

**Submitted in part fulfilment for the degree of BEng.**

# **An App for Visualisation of Avalanche Hazard**

Chongyang Shi

DRAFT 1 - April 12, 2017

Supervisor: Dr William A. P. Smith

Number of words = 12604, as counted by texcount -inc project\_report.tex.  
This includes the body of the report, but not acknowledgements and bibliographies.

## **Abstract**

This project produces a functional web application to visualise potential avalanche hazards. Within a virtual 3D terrain reality, avalanche hazards along popular winter mountaineering routes can be observed. The project has a primary focus on Scottish mountains while providing excellent adaptability for other regions with appropriate data sources. Avalanche hazards were calculated from a combination of professional avalanche forecasts and spatial analysis of mountain terrain models. Both the accuracy and the usability of the application were evaluated through statistical comparisons with past avalanche records and user experimentation. Statistical evaluations conducted on our static risk model demonstrates better or comparable performance to past efforts. Additional features such as locating historical avalanches and route planning are also provided in the application. The safety and ethical considerations of the application being put into real world use are discussed.

## **Acknowledgements**

I would like to thank Dr William Smith for coming up the original idea for this project, as well as his excellent guidance throughout the course of this project, making much of this possible.

I would also like to thank Dr Christopher Power and Dr Lilian Blot for their support throughout my academic study, as well as all three academics' references enabling my future academic studies.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Overview of the project . . . . .	5
1.2	Motivations behind the application . . . . .	5
1.3	Stages of the project . . . . .	6
1.4	Considerations and statements on ethics . . . . .	6
1.5	Structure of this report . . . . .	7
<b>2</b>	<b>Literature Review</b>	<b>8</b>
2.1	Background on snow avalanches . . . . .	8
2.2	Avalanche forecasting . . . . .	9
2.3	Modelling and visualisation of avalanche hazard in GIS . . . . .	10
2.4	Colour representation of risks . . . . .	12
2.4.1	Saturation-based colouring . . . . .	12
2.4.2	Hue-based colouring . . . . .	13
2.5	Cost-based terrain pathfinding . . . . .	13
2.6	Usability aspects of the application . . . . .	14
<b>3</b>	<b>Problem Analysis</b>	<b>15</b>
3.1	Preliminary Stage: requirement gathering . . . . .	15
3.2	Stage 1: the essential framework . . . . .	16
3.3	Stage 2: the static risk integration . . . . .	16
3.4	Stage 3: pathfinding and past avalanches . . . . .	17
3.5	Stage 4: evaluations . . . . .	17
<b>4</b>	<b>Theories and Methodologies</b>	<b>18</b>
4.1	Dynamic risk factors of avalanche release . . . . .	18
4.2	Static risk factors of avalanche release . . . . .	18
4.2.1	Surface fitting with a polynomial . . . . .	19
4.2.2	Modelling avalanche risk from slope . . . . .	19
4.2.3	Modelling avalanche risk from curvature . . . . .	20
4.2.4	Modelling avalanche risk from terrain roughness based on aspect and slope .	21
4.2.5	Combining static risk models . . . . .	23
4.3	Colour visualisation of risk . . . . .	23
4.4	Combining the static risk model and the dynamic risk model . . . . .	23
4.5	A* pathfinding in a risk-adjusted environment . . . . .	24
4.5.1	Vertical distance between two points . . . . .	24
4.5.2	Balancing costs in distance and risk . . . . .	25
4.5.3	Heuristic function for risk-adjusted A* search . . . . .	25
4.5.4	Overview of the adapted A* algorithm . . . . .	26
4.6	Statistical evaluation of forecast accuracy . . . . .	27
<b>5</b>	<b>Design and Implementation of the Application</b>	<b>28</b>
5.1	Architecture of the system . . . . .	28
5.2	Data retrieval and processing . . . . .	29
5.2.1	Dynamic risk data . . . . .	29

## *Contents*

5.2.2	Altitude, aspect and static risk data . . . . .	30
5.2.3	Past avalanches . . . . .	31
5.3	The freshly-implemented API servers . . . . .	31
5.3.1	Imagery APIs . . . . .	32
5.3.2	Data APIs . . . . .	33
5.4	The user interface . . . . .	33
<b>6</b>	<b>Evaluation and Testing of the Application</b>	<b>36</b>
6.1	Evaluation and testing of usability . . . . .	36
6.2	Evaluation of the risk model . . . . .	36
<b>7</b>	<b>Conclusions</b>	<b>41</b>
7.1	Summary of achievements . . . . .	41
7.2	Potential of future work . . . . .	41

# 1 Introduction

## 1.1 Overview of the project

The purpose of this project is to produce a functional web application to visualise potential avalanche hazards in Scottish Highlands. Through the application, the user is able to freely explore the virtual 3D terrain reality and observe avalanche hazards along popular mountaineering routes. Useful tools such as a route planner and a past avalanche locator are also included in the application.

A model for calculating localised avalanche hazards based on carefully sourced and processed avalanche forecasts [1] and terrain model data [2] was produced. The accuracy of the model has been statistically evaluated against past avalanche records [3, pp. 143-151][4]. Usability and effectiveness of the application guiding the user away from hazardous locations were evaluated through test uses conducted with experienced mountaineers.

The project incorporated tools and techniques from various aspects of Computer Science, including software engineering, geographic information system (GIS) modelling, human-computer interaction, computer graphics and algorithms. Many of these tools and techniques are associated with current research in these areas.

At the time of writing, the application can be accessed at [avalanche.eboronet.com](http://avalanche.eboronet.com).

## 1.2 Motivations behind the application

As a front-runner of digital avalanche forecasts, the Scottish Avalanche Information Service (SAIS) [1] have been providing frequent and reliable winter avalanche forecasts for decades in Scotland.

In addition to written observations on conditions of the snowpack and weather, a typical SAIS avalanche forecast also includes a compass rose, providing avalanche risk levels on the scale of 1 to 5 for slopes of different aspects above and below a transition threshold altitude, shown in Figure 1.1. However, while this compass rose provides a comprehensive overview on avalanche risks across the forecast region, it is not straightforward to interpret. The user will need to work out the surface aspects and altitudes along their route, and mentally infer the risk levels of different locations along their route. This is a tedious task, as it is not only easy to underestimate the risk due to risk levels varying sharply between neighbouring aspects and altitudes (as shown in Figure 1.1), but also requires the user not to be unnecessarily cautious – avoiding better routes with less-than-anticipated risk.

Therefore, the primary purpose of this project is to vastly simplify this process, by generating a coloured image based on the altitude and surface aspects at each point, which is then laid on top of a computer-generated terrain model in the 3D terrain viewer. Hence the information supplied by the compass rose can be visualised in the most straightforward way, allowing the user to freely navigate the 3D model and observe the hazard levels at each location along their route. The avalanche hazard of the same area mapped in Figure 1.1 is visualised as shown in Figure 1.2 (also shown on the right is the final application with high risk terrain areas accentuated and an improved colour coding).

