrain became flatter; the steepest slope is now 54°. TWI values derived in flat areas still have a higher uncertainty but the

[1] http://www.swisstopo.admin.ch/internet/swisstopo/de/home/products/height/dhm25.html

lowest error can be observed around 35° when using a spatial autocorrelation range of 125 m (Figure 5.33(a)). The influence of slope decreases when a range of 500 m is used (Figure 5.33(c)). The error also considerably decreases in convex terrain when using a range of 125 m (Figure 5.33(b)). With a range of 500 m, errors are only slightly higher in concave compared to convex reliefs while errors are highest neither in convex nor in concave terrain (Figure 5.33(d)).

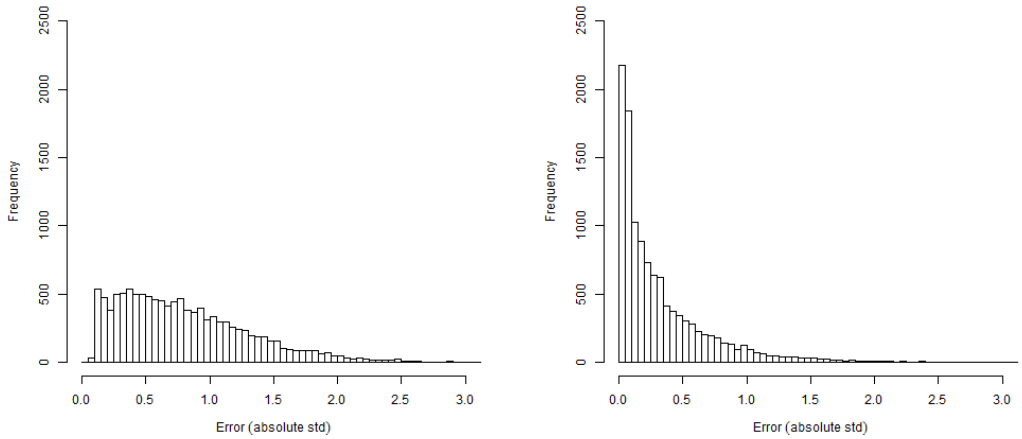


(a)



(b)

Figure 5.31: Grids showing the absolute standard deviation of the D8 flow routing algorithm based TWI uncertainty analysis using 400 simulation runs, a spatial resolution of 25 m, a RMSE of 4.3 m and a spatial autocorrelation range of (a) 125 m and (b) 500 m. The black cells represent no data. TauDEM does not compute the flow accumulation for contaminated cells because their flow accumulation is affected by cells which are not part of the DEM.

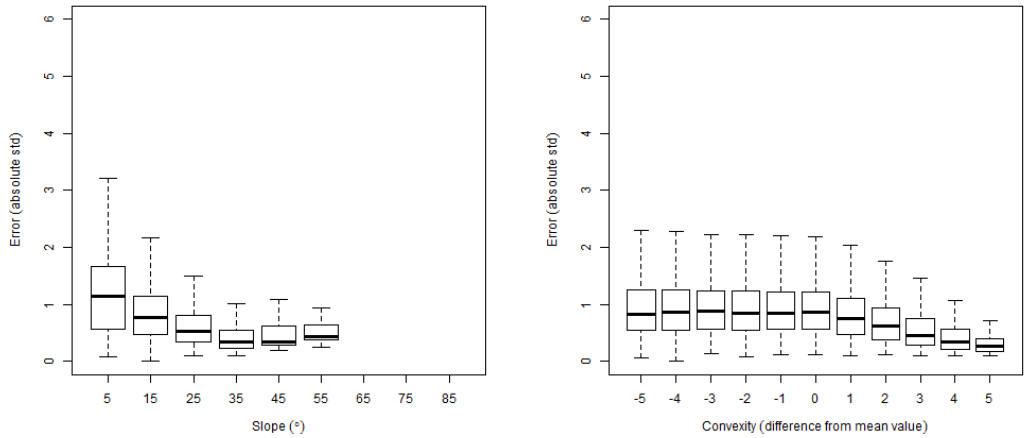


(a) (b)

Figure 5.32: Histograms showing the distribution of errors of the TWI computation based on the D8 flow routing algorithm. The uncertainty analysis was done with 400 simulation runs, a spatial resolution of 25 m, a RMSE of 4.3 m and a spatial autocorrelation range of (a) 125 m and (b) 500 m.

Resolution [m]	Range [m]	#Runs	Mean	STD
25	125	400	0.8	0.53
25	500	400	0.33	0.38
2	40	400	0.65	0.59

Table 5.5: Statistical measurements of the TWI histograms (Figures 5.32 and 5.11(a)).



(a) (b)

(c) (d)

Figure 5.33: Box plots showing the spatial structure of error of the TWI computation based on the D8 flow routing algorithm. The uncertainty analysis was done with 400 simulation runs, a spatial resolution of 25 m, a RMSE of 4.3 m and a spatial autocorrelation range of (a,b) 125 m and (c,d) 500 m.

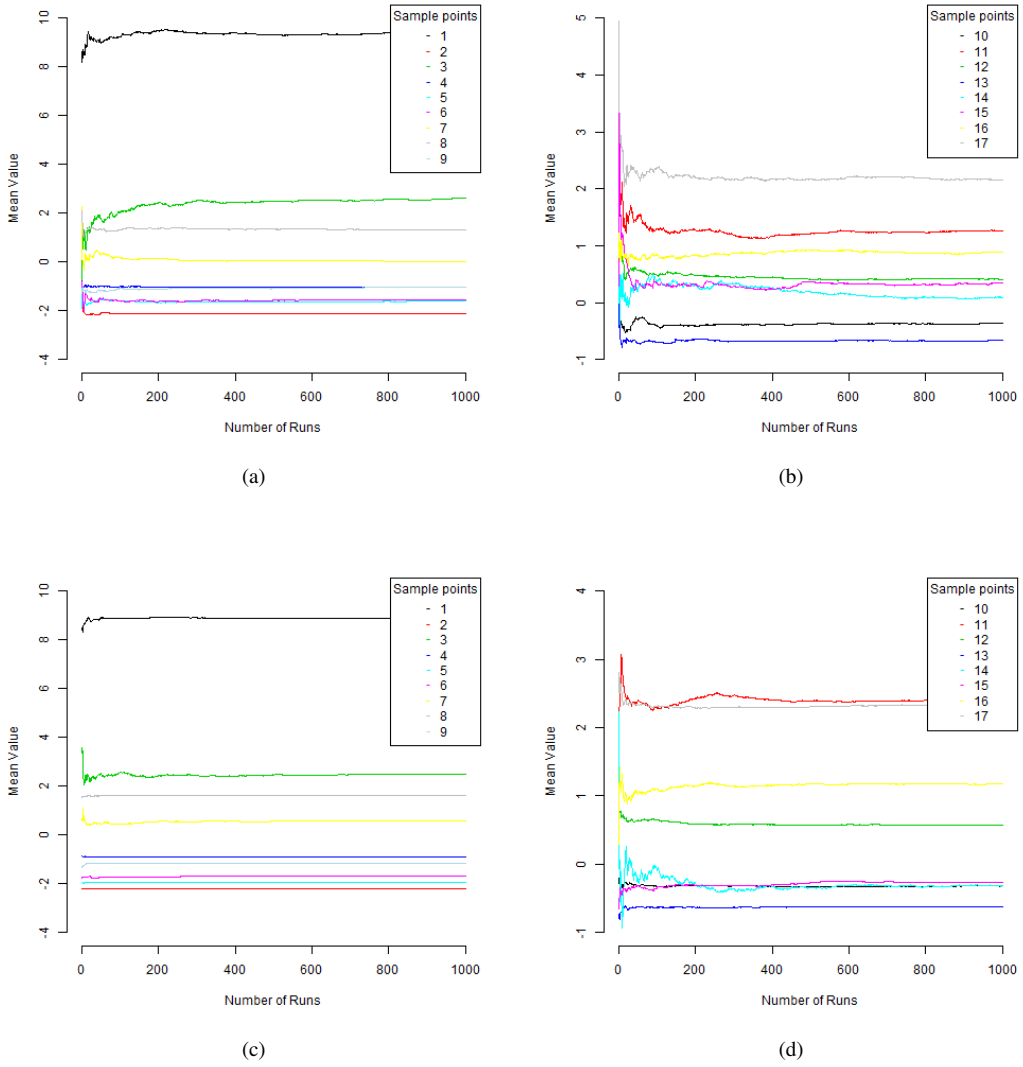
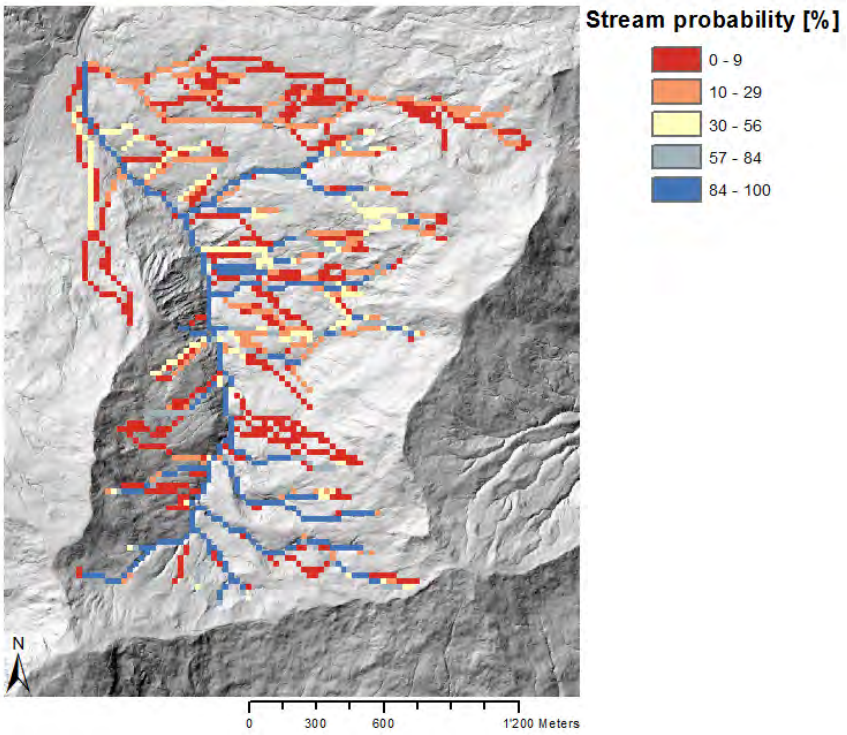


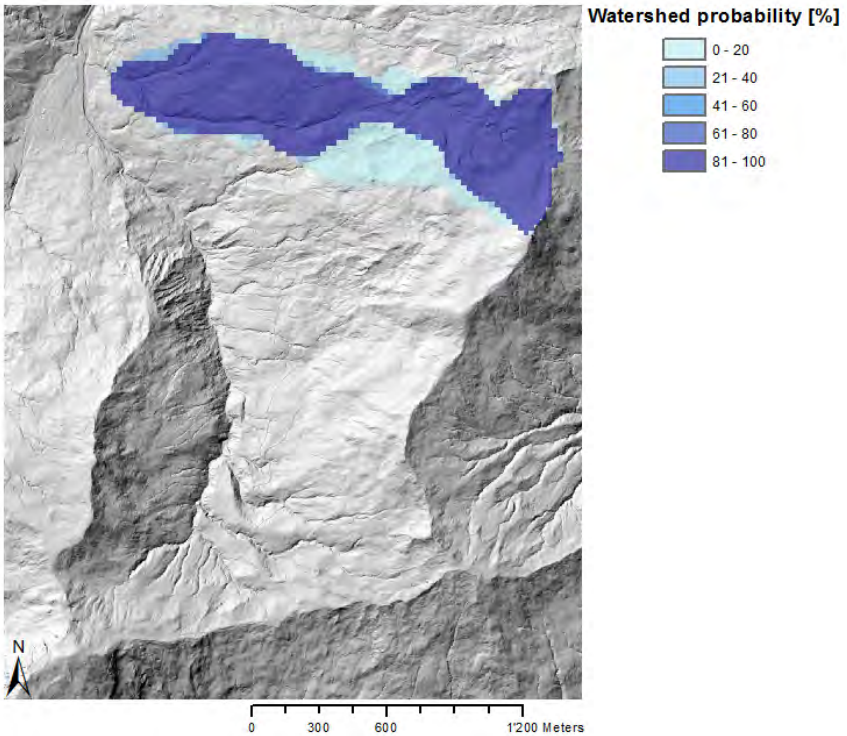
Figure 5.34: Graphs showing the convergence of the D8 based TWI calculation using the 25 m DEM with a spatial autocorrelation range of (a,b) 125 m and (c,d) 500 m.

