Student thesis series INES nr 391

Development of methods for flood analysis and response in a Web-GIS for disaster management

Ludvig Forslund

2016
Department of
Physical Geography and Ecosystem Science
Lund University
Sölvegatan 12
S-223 62 Lund
Sweden

Ludvig Forslund (2016).

Development of methods for flood analysis and response in a Web-GIS for disaster management

Utveckling av metoder för analys och respons av översvämningar i ett Web-GIS för katastrofhantering

Master degree thesis, 30 credits in *Geomatics*
Department of Physical Geography and Ecosystem Science, Lund University

Level: Master of Science (MSc)

Course duration: *January 2016* until *June 2016*

Disclaimer

Development of methods for flood analysis and response in a Web-GIS for disaster management

---

Ludvig Forslund

Master thesis, 30 credits, in *Geomatics*

Supervisor: Ali Mansourian
Department of Physical Geography and Ecosystem Science

Exam committee:

Anna-Maria Jönsson
Department of Physical Geography and Ecosystem Science

Petter Pilesjö
Department of Physical Geography and Ecosystem Science

Abstract

In this thesis, two methods were developed for a web-based GIS to make the spatial perspective of flood disaster risk reduction and response accessible. Flood disaster can cause major disruption to the street network, which further affects the emergency services and evacuation work in finding the safest and fastest routes. Thus, a method for dynamic street network analysis was developed and implemented into a Web-GIS for disaster risk management, for demonstrational and testing purposes. The dynamic routing calculations were performed within a PostgreSQL database using OpenStreetMap data as the street network, and polygons defining the areas with floods or other obstructions. Testing was performed for the relative server response time by measuring the response for an increasing amount of equally sized polygons as flooded areas. The results indicated that the method performed in a suitable way for a web environment.

A second method aimed at analysing affected areas from a fluvial flood was also developed. Since the intention was to perform the flood modelling within a Web-GIS client, the focus of this method was to develop and utilize algorithms with low time complexity. Thus, a sequenced flat-surface modelling was created for calculating the water-increase of a sloping watercourse. Performance of the method was further assessed through implementation into the Web-GIS client, where the results indicated on a high performance in terms of time and accuracy.

Throughout this thesis the relevance of the developed methods was put into the context of current international and Swedish flood disaster risk management. While much of the work in the developed world revolves around building infrastructural resilience, the developing world finds itself disproportionately vulnerable to natural hazards, and thus in larger need of methods for disaster response. The methods developed in this thesis were extra relevant to users with a low GIS experience working with disaster risk reduction and response in flood prone areas.

Keywords: Physical geography, Web-GIS, Dynamic street network analysis, Fluvial flood analysis, Flood disaster response

Sammanfattning

I detta examensarbete utvecklades två metoder för implementering i ett webbaserat GIS i syfte att tillgängliggöra geografiska verktyg för katastrofförebyggande arbete samt som verktyg för respons under pågående katastrof. Fokus ligger på översvämningskatastrofer, vilka kan orsaka stora störningar i trafiknätet som i sin tur påverkar möjligheten att finna rätt och snabbast väg för till exempel räddningstjänsten. En av de utvecklade metoderna behandlar dynamisk gatunätsanalys som vidare implementeras i en webbaserad GIS-klient för demonstration och representativ prestandautvärdering. Beräkningarna sker med OpenStreetMap-data i en PostgreSQL-databas samt med polygoner för beskrivningen av översvämmade områden. Applikationen och metoden testas genom att mäta server-responstiden för ett ökande antal likformade översvämningspolygoner. De genererade resultaten visar att metoden är lämplig för användning i webbmiljö.

Vidare utvecklas en metod för modellering och analys av fluvial översvämning och dess påverkan på omkringliggande områden. Även här är intentionen att utveckla en lämplig metod för webbmiljö vilket medför att fokus läggs på användning och utveckling av tidsoptimerade översvämnings-algoritmer. För detta utvecklas en sekvenserad modellering av plana ytor för att effektivt kunna beräkna en ökning av vattenytan i ett sluttande vattendrag. Utvärdering av metodens prestanda och noggrannhet testades genom implementering i den webbaserade GIS-klienten, vilket visade godkända resultat för metodens syfte.

I detta examensarbete studeras också de utvecklade metodernas relevans inom internationell samt svenskt katastrofberedskapsarbete. Arbetet i de utvecklade länderna kretsar kring att bygga infrastruktur för att öka motståndskraften där utvecklade metoder kan bistå som analysgrund. Utvecklingsländerna har lägre motståndskraft där utvecklade metoder i denna uppsats kan fungera som respons till pågående katastrof. Metoderna är främst utvecklade till en användargrupp med låg erfarenhet av GIS och som arbetar med katastrofberedskap och respons i översvämningsdrabbade områden.

Table of contents

Acknowledgements

I would like to express my sincere gratitude to my supervisor Ali Mansourian for guiding me through this work. A big thanks to Anton Lundkvist for your extensive technical support with WGRAS and Web-GIS in general. I would also like to thank all the hospitable lecturers, researchers and PhD-students at the GIS-centre for letting me perform my studies there and being very inspirational and kind to me. Last but not least, I would like to thank Maja Jensen for reading, commenting and all in all made this effort reachable.

Abbreviations

AJAX	Asynchronous JavaScript and XML, web development techniques for sending and retrieving data from a server as a background process.
API	Application Programming Interface, a set of pre-developed protocols for building software applications.
DEM	Digital Elevation Model, a three-dimensional representation of the surface of a landscape in terms of elevation.
GDAL	Geospatial Data Abstraction Library, an open source raster and vector geospatial processing library available for python.
HOT	Humanitarian OpenStreetMap Team, coordinates the volunteers contributing to OSM during major disasters.
LIDAR	Light Detection And Ranging, measures distance using laser for producing high-resolution digital elevation models (see DEM).
OGC	Open Geospatial Consortium, international organisation committed to develop standards for interoperability within the spatial data community.
OSM	Open Street Map, a collaborative project creating a free and editable map of the world.
OWS	OGC Web Services, name of the standards created for use in World Wide Web applications.
PL/pgSQL	Procedural language for PostgreSQL, enables scripting of several SQL statements forming a function to trigger more complex procedures.
SQL	Structured Query Language, a domain-specific language for managing data in a relational database management system.
VGI	Volunteered Geographic Information, products of open geographical data created voluntarily by individuals.
WGRAS	Web-GIS for Risk Assessment System, the name of the application operating as a platform for the work of this thesis.
WCS	Web Coverage Service, standard for sharing raster data over the web.
WFS	Web Feature Service, standard for sharing vector data over the web.
WMS	Web Mapping Service, standard for sharing maps over the web.
WPS	Web Processing Service, standard for performing geospatial processing over the web.

1. Introduction

Disaster is the following events of a natural hazard such as flood, earthquake, hurricane, tsunami and landslide. The impact on people and societal infrastructure is extensive, resulting in deaths, displacement of people and economical loss. These impacts are disproportionately high in developing countries, mainly because resilience is low. Disaster is a combination of natural hazards and poor preparedness and thus preventable even though hazards are naturally occurring. Further, the severity of disasters globally has risen due to increase in population density in areas especially prone to natural hazards (UNISDR 2015). With climate change, there is a high probability that future natural hazards will have increased frequency and intensity, thus more extensive disasters in unprepared areas (IPCC 2014).

Disaster management focuses on reducing impact of hazards as well as increasing resilience and infrastructure for effective response and recovery. Interest and use of GIS, and in particular Web-GIS, has increased for analysis of disaster risk reduction and response. Geospatial technology can be tailored to support all stages of the disaster management cycle with methods for scenario simulation, identification of vulnerabilities and streamlining of emergency response (Peggion et al. 2008). Further, through the efforts of the VGI-community (volunteered geographic information) during recent disaster events, to the growth of specialized Web-GIS applications, GIS has been made accessible for non GIS-experts.

The work of this thesis focuses on the development of two methods for supporting the analytical work of flood disaster risk and response. A major impact from floods is the blocking of the street network through inundation or damage. The need for a reliable route is important for emergency services and evacuation in order to safely and efficiently continue the disaster response. Therefore, it is important to gather information on the extent of the floods as well as implementing methods to process such dynamic data.

The second focus of this thesis is web-based fluvial flood analysis to support the analytical work of disaster risk management. By modelling a possible water-level increase of a watercourse, the impact on adjacent infrastructure, such as buildings and roads, can be examined. However, traditionally these operations are very complex in terms of processing time and data-input, and thus not suitable for a web-based application. The main objective is therefore to find a simple and efficient method which will be elaborated throughout this thesis.

Since the developed methods are intended for web-based GIS, fundamental parts of the method development are the user-oriented accessibility as well as the suitability of the methods for this platform. Therefore, these methods will be demonstrated through implementation of tools into the Web-GIS for Risk Assessment System (WGRAS). This system is currently in research and development at the GIS Centre in Lund in collaboration with the World Health Organisation (WHO) regional office for Europe in Copenhagen.

1.1 Problem statement

During major disaster events, data describing the extent of damage from the flood, earthquake, hurricane or landslides is often available, however tools to manage and respond to such events are few. When the landscape undergoes temporary change, the standard geographic methods and spatial data becomes obsolete. Therefore, accessible methods dynamic to variations are needed (Nayak and Zlatanova 2008).

One of the most affected parts of society during floods is the road infrastructure used for transportations and emergency services. The standard methods for route-planning are not designed for taking obstructions in the road network into account and thus not show alternative driving directions. Hence, a method for dynamic street network analysis and how it could be implemented in a Web-GIS, is of interest. Here, objectives such as performing computations of large datasets and solutions for increasing time performance, are also in focus.

Flood disaster management is not only a matter of combatting present events, but also of analysing future events and the ability to respond. However, performing flood analysis and inundation modelling is very computational intensive, and thus not suitable for a Web-GIS implementation. Instead, simpler methods can be implemented, not including variables such as the exact water inundation-level and runoff speed, but that instead enables analysis of how a flooded watercourse affects the surrounding area. Thus, not analysing the probable water-level increase of a watercourse during a flood event, but what if the water-level increases with x metres. The main challenge is to model the flooding of a sloping watercourse without implementing very time-consuming algorithms.

1.2 Aim and objectives

The main aim of this thesis is to develop methods for flood disaster analysis, planning and response. These methods will further be evaluated and demonstrated by implementation into a Web-GIS for disaster management. Therefore, the methods developed will be focused towards suitability in a Web-GIS environment. In order to meet this aim, a set of objectives are further defined:

Objectives:

- Develop a method for dynamic street network analysis performed in a Web-GIS system architecture.
- Develop a method fluvial flood analysis using time efficient algorithms suitable for Web-GIS implementation.

Since the work of this thesis is in line with an on-going initiative at the Lund University GIS Centre, a set of requirements are formulated:

Requirements:

- The developed methods must be based on open-source data and software.
- The focus of the resulting methods should be usability and accessibility.
- Methods should be applicable on a global scale.
- The required level of pre-knowledge in GIS should be low.

1.3 Users

During the work of this thesis, the methods are developed and designed towards a certain user group. A user in this group is not a GIS-expert but instead an expert within other fields such as health, evacuation planning and emergency operations with the need of spatial analysis. Hence, the methods are developed with regards to the previous knowledge of the potential user group, where streamlining versus the user’s freedom of choice is evaluated at every instance. Since WGRAS is a project in collaboration with the WHO, the work of WHO is placed as a context for the methods developed through the work of this thesis. Key factors of disaster management emphasised by the WHO are displacement of population, impact on vital infrastructure and assisting in emergency planning (Menne and Murray 2013). Therefore, the perspectives of WHO were present already during the selection of chosen methods.

2. Background

In this section the theoretical framework involved in the development of mentioned methods will be introduced. Further, a review of disaster risk management in general and within the study area in particular will be presented. Finally, a thorough introduction of the Web-GIS client WGRAS, which serves as the platform for implementation and testing, will be described.

2.1 Web-GIS

A Web-based Geographical Information System (GIS) is fundamentally a GIS accessible through a web browser. It is performed by implementing management and visualization of spatial data into a web-environment through the combination of GIS and web technologies. Web-GIS was initially intended as a tool for web mapping and data download, however, through the years more focus has been put upon advanced applications targeted towards certain fields of research and purposes (Yang et al. 2005). Within these applications, processing and analysis of spatial data is implemented through customized methods (Aye et al. 2016).

One of the reasons for implementing GIS in a web-environment is to facilitate sharing of spatial data and increase GIS accessibility. However, structure of spatial data is complex and solutions regarding the interoperability between different data sources is essential. Therefore, the Open Geospatial Consortium (OGC) has developed a set of standards (OWS) for the communication of maps and spatial data over the Hypertext Transfer Protocol (HTTP). The OWS standards include the Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), Web Processing Service (WPS), extensively used within the Web-GIS community for communication of vector and raster data (OGC 2016).

2.2 Disaster

According to the UNISDR (2009), disaster risk is defined as a disaster event with the potential of losses in lives, health status, assets and services, affecting a society in an varying time period. A disaster can further be described as *"a serious disruption of the functioning of a community or a society involving widespread human, material, economic, or environmental losses and impacts, which exceeds the ability of the affected community or society to cope using its own resources."* (UNISDR 2009). Disaster events can origin from three different sources such as natural disasters, technological disasters and man-made disasters (Mansourian et al. 2006). Further, floods are natural disasters and defined as the temporary covering by water of land not normally covered by water (European Parliament 2007). The two most common causes of flood disaster are listed below.

- Pluvial flood – An event of extreme downpour in urban areas with low infiltration capacities and exceeds the dimensioning of the sewage system.

- Fluvial flood – An indirect type of flooding which triggers an increase in water-level elevation of a watercourse downstream and causes the river to exceed its banks. These events are often caused by heavy downpour or snowmelt somewhere within the catchment area (Apel et al. 2016).

Regardless of the origin of disasters, the resilience of society can be increased by implementing strategies and improved coping capacities (UNISDR 2009). Such measures can be applied before a disaster, with management and development of infrastructure, to reduce disaster vulnerability (UNISDR 2015). In the case of flood disaster, work should be focused on analysing risks throughout flood plains and to implement measures reducing the impact factor of floods as well as methods for response (European Parliament 2007). Following two sections will describe how this is practically implemented on a regional level, as well as how volunteered geographic information (VGI) provides valuable data in response to disasters.