## 1 Introduction

In later stages of the project, we also seek to improve hazard representation through constructing a custom risk model based on information from the compass rose and terrain spatial data; and to improve the functionalities of the application by implementing features such as pathfinding.

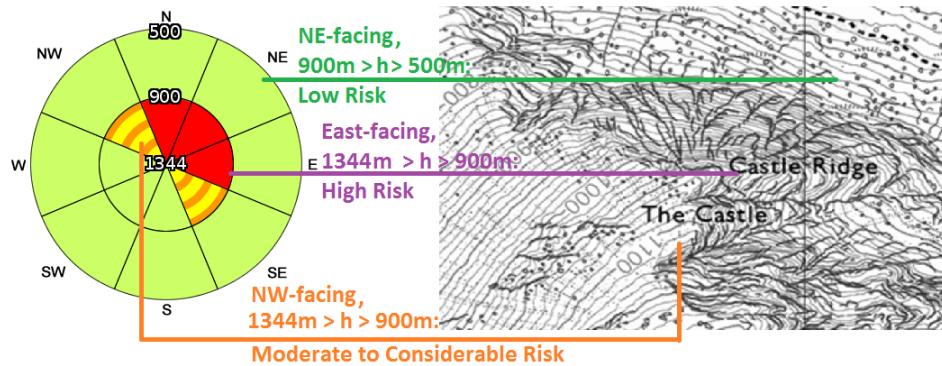


Figure 1.1: An example compass rose produced by the SAIS on January 6, 2016 for Lochaber (left), showing a steep transition of risk levels; and an example manual inference of risk levels based on the compass rose and contour lines near Ben Nevis, Lochaber (right). [5]

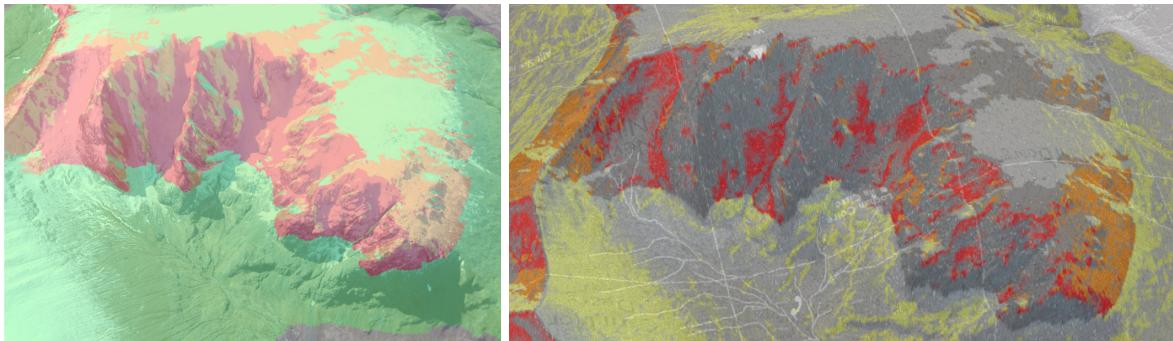


Figure 1.2: The avalanche hazard visualisation of the same area mapped in Figure 1.1, generated in the 3D terrain viewer using the same avalanche forecast by both the essential framework (dynamic risk only, left) and the final product (both static terrain and dynamic risk with new colour coding, right).

## 1.3 Stages of the project

In order to effectively track and manage the workload, the course of the project was divided into a requirement gathering stage and four development stages: the essential framework, static risk integration, risk-adjusted pathfinding, and evaluation & testing. As the product from each stage is able to function independently of work done in later stages, unforeseen difficulties encountered in the research and development process can be effectively mitigated. The stages are described in Chapter 3.

## 1.4 Considerations and statements on ethics

The application was designed as a tool to improve winter mountaineering safety, therefore evidently the foremost consideration on ethics is regarding the accuracy and robustness of avalanche hazards projected by the application. If a location with imminent avalanche threat is reported as safe by the

application, a user relying on the guidance of this application could be placed in serious danger. Conversely, if the application indicate avalanche warnings to users at a location where an avalanche is unlikely to occur, users may take unnecessary, time-consuming diversions that could expose the user to greater danger if daylight fades.

Due to the constraints placed on the lifecycle of this project, it is difficult to conduct an extensive peer review process to examine the accuracy of hazard projections. And while most open source libraries used during the development process are widely-used and have been well-tested, software errors may still occur and affect the safety assurance of the product. Furthermore, the lack of a known commercial or academic product with comparable functionalities eliminates the possibilities of cross-testing the software product.

As a result, while the application is functional and has shown promise during our own evaluations, it is not considered as a product ready for real-life use – a warning of which is displayed when a user accesses the application.

## **1.5 Structure of this report**

Chapter 2 reviews relevant literature. Chapter 3 describes the problem analysis and requirement gathering conducted in the planning stage. The theories studied and developed for the application, as well as the processing of available data are discussed in Chapter 4. Chapter 5 describes in detail the features and internal architectures of the application, as well as its development process. This is followed by Chapter 6, which describes the evaluation and testing conducted on the application as well as its risk model. Finally, Chapter 7 concludes the report and discusses the potential uses and future developments of the application.

## 2 Literature Review

We first consider and evaluate past research and efforts made to understand and predict avalanches, as well as efforts in visualising these knowledge and predictions. While the term avalanche can be applied to various types of moving substances, such as rock and mud [6, p. 1], only snow avalanches are studied by this project.

The literature review starts with an overview of the cause, classification and consequences of snow avalanches in Section 2.1 to motivate the need of accurately visualising avalanche risks. This is followed by comparisons and critiques on existing methods and models for forecasting avalanches in Section 2.2, which can be potentially applied in analysing avalanche forecasts and topographical data in Chapter 4. Past efforts in modelling and visualising avalanches risks are then discussed in Section 2.3, which provides reference theories in designing risk analysis models used in Chapter 4, and statistical evaluations in Chapter 6. Section 2.4 looks into the theories behind colour representations of risk, providing a basis for the colour design of the visualisation in Chapters 4 and 5. Section 2.5 discusses existing methods in cost calculations in 3D space, and compares researches in the pathfinding algorithm A\*, for algorithmic designs of the route planning tool in Chapter 4. Finally, Section 2.6 covers existing Human-computer Interaction researches in 3D navigation and field usage, guiding the interaction design of the application.

### 2.1 Background on snow avalanches

Accumulated snow is usually not a stable structure – snow found in avalanches are usually 80% air, and the snow cover gradually deforms as environment temperature approaches the melting point of water [7, p. 16]. This instability fundamentally allows frequent occurrence of avalanches.

Snow avalanches occur when large masses of snow or ice move rapidly down a mountainside or over a precipice, often triggered from the snow cover [6, p. 1]. Avalanches can either be triggered naturally due to combinations of certain geographical [6, p. 17] and meteorological [6, p. 23] conditions, or triggered by human activities nearby [8]. Often the latter is constructive to the former when an avalanche is triggered [7, p. 17][3, p. 48].