The convergence test was performed once again with the 25 m DEM. The results are shown in Figure 5.34. The graphs highlight that the TWI computation appears to converge after 600 runs using a spatial autocorrelation range of 125 m. However, some sample points are still not 100% stable. In this case, significantly more runs are needed compared to the 2 m DEM where the average TWI appeared to converge after 400 simulation runs (Figure 5.2) to achieve an appropriate uncertainty assessment. When using a spatial autocorrelation range of 500 m the number of runs needed to achieve stable results decreases and the result appears to converge already after 400 simulation runs which is equal to the result obtained by using the high resolution DEM.

The stream network for the Zwackentobel and the watershed for the Erlenbach ($zw04$) were extracted using the 25 m DEM and a spatial autocorrelation range of 500 m. The uncertainty maps of the stream network and watershed are illustrated in Figure 5.35. In general, the results are mostly similar to the one obtained by using the 2 m DEM (Section 5.2). Figure 5.36(a) shows the convergence test for the watershed extraction. The watershed calculation appears to converge once again at around 150 simulation runs. However, the results are very unstable below 100 runs which is considerably higher compared to the result using the 2 m DEM (50 runs) (Figure 5.3). The distribution shown in Figure 5.36(b) is also very similar. The outliers are mostly upwards, overestimating the area of the watershed. Interesting is that the average size of the area is almost the same after 250 simulation runs (Tables 5.6 and 5.3). The differences are negligible with 736989 (2 m DEM) to 737797 m^2 (25 m DEM). However, the error is substantially higher ($\pm 6\%$ to $\pm 2\%$), explaining the differences in the convergence graphs. The spatial pattern of uncertainty for the stream network extraction is illustrated in Figure 5.37. The slope does not appear to have an influence on the uncertainty of stream extraction. This result is similar to the one obtained for the 2 m DEM (Figure 5.20(b)). In contrast to the 2 m DEM result (Figure 5.20(a)), the error of mapping streams increases with higher convexity of the terrain.

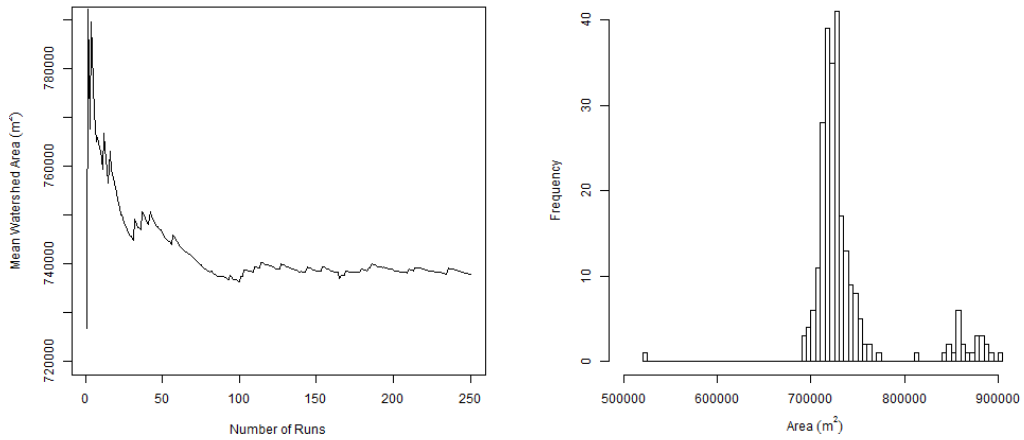


(a)



(b)

Figure 5.35: Grids showing the extracted stream network for the Zwackentobel (a) and the watershed for the Erlenbach (b) using the 25 m DEM (RMSE = 4.3, range = 500 m).



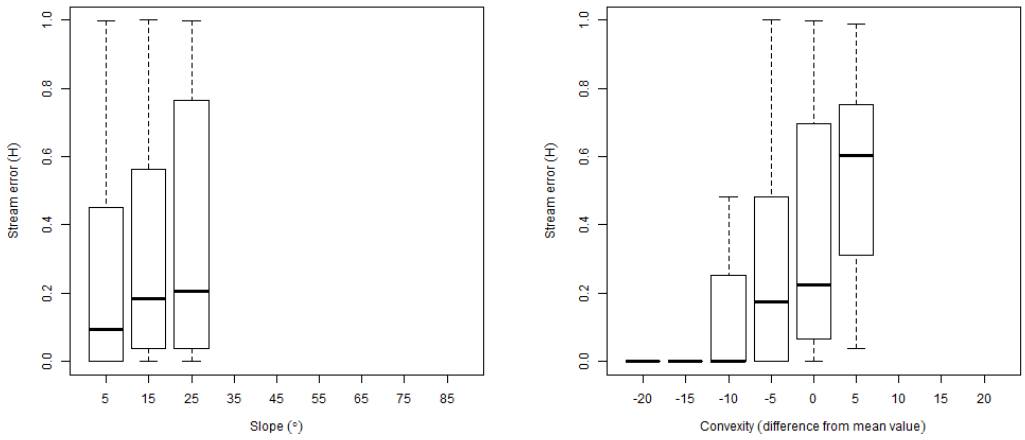
(a) (b)

Figure 5.36: (a) Graph showing the convergence of the watershed extraction. (b) Histogram illustrating the distribution of the calculated watershed areas. Statistical measurements are shown in Table 5.6 along with the 2 m resolution results.

Resolution [m]	Range [m]	#Runs	Mean (m^2)	STD (m^2)	RSTD	Min (m^2)	Max (m^2)
25	500	250	737797	47129	0.06	523100	901900
2	40	250	736989	17735	0.02	709800	858700

Table 5.6: Statistical measurements of the watershed histograms (Figures 5.36(b) and 5.24(a)).

Finally, one might conclude that the uncertainty of derived products in coarser resolution DEMs is very similar to the uncertainty observed when using high resolution DEMs. The analysis showed that the spatial autocorrelation range has such a large impact on the uncertainty assessment so that the spatial resolution of the DEM plays a minor role. However, this conclusion could be very case-specific and may change using different model inputs such as the RMSE or DEM.



(a) (b)

Figure 5.37: Box plots showing the spatial structure of error of the stream network extraction. They show the relationship between the stream error (H) and (a) slope and (b) convexity.

5.5 Possible Application in Recent Hydrological Research

Benjamin Fischer from the H2K unit of the University of Zurich investigates the relationship between DEM derived indices and field measurements such as the quality of water. He, for example, uses the TWI as a topographic index and the amount of ^{18}O in the water, both measured at multiple sampling sites. His analysis showed that there is no significant relation between the TWI and the amount of ^{18}O in the water (personal communication, January 7, 2013). However, it might be that uncertainties in the data may have affected the results and lead to an incorrect conclusion. The proposed uncertainty approach allows to at least improve the quality of the TWI data. It is possible to show how considerably a TWI value varies and it is possible to provide, for example, a final averaged TWI value for each particular sampling site. Using this data may lead to a different conclusion in his investigation (Figure 5.38). In this case, the uncertainty analysis using the mean value did not reveal any significant under- or overestimated TWI values. The plots look almost exactly the same (Figures 5.38(a) and 5.38(c)). Hence, the conclusion remains the same so far. However, the TWI varies substantially at some sample points (Figure 5.38(b)). Removing such uncertain sampling points from the correlation analysis probably leads to a different result (Figure 5.38(d)). But even this method does not change the conclusion that there is no significant correlation between ^{18}O and the TWI. However, this example shows that results of digital terrain analysis are uncertain and that these uncertainties may affect investigations building on digital terrains analysis.

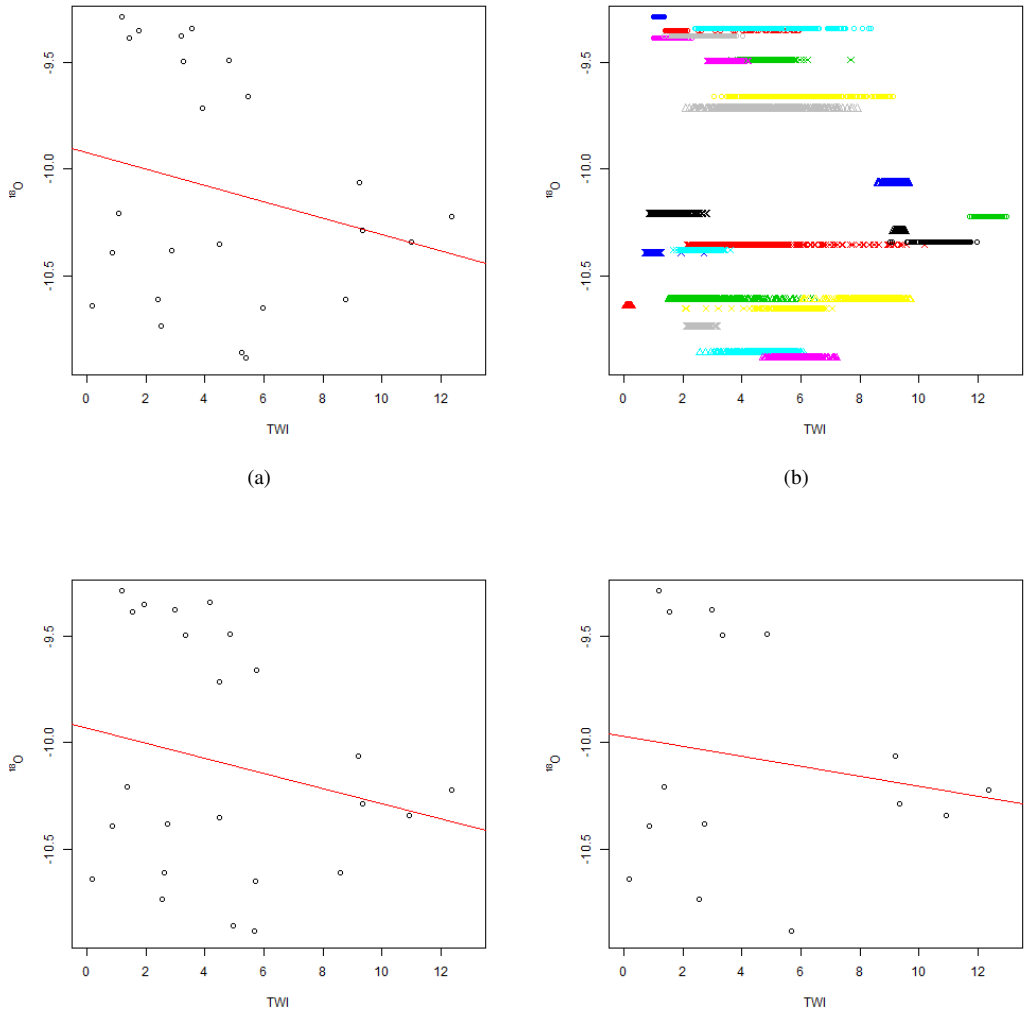


Figure 5.38: Plots showing the relationship between the TWI (MD∞) and the amount of ^{18}O in the water using 24 sampling sites: (a) No uncertainty analysis. Uncertainty analysis (#Runs=400, RMSE=0.5, R=40) illustrating (b) all calculated TWI values for each sampling ponit, (c) the mean TWI for each sampling point and (d) the mean TWI for each sampling point having a standard deviation lower than 0.5. In addition, regression lines were added to the plots.

6 Discussion

This chapter discusses the achieved results of this thesis. First, the proposed uncertainty analysis approach based on TauDEM is evaluated and suggestions for improvements are made (Section 6.1). Section 6.2 discusses the findings for hydrological applications. In the following Section 6.3 the research questions given as the aims of this study are answered.