2.2.1 Current work with flood disaster management in Sweden

During the Sunday of 31 August 2014, the city of Malmö experienced an event of intense precipitation resulting in extensive floods. For the duration of 24 hours, a record of 100 mm rained with 31.7 mm during the most intense hour (SMHI 2014). Cost from damages caused by the resulting floods is estimated to 250 million Swedish crowns (Sydsvenskan 2014). According to SMHI (2014), a rain of this magnitude in Malmö has a recurrence interval of 100 years. From the experience of this and similar events in Sweden, a larger focus on flood disaster risk reduction and management has been highlighted. Also, the warnings of an increased flood frequency and sea-level rise in future climate (IPCC 2014) has contributed to an increased focus on preparedness and management.

The Swedish Civil Contingencies Agency (MSB) has developed a national work plan following the requirements set in directive 2007/60/EC (European Parliament 2007) for flood disaster management (MSB 2013). In these directives, a large emphasis is put on production of flood risk maps and plans as well as methods and tools for analysis and response. MSB (2011) is responsible for identifying areas with considerable flood risk in Sweden, where a vulnerability assessment is produced based on the amount of people affected within each high risk area. MSB (2013) further assign county governments the role in providing local risk assessments and take action in lowering the impact of pluvial and fluvial floods on society. In Scania, where the city of Malmö is situated, the local county government has made an investigation on preventive measures for improving the community planning preparedness for floods (Persson et al. 2012). Examples of suggested measures revolve around keeping a sufficiently dimensioned storm water management, regulations on hard surfaces (concrete, tarmac), limiting barriers for urban runoff and green roofs (Persson et al. 2012).

After the flood events in 2014, the city of Malmö has initiated a plan for extreme precipitation events, where the goal is to implement flood resilience in the urban planning. However, this framework is yet to be published, although some key solutions have been presented by Malmö stad (2016). Here, the efforts revolve around blue-green hydrological solutions using open storm water ditches and pools. Other suggested solutions regard leading the urban runoff away from buildings and infrastructure to areas with high permeability and proper dimensioning for storage of water. However, approaches and methods during an ongoing flood are not yet described.

2.2.2 VGI for disaster response

Volunteered Geographic Information (VGI) is the concept of free and open geographic data and follows the web phenomenon of user created content. Any person with access to the internet can edit the collective database and provide new spatial information. Volunteers are also reviewing the provided content to confirm that it holds accuracy and significance (Goodchild 2007). VGI has thus transformed and challenged the traditional way of obtaining spatial data by freeing the end-user from cost barriers and copyright restrictions (Lin 2016). Hence, the number end-users with access to spatial data has grown and led to an “democratization of GIS” (Goodchild 2007). The content is created within different frameworks, where standards are defined for categorization and how to properly attribute the digitized features.

One such framework, and the most popular, is the Open Street Map (OSM) project introduced in 2004 with an open content license. OSM is one of the key driving forces of VGI and aims at building and distributing a free editable map of the world. Initially, OSM was dedicated to mapping streets but has over the years transformed into additionally mapping buildings and land use, among others (Lin 2016).

Because of the large group of active contributors, OSM has shown to be especially efficient during disaster response. A prominent example is the events following the Haiti earthquake in 2010. 600 volunteer mappers joined the humanitarian response where digitization of roads and buildings were made from available satellite images (Soden and Palen 2014). Initially, the OSM dataset had almost no coverage of the Haitian island, and the standard web map applications (e.g. Google Maps and Bing Maps) had even less. During the month following the earthquake event, over 10,000 edits were contributed to the area in and around the Haitian capital, Port-au-Prince, alone. These spatial data played a significant part to disaster response and aid on the ground, where OSM was relied upon as the main supplier of geographical information (Zook et al. 2010). Following the Haitian response of the VGI-community, the Humanitarian Open Street Map Team (HOT) was established for future disaster response in poorly mapped areas. It has since been a central collaborator in disaster response during disasters and epidemic disease outbreaks (Soden and Palen 2014).

2.3 Network analysis

Network analysis is first introduced by Leonard Euler in 1736 when trying to solve the so-called Königsberg bridge problem (Evans and Minieka 1992). It has since found its way into many practical applications such as psychology, electrical and computational engineering, transport planning , management and social research (Evans and Minieka 1992). In this thesis, network analysis will be studied and utilized in its geographical context, finding shortest paths between point x and y in a physical road network.

In network analysis, a graph consists of two parts describing interconnectivity between entities, which are referred to as vertices (points) and edges (lines) (Evans and Minieka 1992). Further, two vertices a and b, in a graph G are connected if an edge $e = ab$ also exists in graph G. Then edge e joins vertices a and b (Worboys and Duckham 2004). In this state, graph G simply explains whether two vertices are connected or not. However, extensions such as directed and labelled can be introduced to graph G, which gives edges properties that further explain the connection between vertices. A Directed graph limits the connectivity of edges to specific allowed directions as the following examples: $e = a \rightarrow b$, $e = a \leftarrow b$ or $e = a \leftrightarrow b$ (Worboys and Duckham 2004). Labelled graphs assign a weight to each edge defining the cost of travelling from a to b through edge e (Worboys and Duckham 2004).

When studying the spatial relationships of road networks, a *vertex* and *edge* can for simplicity instead be explained as an *intersection* and *road segment*, respectively. A vertex is hence created for every intersection between two or more roads (edges). Since road networks often include restrictions such as one-way roads, a directed graph is utilized to limit the allowed drivable direction (Worboys and Duckham 2004). Further, in street network analysis, labelling of edges primarily represents the distance of road segments, but can also be combined with a set of factors such as speed limit, type of road and traffic intensity (Worboys and Duckham 2004).

2.3.1 Shortest path algorithm

When a graph (Figure 1a) with vertices and edges is created for a street network, a method is needed for finding the shortest path between two points *A* and *D*, where *A* and *D* rarely are direct neighbours (Figure 1b). Several algorithms exist for solving this task such as the A-star (Hart et al. 1968) algorithm and the Bellman-Ford algorithm (Bellman 1958). However, for the work of this thesis, Dijkstra's algorithm (Dijkstra 1959) is studied and applied. Dijkstra's algorithm is first and foremost based on finding the shortest path to all vertices in a graph with a specified starting vertex *A*. The algorithm starts by setting the total distance from *A* to all vertices in graph *G* to infinity except the starting node *A*, which is set to zero. In addition to distances, an instance also keeps track on whether a vertex has been "visited" or not, meaning the adjacent connected vertices of a vertex has been evaluated. If a vertex has been evaluated, a path has been found to that vertex and the initial infinite weight can be recalculated. The algorithm then proceeds by visiting the unvisited but evaluated vertices, and recalculate the total distance to each vertex if a shorter path is found. When all vertices are visited, the algorithm will terminate and the shortest path between *A* and *D* can be reconstructed (Figure 1c).

In terms of performance, Dijkstra's algorithm computes with a time complexity of $O(n^2)$, where n is the number of vertices analysed (Worboys and Duckham 2004). Since the computational complexity of Dijkstra's algorithm is dependent on the number of vertices analysed, the algorithm can be terminated once the destination vertex *D* is visited. In some cases, this method lowers *n*. However, it cannot be guaranteed that *D* is not the last visited vertex in the network (Worboys and Duckham 2004).

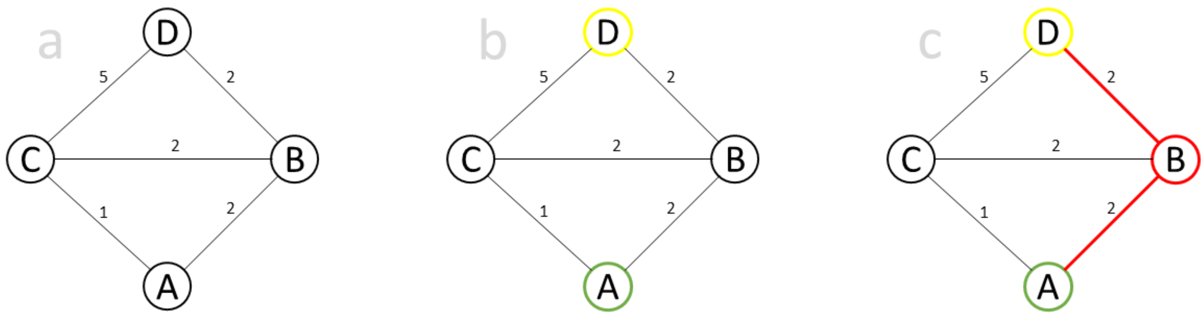

Figure 1: Graphical description of network analysis. (a) consisting of vertices A, B, C, D and edges AB, AC, BC, BD & CD, where labels correspond to the weight of each edge. (b) represents a start-vertex A (green) and end-vertex D (yellow). (c) shows the shortest path between start-vertex A and end-vertex D through vertex B.

2.4 Flood modelling

Traditionally, GIS based flood analysis is very data and computationally intensive. It demands storage and processing of high-resolution digital elevation raster data, as well as complex modelling including input data regarding soil, drainage infrastructure, watercourse discharge, evaporation, precipitation and land cover. When analysis is performed on this level, the aim is to properly model the propagation of a flood inundation over time (Zhang and Pan 2014).

When modelling the propagation of a flood using a digital elevation model, two different approaches regarding the dynamics of water can be used: flat-water modelling and physical-based modelling. The main difference between the two is how the inundating waterbody is modelled in-between each raster DEM cell. The flat-water method focuses on DEM depressions and the accumulation of water within these during a flood event. Here, the inundation elevation is governed by the inflow (precipitation, run-off) and outflow (drainage, soil infiltration) of the depression. The water inundation elevation of each depression is treated as a static surface, where neighbouring "dry" cells that are lower than the inundation elevation are set to "wet", and vice versa if the inundation elevation decreases (Chen et al. 2009). The physical-based modelling approach instead focuses on the progression of a flood-wave through the urban environment. It thus treats the flooding as a flow rather than an accumulation and considers the dynamic water mass balance of each cell. For calculating the flow in-between DEM cells, the method applies Manning's equation, where surface roughness and gravity are considered the main driving forces for the flood propagation (Bates et al. 2010; Fewtrell et al. 2011).

Of the two mentioned methods, the flat-water method is traditionally used within pluvial flood modelling, while fluvial flood is modelled with the physical-based method. However, by considering the dynamic water mass in physical-based modelling, the time complexity is greatly increased (Chen et al. 2009).

2.4.1 Flood-fill algorithm

For modelling the flat-water surface, the flood-fill algorithm can be utilized. The flood-fill algorithm is traditionally used by computer sketch programmes in the so-called "paint bucket tool" used to fill each cell with a uniform colour within a defined boundary (Figure 2) (Nosal 2008). In these applications, the boundary is often defined as being of another colour than the cell where the "bucket tool" is applied. For this task, the algorithm performs an eight directional neighbourhood search, visiting all cells with the same colour as the clicked cell, and assigns the colour of the bucket (Nosal 2008). However, cells outside the boundary with the same colour as the clicked cell, are not assigned to the colour of the bucket since these are never "seen" and visited by the search window (Figure 3).

The application of the flood fill algorithm in GIS modelling is further explained in the method chapter.

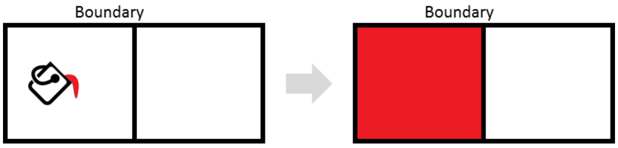

Figure 2: Graphical demonstration of the "paint bucket tool" using the flood-fill algorithm.

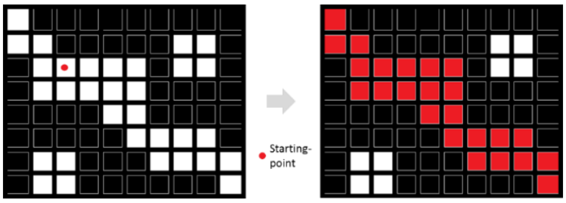

Figure 3: Cell-level visualization of the flood-fill algorithm. The tool is applied in a cell (red point) and the connected white areas change colour. However, white areas at other side of boundary (black cells) are unaltered.

2.5 WGRAS

The Web-GIS for Risk Assessment (WGRAS) is a project at the Department of Physical Geography and Ecosystem Science as a collaboration between the GIS-centre and the WHO regional office for Europe in Copenhagen (LUGISC 2016; WHO 2016). The project aims at building a Web-GIS for disaster risk management with focus on interoperability and accessibility. Here, accessibility refers to a higher level of interactivity, not only for displaying and viewing spatial data, but also the ability to manipulate and analyse.

This project involves a handful of researchers at the GIS Centre, where the work of this thesis has contributed with research on flood analysis and response methods.

Following sections will describe the WGRAS system architecture to clarify and explain the instances where the developed methods are processed within.

2.5.1 Client

The client of a web application is executed in the browser when a user requests access to the webpage. The client is hence the user interface and is developed as a Single Page Application in the WGRAS system architecture (Figure 4), meaning all functionality is accommodated within one HTML-document. It is developed in JavaScript using the Ext Js library (ExtJs 2016), which works as the framework and design of the HTML document, where functionality and tools can be developed for user interaction. A large portion of this framework is the map interface, which is developed using the OpenLayers 3 web mapping API (OpenLayers 3 2016). OpenLayers is open source and enables visualization and management of spatial data in a web environment.

Besides the user and map interface, the client has a manager instance, where a user with administrator privileges has the ability to add, update and delete layers in the database. This instance of the WGRAS application is still under development, but as for the time of the work of this thesis, the user is able to upload shapefiles to the PostgreSQL database as PostGIS tables.

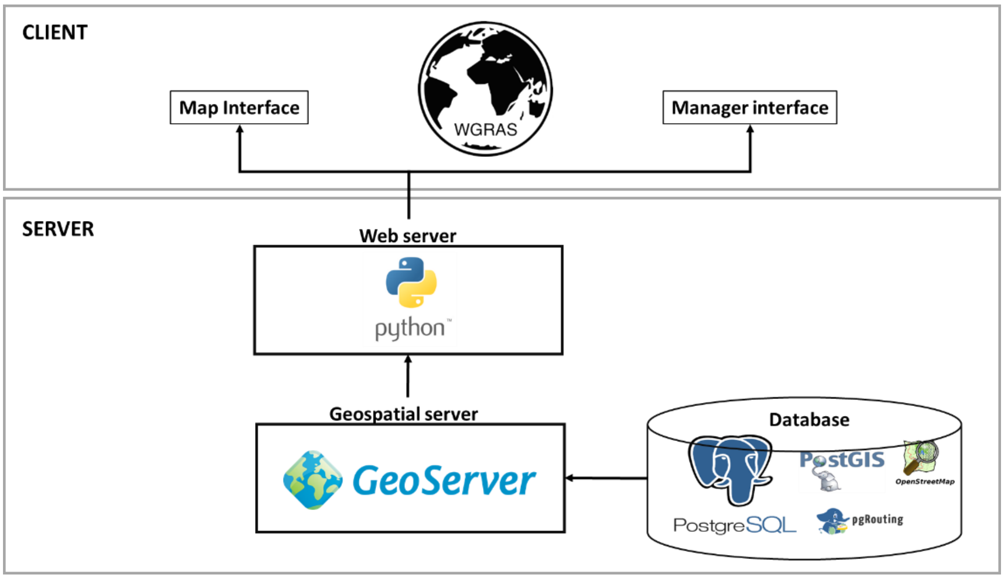

Figure 4: System architecture of the WGRAS application.