McClung and Schaefer [7, p. 73] classified snow avalanches into two general types: loose snow avalanches, which usually involves surface snow and starts from a single area of the slope; and slab avalanches which are results of failures in the snow cover, featuring blocks of snow falling down the slope destructively. This classification is in agreement with most schemes, such as the avalanche classification by the UNESCO [9], which also diversifies the classification by types of originating zones, paths and deposition zones. The diversified classification has been further improved by recent researches [6].

Regardless of type and triggering source, avalanches are often deadly to human presence nearby, especially when travelling outside areas with built-up defense [6] (known as *backcountry*), as is often the case for mountaineers. During the 45 years leading up to 1999, a total of 440 fatalities from avalanche incidents were recorded in the United States [10]; while in the UK, avalanches in Scottish mountains alone claimed at least 73 lives during a similar time period [3].

With a prosperous winter sport industry bringing millions of people to mountain resorts every year [11], accurate forecasting of avalanches is becoming increasingly critical.

## 2.2 Avalanche forecasting

Avalanche forecasting is defined as predictions of current and future instability of snow in space and time relative to a given triggering level for avalanche initiation [12, p. 131]. When applied, the main concerns of such predictions are risks to humans or property.

A conventional approach to avalanche forecasting involves extensive field observations and testings of the snow cover by experienced forecasters and mountain guides, cross-referencing of historic meteorological and avalanche records, and utilisation of statistical and information theory methods on data collected to reduce decision uncertainties, as summarised by LaChapelle [13] in 1980. This is still very similar to the approach the Scottish Avalanche Information Service (SAIS) use today [14, p. 5]. In this style of approach, final decisions on hazard levels are heavily reliant on the experience and judgement of the experienced forecasters. LaChapelle noted [13, p. 76] that while inaccuracies do occur in forecasters' decisions, complete failures are rare. McClung [7] further analysed the influence of human perceptions in avalanche forecasting, and gave a formalised decision-making process to eliminate biases and improve accuracy.

However, human avalanche forecasting can sometimes be impractical or cost-prohibitive, due to area inaccessibility or vast size, as observed by Bühler *et al.* [15] in the northwestern Himalaya. McClung [16] has explored as early as 1994 that computational methods can be used to improve data analysis and provide supplementary opinions through machine learning methods. Utilisation of computational methods also allow rapid coverage of new areas without undertaking increased human observations, as discussed by Lehning [17]. Notable computational models on different types of data include SNOWPACK and the NN method.

SNOWPACK, a snow profile comparison method was first published by Lehning *et al.* [18] in 2001. The method attempts to establish a numerical profile of snow cover conditions based on various measured properties of the cover, such as the size and type of snow grains, temperature and density of the environments. An agreement score could then be established by comparing the numerical profile with an observed profile in the field. If high agreement scores are reached by the experiments, the numerical profile becomes suitable for use in avalanche forecasts. While SNOWPACK also considers dynamic environmental factors such as the influence of solar radiation and warm winds on the stability of snow layers [19, p. 124], it does not consider the influence of static terrain features on the stability of snow, such as the slope and the aspect.

However, attempts to apply SNOWPACK in other regions with different snow conditions have identified weaknesses in the assumptions made, which was originally developed for the Swiss Alps. A notable attempt was made by Hirashima *et al.* [20] from 2005 to 2006 in Tsunan, Japan. While numerical profiles computed were in reasonable agreement with the observed snow profiles in Tsunan, it was found that the stability index estimators embedded in the model was not suitable for the shear strength of Japanese snow, and an alternative method for calculating the index had to be adopted. This suggested that adaptation of SNOWPACK to a new region may require appropriate changes to the model based on local snow conditions.

While SNOWPACK was developed as a specialised method for analysis of snow covers, more generalised models such as the Nearest Neighbours (NN) method have also been developed. The NN model was initially developed by Buser [21] in 1983, which has since been supplemented by various studies, such as Purves *et al.* [22] in 2003, and Singh and Ganju [23] [24] since 2004. The NN model constructs a vector-space consisting of measurements on weather and terrain conditions on different days of observation, and applies a k-nearest-neighbour method to calculate the likelihood of an avalanche based on a new day's measurements. While measurements may be on different units and scales, which makes distance measurement difficult, this is partly compensated by applying scaling and weighting [21, p. 599]. As the NN method has difficulties with high dimensional data, an alternative method based on Support Vector Machines (SVMs) have been proposed and developed

## 2 Literature Review

[18] [25]. Another deficiency of the NN method is the increased uncertainty when there is not sufficient data available in the vector space [25, p. 379].

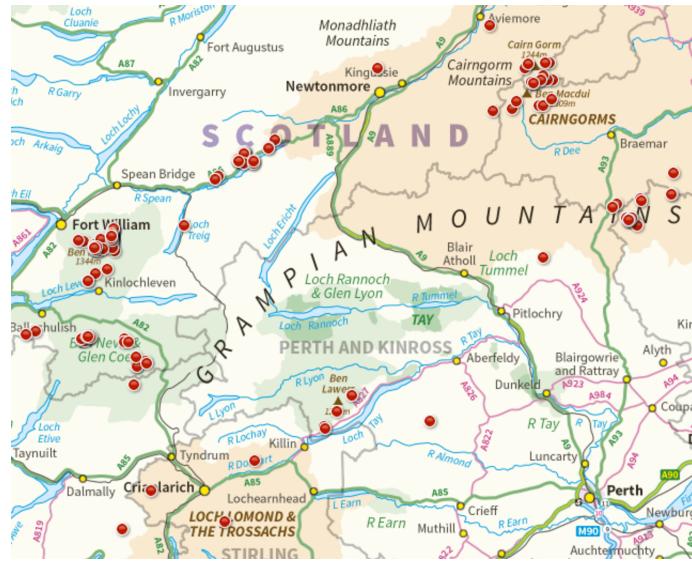


Figure 2.1: Reported avalanches in the Cairngorms and the Glencoe regions during winter sports season of 2015-2016, generated by the SAIS.[4]

The most notable application of the NN method is by the SAIS. Avalanche forecasts from the SAIS that are partially dependent on the NN method achieved a weighted accuracy of between 71% and 82% on visible winter days between 1998 and 2002 [22, p. 351]. With up to 205 avalanches reported during the winter season of 2015-2016 [1] (as partly shown in Figure 2.1), the SAIS plays a very significant role in saving lives from avalanches. Data from SAIS forecasts would also be a critical component in data sourcing and hazard modelling during this project, and is directly used to derive the hazard overlay in **Stage 1** of the project.