6.1 Proposed Uncertainty Analysis Approach

The programming task was one of the main focuses of this thesis. The goal was to provide an uncertainty analysis approach with highly efficient algorithms to evaluate various hydrological models regarding their uncertainty in digital terrain modeling using high resolution DEMs. With this approach we are getting closer to the final goal of having an "uncertainty button" in a GIS (see Heuvelink, 2006). Compared to other uncertainty analysis approaches in digital terrain analysis (e.g. Wechsler, 2007; Hengl et al., 2010) the approach proposed in this thesis provides the user with a suite of efficient hydrological algorithms paired with numerous statistical evaluation options enabling error propagation analysis for large datasets independent of a GIS software suite. Tesfa et al. (2011) showed that the execution time of the algorithms provided by TauDEM is considerably lower for common hydrological algorithms compared to commercial GIS software suites such as ArcGIS which is confirmed by the performance evaluation in Chapter 4. In addition to the efficiency, the parallelization can improve the execution time of error propagation analysis even further. With the test system described in Chapter 4, an acceleration of up to a factor of three was achieved by running the algorithms in parallel on four processing cores. This is very crucial for uncertainty analysis in digital terrain modeling using the computationally very demanding Monte Carlo simulation approach in combination with large datasets such as LiDAR derived DEMs. The time needed for such analysis is seen as one of the main reasons why error propagation analysis receive little attention, especially when working with complex LiDAR data. With the proposed approach, this aspect should become less important.

However, the uncertainty analysis approach proposed by this thesis is also not the perfect and desired "uncertainty button". Building a geostatistical model for building error surfaces representing the DEM error includes a number of parameters affecting the results of the error propagation analysis. Often there are no control points available to evaluate the spatial structure of error in a detailed way. Usually, users are only provided with a statistical estimator such as the RMSE describing the error in the DEM. Hence, the generation of accurate error surfaces is a challenge. Without knowledge of the spatial structure of error, it is not possible to fit a correlogram or (semi)variogram like it is used in SGS approaches as demonstrated in some recent studies (e.g. Hengl et al., 2010). Therefore a process convolution approach (e.g. Oksanen and Sarjakoski, 2005b; Oksanen, 2006; Wechsler, 2007; Hebeler and Purves, 2008) using an uncalibrated correlation function should be

used. Because the study mainly focuses on high resolution DEMs derived from LiDAR data, the geostatistical model could be improved by considering knowledge of spatial structure of error in LiDAR derived DEMs. This approach is based on the method proposed by Ulmer (2007) with slight modifications to some parameters. However, because of the lack of control points, a validation if the resulting error surfaces including the spatial structure of LiDAR derived DEM error are more realistic could not be carried out. The results only show that such adjustments in the geostatistical model have some impact on the error propagation analysis (Chapter 5). Besides the issues in the methodology, there also exist technical problems which are outlined in Chapter 4. Memory leaks in the watershed and stream network computation limit the maximum amount of possible simulation runs and an inefficient algorithm to remove flats on DEMs slows down both of the implemented multiple flow direction routing algorithms MD8 and MD∞.

The proposed uncertainty analysis approach could be improved by a validation of the implemented spatial structure of error concept. Actually, all parameters used in this process were based on literature and were not verified by own evaluations. The proposed approach could also be extended by a SGS method, in case that control points would be available. This may lead to more realistic error surfaces improving the quality of uncertainty analysis. A comparison between the SGS and the process convolution approach could also be interesting. In addition, the previously mentioned technical issues should be resolved for a fully functional uncertainty analysis.

6.2 Advantages in Hydrological Applications

In Section 5.5, some results showing the application of uncertainty analysis in recent hydrological research are presented. As shown by Seibert et al. (2007), it is indeed quite frequent that scatter can be observed in the data affecting topographic correlation analysis. This scatter may also be a result of DEM uncertainty affecting the computed topographic indices such as the TWI. Benjamin Fischer from the H2K unit of the University of Zurich provided me with some data (^{18}O and TWI for multiple sample sites) which do not reveal any correlations. Recomputing the TWI values with the proposed uncertainty analysis approach allowed to determine all uncertain sample sites and removing them from the data. The removal of uncertain sample sites may reduce the scatter. However, in this example it does not change the conclusion that there is no correlation between ^{18}O and the TWI. However, the approach used seems to be suitable to improve such correlation analysis.

6.3 Research Questions

After the results have been obtained and interpreted, the research questions can be answered and discussed and responses can be given to the hypotheses formulated in the beginning of this thesis.

Research question 1: *Can the time needed for uncertainty analysis be reduced considerably with parallelized algorithms?*

Performance tests are done and discussed in Chapter 4. The results show that most algorithms used in the uncertainty analysis approach are very efficient and scale pretty well with additional processes. Some algorithms such as the D8 flow routing algorithm run up to three times faster when using four

processes instead of only one. However, the execution time does not scale linearly with the number of processes. The performance increases less considerably with each additional process. One reason can be found in Amdahl's law (Amdahl, 1967), saying that parallel code will effectively be bottlenecked by the slowest serial parts of the code. If there is infinite parallel computing power, the code runs as fast as the serial portions of the code. For example, writing and reading data is a serial part of this code. Another point are the communication costs between the processes. The processes can not run all algorithms independently from each other. But the results are in line with the findings of Tesfa et al. (2011) which is not surprising given the fact that the same parallelization approach and some identical algorithms were used. Due to the limits of the test system, performance tests using more than four processes could not be performed although the proposed analysis approach would support significantly more processes. But it can be expected that the results would correspond with those presented by Tesfa et al. (2011) meaning that the performance gain will slightly decrease with additional processes. Finally, it can be said that parallelization accelerates common hydrological algorithms and the proposed error propagation approach considerably. Compared to other uncertainty analysis studies (e.g. Hengl et al., 2010), the proposed approach does in most instances not suffer from performance issues, thus enabling users to assess the uncertainty of their hydrological products even on LiDAR based DEMs. However, the flat removing algorithm used in the MD8 and MD∞ flow routing algorithm computations causes some performance issues if the DEM is very large or contains many flats. More efficient algorithms exist to solve this issue (e.g. Magalhães et al., 2012), but have not been implemented due to the lack of time. However, the proposed uncertainty analysis approach was produced in a modular way easily allowing an extension with more efficient algorithms.

Working hypothesis 1: *Considerably lower execution times with an increasing number of parallel processes will be expected.*

This hypothesis can be partially verified according to the results outlined in Chapter 4. Digital terrain analysis algorithms are very suitable for parallelization approaches due to the lattice representation and the neighborhood operations (see Tesfa et al., 2011). In many algorithms, the result of a cell can be computed independently of the results of other cells which makes the parallelization relatively easy and efficient. However, the execution time does not decrease linearly with the number of processes used.

Research question 2: *How many realizations in a Monte Carlo simulation are sufficient for common hydrological algorithms? How well do they converge with an increasing number of runs?*

The convergence of different hydrological models was analyzed in Section 5.1. The tests were done using up to 1000 simulation runs. The classification of the topography with the method proposed by Iwahashi and Pike (2007) has proven to be very useful to analyze the convergence of hydrological products in combination with sampling points located in the different terrain classes. In this case (Zwackentobel DEM, RMSE = 0.5 m, R = 40 m, spatial resolution 2 m), the TWI based on the D8 flow routing algorithm appears to converge after 400 simulation runs. The results also show that this number can vary considerably in space. In convergence tests, the behavior of the sample points appears to be related to the terrain classes proposed by Iwahashi and Pike (2007). In some sample points, usually located in flat and rough terrain, the result still does not appear to converge after 1000 simulation runs. This shows that more simulation runs always lead to a more stable and thus more reliable uncertainty analysis results, especially in flat and rough terrains. But users do not have time to run 1000 simulation runs to achieve very stable results, hence a threshold is needed at which the

quality of the results is adequate enough which also depends on the users view. What can definitely be said is that less than 100 simulation runs lead to inappropriate results, even when using high resolution DEMs. The evaluated values are too noisy below 100 simulation runs. This confirms the results of other studies which indicated that in most cases the number of simulation runs should be at least 100 (e.g. Heuvelink, 1998).

The same analysis was performed using the TWI based on the MD∞ flow routing algorithm. In this case, the calculations appear to converge already after 200 simulation runs. Also, not that many outliers can be observed as before, all sample points reach very stable results before the 1000 simulation runs mark. However, below 100 runs the results are once again very noisy. So even with multiple flow direction algorithms more than 100 simulation runs are absolutely necessary for a valid uncertainty assessment. But as Oksanen (2006) mentioned, convergence analysis is always case-specific because it depends on the used terrain analysis algorithm, the parameters of the uncertainty model and the topography of the used DEM. But such analysis can deliver a rough guideline for users if they do not want to perform an own convergence test for their specific case. However, the proposed method using sample points has proven to be a suitable method to investigate the convergence of the TWI in error propagation analysis.

In addition, a convergence test for the watershed computation was done. Instead of using sample points as Oksanen (2006) did, the convergence of the watershed area size was investigated. In this case, the area already appears to converge after 150 simulation runs. The area sizes are very noisy below 50 simulation runs. A lower threshold can be observed compared to the previously mentioned TWI calculations and the results provided by Oksanen (2006) (500 simulation runs for watersheds). This result may be based on the fact that the Zwackentobel is very steep. The watershed computation relies more on the global relief than on the local neighborhood conditions, so that the watershed area is very stable in the study site. Because this result could be completely different for other DEMs or watershed locations, more research is needed to give a final conclusion.

Working hypothesis 2: *The results will vary spatially depending on the terrain characteristics.*

It can indeed be verified that the point at which a result converges differs in different terrain classes (Iwahashi and Pike, 2007). Basically the results prove that sample points showing a high uncertainty also exhibit noisier convergence graphs and that they appear to converge considerably later. Such sample points are often located in terrain classes describing rough and flat terrain. However, because only one sample point was chosen for each terrain class (Iwahashi and Pike, 2007), not enough data is available to conclude that there is a direct relation between the terrain classes (Iwahashi and Pike, 2007) and the convergence of sample points located in these classes. This has to be clarified in further investigations.

Research question 3: *How sensitively do the different hydrological algorithms behave in uncertainty analysis?*

This analysis was done in Section 5.2. First, the sensitivity of the flow routing algorithm was analyzed using the relative standard deviation as the statistical uncertainty estimator. The relative standard deviation is useful to describe uncertainty because it shows the percentage deviation of an uncertain value (e.g. Temme et al., 2009). As expected, the D8 flow routing algorithm shows the highest errors. The uncertainty map of the D8 flow accumulation (Figure 5.4(a)) looks very noisy and shows the typical linear features of the D8 flow routing algorithm. This behavior can be explained by how the D8 flow routing algorithm works compared to the multiple flow direction

algorithms (see Seibert and McGlynn, 2007). They basically show a very similar map looking considerably smoother. Between the MD8 and the MD∞ flow routing algorithm almost no differences can be observed (Figure 5.4). The highest errors can be observed for all flow routing algorithms in the upper-left corner of the DEM. This is due to the fact that errors are typically larger in flat and rough areas (Figures 5.6 and 5.8) and the upper-left corner is such a flat and rough area (Figures B.1(b) and B.1(c)). The results show that very flat areas are usually also very rough as shown in Figure 3.4(f). The errors reach up to $\pm$1990% in this area. Such high errors are not surprising. When the variation in the flow accumulation computation is very high but the mean value is still very low, such high errors may occur. Similarly high relative standard deviations were also observed by Temme et al. (2009) for soil redistribution (up to 1000%) and Wechsler (2007) for the D8 based upslope area computations (up to 2170% on average depending on the random field method used). The results also show that errors are not directly related to the convexity of the terrain, except for the D8 flow accumulation, where errors tend to increase in highly concave reliefs (Figure 5.7). Interesting is the fact that the multiple flow direction algorithms are extremely sensitive to the roughness of the terrain. If the terrain is very rough, the multiple flow direction algorithms produce a high amount of errors, even higher than the usually more sensitive single flow routing algorithm D8. This behavior cannot be explained and may be a case-specific exception. When the terrain is classified according to Iwahashi and Pike (2007), the average error of the D8 flow routing algorithm is generally more than twice as high across all classes compared to the multiple flow direction algorithms (Figure 5.9). In literature, different flow routing algorithms were often investigated regarding their accuracy to model flow in a realistic way (e.g. Zhou and Liu, 2002). These studies showed that multiple flow direction algorithms have many advantages. On the other hand, this thesis analyzed the behavior of different flow routing algorithms regarding their sensitivity in uncertain DEMs with the conclusion that the D8 is the most sensitive flow routing algorithm followed by the D∞ which is already considerably more robust. The MD8 and MD∞ flow routing algorithms which behaved almost the same way in the uncertainty analysis, are the most robust in the uncertainty analysis and thus are more suitable for extracting hydrological parameters from uncertain DEMs (see Seibert and McGlynn, 2007).