2.5.2 Server

The server is the instance where the HTTP-request from the client is processed and supervised. The server is therefore the storage of all components needed to build the web application in the client-side web browser. Structure of a server is different based on the purpose of the application where the WGRAS system architecture (Figure 4) is based on a typical Web-GIS application. In following sections, the three main instances of the WGRAS server will be further described.

2.5.2.1 Web server

When a client is granted access to the web-application, the web server returns the code back to the requesting client where it is published in the web browser. Based on the following interactions of the user, additional functionality is served and processed using the other instances of the server. Thus, the web server can be described as the portal of the server handling the communication between the server and the requesting client. The web server can also be customized to process any functionality desired by the developer. In WGRAS, the web server is developed in Python 2.7.11.

2.5.2.2 Geospatial server

Within a Web-GIS server, a geospatial server is required to publish the spatial data to the client. In WGRAS, GeoServer is applied which is an open source Java server software developed for spatial data publishing (GeoServer 2015). Its main goal is to build a software designed for interoperability within the domain of spatial data regardless of source (GeoServer 2015). Thus, GeoServer uses, and is the main implementer of, map publishing OWS standards created by OGC (GeoServer 2015). GeoServer also supports several vector file formats such as shapefile, GML, PostgreSQL tables, as well as raster formats such as GEOTIFF, ArcGrid and GDAL image formats (GeoServer 2015).

GeoServer also provides so-called SQL view requests. An SQL view request serves as any other layer request from GeoServer, which is; locate the requested layer geometry in the database and send it back as a WMS image. However, when a layer is defined as an SQL view, the selection can be customized to meet a defined criterion. (GeoServer 2015). Further, SQL views can be parameterized so that each request can provide information on how and what data to selects from the database. For example, if a SQL view is defined for selecting the five largest cities in a country from a database table of cities around the world, a parameter can be set for sending information on what country to select data from.

GeoServer SQL views also enables the possibility to define *validation regular expressions* for each parameter to protect the database from SQL-injections. This to ensure that only the intended type of information is permitted to pass through to the database. Otherwise, malicious code can be forwarded and harm the database.

2.5.2.3 PostgreSQL database

PostgreSQL is an open source object-relational database system for storing and retrieving data (The PostgreSQL Global Development Group 2016a). In WGRAS, PostgreSQL serves as the general database of the web server as well as a spatial database for storing and processing vector layers.

By extending with PostGIS (The PostGIS Development Group 2015), storage of geographic features and spatial queries are added to the PostgreSQL database. Further, the pgRouting (pgRouting Community 2015) extension can be added to the database where shortest path algorithms can be applied on already imported street networks. In this extension an improved version of Dijkstra's algorithm is implemented with an enhanced time complexity of $O(V*log(V) + E)$, where V and E are number of analysed vertices and edges respectively (pgRouting documentation 2016).

When building a database application for more complex computations, a procedural language needs to be utilized since standard SQL statements are executed individually. In PostgreSQL, PL/pgSQL is incorporated to enable development of functions for scripting a procedure that can be triggered with input arguments (The PostgreSQL Global Development Group 2016b). PL/pgSQL thus enables procedural language elements such as conditions and for-loops as well as declaration of variables and constants. This makes it possible to execute several SQL statements and use the result for executing PostGIS functions or pgRouting functions (The PostgreSQL Global Development Group 2016b).

3. Methods

When it comes to research and development of methods for Web-GIS, a key aspect is how to perform the processing and user interaction. If the intention is to develop methods for a Web-GIS environment, it must also be properly tested within this context. Thus, the two methods developed as well as the details about the implementation is described in this section. For explanatory reasons, there will be a distinction between the method alone and the applied tool within the WGRAS system. Hence, the WGRAS system works as a platform for testing of performance and demonstrations throughout this thesis.

3.1 Study Area

Even though developed methods should work globally, a study area was chosen for testing and evaluating the relevance of the method when applied. The city of Malmö, which is located on the west coast of Scania, the southernmost county of Sweden, was chosen as the study area for this thesis (Figure 5). Malmö is the third largest city in Sweden and Scania is the second most densely populated region (SCB 2015). Malmö has a humid continental climate with an annual average temperature of 8.1 ºC and 602 mm precipitation (SMHI 2016).

For the fluvial flood modelling, a subsection of the Sege å watercourse was used as study area for the developed method and algorithm. The selected subsection was in the city of Malmö's periphery between coordinates [55°37'09.8" N, 13°06'45.3" E] and [55°37'14.0" N, 13°03'48.6" E], close to the outlet into the sound of Öresund.

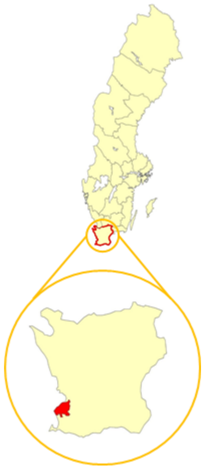

Figure 5: Map of Sweden highlighting Scania and the city of Malmö.

3.2 Data

3.2.1 OpenStreetMap

The dynamic street network analysis, was implemented using Open Street Map (OSM) data obtained from <u>www.geofabrik.de</u>, where it was possible to download coverage by continent or country. For the development of the dynamic network analysis, the most recent, as of 26th January 2016, OSM-data for Sweden was downloaded.

3.2.2 Digital Elevation Model

Through the use of airborne LIDAR, a nationwide point cloud was produced between 2009 and 2013 by the Swedish land survey. Further, the point cloud was interpolated into a 2x2 metres continuous grid with a standard error of 0.05 metres in elevation and 0.25 metres in the plane (Lantmäteriet 2015). Cells that represents water surfaces were also derived from the laser scanning as well as complemented with the measured centreline elevation of watercourses (Lantmäteriet 2015). However, watercourses narrower than 6 metres and lakes smaller than

0.25 km^2, were not included (Lantmäteriet 2015). Since the studied watercourse was 10-15 metres wide in average, it was included in the raster image.

During development of the flood analysis tool, a subset of this LIDAR data covering the study area was downloaded the 15th March 2016 from the Swedish University of Agricultural Science's (SLU) GET-service at *atlas.slu.se*. The DEM raster Tiff image was delivered with SWEREF 99 TM as the plane reference system and RH 2000 as the height reference system (Lantmäteriet 2015). According to the supplied metadata, the elevation cells within the study area were scanned the 7th April 2010.

3.3 Preparation of the datasets

3.3.1 Street network

The downloaded OSM-file contained many features additional to road data such as buildings, railroads, rivers, land use and administrative boundaries. Therefore, street data had to be extracted since these are the only features of interest for a routeing application. Further, road features needed to be prepared for successfully doing a network analysis by calculating the network topology. With the use of Osm2pgRouting (Daniel Wendt 2016) the vertices were created at road intersections, and edges attributed with a source and target vertex. An XML-file, readable by Osm2pgRouting, was created to configure what road-classes to import to the PostgreSQL database. Since the intended application of this thesis only handled dynamic routeing for cars, road features defined as cycle or pedestrian paths were excluded. Further, weights were assigned to road classes where the larger the road the smaller the weight (Table 1).

Table 1: Road classes imported to the PostgreSQL database and their corresponding assigned weights.

Road class	**Weight**
Living street	1.5
Secondary	0.8
Tertiary	0.8
Primary	0.6
Primary link	0.6
Trunk	0.4
Trunk link	0.4
Motorway junction	0.4
Motorway link	0.4
Motorway	0.3

Since vertices and edges are added to the database separated by country, a method had to be created to keep track of which country the routeing was to be performed in. Thus, a country specific two-letter prefix, following the international standard ISO 3166-1 alpha-2, was added to each table of *vertices* and *edges* in the database. When, for example, the Swedish road network was added to the PostgreSQL database, the two tables containing vertices and edges, had the following names: "*se_vertices*" and "*se_edges*".

After import, a column named *flooded* was added to the *edges* table, which was utilized later in the dynamic street network analysis.

While importing the routeing topology with Osm2pgRouting, some information describing the road features was lost. For the dynamic network analysis tool, it was decided that the information whether a road feature was a bridge or not had significance. For this, Osm2pgsql (Burgess et al. 2014) was used to import the raw data from the same OSM-file. As with the Osm2pgRouting above, an XML-file was created to select the desired information. In this case, the column describing if a road feature is a bridge or not and the column with the feature-ID, was the two imported. After import the bridge column was added to the *edges* table by matching the feature id.

3.3.2 Digital Elevation Model

After the download of the DEM, the GEOTIFF was added to the file database in GeoServer, ready for visualization in WGRAS or data extraction with the available Web Coverage Service (WCS).

3.4 Dynamic street network analysis

The objective here was to develop a suitable, efficient and reliable method for performing dynamic network analysis within a Web-GIS application. The method was designed to route around areas that are defined as flooded or has some other kind of obstacle (Figure 6).

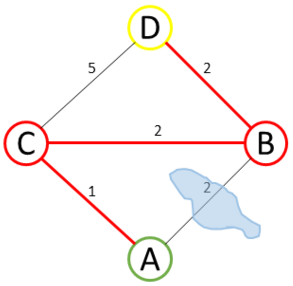

Figure 6: Same graph as in Figure 1, showing the shortest path between A & D through C & B. Blue polygon represents a flooded area making edge AB unrouteable.

The dynamic street network analysis was performed through three of the instances in the WGRAS system architecture (Figure 7). First, the map interface where *flooded areas* and the coordinates to route between, were specified by the user. Secondly, the GeoServer SQL view, managed the connection to the PostgreSQL database and published the returned geometry back to the map interface. Finally, the data management and network analysis was performed in the PostgreSQL database.

The rest of this section describes the detailed method in correspondence to the system sections of the WGRAS system.

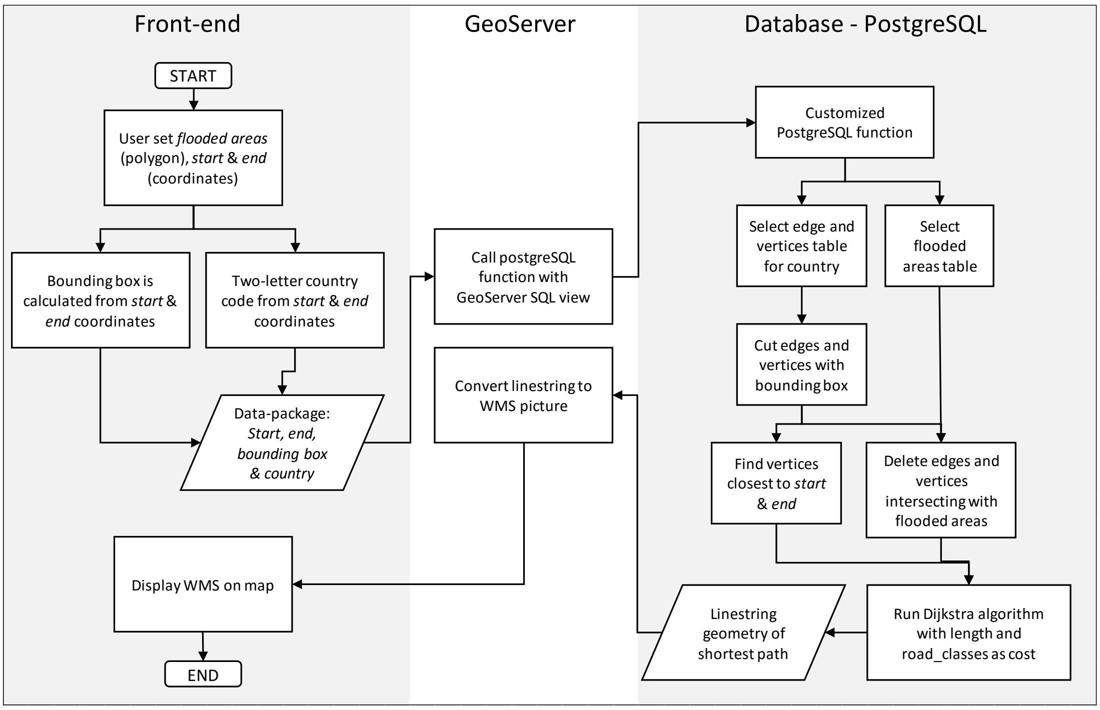

Figure 7: Flow chart of the Dynamic street network analysis through the WGRAS system architecture.

3.4.1 Client

A user interface and underlying functionality was developed in the client map application. Here, the user was able to specify the coordinates to route between by clicking in the map interface or by typing the address. The interface also enabled selection of *flooded areas* which were avoided during routeing. *Flooded areas* were stored as PostGIS polygons in the database which could be added by the user through the WGRAS manager interface.

When a user entered an address text as a location identifier, the location of the address had to be converted into its coordinate. This process was called geocoding (Ratcliffe 2001), which was performed in WGRAS with Nominatim, an open source geocoding API developed together with OpenStreetMap (Quinion et al. 2016). Nominatim was also used to find the country specific two-letter code, which is derived from a coordinate through reverse geocoding (Quinion et al. 2016).

After a user hit the *Route* button, all coordinates (*start*, *end*, *bounding box*) were packaged into an array together with the *flooded areas* name-string and country specific two-letter code. These parameters were further sent on as a WMS-request to the GeoServer SQL view.

3.4.1.1 Dynamic Bounding box

Since the time complexity of the Dijkstra algorithm is very dependent on the number of vertices and edges analysed, a method for limiting these to the least possible was developed. Hence, a bounding box dynamic to the user input coordinates was created using the following method.

A spatial bounding box is specified using two coordinates (x_{Min}, y_{Min} & x_{Max}, y_{Max}) and is referred to as the minimum bounding rectangle surrounding a geographical feature (Caldwell 2005). Since this application uses two coordinates declaring the route start and end, a bounding box would simply be created by ordering the coordinates by magnitude in respective dimensions as in equation (1) (Figure 8a).

$$\begin{aligned} x_{Min} &= \min(x_1, x_2) \\ y_{Min} &= \min(y_1, y_2) \\ x_{Max} &= \max(x_1, x_2) \\ y_{Max} &= \max(y_1, y_2) \end{aligned} \tag{1}$$

However, if the street network analysis was limited to an area inside the coordinate pair alone, a risk of excluding possible shortest routes was probable. This risk was significantly higher when routeing over small distances. Further, when a bounding box was based on two point-coordinates as in equation (1), there was a risk of a resulting bounding box with only one dimension, where $x_1 = x_2$ or $y_1 = y_2$, i.e. no area.