In addition to analysis of snow and terrain conditions and past avalanche records, studies of vegetations in mountainous areas can also provide important information on recurrence of avalanches. Christophe *et al.* [26] developed a model based on observation of tree-rings. As avalanches affect the growth of trees and other vegetations on the mountain slopes, growth condition of trees would provide useful insight into the magnitude and frequency of past avalanches in the area. One downside of the model is its unsuitableness for direct use in forecasting temporal avalanches, as the formation of general vegetation conditions require a long period of time. However, for the purpose of this project, this model could potentially be used for establishing a static risk factor for each point in the terrain, which is to be included in the overall risk model.

## 2.3 Modelling and visualisation of avalanche hazard in GIS

In order to effectively store large amounts of geographical data for computing hazard map overlays, a geographic information system (GIS) [27] is commonly used to organise avalanche hazard data.

While a majority of implemented avalanche forecasting methods – such as those discussed in Section 2.2 – provide generalised risk levels with respect to unique sets of environmental conditions, these generalised risk levels cannot be easily converted into localised hazards in a GIS, as these forecasting methods only operate on smaller subsets of parameters contributing to an avalanche. To evaluate the avalanche hazard of a specific location (e.g. Red Burn in Ben Nevis, Lochaber), a systematic method is often developed to calculate risk for each point in the region, taking into account as many

### 2.3 Modelling and visualisation of avalanche hazard in GIS

hazard parameters as possible, which will be demonstrated by past researches discussed in this section.

According to Bühler *et al.* [15], parameters influencing avalanche hazards can generally be placed in one of three categories: (a) terrain topography features at each point in the modelled area; (b) meteorological data of the general area; and (c) characteristics of the snow cover in the area.

Terrain topography features, such as aspect and slope [28, pp. 17-18], convexity and concavity [29, p. 267], and roughness [30, p. 12] can be computed directly from a terrain height-map raster. For the main area of study in this project – Scotland, the government-owned Ordnance Survey has mapped the entire terrain of the British Isles (excluding Republic of Ireland) into a raster with resolution of 5 meters [31]. The data is updated frequently, and has a root-mean-square error of 2.5 meters in mountainous area. The level of resolution and precision is considered as capable of producing good representations of morphologic complexity and surface topography by Rayburg *et al.* [32]. As terrain features outside urban areas are not subject to frequent change, once the aforementioned terrain features are computed, they do not need to be re-computed until the source terrain data updates.

Meteorological and snow profile data, on the other hand, are largely dependent on the current and recent weather conditions in the forecasting regions. Commonly examined parameters by past researches include temperature change, snow accumulation [33], humidity [34], and wind direction [35]; as well as snow profile parameters such as snow grain size and density (as analysed in the SNOWPACK model [18]). However, as most past researches relied on weather forecasts in the regions in which they were conducted, not all parameters they used are available in Scotland. Fortunately, each SAIS avalanche forecast is also accompanied by a snow profile, containing measured meteorological and snow conditions observed during the field work of that forecast [36].

A number of efforts have been made to produce GIS models of avalanche hazards based on some or all of the aforementioned hazard parameters.

Grêt-Regamey and Straub [40] utilised a Bayesian network to study the relation between snow accumulation and flow model of avalanches, and visualised risks of avalanche impacts in an area with the trained network [41]. A more comprehensive risk model was used by Cappabianca *et al.* [37] to produce a similar visualisation, as shown in Figure 2.2 (top-left).

However, neither of the two studies above considered the full topography features of the triggering zones (e.g. lacking convexity and terrain roughness), some of which significantly affect the probability of release, as shown by Ghinoi and Chung [42]. Bühler *et al.* [15] devised a considerably accurate release zone identification algorithm based on terrain topography features alone, as visualised in Figure 2.2 (top-right).

More recent researches have also included meteorological analysis in the risk model. A notable example is by Veitinger *et al.* [39] in 2016: in addition to topography features such as slope and roughness, wind sheltering effects of the terrain was computed based on the direction of wind, and included in the fuzzy logic computations of per-point risk. The algorithm's visualisation of avalanche release risk on a slope where a real avalanche took place is shown in Figure 2.2 (bottom-right). While evaluations showed that little accuracy improvement was achieved in the general case due to potential mapping errors, the algorithm was more accurate than its predecessor in locating frequent avalanche release areas.

Most hazard map visualisations of the aforementioned studies are in 2D only and are static once computed, providing no interactive capabilities. To our knowledge, only Pistocchi and Notarnicola [38] constructed a static 3D model with an avalanche risk overlay, as shown in Figure 2.2 (bottom-left). The risk model by Pistocchi and Notarnicola considers a mix of static terrain features (slope, aspect, curvature, etc.), land features (forest cover) and dynamic features (snow cover duration), and deploys both Weights of Evidence and Logistic Regression to compute the avalanche release potential of each

## 2 Literature Review

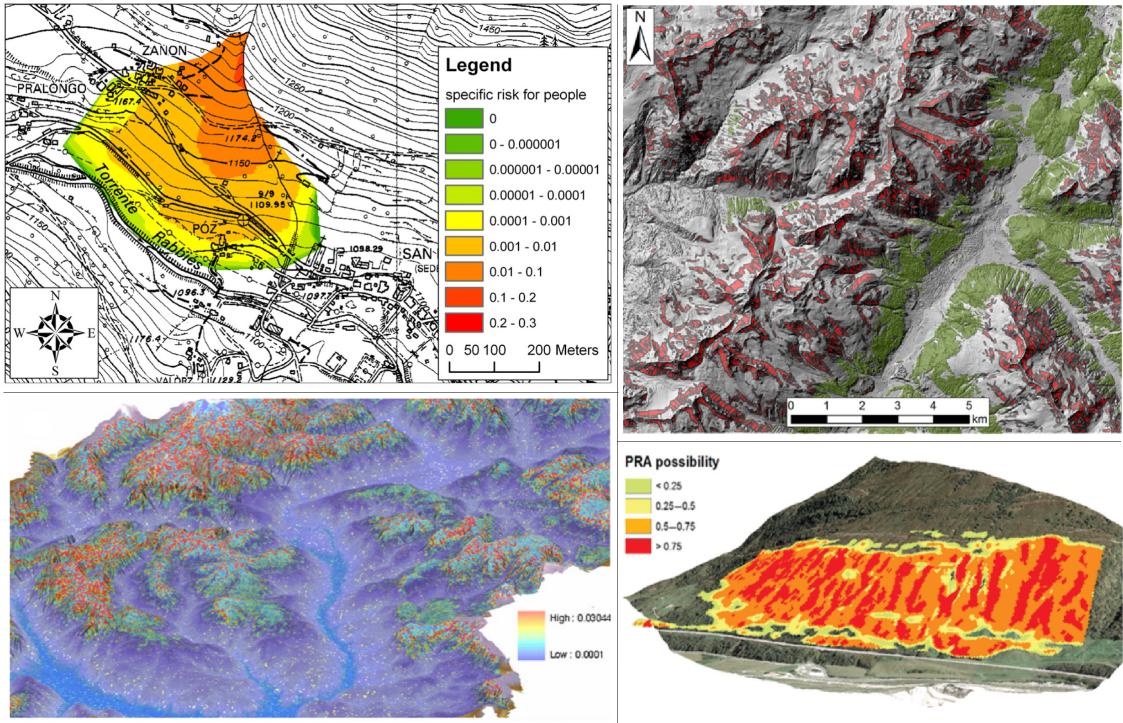


Figure 2.2: Hazard maps produced by prior efforts in visualising avalanche hazards: flow model and snow profile analysis by Cappabianca *et al.* (top-left) [37]; data-driven analysis by Pistocchi and Notarnicola (bottom-left) [38]; terrain topography analysis by Bühler *et al.* (top-right) [15]; and terrain topography and wind shelter analysis by Veitinger *et al.* (bottom-right) [39].

area. At best cases, this model predicts about 70% of avalanches in the 20% of area classified at highest hazard.