The same analysis was done calculating the TWI based on the D8 and MD∞ flow routing algorithm. These analyses incorporate the uncertainty of flow accumulation and local slope. The standard deviation was used to describe the uncertainty. The relative standard deviation tends to produce extreme values and may adversely affect statistical analysis. The observed average error for the D8 based TWI is 0.65 (Table 5.2). This result corresponds with the results of Wechsler (2007) who found an average error of 0.4 to 0.6 depending on the random field method used. Once again, low slopes lead to higher errors and the D8 based TWI is more sensitive to elevation uncertainties (Figure 5.13). The convexity has a similar impact on the uncertainty as shown before in the flow accumulation calculations (Figure 5.14). Errors are slightly higher in concave reliefs in both cases. The MD∞ based TWI reacts very sensitively to the roughness of the terrain, whereas in this case, the D8 based TWI behaves very robustly, but shows a relative constant and high error (Figure 5.15). When we take a look at Figure 5.16, we can see once again that in almost all terrain classes the error of the D8 based TWI is approximately twice as high compared to the MD∞ based TWI. Because the TWI is calculated from the slope and the flow accumulation, it is of interest to analyze if the uncertainty of the final TWI map is influenced more strongly by the slope or by the flow accumulation error. Figure 5.17 shows that the uncertainty of the TWI calculation mainly depends on the flow accumulation uncertainty. There cannot be seen a direct correlation between slope error and

TWI error. The flow accumulation errors are too huge and thus have the strongest impact on the TWI uncertainty assessment result. The calculated TWI values are not always normally distributed and can vary over a large range (Figure 5.12). Thus, results of hydrological models can be considerably over- or underestimated if DEM uncertainty is ignored.

In addition, the sensitivity of watersheds and stream networks was analyzed. To evaluate the impact of DEM uncertainty on watersheds, the probability of a cell belonging to the watershed was computed. The same approach was also used by Oksanen and Sarjakoski (2005b). This results in a map showing the watershed in a continuous form which allows to identify uncertain regions (Figure 5.23). Apparently, uncertain regions can always be found near the boundaries of the watershed. These uncertainties can lead to huge changes of the watershed size depending on the study site. Whereas the spatial extent of watershed $zw04$ varies by 20%, the spatial extent of watershed $zw11$ varies by over 519% between their smallest and largest extents (Table 5.3). Similar results were achieved by Hebeler and Purves (2009) who observed variations of up to 320% in their study site. Cells located in the center of the watershed usually always have almost a 100% chance to belong to the estimated watershed. This is also the reason why analyzing the spatial structure of error in this case does not make any sense. The watershed uncertainty is directly related to the stream errors. If a stream changes its direction in a realization and enters the watershed, the watershed is extended by the catchment area of this stream and whether a stream changes its direction or not depends on the local terrain characteristics. Usually very flat terrain leads to uncertain stream extractions (Figure 6.1). If a watershed is located in a flat relief the errors may be considerably higher than observed in this case, where the relief is very steep. As illustrated in Figures 5.20 and 5.21, it is difficult to find any spatial patterns of stream error. The problem is that one uncertain location (small cell neighborhood) can change the direction of a stream completely. Now, the topographic properties of all the following cells which belong to the stream do not necessarily have an impact on the stream uncertainty anymore but are classified by the model that way and thus influence the plots of spatial structure making them less meaningful. For example, if a stream reaches a road which happens in the performed analysis (Figure 6.1), the chance is 50% that the stream flows down the road instead of flowing straight through it (in reality the stream flows below the road). Obviously the flat road is the reason for the uncertain extraction of the stream network. However, all cells belonging to this stream which are located downwards the road and below the road are also modeled as very uncertain (probability = 50%, error (H) = 1) disregarding the topographic characteristics. This issue is also related to high resolution DEMs. Coarse resolution DEMs are not affected by such anthropogenic features which cause a lot of uncertainty because they are not detectable in this data. In addition, streams in concave reliefs have in general a high probability, whereas streams extracted in non convex or slightly convex areas show a small probability to be mapped (Figure 5.21). These results are in line with the findings of other studies conducted (e.g. Hengl et al., 2010). The results also identified that streams with lower Strahler stream orders show lower probabilities and can vary considerably in space (Figure 5.18(b)) (see Hengl et al., 2010). In addition, the probability map was transformed into the error space H (Figure 5.18(a)) as it was also done by Hengl et al. (2010). With this representation, locations which are highly uncertain in the stream extraction process are discovered. In contrast to the results provided by Hengl et al. (2010), it could not be confirmed that errors are higher in slightly convex reliefs. In this study, errors are slightly higher in concave reliefs, potentially due to the higher spatial resolution or different terrain characteristics of the study area. But the results show that errors are higher in terrain with high TWI errors and rough landforms (Figure 5.21).

Some locations in the field which were evaluated as very uncertain regions were also investigated. Figure 6.2 shows such an area. Due to the wet soil and the steep terrain, such plateaus can occur in erosive processes. In these cases the stream is distributed over the plateau and flows in small channels and accumulates again at some point to continue flowing downwards as one stream. These small channels cannot be detected by the LiDAR system and thus the modeling of flow is very uncertain in such locations. However, the uncertainty analysis shows that in this case the stream is modeled correctly with a probability of over 80% after the plateau. Without an uncertainty analysis, the possibility that a stream is mapped correctly in such areas and beyond may be very low.

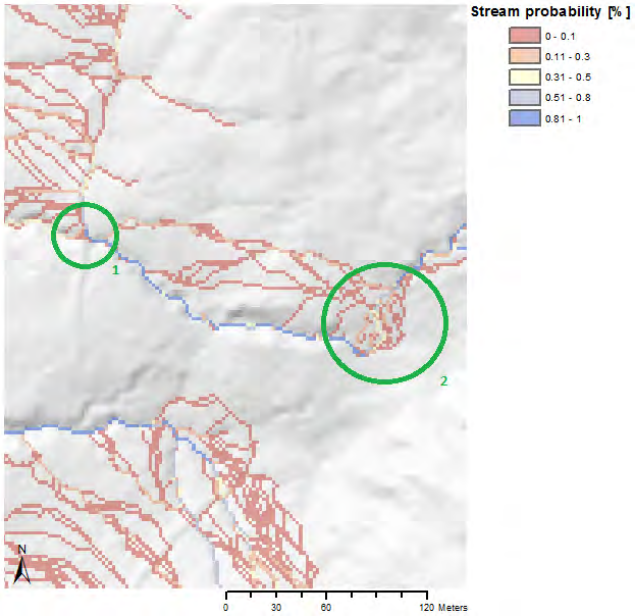


Figure 6.1: Map showing the stream network probability. (1) The location where the stream crosses the road. (2) A very flat plateau introducing high uncertainty in the stream network extraction process (see Figure 6.2).

Figure 6.2: A photo of a location which leads to high uncertainty in all evaluated hydrological algorithms. The location is marked in Figure 6.1 at position 2. This area is almost perfectly flat and the water flows in many small channels across it.

Working hypothesis 3: *Multiple flow direction algorithms will be less sensitive than the single flow direction algorithm and the hydrological algorithms will show the highest errors in flat terrain.*

It could be verified that the multiple flow direction algorithms are more robust than the single flow direction algorithms which was implied by Seibert and McGlynn (2007). Almost a twice as high mean error (std) can be observed in the hydrological models when using a single flow direction algorithm. Also, the multiple flow direction algorithms are very robust in different topographic conditions as long as these conditions are not too extreme. A very good example is the roughness; whereas errors are almost zero in less rough terrain, the error extremely increases in highly rough terrain when using multiple flow direction algorithms. The single flow algorithm still shows a similar trend, but it already has a considerable error in less rough terrain and does increase at a lower rate in rougher terrain. The second part of this hypothesis can also be verified. The results presented in Section 5.2 show that the uncertainty assessments of all hydrological algorithms do highly depend on the steepness of the terrain with highest errors occurring in the flattest areas.

Research question 4: *What are the impacts of the input parameters of the uncertainty model on the uncertainty analysis results?*

Oksanen and Sarjakoski (2005b) already showed that the uncertainty model parameters *spatial autocorrelation range* and *vertical error* have an impact on the results of an uncertainty assessment of constrained derivatives (slope, aspect) using the process convolution approach. They also stated that the appropriate shape of the spatial autocorrelation model, either exponential or Gaussian, was not as important as the other uncertainty model parameters. These conclusions were extended by analyzing additional hydrological models using a high resolution DEM with an uncertainty model exploiting the spatial structure of LiDAR elevation errors.

The spatial structure of error in LiDAR derived DEMs as it was implemented in the proposed approach leads to differences in the derived results (Figures 5.25, 5.26). However, the problem in this study is that control points are not available to validate whether this approach leads to more accurate error surfaces or not. Therefore, further studies are needed to validate this approach. The second parameter analyzed was the spatial autocorrelation range (Figures 5.27, 5.28). The results are in line with the study from Oksanen and Sarjakoski (2005b). A range of 0 m dramatically increases the uncertainty in all hydrological algorithms, whereas a higher range smooths the error surface and thus reduces the uncertainty of the hydrological algorithms. The spatial autocorrelation range also has a direct influence on the convergence test. The higher the range the less simulation runs are needed to reach stable results. A too low spatial autocorrelation range could not only lead to overestimated errors, it also needs more simulation runs if stable results should be achieved. On the other hand, too high spatial autocorrelation ranges can underestimate the uncertainty. An interesting observation is that the single flow direction algorithm D8 with a spatial autocorrelation range of 0 m starts to vary so considerably that it approximates a multiple flow direction algorithm when the mean of the TWI realizations is computed. As illustrated in Figure 5.30(b), the TWI shows a very smooth surface which is literally uncharacteristic of the D8 flow routing algorithm. This might be an alternative method to compute the TWI.

Finally, it can be said that the election of the uncertainty model parameters is crucial. It can lead to over- or underestimated uncertainty assessments and influence the results of convergence tests. The main issue of the process convolution based uncertainty analysis remains the determination of

these parameters. Only studies can deliver rough guidelines for non- expert users allowing them to execute appropriate and valid uncertainty analysis.

Working hypothesis 4: *Input parameters changing the error surface in the expected sensitive flat areas will have an influence on the derived uncertainty assessments.*

This hypothesis could be verified too. Especially the spatial autocorrelation range has a considerable impact on the uncertainty analysis results. As already shown by Oksanen and Sarjakoski (2005b), the spatial autocorrelation range is the dominating parameter of a process convolution based uncertainty model. Input parameters regarding the spatial structure of LiDAR DEM error play a minor but not necessarily an insignificant role.

Research question 5: *Do coarser spatial resolutions of DEMs change the findings of the above asked research questions?*

Using coarse resolution DEMs has some benefits. First, the computation time of the algorithms is reduced considerably compared to high resolution DEMs. Resampling the 2 m DEM to 25 m increases the performance in this study approximately by a factor of 40 in the D8 flow routing algorithm based TWI calculation. Second, coarse resolution DEMs often have a significantly lower cost compared to high resolution LiDAR data and are more available. The third advantage found in this study is that resampling a high resolution DEM to a coarser spatial resolution provides the possibility to estimate an error surface. The statistical uncertainty estimator RMSE can be derived from this map and can be used to calibrate the uncertainty model appropriately. However, for the proposed method, the spatial autocorrelation range still has to be estimated. Using this methodology to calibrate the uncertainty model, slightly higher uncertainties are observed with a spatial autocorrelation range of 125 m in the hydrological models compared to the analysis using the 2 m DEM. In contrast, when using a spatial autocorrelation range of 500 m, the errors observed are, at least for the TWI calculation, considerably lower. The results of the convergence tests also highly depended on the spatial autocorrelation range. For the extraction of watersheds and stream networks similar results can be observed as for the 2 m DEM, except that the error of mapping streams is this time larger in convex instead of concave terrain. In this case, the results are in line with the result obtained by Hengl et al. (2010) who also stated that mapping streams in slightly convex areas is slightly more uncertain using a 30 m DEM. Finally, the conclusion is that the uncertainty in hydrological models highly depends on the uncertainty model, its parameters, and the quality of the DEM. The spatial resolution of the DEM plays a minor role in an uncertainty assessment. But at least some spatial structures of error appear to change at different scales.