After calculating the minimum bounding rectangle (equation 1), a proportion of the height (x) and width (y) was added respectively as in equation (2) (Figure 8b).

$$\begin{aligned} New\_x_{Min} &= x_{Min} - \Big(\big((x_{Max} - x_{Min}) * 0.3\big) + 5000\Big) \\ New\_y_{Min} &= y_{Min} - \Big(\big((y_{Max} - y_{Min}) * 0.3\big) + 5000\Big) \end{aligned} \tag{2}$$

$$New\_x_{Max} = x_{Max} + \left(\left((x_{Max} - x_{Min}) * 0.3\right) + 5000\right)$$
$$New\_y_{Max} = y_{Max} + \left(\left((y_{Max} - y_{Min}) * 0.3\right) + 5000\right)$$

The coordinates processed during this step had the Web Mercator projection (EPSG: 3857) as spatial reference, where units were in metres. Therefore, the *5000* constant in equation (2) regulated the minimum allowed width and height of the bounding box to above 2 * 5000 metres. On the other hand, when the two *start* and *end* coordinates were separated by several tens of kilometres, the *5000* metre constants had a small impact on the widening of the bounding box. Thus, a 30% (equation 2) increase of the width and height made a substantial change on a large minimum bounding rectangle, however on a small, the change was insignificant. Thus, these two constants regulated the added margins dynamically based on whether the minimum bounding rectangle was small or large.

Further, the rectangle was squared (Figure 8c) to cancel out the risk of the one-dimensional bounding box explained above, which was performed by adding the difference between the height and width to the smaller one of the two (Figure 8d).

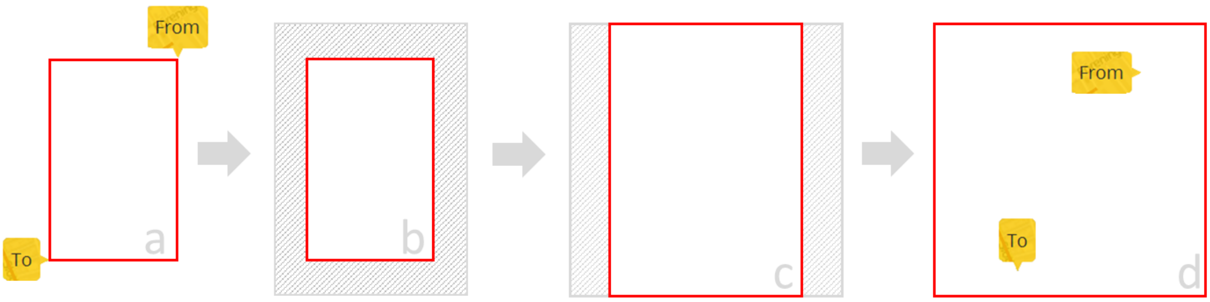

Figure 8: Graphical representation of the dynamic bounding box calculations. "To" and "From" represents the point-coordinates for routing between, used as input to the dynamic bounding box.

3.4.2 GeoServer

An SQL view was created in GeoServer to receive the information (*start-coordinate*, *end-coordinate*, *bounding box-coordinates*, *country-code*, *flooded areas*) as parameters and pass it forward to a developed PostgreSQL function. A *validation regular expression* was defined for each parameter, where only decimal numbers were allowed for coordinates and only alphabet characters for text variables (*country code*, *flooded areas*).

3.4.3 Database processing

All processing within the database was gathered in one PL/pgSQL-function developed and installed into the PostgreSQL database. It was designed to accept the parameters (*start-coordinate*, *end-coordinate*, *bounding box-coordinates*, *country-code*, *flooded areas*) as input arguments, and the calculated shortest path geometry as output.

In the initial stage of the function, edges and vertices with the country-specific prefix matching the *country-code*, were located. Further, the edges and vertices within the *bounding box* were selected and put into temporal tables, only saved throughout the current call of the function. Edges and vertices were further manipulated by setting the *flooded* column to *True* for each edge and vertex intersecting with the *flooded areas* polygons. However, if an edge was defined as a bridge, the flooded column would stay *False*. This made bridges de facto non submersible within this application, which was not true in a real flood scenario. However, since the elevation of bridges rarely falls below its surrounding street network, a bridge is not flooded until its connected roads are. Thus, if the road segments leading up to a bridge were flooded, the bridge was indirectly flooded since it was no longer routable.

Exception of bridges was implemented to ensure that the application would not be sensitive to what kind of water bodies were included in *flooded areas*. If, for example, *flooded areas* also included water bodies under normal conditions, such as rivers or canals, the bridges passing over should not be defined as flooded.

Network analysis had to be performed between vertices within the imported street network. Since the *start* and *end* coordinates could be at any location, the closest network vertex needed to be located. By calculating the Euclidean distance between each vertex and the coordinate (*start*, *end*), the vertex with the smallest distance was selected as the street-network-equal to the input coordinates.

The Dijkstra's algorithm was applied for network analysis since it has shown to be the most efficient in one-to-one routeing in average sized road networks (Zhan and Noon 1998). The algorithm was assigned to perform a network analysis between the two calculated vertices within the temporal street network. Further, the vertices and edges defined as *flooded* were not included in the analysis. Traversal cost for each edge was set to the multiplied product of the edge length and road class weight (Table 1).

The set of continuous edges forming the shortest path between the input coordinates was hence returned back to the GeoServer SQL view as a PostGIS geometry.

3.5 Fluvial flood analysis for Web-GIS

The purpose of this section was to develop a method for fast and simple modelling of water-level increase of a watercourse. As with the other work of this thesis, the intension was to implement this method into the WGRAS system architecture through an intuitive and accessible tool. However, prior to this implementation, little work had been done on raster processing within the WGRAS framework. Hence, the structure of the work dedicated to this part of this thesis was divided into two parts:

- Design method for raster image processing within the WGRAS system architecture.
- Development of a method for fluvial flood analysis where only a digital elevation model was used as input data.

3.5.1 WGRAS raster image processing

A substantial amount of time was dedicated to finding a solution to store and process raster data. Since this practise was by its nature very computational intensive, a good method was necessary for legitimizing an implementation in a Web-GIS. Here, processing time was of absolute importance, where tools demanding more than half a minute would probably confuse the user whether the tool was functioning or not.

In the WGRAS system architecture there were several possible solutions for storing and processing raster data. One solution was to store the data in the PostgreSQL database as a PostGIS raster layer and perform the cell processing using PostGIS raster functions. However, this proved to be difficult since the support for PostGIS raster layers in GeoServer was underdeveloped, meaning the connection to the map interface was not possible. A second solution was to store the raster as a file in the GeoServer file depository and perform the processing as a Web Processing Service (WPS), by developing customized raster functions within the GeoServer framework. A good solution in terms of efficiency since storage and processing was performed within the same instance, GeoServer. However, the web server was developed in Python and therefore convenient for further customization. Also, rather than extending the functionality of GeoServer using Java, the functionality of the WGRAS application was gathered under the same instance, using the same programming language. This led to the third and chosen solution described below.

From the GeoServer file repository, the raster GEOTIFF could be displayed on the WGRAS map interface through WMS and imported through Web Coverage Service (WCS) to the web server for processing. WCS also enabled subsetting of the raster image by defining a bounding box of the area of interest, making it more time efficient in retrieving the raster image from GeoServer. The imported raster was temporally saved during processing in the web server filesystem.

For raster calculations in the web server, the python version of the open source library GDAL (GDAL Development Team 2015) was used. Through GDAL, it was possible to perform raster calculations and statistics on the WCS-imported raster image(s). If the result from calculations was in vector format, it was automatically added to the PostgreSQL database as a PostGIS layer. If it was a raster, the image was added back to the GeoServer file-repository.

3.5.2 Modelling and implementation

The method of this section was developed with concerns of reasonable implementation into a Web-GIS system. Therefore, it was decided to keep the amount of input data and complexity as low as possible. As described earlier, the only data used was a digital elevation model with a resolution of 2 metres, where each cell represented the average elevation of its 4 m^2 area. Since it is impossible to determine, based on this information, whether a cell is a river or not, some data was specified by the user.

3.5.2.1 Flood-fill algorithm for flood analysis

The flood-fill algorithm was applied for flat-surface modelling by, instead of colours, calculating the floating-point elevation values of a DEM. Thus, the boundary which the search window was enclosed within was instead defined as a certain elevation threshold. For example, if the starting point for the algorithm was set at the water surface of a lake, the spatial boundary was defined as the elevation of the starting point. In this case, the search window only visited cells lower or equal to the elevation of the water surface, i.e. the boundary for the algorithm marked the edge of the lake. Further, if a flood inundation analysis was performed for the lake, the threshold value could simply be altered with a water-level increase as in equation (3),

$$Threshold\ value = E + WI \quad (3)$$

where *E* was the starting point cell elevation and *WI* was the water-level increase. Thus, the boundary of the algorithm was enlarged covering not only the water surface, but also the surrounding landscape below and equal to the threshold elevation.

3.5.2.2 Staircase flood-fill

When the flat-surface modelling was applied, through the flood-fill algorithm, on rivers and streams, a problem occurred. Since a watercourse always had an inclination in its flow direction, the above implementation of the flood-fill algorithm make the flooding downstream exaggerated (Figure 9a). This problem was caused by the calculation of the threshold elevation forming the boundary of the flood-fill algorithm. When the threshold elevation was calculated from a single upstream water-elevation (equation 3), the relative water-level increase downstream was equal to the threshold elevation plus the elevation drop along the watercourse. Therefore, a method performing flood-fill as a staircase operation, was developed.

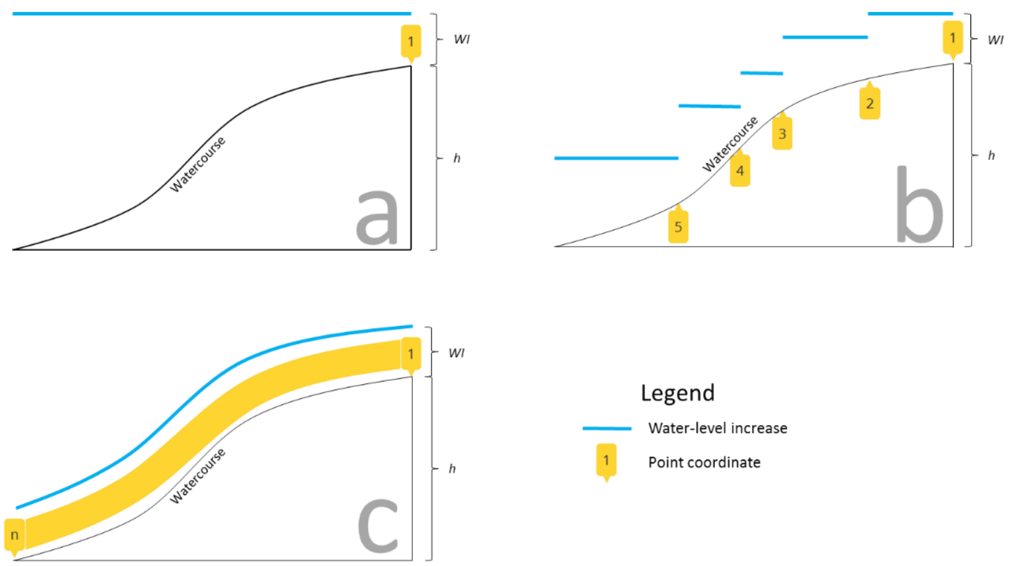

Figure 9: Describes difference in results between standard flood fill (a) and staircase flood-fill (b, c). Performed with a water-level increase WI, for a river with elevation drop h.

By calculating the water-level increase stepwise, the threshold elevation was updated at every point-coordinate before the algorithm proceeded downstream (Figure 9b). The steps for this procedure are listed below:

1. Find cell coordinate for every point coordinate.
2. Choose the first point upstream.
3. Calculate threshold elevation as in equation (3) for current point.
4. Assign boundary at next point perpendicular to the flow direction (Figure 10).
5. Execute flood-fill algorithm for the threshold elevation. Visited cells and next-point-boundary cells are not visited.
6. Remove next-point-boundary.
7. Choose the next point downstream.
8. Repeat 3-8 until all points are processed.

The river was treated as a set of lakes varying in size based on the number of points per length unit. Therefore, the quality of the flood analysis was based on number of points per metre in elevation drop (Figure 9c).

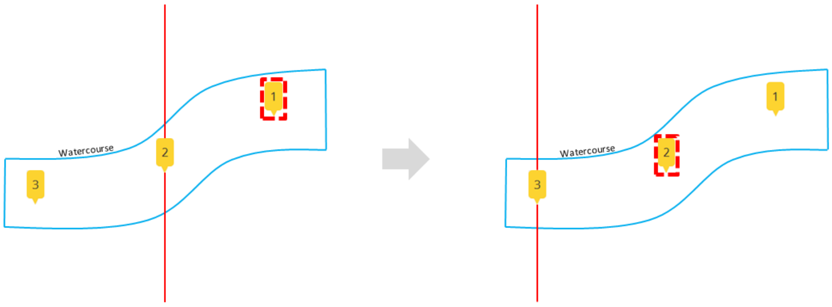

Figure 10: The watercourse seen from above with three point-coordinates assigned for modelling. Point marker with a dashed red frame was the current point and the next-point boundary (red line) was set perpendicular to the flow direction at the following point-coordinate.

3.5.2.3 Client

A user interface was developed for flood analysis within the client map application. A user was able to draw a bounding box of study area and identify the watercourse segment for analysis. Identification was based on coordinate point values clicked by the user in the centre of the watercourse, starting upstream. Further, the water-level increase, compared to normal conditions, was set in metres by the user. The data was hereafter sent through an AJAX-request to the server for processing.

3.5.2.4 Server processing

At the server, the processing started with requesting and saving the chosen DEM from GeoServer. For this WCS-request, the user-drawn bounding box was used for selecting the area of interest from the larger DEM stored in GeoServer. Further, the staircase flood-fill algorithm was executed with the point-coordinates and water-level increase, starting with the most upstream point. The resulting polygon of flooded areas was saved to the PostgreSQL database and published to GeoServer with a user-specified name.