Risk visualisations produced by all past efforts discussed in this section are static imageries. For field uses, capabilities of displaying additional image layers and map features "on the fly", as well as the ability to observe from different angles and distances could be very helpful to users.

## 2.4 Colour representation of risks

While a realistic imagery of the surface can be produced in the 3D viewer by covering the artificial surface with corresponding aerial photographs, an appropriate colour representation scheme needs to be chosen for the hazard map overlay, which visualises the risk level at each point in the terrain model. Considerations need to be made regarding the choice of the colour scheme, as well as in what form the colours are displayed. While all colour schemes should have a unique representation for each distinct level of risk, how well an average user of the application perceives the colour scheme is a usability issue. The two main colour schemes proposed by existing researches and some attempts of applying them will be discussed in this section.

### 2.4.1 Saturation-based colouring

This is the simplest colour representation scheme. A single colour determined by one hue value is chosen, and in the context of visualising hazards, the colour of choice is often red [43] [44]. The saturation of this colour at different points in the visualisation would vary, dependent on the level of

risk at that point. As summarised by Cheong *et al.* [44, Ch. 2], there have been conflicting conclusions from researches on the effectiveness of saturation-based risk visualisations, when compared with other schemes. Experiments also conducted by Cheong *et al.* showed that the accuracy of user decisions under this scheme is good while multitasking, but otherwise not as good.

While the subject of risk mapped by the visualisation of Cheong *et al.* is wildfire hazard, a known visualisation of avalanche hazard under this saturation-based scheme was by Bühler *et al.* [15] as discussed in Section 2.3, although no human-computer interaction (HCI) evaluation was conducted. This scheme is also used by Spachinger *et al.* [45] for flood risks.

#### 2.4.2 Hue-based colouring

This colour representation scheme is similar to the saturation-based scheme, but uses different hues of colours (e.g. yellow and red) for different levels of risk in the visualisation. The European Avalanche Hazard Scale [6, pp. 93-94] – also used by the SAIS in the UK – applies such a hue-based scheme, as shown in Figure 2.3. This scheme is also used in visualisation of avalanche hazards by most studies discussed in Section 2.3, as shown in Figure 2.2.



Figure 2.3: colour scheme reconstructed from the European Avalanche Hazard Scale, used by the SAIS [46].

However, some studies have considered the hue-based scheme used in Figure 2.3 to be insufficiently effective, most notably by Conger [47], who found that using green to represent low risk deviates from the common perception of green being a safety-indicating colour, which is not always the case in avalanche forecasting. Orange was also found to be sometimes inappropriate in association with a risk, and more difficult to print cartographically [47, Ch. 4.2]. Therefore Conger has suggested that the risk scale should be rescaled into four, in order to abolish the use of orange. However, other researches such as Braun and Silver [48, p. 2212] disagree with Conger's assessment of orange.

HCI evaluations conducted by Cheong *et al.* [44, Ch. 5] concluded that while a hue-based scheme is often most preferred by users, its user decision accuracy is not the best among different colour representation schemes.

Other variations of applying hue-based colouring include an attempt by Gavaldà and Moner [49] to represent mountaineering route hazards with red-yellow-green traffic signals; as well as a method developed by Saito *et al.* [50], using two tones of hue to represent hazard at each level, which in this case is unsuitable for the steep transitions of risk frequently occurring in avalanche hazard maps [50, Ch. 5.1]. Conger [47, Ch. 5.2] also suggested a method of partially filling hazard areas with triangles that are coloured according to the colour scheme. This however would be difficult to implement in practice, due to the size variations of neighbouring points with the same risk level.

### 2.5 Cost-based terrain pathfinding

Based on computations of per-point risk in **Stage 2** of the project, **Stage 3** focuses on implementing the extended functionality of terrain pathfinding, which will involve adapting existing pathfinding algorithms for use under the complicated conditions of backcountry terrain.

As according to Moore *et al.* [51, pp. 4-5], the terrain height map raster from Ordnance Survey, as discussed in 2.3 is essentially a square-grid network model of the real terrain. The network model was constructed from sampling the real elevation data at a resolution of 5 meters, which minimises the risk of missing abrupt changes in elevation. Pathfinding can then be conducted on the model.

## 2 Literature Review

The most widely implemented algorithm for this purpose is the A\* Search Algorithm [52], utilising heuristic analysis to guide a graph search based on Dijkstra's Algorithm [53]. Most practical uses of search algorithms prefer the use of A\* Search to a pure implementation of Dijkstra's Algorithm, as the former's heuristic function, when well-chosen, guides the search to a far quicker (although sometimes non-optimal) arrival at the goal, particularly in a large search space.

A\* is most widely implemented in the development of pathfinding artificial intelligences in computer games [54]. Since its introduction, a significant number of studies have been conducted to improve the performance of A\* search, including iterative-deepening A\* (IDA) [55] which requires less memory at runtime; hierarchical path-finding A\* (HPA) [56], which utilises hierarchical abstraction on large data grids to produce a near-optimal solution at very high performance; and *navigational mesh* (NavMesh) [57], which minimises the structure representation of data to reduce memory use and computation time. However, the hierarchical structure of HPA means that interconnection points between sub-level grids are often picked around obstacles, which do not exist on the elevation-based terrain grid, potentially hindering HPA's performance if applied.

When adapting A\* or its aforementioned variants for computing a path in the terrain and taking into account the avalanche risk level at each point in the terrain raster, both the distance and the risk levels need to be considered by the heuristic function of A\*. Due to the nature of the uneven terrain, the proper walking distances between points can be estimated by Naismith's Rule and its variants [58]. The accuracy of this estimation decreases as the resolution of the terrain raster decreases, which can occur if the terrain has to be down-sampled to reduce search space size.

To represent the points as a data structure, Yap [59] has suggested using a tiled hex grid to organise neighbouring points for optimisation of performance, which can then be analysed with a hierarchical search algorithm such as HPA [56]. However, it is relatively difficult to re-project square grids of the terrain model into hex tiles, as terrain feature displacements can occur.