Working hypothesis 5: *Major differences will not be expected.*

The differences observed vary. The mean error observed in the uncertainty assessment appears to be independent from the spatial resolution of the DEM. However, some spatial structures of uncertainty are different. For example, observed errors in mapping streams are larger using a coarse resolution DEM (see also Hengl et al., 2010), whereas the errors are smaller using a high resolution DEM in convex reliefs. As of now, this hypotheses can be neither verified nor falsified. More research is needed to obtain enough ground data to make a final decision.

7 Conclusion

High resolution DEMs, such as LiDAR derived DEMs, are also not free of errors. As of today, this fact is commonly ignored in geomorphological studies because users often assume the DEM as the true surface rather than a model as its name implies and because too few methods exist to address the impact of the DEM error. In this case, the errors contain a significant stochastic component allowing to use the Monte Carlo simulation approach to investigate the implications of uncertainty in elevation on the derivation of hydrological models. The process convolution approach allows to model error surfaces in a very efficient way without requiring control points. This approach allows error propagation analysis when only little information such as the RMSE indicated by the DEM provider regarding the elevation error is available. The downside is that it uses an uncalibrated correlation function to produce the error surfaces.

Previous studies rarely used high resolution DEMs derived from LiDAR data. This study provides an uncertainty assessment of various hydrological models derived from a high resolution LiDAR DEM using an uncertainty model which was optimized according to the spatial structure of error in LiDAR based DEMs. The methodology and its application presented in this thesis to assess DEM errors demonstrated that it is possible and also worth investigating in error propagation when working with high resolution DEMs.

Numerous hydrological applications utilize DEMs and DEM derived hydrological parameters as input data. The purpose of the proposed uncertainty analysis approach is to provide hydrological users with a methodology of how to evaluate the impact of DEM uncertainty on the derived hydrological models and improve the quality of more advanced hydrological models.

7.1 Achievements

The following points outline what has been achieved with this study:

- An uncertainty model was implemented with the following characteristics:
  - The uncertainty model is based on the Monte Carlo simulation approach and uses the process convolution approach for the spatial autocorrelation of the produced error surfaces. Due to performance reasons, a Gaussian convolution kernel was implemented.
  - The uncertainty model only needs two input parameters: the RMSE and the spatial autocorrelation range describing the error of the DEM.
  - Specific information about the spatial structure of error of LiDAR based DEMs can be used to produce more realistic error surfaces. The error can be adjusted according to the local slope. In addition, the uncertainty model can utilize a canopy mask, adjusting the error at locations covered by dense canopy.

- A complete uncertainty analysis approach was proposed and has the following characteristics:
  - Uncertainty assessments are offered for the digital terrain analysis products slope, upslope area, TWI, STI, SPI, stream network and watershed.
  - The upslope area and compound indices can be calculated with different flow routing algorithms. The D8, D∞, MD8 and MD∞ flow routing algorithms were implemented.
  - The uncertainty assessment is possible even for very large datasets such as LiDAR derived DEMs. The implementation uses mostly very efficient algorithms reducing memory usage and computation time considerably.
  - The computation time of uncertainty analysis was reduced dramatically by fully parallelizing all involved algorithms. The proposed uncertainty analysis can utilize as many processes as a system can provide. It can even exploit the power of computer clusters used in HPC environments as a result of the used MPI programming approach.
  - Coordinates can be read from text files allowing to save the intermediate results of the Monte Carlo simulations for specific locations. These results allow additional analysis such as convergence tests or histogram analyses.
  - The uncertainty assessment provides maps describing the uncertainty by different statistical uncertainty estimators (mean, std, rstd, probability and error (H)). In addition, text files are provided giving the users the possibility to evaluate the uncertainty regarding their spatial structure. The spatial structure of error can be analyzed with the terrain characteristics slope, convexity, canopy cover and elevation. The additional terrain characteristics used in this thesis (roughness and the topography classification according to Iwahashi and Pike (2007)) were done externally and are not directly implemented in the proposed uncertainty analysis approach.
- Numerous uncertainty assessments with different hydrological models for the Zwackentobel study site using a high resolution LiDAR DEM were done and analyzed according to the sensitivity of the algorithms and the spatial structure of error. These analyses revealed that the multiple flow direction algorithms are less sensitive than the single flow direction algorithm. The errors were up to three times as high when using the single flow direction algorithm D8. The errors were in general larger in flat and rough terrain. The analysis of watersheds showed that the area can be considerably over- or underestimated even due to small elevation errors. Stream networks showed high errors especially for Strahler order one or two without revealing any patterns in the spatial structure of error.
- Convergence tests were done for the TWI and the watershed extraction, using multiple sampling points respectively the calculated area size. The points were sampled in different terrain classes (Iwahashi and Pike, 2007). The tests demonstrated differences between the sample points, the used flow routing algorithms and the two hydrological products. Hydrological models using multiple flow direction algorithms converge considerably earlier allowing users to perform fewer simulation runs. Also, watershed computations need fewer simulation runs for stable results mainly due to the fact that watersheds in general depend on large scale topographic features. The differences between the sample points could most likely be explained by their local terrain characteristics.

- The analyses show that the uncertainty model parameters canopy, slope and spatial autocorrelation range have an impact on the result of the uncertainty assessment. A high spatial autocorrelation range reduces the observed error considerably and vice versa. The slope adjustment mainly reduces errors in flat areas whereas the canopy mask increases the error in forested areas and reduces the error in forest free locations.
- The proposed uncertainty analysis approach can also be used with coarse resolution DEMs. A LiDAR DEM (2 m) was resampled to a spatial resolution of 25 m using the nearest neighbor method. The results of the uncertainty assessments were not clear. The analyses show that the uncertainty analysis results mainly depend on the selected spatial autocorrelation range instead of the spatial resolution.

7.2 Insights

The parallelization of common digital terrain analysis algorithms has proven to be an efficient programming technique to accelerate uncertainty analysis using the Monte Carlo simulation approach. High resolution DEMs can be processed in an appropriate time frame allowing hydrological users to analyze their computed products regarding their uncertainty in a valid way (using sufficient simulation runs). The study also showed that the algorithms itself must follow an efficient approach, otherwise even the parallelization cannot compensate for the performance loss caused by inefficient algorithms.

The determination of the uncertainty model parameters is not an easy task. As demonstrated in literature, the error occurring in terrain covered by dense canopy when using LiDAR measurement technique is very case-specific. First, the errors depend on the factors of the LiDAR system itself (foot print size, calibration of the LiDAR system components) and second, on the canopy type. Classifying the canopy cover of a terrain in multiple classes is a very time demanding task and thus we can expect that most users will not utilize such a feature. Hence, the implemented approach only utilizes one class. The error adjustment for canopy covered terrain was evaluated by literature without any additional justification. The determination of the spatial autocorrelation range is also not trivial. Without any hard data such as control points, the range must be estimated. The issue here is that the spatial autocorrelation range cannot be verified, which is also seen as one of the biggest drawbacks of the process convolution approach. The range can be determined by expert knowledge, data from literature or by an evaluation of an own exploratory study. This process is rather critical because the results show that different spatial autocorrelation ranges have a huge impact on the obtained uncertainty assessment results. Obviously higher ranges reduce the observed error in the results. If the proposed method and the estimated spatial autocorrelation range produces valid error surfaces could not be carried out. It may be useful to resample the DEM to a slightly coarser resolution and using the high resolution DEM as reference data. Another possibility is taht the point cloud of a LiDAR mission could be used as reference data. Reference data would allow to improve the uncertainty model (e.g. with a SGS approach).

The uncertainty analysis highlighted that even with high resolution DEMs derived from LiDAR data hydrological products can be influenced considerably by elevation uncertainty. The spatial pattern of uncertainty appeared to be very similar to studies using coarse resolution DEMs. However, LiDAR derived DEMs show a noisier uncertainty map when using the D8 flow routing algorithm. The micro-

scale features of high resolution DEMs have a considerable impact on the very sensitive D8 flow routing algorithm. As evaluated in this study, the multiple flow direction algorithms are more robust and are more suitable for calculating hydrological products using flow routing algorithms in high resolution DEMs. However, anthropogenic features detectable in LiDAR datasets such as bridges cause a lot of uncertainty regardless of the flow routing algorithm used. Resampling the DEM to a coarser resolution in which these features disappear would solve this issue. This solution would also support the conclusion to resample the high resolution DEM to a coarser spatial resolution.

The convergence tests revealed that the point at which the result begins to converge varies in number. Hence, setting a global threshold at which the result is seen as stable and thus as a valid uncertainty analysis is not an easy task. The more sample points are used for the convergence test the more reliable is the obtained threshold. However, the determination of the threshold depends on the users view. If values are needed only for single cells, the proposed approach allows a very easy determination of the number of simulation runs needed for an adequate uncertainty assessment.

The spatial pattern of uncertainty is mostly influenced by slope and roughness. The highest errors are observed in steep and rough terrain, whereas the errors decrease approximately exponential when slope and roughness decrease. Notice that flat areas are also rough, whereas steep terrain shows a lower degree of roughness. The convexity did not seem to have that large an impact. However, errors observed were slightly larger in concave terrain when using the high resolution DEM. This was not expected, because Hengl et al. (2010) observed slightly larger errors for mapping streams in convex reliefs. Thus, this spatial pattern appears to be case-specific. A reason could be that a high resolution DEM was used compared to the 30 m DEM used by Hengl et al. (2010). This assumption is supported by the results obtained using the resampled DEM (25 m) which also show larger errors for mapping streams in convex terrains.

The proposed uncertainty analysis approach seems to be suitable to improve topographic correlation analysis in recent hydrological research (see Seibert et al., 2007). By computing the mean value of the DEM derived parameters or by completely removing the uncertain sample sites from the data, the scatter observed in correlation analysis could be reduced. This method may improve the quality of such correlation analysis.

7.3 Future Work

The uncertainty was visualized with statistical estimators found in literature without any real justification. Possibly better methods exist or could be developed to measure and visualize uncertainty in a more appropriate way. Ehlschlaeger et al. (1997), for example, proposed a method to visualize uncertainty using animations. This could lead to new findings or simplify the interpretation of the results by hydrological users. Further, the proposed uncertainty analysis could be extended by additional hydrological algorithms such as the maximum stream length calculation. The uncertainty assessments for these algorithms may differ from the algorithms analyzed in this thesis.

There are other methods to parallelize hydrological algorithms (e.g. Zhan and Qin, 2011). It is of interest to find the most efficient approach to accelerate hydrological algorithms. There is always a need for more performance to either process very large and high resolution DEMs or execute as many Monte Carlo simulation runs as possible to achieve very stable results for uncertainty assessments.

As a next step, the implementation of the SGS approach (e.g. Hengl et al., 2010) would be logical because the process convolution method reached its limits. If control point data is available, the SGS approach could lead to more realistic error surfaces due to the use of calibrated correlation functions. The estimation of the very critical uncertainty model parameters such as the spatial autocorrelation range would not be necessary anymore.

Another open question is how much of the uncertainty analysis should be automated? To achieve the goal of an "uncertainty button" in a GIS, the process must be automated. Many parameters must be defined in the background without the user's awareness, otherwise only experts are able to execute an uncertainty analysis. But the question is if such a black-box approach really leads to reasonable uncertainty assessments or if it leads to more misinterpretations because of unsuitable parameters automatically chosen by the model. This question is very crucial and a big issue of the current state of uncertainty analysis. Studies are needed to investigate this issue and providing software developers with a framework defining and describing the grade of automation.

Bibliography

Abrahamsen, P. (1997). A review of Gaussian random fields and correlation functions. Technical report, Norwegian Computing Center, Oslo.

Achour, H., Rebai, N., Driessche, J. V. D., and Bouaziz, S. (2012). Modelling Uncertainty of Stream Networks Derived from Elevation Data Using Two Free Softwares: R and SAGA. *Journal of Geographic Information System*, 4:153–160.

Adams, J. and Chandler, J. (2002). Evaluation of Lidar and Medium Scale Photogrammetry for Detecting Soft-Cliff Coastal Change. *The Photogrammetric Record*, 17(99):405–418.

Aerts, C. J. H., Heuvelink, G. B. M., and Goodchild, M. F. (2003). Accounting for spatial uncertainty in optimization with spatial decision support systems. *Transactions in GIS*, 7:211–230.

Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference*, AFIPS '67 (Spring), pages 483–485, New York, NY, USA. ACM.

Andersen, H.-E., Reutebuch, S. E., and McGaughey, R. J. (2005). Accuracy of an IFSAR-derived digital terrain model under a conifer forest canopy. *Canadian Journal of Remote Sensing*, 31(4):283–288.