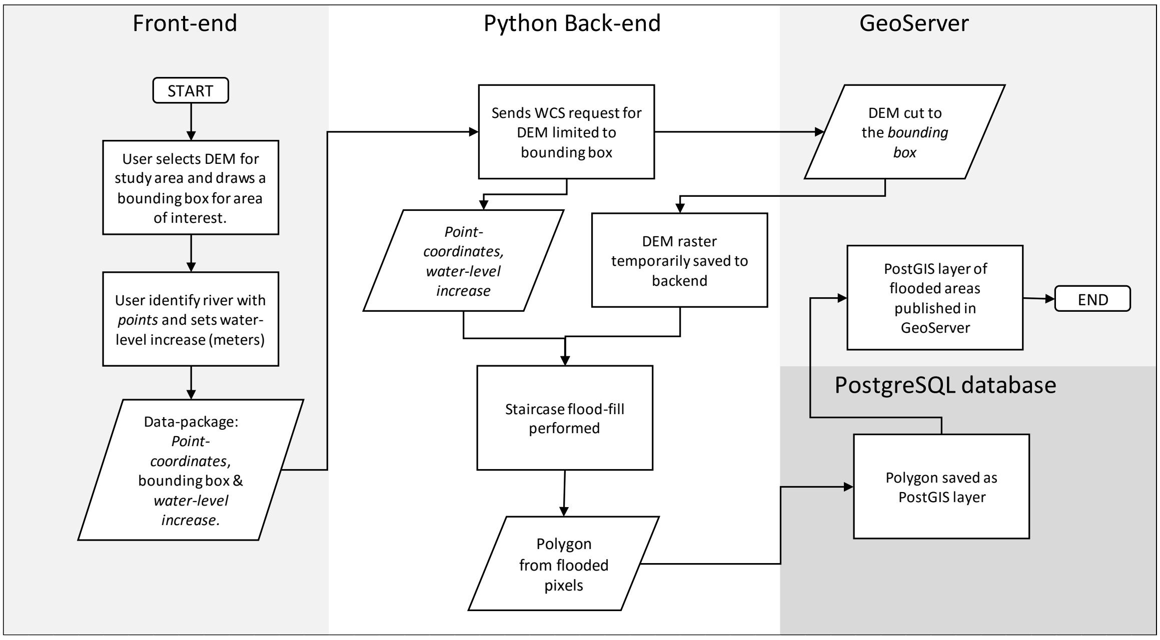

Figure 11: Flow chart of the Flood analysis through the WGRAS system architecture.

3.6 Performance assessment

A set of tests were performed to analyse the performance of the two developed methods and their corresponding applications. Focus in this section was both to assess the precision of the modelled results as well as evaluating the performance in terms of executional time.

3.6.1 Dynamic street network analysis

The dynamic street network analysis performance, in terms of time, was empirically tested using four different layers of 1, 10, 100 and 1000 circular test-polygons. Test-polygons were randomly distributed within the study area and were assigned a radius of 50 metres. Ten routings between the same start- and end-coordinate were performed for each test-polygon layer used as flooded areas. Average time from starting the routeing, i.e. sending the data to the GeoServer SQL-view, to a visible shortest-path in the map interface, was measured using the built-in network analyst in Firefox version 46.0.1, measuring response time of a server-request.

3.6.2 Flood analysis

In this section two different performance assessments were performed. The first measuring the elevation drop fault of the algorithm, and the second measuring the performance in terms of time complexity.

3.6.2.1 Measuring elevation drop fault

For measuring the fault due to elevation drop (Figure 9b) with the staircase flood-fill algorithm, the resulting polygon was analysed using zonal statistics in QGIS and a set of 14 randomly distributed test-polygons. These polygons were rectangular in shape and covered the width of the resulting flood polygon as a transversal to the flow direction. Further, the maximum and minimum elevation was derived using zonal statistics of the DEM-cells covered by the test-polygons. Since the elevation of the watercourse-cells were the lowest within each polygon, the highest relative elevation classified by the algorithm as flooded could be calculated (equation 4),

$$E_{Diff} = E_{Max} - E_{Min} \tag{4}$$

where E_{Diff} was the highest elevation difference within each test-polygon. Hence, if the elevation difference was higher than the input water-level increase, the result from the staircase flood-fill was more similar to a set of small lakes (Figure 9b) rather than a continuous flooded watercourse (Figure 9c). This was performed to study the optimal number of point-coordinates per metre of elevation drop.

3.6.2.2 Empirical testing of time complexity

A time complexity assessment was performed empirically for the developed staircase flood-fill algorithm. The algorithm was executed 21 times for water-level increases between 0 - 4 metres with incremental steps of 0.2 metres. Number of cells processed and execution time was measured for each water-level increase. Even though the absolute execution time was irrelevant, the relative processing time of different water-level increases was evaluated. Since different water-level increases sets the number of cells processed in the DEM, this testing evaluated the time complexity. Testing was performed on a Windows 10 64-bit operating system with a 2.8 GHz Intel i5-6200U processor with two cores and four threads, 8 GB of 1866 Hz RAM and an SSD hard drive.

4. Results

The result from the two methods studied and developed in this thesis, will be presented in this section. Since the methods were implemented as tools in the WGRAS system architecture, the results were produced through this application by using a virtual server on a Windows machine.

4.1 Dynamic street network analysis

An example of the dynamic street network analysis tool can be seen in Figure 12, where a routeing was performed between two street addresses in the study area. The polygons used as *flooded areas* represent a 2.2-metre water-level increase of the central canals in Malmö, and was calculated using the flood analysis tool. For this scenario the application calculated the shortest path via the top-left bridge (Figure 13), which was not defined as flooded.

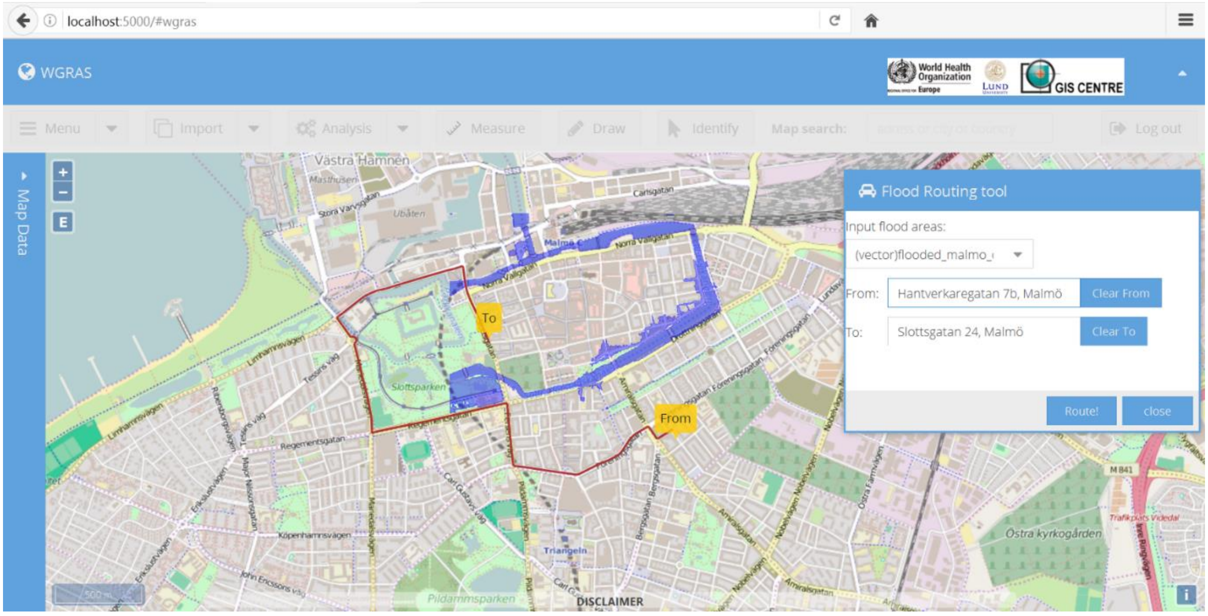

Figure 12: Results in the WGRAS map interface when routing between Hantverkaregatan 7b and Slottsgatan 24 in the study area. Dark blue areas represent flooded areas where the central canals have increased 2.2 metres in water-level elevation.

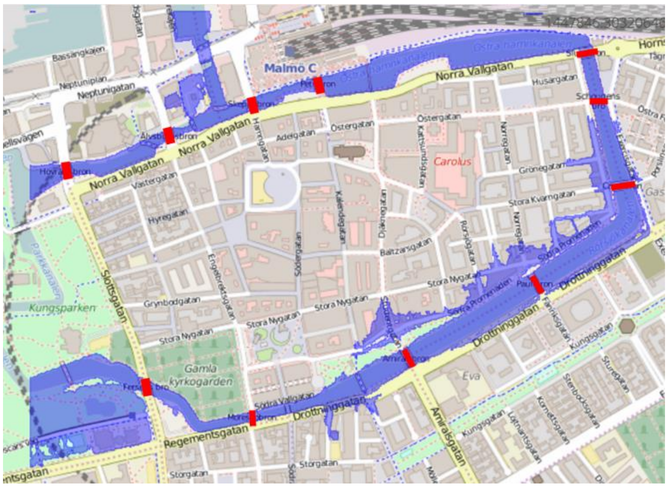

Figure 13: Zoomed in version of the flooded areas used for dynamic street network analysis in Figure 12. Red lines marks bridges leading over the flooded canals.

4.1.1 Dynamic bounding box

The dynamic bounding box was calculated every time two point coordinates were set for the dynamic street network analysis. When two point coordinates were set within the study area of Malmö, the bounding box would always cover a significant part of the city (Figure 14). Similarly, when coordinates with a larger separation were used, the bounding box covered a relatively larger area than the minimum bounding rectangle (Figure 14).

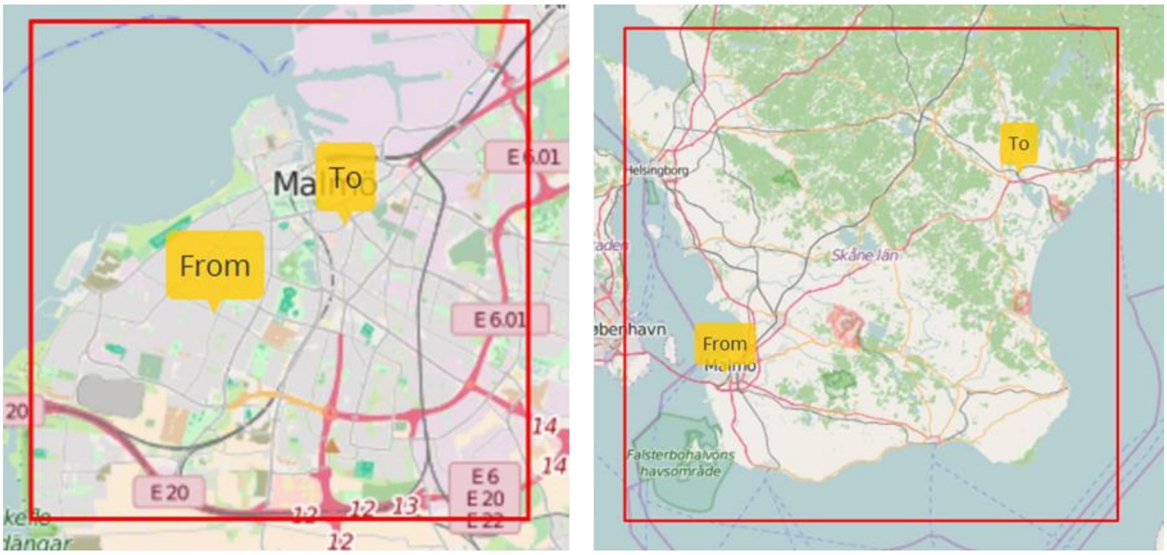

Figure 14: Two examples of the dynamic bounding box (Red box) calculated within the client of the WGRAS application. Picture to the left shows the bounding box of two points set with a separation of 5 kilometres within the study area. Figure to the right shows the bounding box of two points with a separation of 85 kilometres.

4.1.2 Performance

The results from the server response time assessment can be seen in Figure 15. Here, a function of execution time and number of *flooded features* was estimated to $y \approx 5.2x + 588$ milliseconds, where x was the number of *flooded features*. It was not possible to analyse the server response with *10,000* features since almost all roads within the study area were defined as flooded, and thus no routeing could be performed.

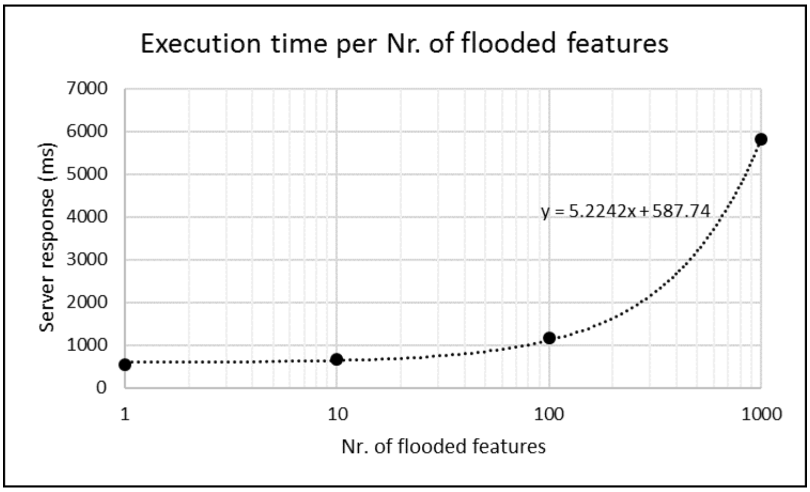

Figure 15: Chart of the server response time assessment performed for 1, 10, 100 & 1000 flooded areas. Calculated trend line (black dashed line) followed a linear pattern where y ≈ 5.2x + 588 milliseconds. The x-axis is scaled logarithmically which makes the trend line appear non-linear even though it is linear.

4.2 Fluvial flood analysis

The fluvial flood analysis was performed for a subsection of Sege å. Results were produced for the single-point Flood-fill algorithm (Figure 16) as well as the developed multi-point Staircase flood-fill (Figure 17 & Figure 18), for comparison between the two methods.