## 2.6 Usability aspects of the application

As the project requires that the front-end application to be built is usable both on a mobile device with touchscreen and on a desktop computer with a mouse, best practises of design from existing HCI research on both types of user interfaces need to be considered, as well as the principles for a system that would potentially be used in the field rather than in office.

Nielson and Olsen [60] proposed a formal method of manipulating the view of a 3D object (such as the virtual terrain globe in this application) with a 2D interface. This allows translations, rotations and scalings of the object to be performed by a single 2D event device, such as a mouse. In practise, methods of interactions have been mapped to specific combinations of actions of the mouse wheel and mouse keys [61, pp. 195-197]. However, not all users find mouse interactions straightforward when navigating a 3D scene, and their satisfaction of the mouse interface largely depends on their proficiency with the mouse, as demonstrated by Dubois *et al.* [62].

On modern mobile devices, touchscreen is often the only interface available for the user to interact with the application. With mouse clicks replaced by touch interactions [61, p. 191], manipulations of the 3D object in the viewer are often less straightforward, as studied by Ku and Chen [63], which provided some recommendations on designing touchscreen-based interactions of a 3D object viewer.

For an application that could be used in the field under time and user attention constraints, a study by Pascoe *et al.* [64] shows that the application should have context awareness and utilise interaction designs which minimise distraction to the user. Another study by Albinsson and Zhai [65] explored methods to improve the precision of touchscreen interactions under rough inputs from human fingers.

# 3 Problem Analysis

This chapter covers the stage-by-stage (1.3) plannings for the implementation of the project. This includes a summary of requirements for the application, use of existing software tools and libraries. Existing and new theories and methodologies required for each stage of the development will be discussed in detail in Chapter 4.

## 3.1 Preliminary Stage: requirement gathering

While the project specification has provided clear descriptions on the desired key functionalities, great flexibility is also given to allow varying methods to achieve these functionalities, for which user opinions form a significant contributor. As the supervisor of the project is an experienced mountaineer, extensive consultations with the supervisor established the primary source of user opinions. Owing to the flexibility afforded, requirements are described in lieu of formal functional and non-functional specifications.

**Goal of the application:** to visually represent avalanche hazard levels on a map, with which the user will be able to navigate the terrain with minimised avalanche risk.

**Usability requirements:** The user must be able to use the application both on a desktop computer (mouse and keyboard) and on a mobile device (touchscreen). Due to the storage limitations of contemporary mobile devices, Internet accessibility is assumed to be available for live transmission of data. The performance requirement of the application shall be low enough to be usable on most modern smartphones and computers.

**Interaction requirements:** For the essential framework (**Stage 1**), the user shall be able to freely navigate a virtual 2D or 3D scene, on which avalanche risk levels are overlaid on top of an accurate terrain map. The colours used for the avalanche risk levels should follow the scheme used by the SAIS. For the static risk integration in **Stage 2**, a new colour coding for risk levels should be developed, and it shall be more distinctive and representative of the risk levels implied. A button should also be added to allow the user to switch the application back to **Stage 1**'s dynamic risk-only overlay.

In **Stage 3**, a new tool should be added to the user interface. Through the tool, the user shall be able to specify a start point and an end point on the terrain for pathfinding. A slider should be made available so that the user can configure the weight of distance and risk when considering a path. The user should also be able to clear the existing paths plotted on the terrain with another button. In addition, this stage shall also add a past avalanche locator tool, which will plot on the terrain one red dot for each historic avalanche recorded by the SAIS. When clicking on the dot the user shall see the time and description of that avalanche. The user should be able to switch the past avalanche time frame, such as in the past month or year.

**System requirements:** In order to generate the visualisation to the user, the system must be able to process the data required. In **Stage 1**, the system should be able to retrieve and store the data from the SAIS through web crawling, and pick the correct data for each pixel on the map by reading the height and aspect value from raster files. For the static risk integration in **Stage 2**, the system should be able to compute a static risk value based on various terrain properties for each pixel on the map.

### 3 Problem Analysis

The static risk value should be converted by the system for suitable visualisation in the new colour coding.

In **Stage 3**, the system should be able to compute a path between two points on the terrain specified by the user, based on a specified weighting between minimising path and risk. The system should return a set of coordinates representing the path to be plotted. The pathfinding operation should take a short amount of time for any reasonable input by the user.

**Development requirements:** The project should be developed in a version control system such as Git, and a detailed deployment documentation should be provided to allow any experienced system administrator to redeploy the application. The code written should be well-documented for future work.

## 3.2 Stage 1: the essential framework

While the interaction requirements do not specify whether the risk map should be a 2D or 3D scene, it is decided in the first stage of development that a 3D terrain will be produced by the application, allowing the user the maximum flexibility in navigating and observing the model.

In order to implement the essential framework required for **Stage 1**, it is necessary to first choose the programming languages and associated libraries for the functionalities.

Due to the requirement of making the application available to both desktop and mobile devices, the easiest way to develop for both platforms would be to produce a web application, taking advantage of the WebGL library to render the 3D terrain in web browsers. The Cesium [66] framework was chosen to for the front-end of the application, as it already supports almost all the functionalities desired by the requirements, as well as excellent extensibility to implement the rest. The framework uses HTML5, CSS3 and JavaScript. The user interface will be described in 5.4.

For the back-end of the application, Python was chosen as the programming language used. In addition to the author's existing expertise with the language, Python also provides a wide range of libraries for interfacing with the required components. The dynamically-typed syntax also allows rapid prototyping and testing of the application.

The main Python libraries chosen for the operational components of the application are: Selenium for SAIS data retrieval, Flask for building the API server, Pillow for imaging the overlay colours, and NumPy for associated numerical computations.

With languages and libraries chosen, the implementation of the essential framework will be relatively straightforward. The back-end will simply read the dynamic risk data from the SAIS (Sections 4.1 and 5.2.1) based on height and aspect values of each point, generate overlay images and return them to the front-end. The full operation of the system is described in 5.1.

## 3.3 Stage 2: the static risk integration

As **Stage 2** focuses on improving the data model and colour coding, no architectural change will be required at this stage.

To take into account of static risk factors, additional computations will need to be conducted, the results of which will supplement the image generation process developed in the previous stage. Some techniques used in this stage will be adapted from existing techniques surveyed in Sections 2.2 and 2.3, while others will need to be developed anew, to be discussed in 4.2.

### *3.4 Stage 3: pathfinding and past avalanches*

Minimal changes will need to be made to allow the application to read more data, and compute them differently. The combination of dynamic and static risk models will be discussed in 4.4. Changes to the risk colour coding at this stage will be covered in 4.3.

## **3.4 Stage 3: pathfinding and past avalanches**

The main addition to the application in **Stage 3** is the risk-adjusted pathfinding tool. Another tool for showing the locations of past avalanches is also added in this stage to assist the user in assessing the route provided.

The user interface changes will involve the addition of pathfinding panel according to the requirements, as well as the ability to display the locations and information of past avalanches.