Atkinson, P. M. and Foody, G. M. (2006). *Uncertainty in Remote Sensing and GIS: Fundamentals*, pages 1–18. John Wiley & Sons, Ltd.

Bater, C. W. and Coops, N. C. (2009). Evaluating error associated with lidar-derived DEM interpolation. *Computers & Geosciences*, 35(2):289–300.

Beraldin, J.-A., Balis, F., and Lohr, U. (2010). *Laser Scanning Technology*, pages 1–42. Whittles Publishing.

Beven and Kirkby (1979). A physically based, variable contributing area model of basin hydrology. *Bulletin of the International Association of Scientific Hydrology*, 24(1):43–69.

Brown, J. D. and Heuvelink, G. B. M. (2006). *Assessing Uncertainty Propagation through Physically Based Models of Soil Water Flow and Solute Transport*, pages 1181–1195. John Wiley & Sons, Ltd.

Burch, H. (1994). Ein Rückblick auf die hydrologische Forschung der WSL im Alptal. *Beiträge zur Hydrologie der Schweiz*, 35:18–33.

Burrough, P. (1986). *Principles of Geographical Information Systems for Land Resources Assessment*. Oxford University Press, Oxford.

Burrough, P. and McDonnell, R. (1998). *Principles of Geographical Information Systems*. Spatial Information Systems and Geostatistics. Oxford University Press, USA.

Callow, J. N., Niel, K. P. V., and Boggs, G. S. (2007). How does modifying a DEM to reflect known hydrology affect subsequent terrain analysis? *Journal of Hydrology*, 332(1):30–39.

Creed, I. F. and Sass, G. Z. (2011). Digital Terrain Analysis Approaches for Tracking Hydrological and Biogeochemical Pathways and Processes in Forested Landscapes. In Levia, D. F., Carlyle-Moses, D., and Tanaka, T., editors, *Forest Hydrology and Biogeochemistry*, volume 216 of *Ecological Studies*, pages 69–100. Springer Netherlands.

Cressie, N. and Pavlicová, M. (2002). Calibrated spatial moving average simulations. *Statistical Modelling*, 2(4):267–279.

DeBarry, P. (2004). *Watersheds: processes, assessment, and management*. Wiley.

Ehlschlaeger, C. R., Shortridge, A. M., and Goodchild, M. F. (1997). Visualizing spatial data uncertainty using animation. *Computers & Geosciences*, 23(4):387–395.

Evans, J. S. and Hudak, A. T. (2007). A multiscale curvature filter for identifying ground returns from discrete return lidar in forested environments. *IEEE Transactions on Geoscience and Remote Sensing*, 45(4):1029–1038.

Florinsky, I. V. (1998a). Accuracy of local topographic variables derived from digital elevation models. *International Journal of Geographical Information Science*, 12(1):47–62.

Florinsky, I. V. (1998b). Combined analysis of digital terrain models and remotely sensed data in landscape investigations. *Progress in Physical Geography*, 22(1):33–60.

Fowler, R. J. (2001). *Topographic Lidar*, pages 207–236. American Society for Photogrammetry and Remote Sensing, Maryland.

Freeman, T. (1991). Calculating catchment area with divergent flow based on a regular grid. *Computers & Geosciences*, 17(3):413–422.

Goodchild, M., Steyaert, L., Parks, B., Johnston, C., Maidment, D., Crane, M., and Glendinning, S. (1996). *GIS and Environmental Modeling: Progress and Research Issues*. John Wiley & Sons.

Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation*. Applied Geostatistics Series. Oxford University Press, USA.

Grimaldi, S., Nardi, F., Benedetto, F. D., Istanbulluoglu, E., and Bras, R. L. (2007). A physically-based method for removing pits in digital elevation models. *Advances in Water Resources*, 30(10):2151–2158.

Gruber, S. and Peckham, S. (2009). Land-Surface Parameters and Objects in Hydrology. In Hengl, T. and Reuter, H. I., editors, *Geomorphometry Concepts, Software, Applications*, volume 33 of *Developments in Soil Science*, chapter 7, pages 171–194. Elsevier.

Hebeler, F. and Purves, R. (2008). Modelling DEM data uncertainties for Monte Carlo simulations of ice sheet models. In Stein, A., Shi, J., and Bijker, W., editors, *Quality Aspects in Spatial Data Mining*, pages 175–196. CRC Press.

Hebeler, F. and Purves, R. (2009). The influence of elevation uncertainty on derivation of topographic indices. *Geomorphology*, 111(1):4–16.

Hegg, C., McArdell, B. W., and Badoux, A. (2006). One hundred years of mountain hydrology in Switzerland by the WSL. *Hydrological Processes*, 20(2):371–376.

Hengl, T. and Evans, I. (2009). Mathematical and Digital Models of the Land Surface. In Hengl, T. and Reuter, H. I., editors, *Geomorphometry Concepts, Software, Applications*, volume 33 of *Developments in Soil Science*, chapter 2, pages 31–63. Elsevier.

Hengl, T., Heuvelink, G. B. M., and van Loon, E. E. (2010). On the uncertainty of stream networks derived from elevation data: the error propagation approach. *Hydrology and Earth System Sciences*, 14(7):1153–1165.

Hengl, T. and MacMillan, R. (2009). Geomorphometry - A Key to Landscape Mapping and Modelling. In Hengl, T. and Reuter, H. I., editors, *Geomorphometry Concepts, Software, Applications*, volume 33 of *Developments in Soil Science*, chapter 19, pages 433–460. Elsevier.

Heuvelink, G. B. (1998). *Error Propagation in Environmental Modelling with GIS*. CRC Press.

Heuvelink, G. B. M. (2006). *Analysing Uncertainty Propagation in GIS: Why is it not that Simple?*, pages 155–165. John Wiley & Sons, Ltd.

Heuvelink, G., G. B. M., Burrough, P., P. A., and Stein, A. (1989). Propagation of errors in spatial modelling with GIS. *International journal of geographical information systems*, 3(4):303–322.

Higdon, D. (2002). *Space and space-time modelling using process convolutions*, pages 37–56. Wiley.

Hodgson, M. E. and Bresnahan, P. (2004). Accuracy of Airborne Lidar-Derived Elevation: Empirical Assessment and Error Budget. Photogrammetric Engineering and Remote. *Photogrammetric Engineering & Remote Sensing*, 70(3):331–339.

Hodgson, M. E., Jensen, J. R., Raber, G. T., Tullis, J. A., Davis, B. A., Thompson, G., and Schuckman, K. (2005). An Evaluation of LIDAR-Derived Elevation and Terrain Slope in Leaf-off Conditions. *Photogrammetric Engineering & Remote Sensing*, 71:817–823.

Holmes, K., Chadwick, O., and Kyriakidis, P. (2000). Error in a USGS 30-meter digital elevation model and its impact on terrain modeling. *Journal of Hydrology*, 233:154–173.

Hwu, W. (2011). *GPU Computing Gems Emerald Edition*. Applications of GPU Computing Series. Elsevier Science.

Irfan, A. M., Zhao, Z., Bourque, C. P.-A., and Meng, F.-R. (2012). GIS-evaluation of two slope-calculation methods regarding their suitability in slope analysis using high-precision LiDAR digital elevation models. *Hydrological Processes*, 26(8):1119–1133.

Iwahashi, J. and Pike, R. J. (2007). Automated classifications of topography from DEMs by an unsupervised nested-means algorithm and a three-part geometric signature. *Geomorphology*, 86(3):409–440.

Joerg, P. C., Morsdorf, F., and Zemp, M. (2012). Uncertainty assessment of multi-temporal airborne laser scanning data: A case study on an Alpine glacier. *Remote Sensing of Environment*, 127(0):118–129.

Kenward, T., Lettenmaier, D. P., Wood, E. F., and Fielding, E. (2000). Effects of Digital Elevation Model Accuracy on Hydrologic Predictions. *Remote Sensing of Environment*, 74(3):432–444.

Kern, J. (2000). *Bayesian Process-convolution Approaches to Specifying Spatial Dependence Structure*. Duke University.

Kraus, K. and Pfeifer, N. (1998). Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(4):193–203.

Kumar, V. (2001). *Introduction to parallel computing*. Harlow : Prentice Hall.

Kyriakidis, P. C., Shortridge, A. M., and Goodchild, M. F. (1999). Geostatistics for conflation and accuracy assessment of digital elevation models. *International Journal of Geographical Information Science*, 13:677–707.

Li, Z., Zhu, Q., and Gold, C. (2005). *Digital Terrain Modeling: Principles and Methodology*. CRC Press, Boca Raton, Florida.

Lichti, D. and Skaloud, J. (2010). *Registration and Calibration*, pages 83–129. Whittles Publishing.

Magalhães, S. V. G., Andrade, M. V. A., Randolph Franklin, W., and Pena, G. C. (2012). A New Method for Computing the Drainage Network Based on Raising the Level of an Ocean Surrounding the Terrain. In Gensel, J., Josselin, D., and Vandenbroucke, D., editors, *Bridging the Geographic Information Sciences*, Lecture Notes in Geoinformation and Cartography, pages 391–407. Springer Berlin Heidelberg.

Marsaglia, G. and Bray, T. A. (1964). A convenient method for generating normal variables. *SIAM Review*, 6(3):260–264.

Montgomery, D. and W.E., D. (1994). A physically based model for the topographic control on shallow landsliding. *Water Resour. Res.*, 30(4):1153–1171.

Montgomery, D. R. (2001). Slope Distributions, Threshold Hillslopes, and Steady-state Topography. *American Journal of Science*, 301(4-5):432–454.

Moore, I., Gessler, P., and Nielsen, G.A.and Petersen, G. (1993). *Terrain attributes: estimation methods and scale effects*, pages 189–214. London: Wiley.

Moore, I. D. and Burch, G. J. (1986). Physical Basis of the Length-slope Factor in the Universal Soil Loss Equation. *Soil Science Society of America Journal*, 50(5):1294–1298.

Moore, I. D., Grayson, R. B., and Ladson, A. R. (1991). Digital terrain modelling: A review of hydrological, geomorphological, and biological applications. *Hydrological Processes*, 5(1):3–30.

Nackaerts, K., Govers, G., and Orshoven, J. V. (1999). Accuracy assessment of probabilistic visibilities. *International Journal of Geographical Information Science*, 13(7):709–721.

Nelson, A., Reuter, H., and Gessler, P. (2009). DEM Production Methods and Sources. In Hengl, T. and Reuter, H. I., editors, *Geomorphometry Concepts, Software, Applications*, volume 33 of *Developments in Soil Science*, chapter 3, pages 65–85. Elsevier.

O'Callaghan, J. F. and Mark, D. M. (1984). The extraction of drainage networks from digital elevation data. *Computer Vision, Graphics, and Image Processing*, 28:323–344.

Oksanen, J. (2006). *Digital elevation model error in terrain analysis*. PhD thesis, Faculty of Science, University of Helsinki.

Oksanen, J. and Sarjakoski, T. (2005a). Error propagation analysis of DEM-based drainage basin delineation. *International Journal of Remote Sensing*, 26(14):3085–3102.

Oksanen, J. and Sarjakoski, T. (2005b). Error propagation of DEM-based surface derivatives. *Computers & Geosciences*, 31(8):1015–1027.

Olaya, V. (2009). Basic Land-Surface Parameters. In Hengl, T. and Reuter, H. I., editors, *Geomorphometry: Concepts, Software, Applications*, volume 33 of *Developments in Soil Science*, chapter 6, pages 141–169. Elsevier, Amsterdam.

Openshaw, S., Charlton, M., and Carver, S. (1991). *Error propagation: a Monte Carlo simulation*, pages 78–101. Longman Scientific and Technical: London, UK.

Ortega, L. and Rueda, A. (2010). Parallel drainage network computation on CUDA. *Computers & Geosciences*, 36(2):171–178.

Planchon, O. and Darboux, F. (2002). A fast, simple and versatile algorithm to fill the depressions of digital elevation models. *CATENA*, 46(2):159–176.

Qin, C., Zhu, A.-X., Pei, T., Li, B., Zhou, C., and Yang, L. (2007). An adaptive approach to selecting a flow-partition exponent for a multiple-flow-direction algorithm. *International Journal of Geographical Information Science*, 21(4):443–458.

Qin, C.-Z., Zhu, A.-X., Pei, T., Li, B.-L., Scholten, T., Behrens, T., and Zhou, C.-H. (2011). An approach to computing topographic wetness index based on maximum downslope gradient. *Precision Agriculture*, 12:32–43.