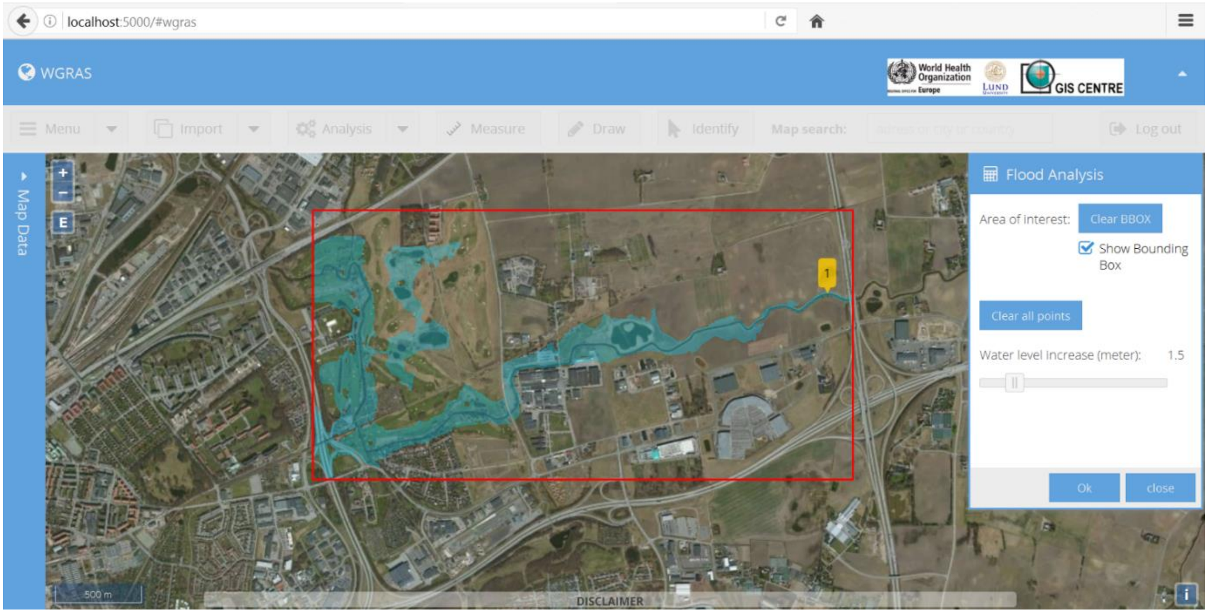

Figure 16: Fluvial flood analysis using the Flood-fill algorithm with only one point-coordinate. The turquoise polygon represents flooded areas with a water-level increase of 1.5 metres at the point-coordinate. Flow direction of the stream was from right to top-left in picture reference.

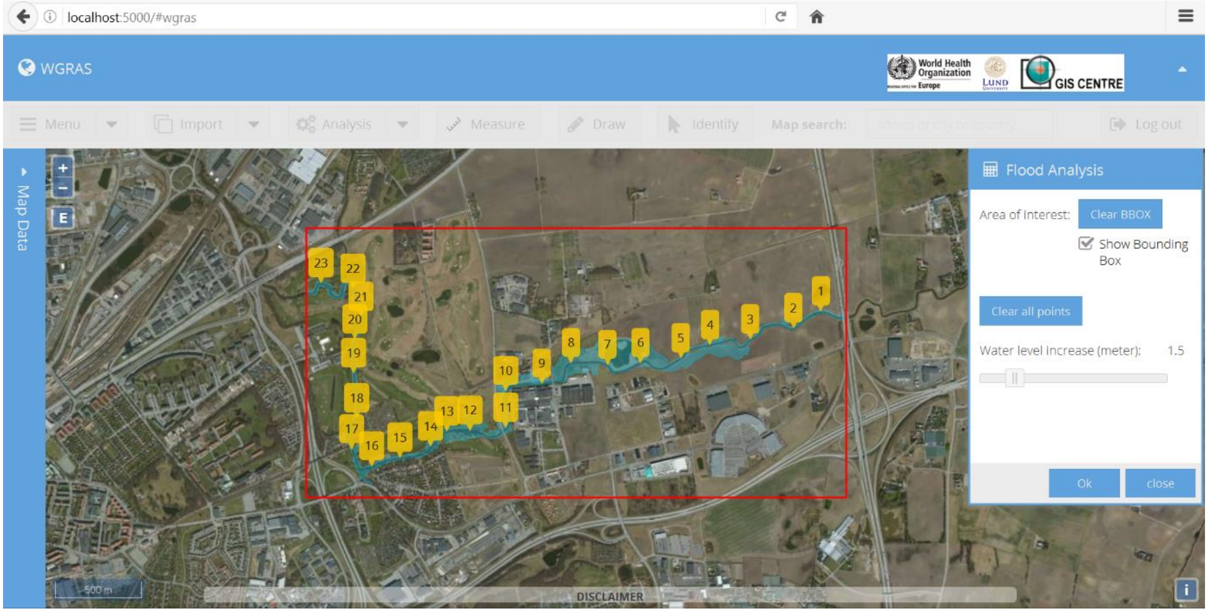

Figure 17: Fluvial flood analysis using the developed Staircase Flood-fill algorithm with 23 point-coordinates within the study area. The turquoise polygon represents flooded areas with a water-level increase of 1.5 metres at every point-coordinate. Please see Figure 18 for better view of the flooded areas without the point-markers. Flow direction of the stream was from right to top-left in picture reference.

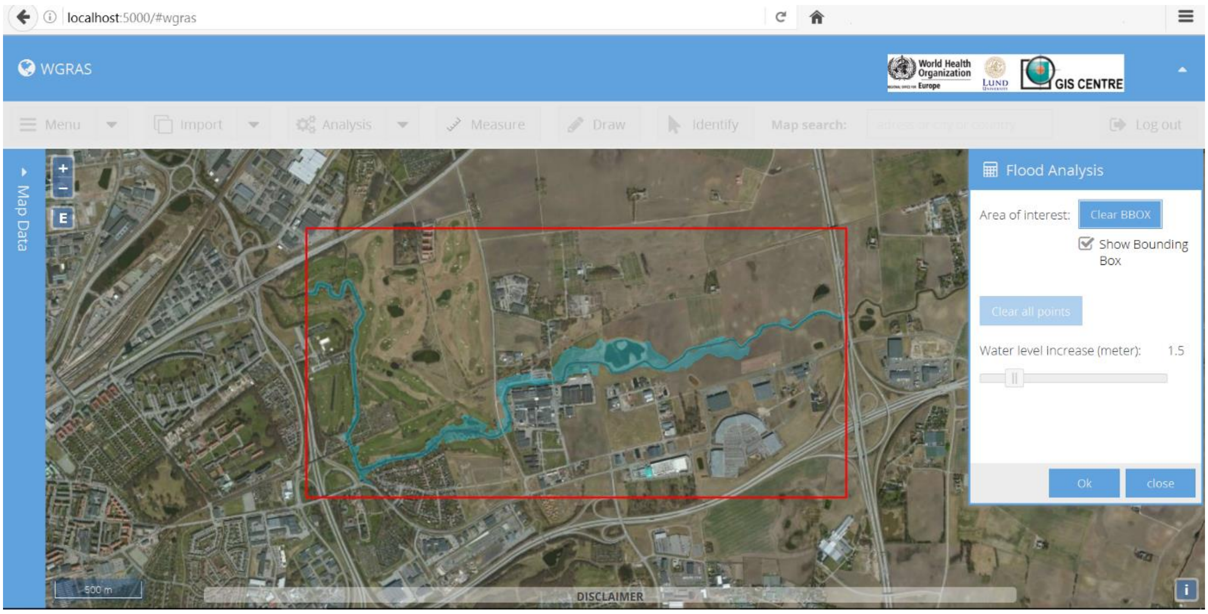

Figure 18: Same as Figure 17 without the point-markers. The turquoise polygon represents flooded areas with a water-level increase of 1.5 metres at every point-coordinate. Flow direction of the stream was from right to top-left in picture reference.

As can be seen when comparing the results in Figure 16 and Figure 18, the difference was the elevation drop fault caused by the use of only one point-coordinate. This difference was due to the total elevation drop of the watercourse being 1.49 metres from the most upstream point to the most downstream point. Hence, the flooded downstream cells derived with the flood-fill algorithm (Figure 16) incorrectly represents a 1.5 metre water-level increase, but instead corresponds to a *1.49 + 1.5 = 2.99* metre increase. With the Staircase flood-fill algorithm (Figure 17 & Figure 18), the water-level increase was instead calculated using the nearest-point-coordinate elevation and therefore more closely corresponds to a ≈1.5 metre increase. This flood fill elevation drop fault was quantitatively measured and can be seen in Figure 19 and Table 2.

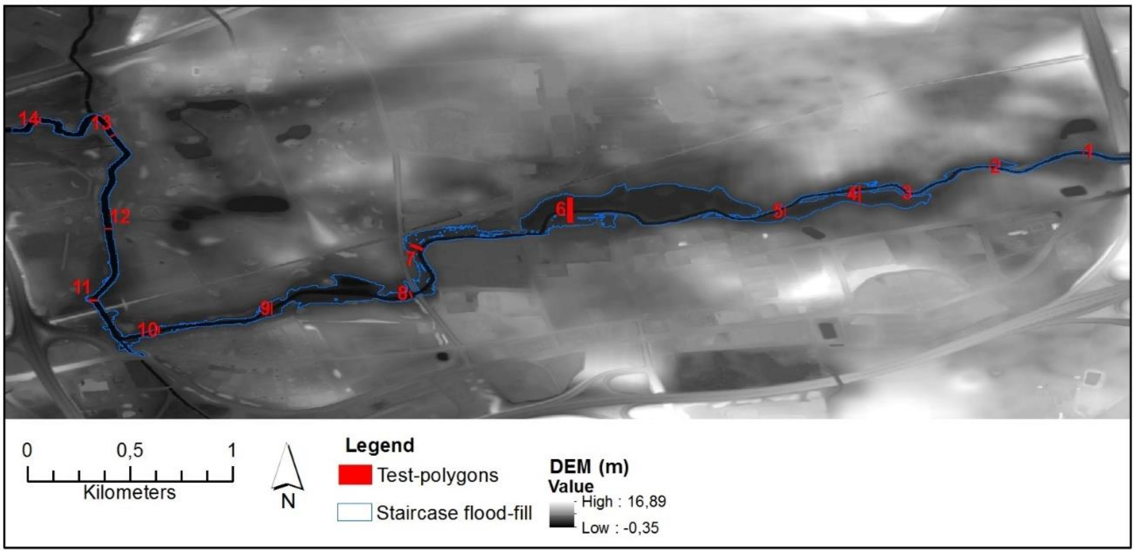

Figure 19: Map showing the distribution of test-polygons (red) for measuring the fault of the flooded areas (blue) calculated with the staircase flood-fill algorithm. Test-polygon labels correspond to Ids in Table 2. The 2-metre resolution DEM used for analysis is also visible in map.

Results hence show that there was a small fault in three of the fourteen test-polygons analysed when using 23 evenly distributed point-coordinates for the study area. For the total elevation drop of 1.49 metres, the average number of point-coordinates per metre of elevation drop is approximately 15, which gives the performance results in Table 2. The same testing was performed with a total of 40 point-coordinates, i.e. 27 point-coordinates per metre of elevation drop, which resulted in only one fault of the fourteen test-polygons.

Table 2: Elevation maximum, elevation minimum & elevation difference within each test-polygon in Figure 19. Values exceeding the input water-level increase are bolded. Elevation difference is calculated as in equation 4.

Id	E_{Min}	E_{Max}	E_{Diff}
1	1.55	2.86	1.31
2	1.46	2.93	1.47
3	1.34	2.53	1.19
4	**1.29**	**2.86**	**1.57**
5	1.15	2.63	1.48
6	**0.88**	**2.46**	**1.58**
7	0.63	2.13	1.5
8	0.56	1.77	1.21
9	0.43	1.83	1.4
10	0.26	1.74	1.48
11	**0.22**	**1.74**	**1.52**
12	0.17	0.82	0.65
13	0.1	1.14	1.04
14	0.06	1.28	1.22

4.2.1 Empirical testing of time complexity

Empirical testing was performed, for the same bounding box and point-coordinates as in Figure 17 and Figure 18. Results in Table 3 show the number of cells processed, proportion of the total raster-subset cells and execution time per water-level increase computed. Execution times were further plotted against number of computed cells in Figure 20, where a linear trend was calculated. The time complexity was thus empirically derived as increasing linearly with size of the input, i.e. O(n).

Table 3: Empirical testing of the performance in terms of time of the Staircase flood-fill algorithm. Results show number of cells, proportion of total raster cells and execution time for different water-level increases (WI).

WI (m)	Nr. of Pixels *n*	Proportion of raster	Execution time (s)
0	1 073	0.1%	0.35
0.2	13 650	1.4%	0.78
0.4	15 544	1.5%	0.85
0.6	18 180	1.8%	0.89
0.8	22 263	2.2%	1.12
1	33 609	3.3%	1.40
1.2	40 356	4.0%	1.78
1.4	52 484	5.2%	2.12
1.6	63 691	6.3%	2.59
1.8	78 872	7.8%	3.10
2	93 781	9.3%	3.58
2.2	121 439	12.1%	4.69
2.4	146 705	14.6%	5.42
2.6	169 305	16.8%	6.12
2.8	208 579	20.7%	7.47
3	247 766	24.6%	8.84
3.2	294 943	29.3%	10.45
3.4	328 573	32.6%	11.35
3.6	420 221	41.7%	14.41
3.8	449 224	44.6%	15.41
4	475 099	47.2%	16.12

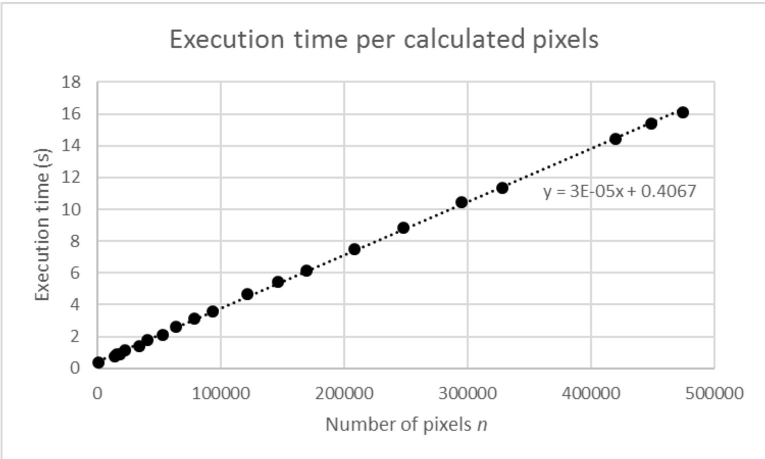

Figure 20: Chart of the execution time in relation to the calculated cells for the staircase flood-fill algorithm. Calculated trend line (black dashed line) followed a linear pattern of $y \approx 3*10^{-5}x + 0.41$.

5. Discussion

The results from the two developed methods and corresponding performance assessments will here be further discussed. Improvements of both methods and alternative solutions to the ones presented in the methods section of this thesis will be highlighted. Finally, the findings of the work of this thesis will be put into context with current work within disaster management.

5.1 Dynamic street network analysis

The results generated by the developed dynamic street network analysis performed as intended, where areas defined as flooded were avoided at every route request. However, some additional features could be implemented to the developed method as well as to the web-application applying the method.