The back-end changes will be more complex. The new pathfinding functionality will require extensive modifications to the standard A\* search algorithm (as described in 2.5) in order to satisfy unique requirements posed by the application, which will be covered in 4.5. This also results in a data API for the API server, to be covered in 5.3.2.

For displaying past avalanches, a new data crawler for past avalanche data will be required (to be discussed in 5.2.3), as well as the addition of another data API (to be covered in 5.2.3).

## **3.5 Stage 4: evaluations**

The final stage of the project involves evaluations on two aspects of the application: the usability of the application and the accuracy of the devised risk model.

The usability evaluation (Section 6.1) will involve user-testing of the application with another experienced mountaineer. During the user-testing, the user will be asked to try functionalities developed across different stages of the project in a simulated scenario, and inquire about their opinions on the user interface, as well as their confidence on the application assisting them to avoid avalanche hazards.

The evaluation of the risk model will be covered in Sections 4.6 and 6.2.

## 4 Theories and Methodologies

This chapter provides the details of the theories in avalanche forecasting, spatial analysis, risk representation and search algorithms applied across different stages of the project. Section 4.1 will be utilised in the construction of essential framework in **Stage 1**, which is built-on in **Stage 2** with Sections 4.2 and 4.3. Section 4.5 provides the basis of algorithm design and implementation in **Stage 3**. Finally, Section 4.6 provides theoretical basis of statistically evaluating the risk model devised against past efforts in **Stage 4**.

### 4.1 Dynamic risk factors of avalanche release

Dynamic factors in avalanche forecasting are considered as factors that change over time, including but not limited to snow precipitation, conditions of snow cover (both well-studied by SNOWPACK [18] as described in Section 2.2), air temperature, wind direction and windspeed (studied by Veitinger *et al.* [39] as described in Section 2.3). These factors also form the major elements of the Nearest Neighbour (NN) forecasting method (2.2), which is used by the project's source of dynamic risk data – the Scottish Avalanche Information Service (SAIS).

Purves *et al.* [22] described the process utilised by the SAIS to forecast avalanche risk in Scottish mountains. Through an NN modelling software tracking observed past avalanches, forecasters will input measurements on temperature, wind and snow conditions, which are then scaled to the same range, and weighted by an autonomous genetic algorithm in order to avoid overfitting on recent avalanche conditions. This process creates a vector in the NN vector-space. The software then provides the input vector's nearest neighbours, the conditions of which then allows experienced forecasters to make a decision on the level of risk for each segment of aspect and altitude in each region. The resulting compass rose-style diagram can be seen in Figure 1.1.

Based on the existing evaluations of the accuracy of SAIS avalanche forecasts [22, p. 351] [67], with an accuracy of above 70% on clear winter days, we have confidence that the dynamic risk model used by the SAIS provides a relatively reliable overview of dynamic avalanche risks across different aspects and altitudes, through a composition of different dynamic factors available to the SAIS observers. Therefore, SAIS risk data were applied directly to the terrain model for the essential framework developed in **Stage 1**.

### 4.2 Static risk factors of avalanche release

Static factors in avalanche forecasting mainly concern the terrain topographic features that rarely change over time. Based on a height map of the terrain, various features can be computed through mathematical formulations. As summarised in Section 2.3, past efforts in modelling avalanche risks based on terrain features have generally covered the following features: altitude, slope, terrain curvature (convexities and concavities) and aspect. While past researches such as Ghinoi and Chung [42, p. 313] have attempted to reclassify altitudes data into distinct altitude bands to take into account the effects of altitude on avalanche risks, this has been equivalently considered in the dynamic risk model (4.1). Therefore, only slope, curvature and aspect are considered in the static risk model of this project.

### 4.2.1 Surface fitting with a polynomial

By definition, slope, curvature and aspect are topography properties relative to a point's neighbourhood, usually within a window of 8 surrounding points in a rectangular grid. Therefore, to calculate these properties for each point, it is necessary to fit a polynomial surface within each such neighbourhood. A commonly used process is to fit a second degree polynomial surface to the neighbourhood. While it is generally agreed that a 2nd order polynomial is the most suitable for the 9-point 3D surface [68] [69] [70], past efforts have applied different configurations of parameters. The configuration chosen for this project is a 9-parameter polynomial by Zevenbergen and Thorne [69, p. 49], which has an order of 2 in each of the x and y directions, as shown in 4.1. This model has the advantage of exactly passing through each point in the neighbourhood with uniquely determined parameters [29, Sec. 6.1.3].

$$Z = Ax^2y^2 + Bx^2y + Cxy^2 + Dx^2 + Ey^2 + Fxy + Gx + Hy + I \quad (4.1)$$

Where coefficients  $A$  to  $I$  are determined by solving a linear system of equations between coordinates of  $(x, y)$  and known  $Z$  values in the height map. The point where the neighbourhood is centered on is  $(0, 0)$ , and the grid is expressed as followed:

$$\begin{matrix} (-1,1) & (0,1) & (1,1) \\ (-1,0) & (0,0) & (1,0) \\ (-1,-1) & (0,-1) & (1,-1) \end{matrix}$$

As the center pixel  $(0, 0)$  is the point where we would like to determine the values of slope, curvature and aspect, which are all derived from the first or second derivatives of the surface polynomial [42, p. 312]. Only constant terms in the derivatives will be required for the calculations, as explained in 4.2.2, 4.2.3 and 4.2.4.

### 4.2.2 Modelling avalanche risk from slope

The slope angle represents the steepness of the terrain at a point, which can be determined by calculating the magnitude of change (first derivative) in both directions [68, p. 50]. Practically, negation is applied to the value to represent that the slope is down-hill, but this is not required in modelling avalanche risk.

The first order partial derivatives of the surface polynomial described in 4.1 are as followed:

$$\begin{aligned} \frac{\partial Z}{\partial x} &= 2Axy^2 + 2Bxy + Cy^2 + 2Dx + Fy + G \\ \frac{\partial Z}{\partial y} &= 2Ax^2y + Bx^2 + 2Cxy + 2Ey + Fx + H \end{aligned} \quad (4.2)$$

At the center point where the slope angle is required, with coordinates  $(0, 0)$ , it is trivial to determine the magnitude of change:

When  $x = 0, y = 0$ :