Quinn, P., Beven, K., Chevallier, P., and Planchon, O. (1991). The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models. *Hydrological Processes*, 5(1):59–79.

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

Raaflaub, L. D. and Collins, M. J. (2006). The effect of error in gridded digital elevation models on the estimation of topographic parameters. *Environ. Model. Softw.*, 21(5):710–732.

Ravalec, M., Noetinger, B., and Hu, L. (2000). The FFT Moving Average (FFT-MA) Generator: An Efficient Numerical Method for Generating and Conditioning Gaussian Simulations. *Mathematical Geology*, 32:701–723.

Reuter, H., Hengl, T., Gessler, P., and Soille, P. (2009). Preparation of DEMs for Geomorphometric Analysis. In Hengl, T. and Reuter, H. I., editors, *Geomorphometry Concepts, Software, Applications*, volume 33 of *Developments in Soil Science*, chapter 4, pages 87–120. Elsevier.

Santini, M., Grimaldi, S., Nardi, F., Petroselli, A., and Rulli, M. C. (2009). Pre-processing algorithms and landslide modelling on remotely sensed DEMs. *Geomorphology*, 113(1):110–125.

Schaer, P., Skaloud, J., Landtwing, S., and Legat, K. (2007). Accuracy Estimation for Laser Point Cloud Including Scanning Geometry. In *Mobile Mapping Symposium 2007, Padova*.

Schneider, B. (2000). Uncertainty Propagation in Digital Terrain Modelling with Taylor Methods and Interval Arithmetic. In *Proceedings of the Accuracy 2000*, pages 561–568.

Seibert, J. and McGlynn, B. L. (2007). A new triangular multiple flow direction algorithm for computing upslope areas from gridded digital elevation models. *Water Resources Research*, 43:W04501.

Seibert, J., Stendahl, J., and Sørensen, R. (2007). Topographical influences on soil properties in boreal forests. *Geoderma*, 141(1-2):139–148.

Strahler, A. N. (1952). Hypsometric (area-altitude) analysis of erosional topology. *Geological Society of America Bulletin*, 63(11):1117–1152.

Su, J. and Bork, E. (2006). Influence of vegetation, slope, and lidar sampling angle on DEM accuracy. *Photogrammetric Engineering and Remote Sensing*, 72:1265–1274.

Swisstopo (2012). swissALTI3D Ausgabebericht 2012. URL: http://alturl.com/kcs5a (visited on 10-11-2012).

Tarboton, D. G. (1997). A New Method For The Determination Of Flow Directions And Upslope Areas In Grid Digital Elevation Models. *Water Resources Research*, 33:309–319.

Team, R. D. C. (2009). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Temme, A., Heuvelink, G., Schoorl, J., and Claessens, L. (2009). Geostatistical Simulation and Error Propagation in Geomorphometry. In Hengl, T. and Reuter, H. I., editors, *Geomorphometry Concepts, Software, Applications*, volume 33 of *Developments in Soil Science*, chapter 5, pages 121–140. Elsevier.

Tesfa, T. K., Tarboton, D. G., Watson, D. W., Schreuders, K. A., Baker, M. E., and Wallace, R. M. (2011). Extraction of hydrological proximity measures from DEMs using parallel processing. *Environmental Modelling & Software*, 26(12):1696–1709.

Tinkham, W. T., Smith, A. M., Hoffman, C., Hudak, A. T., Falkowski, M. J., Swanson, M. E., and Gessler, P. E. (2012). Investigating the influence of LiDAR ground surface errors on the utility of derived forest inventories. *Canadian Journal of Forest Research*, 42(3):413–422.

Ulmer, M. (2007). Unsicherheitsanalyse und Validierung eines Modells der Hangstabilität. Master's thesis, Universität Zürich.

Van Niel, K. and Laffan, S. W. (2003). Gambling with randomness: the use of pseudo-random number generators in GIS. *International Journal of Geographical Information Science*, 17(1):49–68.

Vaze, J., Teng, J., and Spencer, G. (2010). Impact of DEM accuracy and resolution on topographic indices. *Environmental Modelling & Software*, 25(10):1086–1098.

Veregin, H. (1997). The Effects of Vertical Error in Digital Elevation Models on the Determination of Flow-path Direction. *Cartography and Geographic Information Science*, 24(2):67–79.

Wang, L. and Liu, H. (2006). An efficient method for identifying and filling surface depressions in digital elevation models for hydrologic analysis and modelling. *International Journal of Geographical Information Science*, 20(2):193–213.

Wechsler, S. P. (2007). Uncertainties associated with digital elevation models for hydrologic applications: a review. *Hydrology and Earth System Sciences*, 11(4):1481–1500.

Weibel, R. and Heller, M. (1991). *Digital terrain modeling*, pages 269–297. London: Longman.

Wilson, J. and Gallant, J. (2000). *Terrain Analysis: Principles and Applications*. Wiley, New York.

Wise, S. (2000). Assessing the quality for hydrological applications of digital elevation models derived from contours. *Hydrological Processes*, 14(11-12):1909–1929.

Zhan, L. and Qin, C. (2011). A graph-theory-based method for parallelizing the multiple-flow-direction algorithm on CUDA compatible graphics processing units. In *Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2011 IEEE International Conference on*, pages 137–141.

Zhang, J. and Goodchild, M. (2002). *Uncertainty in Geographical Information*. Research Monographs in Geographic Information Systems. Taylor & Francis.

Zhang, X., Drake, N. A., Wainwright, J., and Mulligan, M. (1999). Comparison of slope estimates from low resolution DEMs: scaling issues and a fractal method for their solution. *Earth Surface Processes and Landforms*, 24(9):763–779.

Zhou, Q. and Liu, X. (2002). Error assessment of grid-based flow routing algorithms used in hydrological models. *International Journal of Geographical Information Science*, 16(8):819–842.

Zhou, Q. and Liu, X. (2004). Error analysis in grid-based slope and aspect algorithms. *Photogrammetric Engineering and Remote Sensing*, 70:957–962.

A Code

Code A.1: Flow routing algorithm: Determining the number of draining neighbors.

```
for (long j = 0; j < ny; j++) {
  for (long i = 0; i < nx; i++) {
    if(dem->isNodata(i,j)){
      output->setToNodata(i,j);
    } else {
      neighbors=0;
      dem->getData(i,j,height);
      for(int k=0;k<8;k++){
        X=i+DX[k];
        Y=j+DY[k];
        dem->getData(X,Y,height2);
        if(!dem->isNodata(X,Y) && height<height2){
          neighbors++;
        }
      }
      output->setData(i,j,1.f);
      neighbor->setData(i,j,neighbors);
    }
    change2->setData(i,j,short(0));
    change->setData(i,j,0.f);
  }
}

//fill que with cells having 0 neighbors
for (long j = 0; j < ny; j++) {
  for (long i = 0; i < nx; i++) {
    if(!dem->isNodata(i,j)){
      if(neighbor->getData(i,j,tempShort)==0){
        temp.x=i;
        temp.y=j;
        que.push(temp);
      }
    }
  }
}
```

Code A.2: MD8 flow routing algorithm.

```
while(!finished) {
    while(!que.empty())
    {
      //Takes next node with no contributing neighbors
      temp = que.front();
      que.pop();
      col = temp.x;
```

```
8       row = temp.y;
9       sum=0;
10      maxSlope = 0;
11
12      //initialize vars
13      dem->getData(col,row,p);
14      output->getData(col,row,FlowAccumVal);
15      neighbor->setData(col,row,short(-1));
16      for(int i=0;i<8;i++){
17        s[i]=na;
18        d[i]=na;
19      }
20      //  FLOW ALGEBRA EXPRESSION EVALUATION
21      if(dem->isInPartition(col,row)){
22
23        //slope
24        for(int i=0;i<8;i++){
25          if(!dem->isNodata(col+xd[i],row+yd[i])){
26            dem->getData(col+xd[i],row+yd[i],p1);
27            s[i]=(p-p1)/dd[i];
28            if(s[i]>maxSlope){
29              maxSlope=s[i];
30            }
31          }
32        }
33        //exponent
34        exp=8.9*min(maxSlope,1)+1.1;
35        //dispersion
36        for(int i=0;i<8;i++){
37          if(!dem->isNodata(col+xd[i],row+yd[i])){
38            if(s[i]>0){
39              //formula
40              d[i]=pow(s[i],exp)*L[i];
41            }
42          }
43        }
44        for(int i=0;i<8;i++){
45          if(!dem->isNodata(col+xd[i],row+yd[i])){
46            if(s[i]>0){
47              sum+=d[i];
48            }
49          }
50        }
51        for(int i=0;i<8;i++){
52          if(!dem->isNodata(col+xd[i],row+yd[i])){
53            if(s[i]>0){
54              d[i]/=sum;
55            }
56          }
57        }
58        //apply flowaccum to neighbors
59        for(int i=0;i<8;i++){
60          if (d[i] > 0 ){
61            a = col + xd[i];
62            b = row + yd[i];
63            if (!output->isNodata(a,b) ){
64              output->addToData(a,b,(FlowAccumVal * d[i]));
```

```
65                neighbor->addToData(a,b,(short)-1);
66                if(b==-1 || b==ny){
67                  change->addToData(a,b,(FlowAccumVal * d[i]));
68                  change2->addToData(a,b,(short)-1);
69                }
70                if(neighbor->getData(a, b, tempShort) == 0){
71                  temp.x=a;
72                  temp.y=b;
73                  que.push(temp);
74                }
75              }
76            }
77          }
78        }
79      }
80      if(size>1){
81        //Pass information output
82        change->addBorders();
83
84        for(int x=0;x<nx;x++){
85          if(rank==0){
86            output->addToData(x,ny-1,change->getData(x,ny-1,temp_));
87            change->setData(x,ny-1,0.f);
88            change->setData(x,ny,0.f);
89          } else if (rank==size-1){
90            output->addToData(x,0,change->getData(x,0,temp_));
91            change->setData(x,0,0.f);
92            change->setData(x,-1,0.f);
93          } else {
94            output->addToData(x,0,change->getData(x,0,temp_));
95            change->setData(x,0,0.f);
96            change->setData(x,-1,0.f);
97            output->addToData(x,ny-1,change->getData(x,ny-1,temp_));
98            change->setData(x,ny-1,0.f);
99            change->setData(x,ny,0.f);
100         }
101       }
102       output->share();
103
104       //Pass information neighbour
105       change2->addBorders();
106       for(int x=0;x<nx;x++){
107         if(rank==0){
108           neighbor->addToData(x,ny-1,change2->getData(x,ny-1,stemp_));
109           change2->setData(x,ny-1,short(0));
110           change2->setData(x,ny,short(0));
111         } else if(rank==size-1){
112           neighbor->addToData(x,0,change2->getData(x,0,stemp_));
113           change2->setData(x,0,short(0));
114           change2->setData(x,-1,short(0));
115         }else{
116           neighbor->addToData(x,ny-1,change2->getData(x,ny-1,stemp_));
117           change2->setData(x,ny-1,short(0));
118           change2->setData(x,ny,short(0));
119           neighbor->addToData(x,0,change2->getData(x,0,stemp_));
120           change2->setData(x,0,short(0));
121           change2->setData(x,-1,short(0));
```

```
122         }
123       }
124       neighbor->share();
125
126       //If this created a cell with no contributing neighbors, put it on the
              queue
127       for(int i=0; i<nx; i++){
128         if(neighbor->getData(i, 0, tempShort)==0){
129           temp.x = i;
130           temp.y = 0;
131           que.push(temp);
132         }
133         if(neighbor->getData(i, ny-1, tempShort)==0){
134           temp.x = i;
135           temp.y = ny-1;
136           que.push(temp);
137         }
138       }
139     }
140     //Check if done
141     finished = que.empty();
142     finished = output->ringTerm(finished);
143   }
```