The solution presented in this thesis where bridges are handled as non-submersible by a flood is not a comprehensive solution. It follows the assumption that a bridge is always flooded later than its neighbouring street network. Probably true in most cases, however a more accurate solution would be to analyse the bridge elevation and thus decide whether its submerged or not. This could be done by analysing a digital elevation model and assign these values to the OSM-features defined as bridges. However, this demands a larger amount of database management as well as the *flooded areas* polygons having a water-level attribute. As previously stated, a key quality of an online route application is performance in terms of time, where mentioned improvement can have a negative impact. For finding a suitable solution for Web-GIS implementation as well as a comprehensive solution to flooded bridges, the method presented in this thesis offers a good compromise between the two.

For the implementation of this method into a Web-GIS application as a tool, such as in WGRAS, an alternative selection of flooded roads would be beneficial. As for now, the method only uses polygons for setting the *flooded areas*. Another additional solution should be studied where separate road segments can be selected within the Web-GIS application and set to flooded. This would not limit the dynamic street network analysis to only process measured or modelled spatial data, but also offer a more user-inclusive tool. During a flood disaster event, if flooded roads are reported as segments of street names instead as a flooded extent, the user can more easily manipulate the street network. In the current method presented in this thesis, this action could be performed by drawing a small polygon on the flooded road segment, however the proposed addition is more intuitive.

The performance assessment of the developed method indicated on a linear relationship between an increasing number of equally shaped flooded polygons and server response time. However, since it was not possible to perform an assessment with further polygons, it can only be speculated whether the linear relationship will persist, increase into an exponential relationship or decrease into a logarithmic relationship. It is complex to theoretically assess where within the database processing the increased number of *flooded* features affects the time performance. However, the time performance of the methods used to intersect the *flooded* polygons with the street network, are significantly affected by the complexity of the polygons. The circular polygons with a 50 metres radius used in the performance assessment were built-up by 50 coordinates points. If instead, for example, rectangular polygons created with five coordinates were used, the complexity of each intersection would be lower. Hence, the performance assessment of the developed dynamic street network analysis is rather complex, and thus might tests for an above normal condition.

5.2 Fluvial Flood analysis

The results from the fluvial flood analysis differ in applicability from the dynamic street network analysis. While the network analysis returns a finished result, the flood analysis is more of a component for additional disaster risk analysis and vulnerability. As demonstrated in Figure 12, the fluvial flood analysis can be used to analyse the effects on the street network in relationship with the network analysis. Depending on other inputs used, the results from the flood modelling can describe impact on buildings, evacuation zones and other vital infrastructure.

A set of improvements to the staircase flood-fill algorithm have been formulated after the first version of the application was finalized. These improvements presented below aim at finding more robust solutions while still preserving the intended simplicity and performance, suitable for implementation in a Web-GIS.

5.2.1 Deriving staircase-points from polyline

As for now, the staircase flood implementation in WGRAS relies on the user's knowledge of the area intended for flood analysis. Since the quality of the resulting polygon depends on the number of points per metre of elevation drop, the user has to adjust the number of points-coordinates assigned in perspective to the elevation drop of the watercourse. Henceforth, the method could instead use a polyline as input, which is also the more conventional representation of a watercourse in GIS-science. The method could therefore be adjusted to the following operation, where staircase points are derived from a polyline:

1. Cut the polyline according to the input bounding box extent.
2. Check the difference in elevation between starting- and ending point of the polyline and thus calculate the total elevation drop of the watercourse section.
3. Assign evenly distributed staircase point-coordinates along the entire polyline corresponding to 27 point-coordinates per metre of elevation drop.

This improvement would avoid the problem of too few points used as input which lowers the quality of the result, as well as too many, making the calculations time consuming. Further, the use of polylines as input data would enable the user to import already digitized spatial layers of watercourses for flood analysis.

5.2.2 Nearest neighbour

Yet another improvement could be developed to increase the precision of the staircase flood-fill algorithm. As for in this version of the application, a next-point boundary, limiting the progress of the search-window downstream of the current point-coordinate, is defined at the location of the following point. However, the result would be more accurate if the calculations for each cell was based on the threshold elevation of the nearest neighbouring point-coordinate. A possible method is listed below:

1. For each analysed cell, find the nearest neighbouring point-coordinate.
2. Retrieve the threshold elevation for this point-coordinate calculated with equation (3).
3. If the elevation of the cell is below the threshold elevation, define as flooded.

However, this demands an efficient calculation of the nearest neighbouring point-coordinate for each analysed cell, otherwise the time complexity of the algorithm will be negatively affected. If, for example, the all-nearest-neighbours algorithm is utilized, the time complexity of the flood-flood fill algorithm, O(n), would be multiplied with O(log v), where v is the number of point-coordinates used (Vaidya 1989). Hence, the total time complexity of the staircase flood-fill algorithm would be O(n*log v), which would have a negative impact on performance.

If the implementation of the nearest neighbour algorithm has a large negative effect on the performance of the staircase flood-fill algorithm, the more naïve next-point boundary method could still be used with a small improvement. Instead of defining the next-point boundary at the exact location of the following point, the boundary could be set at the midway between the current and the following point-coordinate (Figure 21). However, as mentioned, this is a naïve interpretation of the nearest neighbour method, where some cells will be classified with the wrong nearest point-coordinate. Nonetheless, it might still be a better method for keeping a relevant computational time for Web-GIS implementation.

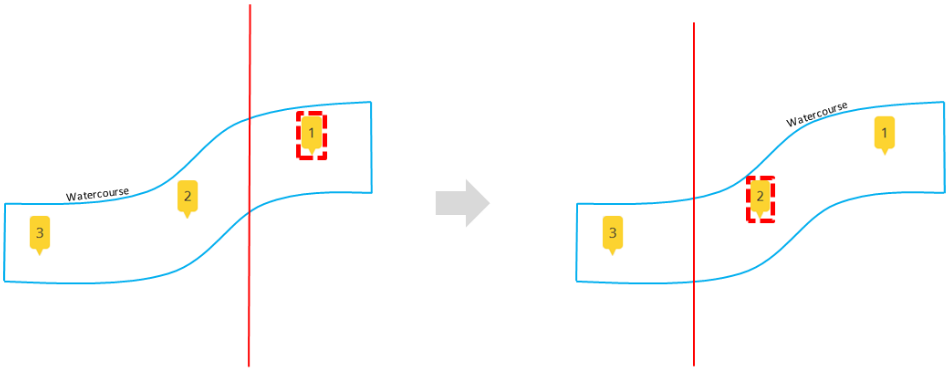

Figure 21: An improved version of the next-point boundary method shown in Figure 10. Again, the watercourse seen from above with three point-coordinates assigned for modelling. Point marker with a dashed red frame is the current point and the next-point boundary (red line) is set perpendicular to the flow direction in-between the current and the following point.

5.2.3 Water-levels and openness of the LIDAR data

The raster image derived from LIDAR sensor data is recorded at a specific date. This is usually not a problem when handling elevation data of a landscape. However, the water-levels of a watercourse are very dependent on season. Thus the water-levels measured during a LIDAR data acquisition are not necessarily a correct representation of the average all year round water-levels. The elevation data was measured the 7th April 2010, where the water flow (m^3/s) of Sege å was doubled compared to the annual average during 2001-2015 (SMHI Vattenweb 2016). How this effects the water-surface elevation of the watercourse must be calculated using the discharge rating curves of Sege å. However, this analysis exceeds the scope of this thesis and is therefore excluded. Although, for further development of the flood analysis method, the water-level deviation of the acquisition date could be specified and taken into consideration.

A main requirement for the work of this thesis is to develop methods using open-source data. This however is only partly met with the fluvial flood analysis method. The data used to test and demonstrate is currently not accessible for everyone, however the method is of course still usable for any type of elevation raster data. Around the world, initiatives concerning open data from the public sector are currently being formulated and taken into action (European Parliament 2013). This will have a large impact on acquisition of high-resolution elevation data, where the nation-wide coverage of the elevation data used for this method, will soon be open to everyone (Lantmäteriet 2013).

5.3 Relevance of developed methods

As reviewed in the background chapter, the current work with flood disaster in Sweden and within the study area is focused on dimensioning infrastructure to probable future rain scenarios. However, when a heavy rainfall hit Copenhagen in 2011, the Swedish Civil Contingencies Agency (MSB) discussed the consequences of a similar rain in Sweden and concluded the following:

"The rain volume [in Copenhagen] *corresponds to a return period of more than 1500 years when compared to Swedish precipitation statistics. It is thus not reasonable to use this event for the dimensioning of our urban drainage systems; however, it teaches us that a similar rain is possible."* (Hernebring and Mårtensson 2013, translated by author).

The cloudburst causing the flood event of 2014 in Malmö was of similar strength, nonetheless the above conclusion is true. Preparing the urban planning and physical infrastructure to withstand even the most extreme precipitation events, is probably not possible. Thus, methods aimed at flood disaster response have to pick up where the societal resilience from disaster risk reduction is under dimensioned. When looking at a global perspective, a small part of the world has the resources and capability to construct flood disaster risk reduction instruments to even withstand annual extreme precipitation events. Thus, floods will recur and methods for handling disaster response is essential to develop together with disaster risk reduction.

Remaining in the global perspective, by designing the dynamic street network analysis to work with OpenStreetMap data, the method is not only applicable all over the world, but especially suitable during disaster response. The volunteered acquisition of OSM-data during major disasters has, besides proven the most effective, also digitized the spatial extent of the disaster damage. Thus, the dynamic street network analysis can be directly linked and updated with the work performed in the VGI community. While the work of this thesis primarily focus on flood disasters, there is practically no changes required to apply the dynamic street network analysis on other types of disasters such as earthquakes and landslides.

6. Conclusions

This thesis presents two methods to support work within disaster analysis and response. The methods are customized for Web-GIS where accessibility and time performance are two important factors. Furthermore, the methods are developed with a non-GIS-expert user group in mind. The intended user group is instead interested of implementing spatial analysis in their respective work with disaster risk management, reduction and response. Further, methods are consistently developed using open-source data and software with the intention of accessibility and global applicability.

The first of the two developed methods concerns calculating the shortest path in an obstructed road network. A central concept for finding a safe and correct route during a flood event as well as assessing the vulnerability of an urban area during the pre-disaster analytical work. The developed dynamic street network analysis method is further demonstrated through implementation in a Web-GIS for disaster management. It is also tested regarding time-performance and proven suitable for use in a web environment. Pros and cons of possible improvements to the method are also discussed, primary regarding the solution for flooded bridges.

The second of the two developed methods concerns modelling of a fluvial flood scenario where results are intended for further vulnerability assessments on vital infrastructure. The method is based on sequenced flat-surface flood modelling for calculating a water-level increase of a sloping watercourse. As stated in the aim, the fluvial flood method is developed with a significant focus on time and accuracy performance of the final product. Just like the dynamic street network analysis method, the fluvial flood analysis method is demonstrated through implementation into a Web-GIS platform and properly tested regarding time and accuracy performance. Further, possible improvements to the developed method are discussed regarding the digitization of the analysed watercourse and placement of the staircase boundaries. Issues concerning the water-levels of the utilized digital elevation model are analysed and discussed, concluding that flow intensity deviations should be considered.

From the perspective of local and international disaster risk management, the complementary relevance of the developed methods is thoroughly evaluated. Floods and other disasters will continue to occur even though major development is performed for disaster risk reduction. This thesis describes two methods to support this development as well as methods to respond to an on-going disaster.

References

Apel, H., O. Martínez Trepat, N. N. Hung, D. T. Chinh, B. Merz, and N. V. Dung. 2016. Combined fluvial and pluvial urban flood hazard analysis: concept development and application to Can Tho city, Mekong Delta, Vietnam. *Nat. Hazards Earth Syst. Sci.*, 16**:** 941-961. DOI: 10.5194/nhess-16-941-2016

Aye, Z. C., M. Jaboyedoff, M. H. Derron, C. J. van Westen, H. Y. Hussin, R. L. Ciurean, S. Frigerio, and A. Pasuto. 2016. An interactive web-GIS tool for risk analysis: a case study in the Fella River basin, Italy. *Natural Hazards and Earth System Sciences*, 16**:** 85-101. DOI: 10.5194/nhess-16-85-2016

Bates, P. D., M. S. Horritt, and T. J. Fewtrell. 2010. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, 387**:** 33-45. DOI: *http://dx.doi.org/10.1016/j.jhydrol.2010.03.027*

Bellman, R. 1958. On a Routing Problem. *Quarterly of Applied Mathematics*, 16**:** 87-90.

Burgess, J., A. Pavlenko, M. Van Oosterhout, S. Hoffmann, K. Krueger, F. Ramm, B. Quinion, M. Amos, et al. 2014. osm2pgsql (Version 0.85) [Software]. Available from: *https://github.com/openstreetmap/osm2pgsql*.

Caldwell, D. R. 2005. Unlocking the Mysteries of the Bounding Box. *ALA Map and Geography Round Table*, Series A, No. 2.

Chen, J., A. A. Hill, and L. D. Urbano. 2009. A GIS-based model for urban flood inundation. *Journal of Hydrology*, 373**:** 184-192. DOI: 10.1016/j.jhydrol.2009.04.021

Daniel Wendt. 2016. OSM2pgRouting (Version 2.1.0) [Software]. Available from: *http://pgrouting.org/docs/tools/osm2pgrouting.html#*.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1**:** 269-271. DOI: 10.1007/bf01386390

European Parliament. 2007. Directive 2007/60/EC of the European Parliament and of the Council of 23 October 2007 on the assessment and management of flood risks. *Official Journal of the European Union*, 288**:** 27-34.

European Parliament. 2013. Directive 2013/37/EU of the European Parliament and the council of 26 June 2013 amending Directive 2003/98/EC on the re-use of public sector information. *Official Journal of the European Union*, 175**:** 1-8.

Evans, J., and E. Minieka. 1992. *Optimization Algorithms for Networks and Graphs, Second Edition*. New York: Taylor & Francis.

ExtJs. 2016. Sencha ExtJs API (version 6) [software]. Retrieved 25-02 2016, from. *https://www.sencha.com/products/extjs/#overview*

Fewtrell, T. J., A. Duncan, C. C. Sampson, J. C. Neal, and P. D. Bates. 2011. Benchmarking urban flood models of varying complexity and scale using high resolution terrestrial LiDAR data. *Physics and Chemistry of the Earth*, 36**:** 281-291. DOI: 10.1016/j.pce.2010.12.011

GDAL Development Team. 2015. GDAL - Geospatial Data Abstraction Library (version 1.11.3) [software]. Retrieved 28-03 2016, from. *http://www.gdal.org/*

GeoServer. 2015. Geoserver User Manual (Version 2.8.0) [Software]. Available from: *http://docs.geoserver.org/stable/en/user/index.html*.