Code A.3: MD∞ flow routing algorithm.

```
1   while(!finished) {
2     while(!que.empty())
3     {
4       //Takes next node with no contributing neighbors
5       temp = que.front();
6       que.pop();
7       col = temp.x;
8       row = temp.y;
9
10      //initialize vars
11      valleySum=0;
12      valleyMax=0;
13      iMax=0;
14      hr=0;
15      hs=0;
16      for(int x = 0; x < 8; x++){
17        r_facet[x]=noData;
18        s_facet[x]=noData;
19        valley[x]=0;
20        portion[x]=0;
21      }
22      dem->getData(col,row,p);
23      output->getData(col,row,FlowAccumVal);
24      neighbor->setData(col,row,short(-1));
25
26      //  FLOW ALGEBRA EXPRESSION EVALUATION
27      if(dem->isInPartition(col,row)){
28        for(int c=0;c<8;c++){
29          i=c;
30          ii=(i+1)%8;
31          dem->getData(col + xd[i], row + yd[i],p1);
```

```
32            dem->getData(col + xd[ii], row + yd[ii],p2);
33            if(!dem->isNodata(col + xd[i], row + yd[i]) && !dem->isNodata(col +
                  xd[ii], row + yd[ii])){
34
35              //Calculate the elevation difference between the centerpoint and
                    the points p1 and p2
36              z1 = p1 - p;
37              z2 = p2 - p;
38
39              //Calculate the coordinates of the normal to the triangular facet
40              nx_ = (yd[i] * z2 - yd[ii] * z1) * gridsize;
41              ny_ = (xd[ii] * z1 - xd[i] * z2) * gridsize;
42              nz_ = (xd[i] * yd[ii] - xd[ii] * yd[i]) * gridsize * gridsize;
43
44              //Calculate the downslope direction of the triangular facet
45              if (nx_ == 0){
46                if (ny_ >= 0){
47                  hr = 0;
48                } else {
49                  hr = M_PI;
50                }
51              } else {
52                if (nx_ >= 0) {
53                  hr = M_PI / 2 - atanf(ny_ / nx_);
54                } else {
55                  hr = 3 * M_PI / 2 - atanf(ny_ / nx_);
56                }
57              }
58
59              //Calculate the slope of the triangular facet
60              hs = -tanf(acosf(nz_ / (sqrt(nx_ * nx_ + ny_ * ny_ + nz_ * nz_))));
61
62              //If the downslope direction is outside the triangular facet, then
                    use the direction of p1 or p2
63              if( hr < i * M_PI / 4 || hr > (i + 1) * M_PI / 4 ){
64                if (p1 < p2) {
65                  hr = i * M_PI / 4;
66                  hs = (p - p1) / (dd[i] * gridsize);
67                } else {
68                  hr = ii * M_PI / 4;
69                  hs = (p - p2) / (dd[ii] * gridsize);
70                }
71              }
72
73              r_facet[c] = hr;
74              s_facet[c] = hs;
75
76            } else {
77              if (!dem->isNodata(col + xd[i], row + yd[i]) && p1 < p){
78                hr = i / 4 * M_PI;
79                hs = (p - p1) / (dd[ii] * gridsize);
80              }
81            }
82          }
83          //Compute the total area of the triangular facets where water is
                flowing to
84          for (int c = 0; c<8;c++){
```

```
85            i = c;
86            ii = (i + 1) % 8;
87
88            if(s_facet[i] > 0 ){                                //If the slope is
                  downhill
89              if (r_facet[i] > (i * M_PI / 4) && r_facet[i] < ((i + 1) * M_PI /
                    4)){    //If the downslope direction is inside the 45 degrees
                    of the triangular facet
90                valley[i] = s_facet[i];
91              } else if ( r_facet[i] == r_facet[ii]){                   //If two
                    adjacent triangular facets have the same downslope direction
92                valley[i] = s_facet[i];
93              } else if (s_facet[ii] == noData && r_facet[i] == ((i + 1) * M_PI /
                     4)){    //If the downslope direction is on the border of the
                    current triangular facet, and the corresponding neigbour's
                    downslope is NoData
94                valley[i] = s_facet[i];
95              } else {
96                ii = (i + 7) % 8;
97                if (s_facet[ii] == noData && r_facet[i] == (i * M_PI / 4) ){
                        //If the downslope direction is on the other border of the
                      current triangular facet, and the corresponding neigbour's
                      downslope is NoData
98                  valley[i] = s_facet[i];
99                }
100             }
101           }
102
103           valleySum += pow(valley[i],MDInfPower);
104           if( valleyMax < valley[i]){
105             iMax = i;
106             valleyMax = valley[i];
107           }
108         }
109
110         //Compute the proportional contribution for each of the triangular
              facets
111         if( valleySum > 0 ){
112           if (MDInfPower < 10 ){
113             for(i=0;i<8;i++){
114               valley[i] = pow(valley[i],MDInfPower) / valleySum;
115               portion[i] = 0;
116             }
117           } else {
118             for(i=0;i<8;i++){
119               if ( i != iMax){
120                 valley[i] = 0;
121               } else {
122                 valley[i] = 1;
123               }
124               portion[i] = 0;
125             }
126           }
127
128           if( r_facet[7] == 0 ){
129             r_facet[7] = 2 * M_PI;
130           }
```

```
131           //Compute the contribution to each of the neighbouring gridcells
132           for(int c=0;c<8;c++){
133             i = c;
134             ii = (i + 1) % 8;
135             if (valley[i] > 0){
136               portion[i] += valley[i] * ((i + 1) * M_PI / 4 - r_facet[i]) / (
                      M_PI / 4);
137               portion[ii] += valley[i] * (r_facet[i] - (i) * M_PI / 4) / (M_PI
                      / 4);
138             }
139           }
140
141           //Apply the flow accumulation to each of the neighbouring gridcells
142           for(int c=0;c<8;c++){
143             if (portion[c] > 0 ){
144               a = col + xd[c];
145               b = row + yd[c];
146               if (!output->isNodata(a,b) ){
147                 output->addToData(a,b,(FlowAccumVal * portion[c]));
148                 if(b==-1 || b==ny){
149                   change->addToData(a,b,(FlowAccumVal * portion[c]));
150                 }
151               }
152             }
153           }
154         }
155         for(int c=0;c<8;c++){
156           a = col + xd[c];
157           b = row + yd[c];
158           dem->getData(a,b,z1);
159           if (p > z1 && !dem->isNodata(a,b)){
160             neighbor->addToData(a,b,(short)-1);
161             if(b==-1 || b==ny){
162               change2->addToData(a,b,(short)-1);
163             }
164             if(neighbor->getData(a, b, tempShort) == 0){
165               temp.x=a;
166               temp.y=b;
167               que.push(temp);
168             }
169           }
170         }
171       }
172     }
173
174     if(size>1){
175       //Pass information output
176       change->addBorders();
177
178       for(int x=0;x<nx;x++){
179         if(rank==0){
180           output->addToData(x,ny-1,change->getData(x,ny-1,temp_));
181           change->setData(x,ny-1,0.f);
182           change->setData(x,ny,0.f);
183         } else if (rank==size-1){
184           output->addToData(x,0,change->getData(x,0,temp_));
185           change->setData(x,0,0.f);
```

```
186           change->setData(x,-1,0.f);
187         } else {
188           output->addToData(x,0,change->getData(x,0,temp_));
189           change->setData(x,0,0.f);
190           change->setData(x,-1,0.f);
191           output->addToData(x,ny-1,change->getData(x,ny-1,temp_));
192           change->setData(x,ny-1,0.f);
193           change->setData(x,ny,0.f);
194         }
195       }
196       output->share();
197
198       //Pass information neighbour
199       change2->addBorders();
200       for(int x=0;x<nx;x++){
201         if(rank==0){
202           neighbor->addToData(x,ny-1,change2->getData(x,ny-1,stemp_));
203           change2->setData(x,ny-1,short(0));
204           change2->setData(x,ny,short(0));
205         } else if(rank==size-1){
206           neighbor->addToData(x,0,change2->getData(x,0,stemp_));
207           change2->setData(x,0,short(0));
208           change2->setData(x,-1,short(0));
209         }else{
210           neighbor->addToData(x,ny-1,change2->getData(x,ny-1,stemp_));
211           change2->setData(x,ny-1,short(0));
212           change2->setData(x,ny,short(0));
213           neighbor->addToData(x,0,change2->getData(x,0,stemp_));
214           change2->setData(x,0,short(0));
215           change2->setData(x,-1,short(0));
216         }
217       }
218       neighbor->share();
219
220       //If this created a cell with no contributing neighbors, put it on the
              queue
221       for(i=0; i<nx; i++){
222         if(neighbor->getData(i, 0, tempShort)==0){
223           temp.x = i;
224           temp.y = 0;
225           que.push(temp);
226         }
227         if(neighbor->getData(i, ny-1, tempShort)==0){
228           temp.x = i;
229           temp.y = ny-1;
230           que.push(temp);
231         }
232       }
233     }
234     //Check if done
235     finished = que.empty();
236     finished = output->ringTerm(finished);
237   }
```

B Figures

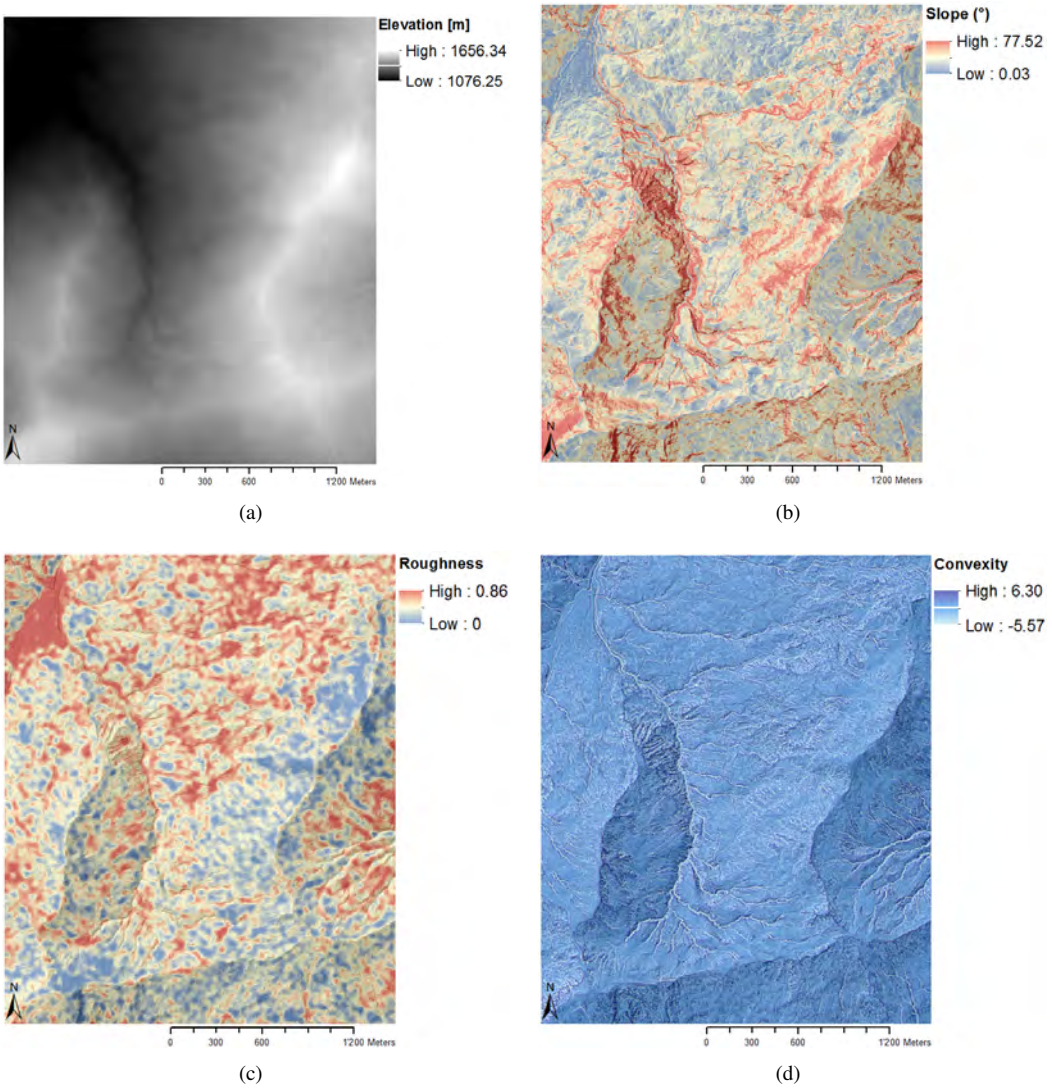


Figure B.1: Grids showing different topographic attributes of the Zwackentobel: (a) Elevation, (b) slope, (c) roughness and (d) convexity.

C Personal Declaration

I hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the thesis.

Zurich, 31st January 2013

Christian Venzin