Goodchild, M. F. 2007. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69**:** 211-221. DOI: 10.1007/s10708-007-9111-y

Hart, P. E., N. J. Nilsson, and B. Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4**:** 100-107. DOI: 10.1109/TSSC.1968.300136

Hernebring, C., and E. Mårtensson. 2013. Pluviala översvämningar: Konsekvenser vid skyfall över tätorter, En kunskapsöversikt. ed. MSB. Göteborg and Lund, Sweden.

IPCC. 2014. Climate change 2014: Impacts, Adaptation and Vulnerability. In *IPCC Working Group II Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press.

Lantmäteriet. 2013. Lantmäteriet gör sig redo för öppna data. Retrieved 05/17 2016, from. *https://www.lantmateriet.se/sv/Om-Lantmateriet/Om-oss/Tidningen-Granssnittet/Tidigare-nummer/Granssnittet-22013/Lantmateriet-gor-sig-redo-for-oppna-data/*

Lantmäteriet. 2015. Product description: GSD-Elevation data, GRID 2+. ed. Lantmäteriet. *https://www.lantmateriet.se/sv/Kartor-och-geografisk-information/Hojddata/GSD-Hojddata-grid-2/*.

Lin, W. 2016. OpenStreetMap in GIScience: experiences, research and applications. *International Journal of Geographical Information Science,* 30**:** 823-824. DOI: 10.1080/13658816.2015.1077965

LUGISC. 2016. WGRAS. Retrieved 05/24 2016, from. *http://www.gis.lu.se*

Malmö stad. 2016. PLAN FÖR MALMÖS VATTEN: Tematiskt tillägg till Översiktsplan för Malmö. ed. Malmö municipality. Malmö.

Mansourian, A., A. Rajabifard, M. J. V. Zoej, and I. Williamson. 2006. Using SDI and web-based system to facilitate disaster management. *Computers & Geosciences,* 32**:** 303-315. DOI: 10.1016/j.cageo.2005.06.017

Menne, B., and V. Murray. 2013. Floods in the European region: Health effects and their prevention. ed. World Health Organisation: Regional office for Europe. Copenhagen.

MSB. 2011. Identifiering av områden med betydande översvämningsrisk. ed. Myndigheten för samhällsskydd och beredskaps författningssamling. Stockholm, Sweden: MSBFS.

MSB. 2013. Myndigheten för samhällsskydd och beredskaps föreskrifter om länsstyrelsens planer för hantering av översvämningsrisker (riskhanteringsplaner); . ed. Myndigheten för samhällsskydd och beredskaps författningssamling. Stockholm: MSBFS.

Nayak, S., and S. Zlatanova. 2008. *Remote Sensing and GIS Technologies for Monitoring and Prediction of Disasters*. Springer Publishing Company, Incorporated.

Nosal, E. M. 2008. *Flood-fill algorithms used for passive acoustic detection and tracking*. New York: Ieee.

OGC. 2016. OGC® Standards and Supporting Documents. Retrieved 05/05 2016, from. *http://www.opengeospatial.org/standards*

OpenLayers 3. 2016. OpenLayers web mapping API (version 3.8) [software]. Retrieved 25-02 2016, from. *http://openlayers.org/*

Peggion, M., A. Bernardini, and M. Masera. 2008. Geographic Information Systems and Risk Assessment. ed. European Commission - Joint Research Centre. Luxemburg: Office for Official Publications of the European Communities.

Persson, P., T. Ehrnstèn, and G. Ewald. 2012. Handbok för klimatanpassad vattenplanering i Skåne. ed. Länsstyrelsen i Skåne län. Malmö, Sweden: Länsstyrelsen i Skåne län.

pgRouting Community. 2015. pgRouting (version 2.1.0) [Software]. Available from: *http://pgrouting.org*.

pgRouting documentation. 2016. pgRouting Manual (version 2.2). Retrieved 03/05 2016, from. *http://docs.pgrouting.org/2.2/en/doc/index.html*

Quinion, B., S. Hoffmann, F. Ramm, M. Spreng, and D. Forsi. 2016. Nominatim (version 2.5.0) [software]. Retrieved 14-03 2016, from. *http://nominatim.openstreetmap.org/*

Ratcliffe, J. H. 2001. On the accuracy of TIGER-type geocoded address data in relation to cadastral and census areal units. *International Journal of Geographical Information Science,* 15**:** 473-485. DOI: 10.1080/13658810110047221

SCB. 2015. Invånare per kvadratkilometer efter region, kön och år. *http://www.statistikdatabasen.scb.se:* Statistics Sweden.

SMHI. 2014. Extremt kraftigt regn över Malmö. Retrieved 11/04 2016, from. *http://www.smhi.se/nyhetsarkiv/extremt-kraftigt-regn-over-malmo-1.77503*

SMHI. 2016. SMHI averages 1961-1990. data.smhi.se.

SMHI Vattenweb. 2016. SMHI vattenweb - Daily hydrological measurements. *http://vattenweb.smhi.se/:* SMHI.

Soden, R., and L. Palen. 2014. From Crowdsourced Mapping to Community Mapping: The Post-earthquake Work of OpenStreetMap Haiti. In *COOP 2014 - Proceedings of the 11th International Conference on the Design of Cooperative Systems, 27-30 May 2014, Nice (France),* eds. C. Rossitto, L. Ciolfi, D. Martin, and B. Conein, 311-326 pp. Cham: Springer International Publishing.

Sydsvenskan. 2014. Nota för skyfallet: En kvarts miljard. Retrieved 11/04 2016, from. *http://www.sydsvenskan.se/malmo/nota-for-skyfallet-en-kvarts-miljard/*

The PostGIS Development Group. 2015. PostGIS (Version 2.2.1) [Software]. Available from: *http://postgis.net*.

The PostgreSQL Global Development Group. 2016a. PostgreSQL (version 9.5) [Software]. Available from: *http://www.postgresql.org/*.

The PostgreSQL Global Development Group. 2016b. PostgreSQL 9.5.2 Documentation. Retrieved 25-03 2016, from. *http://www.postgresql.org/docs/9.5/static/resources.html*

UNISDR. 2009. UNISDR Terminology on Disaster Risk Reduction. ed. United Nations International Strategy for Disaster Reduction. Geneva, Switzerland.

UNISDR. 2015. Sendai Framework for Disaster Risk Reduction 2015 - 2030. ed. UNISDR. Geneva, Switzerland.

Vaidya, P. M. 1989. AN O(N LOG N) ALGORITHM FOR THE ALL-NEAREST-NEIGHBORS PROBLEM. *Discrete & Computational Geometry,* 4**:** 101-115. DOI: 10.1007/bf02187718

WHO. 2016. New geographical tool introduced to assess health risks and response capacities in emergencies. Retrieved 24/05 2016, from. *http://www.euro.who.int/en/health-topics/emergencies/pages/news/news/2016/01/new-geographical-tool-introduced-to-assess-health-risks-and-response-capacities-in-emergencies*

Worboys, M. F., and M. Duckham. 2004. *GIS: A Computing Perspective, Second Edition*. United States of America: CRC Press LLC.

Yang, C., D. W. Wong, R. X. Yang, M. Kafatos, and Q. Li. 2005. Performance-improving techniques in web-based GIS. *International Journal of Geographical Information Science,* 19**:** 319-342. DOI: 10.1080/13658810412331280202

Zhan, F. B., and C. E. Noon. 1998. Shortest path algorithms: An evaluation using real road networks. *Transportation Science,* 32**:** 65-73. DOI: 10.1287/trsc.32.1.65

Zhang, S. H., and B. Z. Pan. 2014. An urban storm-inundation simulation method based on GIS. *Journal of Hydrology,* 517**:** 260-268. DOI: 10.1016/j.jhydrol.2014.05.044

Zook, M., M. Graham, T. Shelton, and S. Gorman. 2010. Volunteered geographic information and crowdsourcing disaster relief: a case study of the Haitian earthquake. *World Medical and Health Policy,* 2**:** 7-33. DOI: 10.2202/1948-4682.1069

Institutionen för naturgeografi och ekosystemvetenskap, Lunds Universitet.

Student examensarbete (Seminarieuppsatser). Uppsatserna finns tillgängliga på institutionens geobibliotek, Sölvegatan 12, 223 62 LUND. Serien startade 1985. Hela listan och själva uppsatserna är även tillgängliga på LUP student papers (https://lup.lub.lu.se/student-papers/search/) och via Geobiblioteket (www.geobib.lu.se)

The student thesis reports are available at the Geo-Library, Department of Physical Geography and Ecosystem Science, University of Lund, Sölvegatan 12, S-223 62 Lund, Sweden. Report series started 1985. The complete list and electronic versions are also electronic available at the LUP student papers (https://lup.lub.lu.se/student-papers/search/) and through the Geo-library (www.geobib.lu.se)

350 Mihaela – Mariana Tudoran (2015) Occurrences of insect outbreaks in Sweden in relation to climatic parameters since 1850

351 Maria Gatzouras (2015) Assessment of trampling impact in Icelandic natural areas in experimental plots with focus on image analysis of digital photographs

352 Gustav Wallner (2015) Estimating and evaluating GPP in the Sahel using MSG/SEVIRI and MODIS satellite data

353 Luisa Teixeira (2015) Exploring the relationships between biodiversity and benthic habitat in the Primeiras and Segundas Protected Area, Mozambique

354 Iris Behrens & Linn Gardell (2015) Water quality in Apac-, Mbale- & Lira district, Uganda - A field study evaluating problems and suitable solutions

355 Viktoria Björklund (2015) Water quality in rivers affected by urbanization: A Case Study in Minas Gerais, Brazil

356 Tara Mellquist (2015) Hållbar dagvattenhantering i Stockholms stad - En riskhanteringsanalys med avseende på långsiktig hållbarhet av Stockholms stads dagvattenhantering i urban miljö

357 Jenny Hansson (2015) Trafikrelaterade luftföroreningar vid förskolor – En studie om kvävedioxidhalter vid förskolor i Malmö

358 Laura Reinelt (2015) Modelling vegetation dynamics and carbon fluxes in a high Arctic mire

359 Emelie Linnéa Graham (2015) Atmospheric reactivity of cyclic ethers of relevance to biofuel combustion

360 Filippo Gualla (2015) Sun position and PV panels: a model to determine the best orientation

361 Joakim Lindberg (2015) Locating potential flood areas in an urban environment using remote sensing and GIS, case study Lund, Sweden

362 Georgios-Konstantinos Lagkas (2015) Analysis of NDVI variation and snowmelt around Zackenberg station, Greenland with comparison of ground data and remote sensing.

363 Carlos Arellano (2015) Production and Biodegradability of Dissolved Organic Carbon from Different Litter Sources

364 Sofia Valentin (2015) Do-It-Yourself Helium Balloon Aerial Photography - Developing a method in an agroforestry plantation, Lao PDR

365 Shirin Danehpash (2015) Evaluation of Standards and Techniques for Retrieval of Geospatial Raster Data - A study for the ICOS Carbon Portal

366 Linnea Jonsson (2015) Evaluation of pixel based and object based classification methods for land cover mapping with high spatial resolution satellite imagery, in the Amazonas, Brazil.
367 Johan Westin (2015) Quantification of a continuous-cover forest in Sweden using remote sensing techniques
368 Dahlia Mudzaffar Ali (2015) Quantifying Terrain Factor Using GIS Applications for Real Estate Property Valuation
369 Ulrika Belsing (2015) The survival of moth larvae feeding on different plant species in northern Fennoscandia
370 Isabella Grönfeldt (2015) Snow and sea ice temperature profiles from satellite data and ice mass balance buoys
371 Karolina D. Pantazatou (2015) Issues of Geographic Context Variable Calculation Methods applied at different Geographic Levels in Spatial Historical Demographic Research -A case study over four parishes in Southern Sweden
372 Andreas Dahlbom (2016) The impact of permafrost degradation on methane fluxes - a field study in Abisko
373 Hanna Modin (2016) Higher temperatures increase nutrient availability in the High Arctic, causing elevated competitive pressure and a decline in *Papaver radicatum*
374 Elsa Lindevall (2016) Assessment of the relationship between the Photochemical Reflectance Index and Light Use Efficiency: A study of its seasonal and diurnal variation in a sub-arctic birch forest, Abisko, Sweden
375 Henrik Hagelin and Matthieu Cluzel (2016) Applying FARSITE and Prometheus on the Västmanland Fire, Sweden (2014): Fire Growth Simulation as a Measure Against Forest Fire Spread – A Model Suitability Study –
376 Pontus Cederholm (2016) Californian Drought: The Processes and Factors Controlling the 2011-2016 Drought and Winter Precipitation in California
377 Johannes Loer (2016) Modelling nitrogen balance in two Southern Swedish spruce plantations
378 Hanna Angel (2016) Water and carbon footprints of mining and producing Cu, Mg and Zn: A comparative study of primary and secondary sources
379 Gusten Brodin (2016) Organic farming's role in adaptation to and mitigation of climate change - an overview of ecological resilience and a model case study
380 Verånika Trollblad (2016) Odling av *Cucumis Sativus* L. med aska från träd som näringstillägg i ett urinbaserat hydroponiskt system
381 Susanne De Bourg (2016) Tillväxteffekter för andra generationens granskog efter tidigare genomförd kalkning
382 Katarina Crafoord (2016) Placering av energiskog i Sverige - en GIS analys
383 Simon Nåfält (2016) Assessing avalanche risk by terrain analysis An experimental GIS-approach to The Avalanche Terrain Exposure Scale (ATES)
384 Vide Hellgren (2016) Asteroid Mining - A Review of Methods and Aspects
385 Tina Truedsson (2016) How does the amount of snow and wind conditions effect water pressure measurements during winter in a lake in western Greenland?
386 Chloe Näslund (2016) Prompt Pediatric Care Pediatric patients' estimated travel times to surgically-equipped hospitals in Sweden's Scania County
387 Yufei Wei (2016) Developing a web-based system to visualize vegetation trends by a nonlinear regression algorithm

388 Greta Wistrand (2016) Investigating the potential of object-based image analysis to identify tree avenues in high resolution aerial imagery and lidar data

389 Jessica Ahlgren (2016) Development of a Web Mapping Application for grazing resource information in Kordofan, Sudan, by downloading MODIS data automatically via Python

390 Hanna Axén (2016) Methane flux measurements with low-cost solid state sensors in Kobbefjord, West Greenland

391 Ludvig Forslund (2016) Development of methods for flood analysis and response in a Web-GIS for disaster management