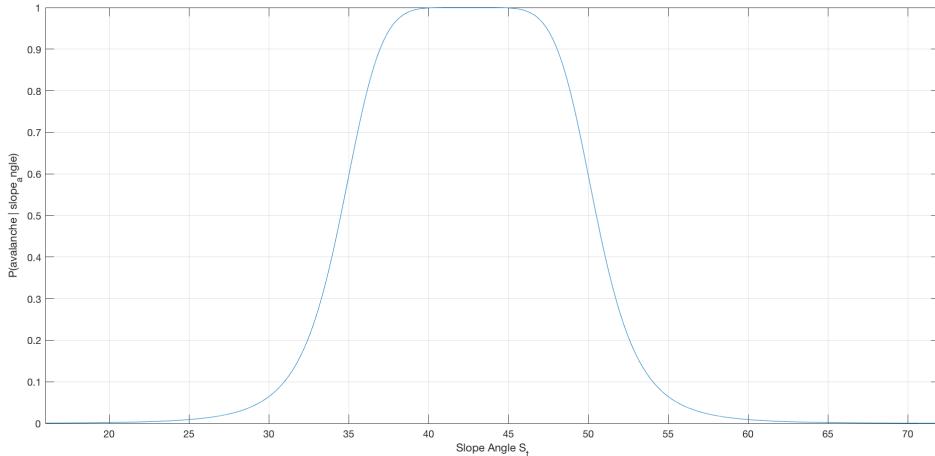
$$\begin{aligned} \frac{\partial Z}{\partial x} &= G, \quad \frac{\partial Z}{\partial y} = H \\ \Rightarrow |SLOPE| &= \sqrt{G^2 + H^2} \end{aligned} \quad (4.3)$$

According to Horn [28], the slope angle is simply the inverse tangent of the magnitude in degree form:

$$S_t = \tan^{-1}(|SLOPE|) = \tan^{-1}(\sqrt{G^2 + H^2}) \quad (4.4)$$

## 4 Theories and Methodologies

The consensus of most researches is that the majority of avalanches occur between a slope angle range of  $25^\circ$  and  $60^\circ$ , with the highest level of risk near the middle of the range [6, p. 42][7, p. 112]. Veitinger and Sovilla [39, p. 2215] modelled this observation as a bell-shaped curve, which has been adapted by our model. Our model made two modifications to the curve. The first was moving the symmetrical line in x direction to  $x = 42.5^\circ$  to make the curve fit better with the McClung [7, p. 112] model, as shown in Figure 4.1.



$$P(\text{avalanche} | S_s) = \frac{1}{(1 + (\frac{S_t - 42.5}{8})^6)} \cdot \frac{1}{C_s} \quad (4.5)$$

Figure 4.1: Adapted bell-shaped curve for modelling avalanche risk of slope angles, normalised to PDF.

The second modification is that the curve can now be normalised to have an integral of 1 over its valid range by dividing with its integral  $C_s$  between  $[0, 90]$  (valid slope angle degrees), thus becoming a probability density function. This will be followed when modelling the other two static risk factors: curvature (Section 4.2.3) and terrain roughness (Section 4.2.4).

### 4.2.3 Modelling avalanche risk from curvature

Another significant static influence on the risk of avalanche release is terrain curvature, which can either be convex, concave or linear, as shown in 4.2. Along the vertical and the horizontal plane, there are two different types of terrain curvatures: profile curvature and planform curvature [68, p. 50]. As the release of avalanches are mainly concerned with gravity in vertical plane, only the profile curvature is considered in our risk model.

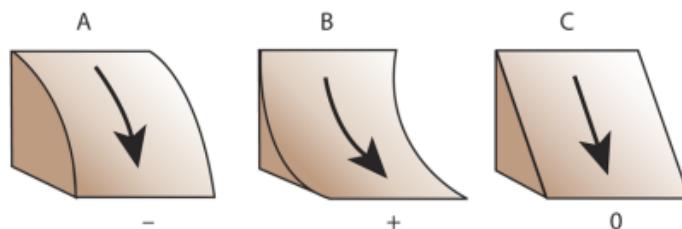


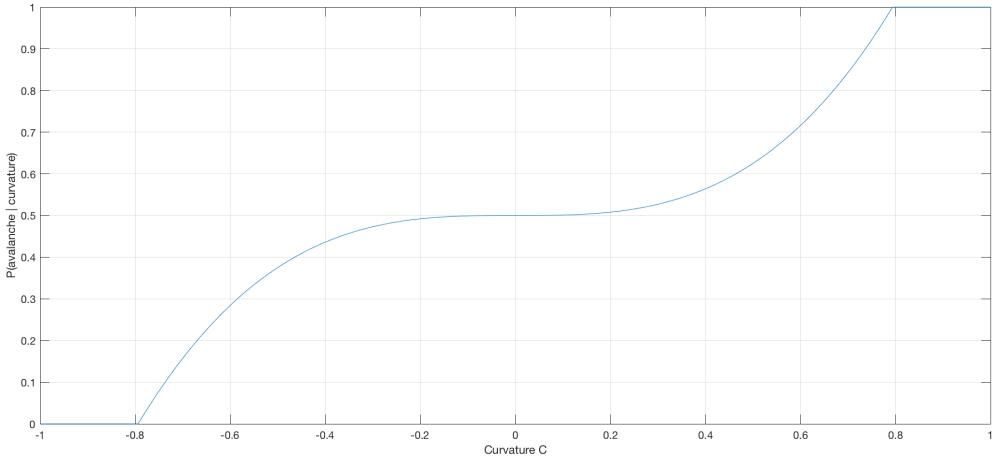
Figure 4.2: An illustration of concave (A), convex (B) and linear (C) terrains. [71, p. 360]

In conformance with the convention, our model defines the value of the profile curvature as one over the radius of the neighbourhood curvature in meters. As with illustrated in Figure 4.2, concave

## 4.2 Static risk factors of avalanche release

curvatures are expressed as negative, convex curvatures are expressed as positive, while linear terrain is defined to have a curvature value of zero.

As observed by McClung [7, p. 115], a majority of avalanches occur on terrains that are without significant curvature (forming a region of similar risk levels near a curvature value of zero), and convex terrains tend to increase the probability of slab avalanches by gliding. Conversely, snowpacks taper off on concave slopes, which reduces the probability of slab avalanches. Therefore, we produced a new probability density function for our model in modelling avalanche risk from curvature  $C$ , as shown in Figure 4.3.



$$Cur = \frac{-2(DG^2 + EH^2 + FGH)}{G^2 + H^2} \quad (4.6)$$

$$S_c = \begin{cases} Cur^3 + 0.5 & -0.7937 \leq Cur \leq 0.7937 \\ 1 & Cur > 0.7937 \\ 0 & \text{otherwise} \end{cases} \quad (4.7)$$

$$P(\text{avalanche} | S_c) = S_c \cdot \frac{1}{C_c} \quad (4.8)$$

Figure 4.3: A spline for modelling avalanche risk of terrain curvature, normalised to PDF with its integral  $C_c$  between  $[-0.7937, 0.7937]$  (uncapped range of terrain curvature).

As shown by (4.7), the numerical model of the spline is based on the curve of function  $y = Cur^3 + 0.5$ , with a limited valid range. Extreme curvatures represented by  $Cur > 0.7937$  or  $Cur < -0.7937$  (curvature radius  $r < 1.26m$ ) will be capped at extreme risk probabilities  $P = 1$  and  $P = 0$  respectively. We believe that the shape of the spline models the aforementioned effects of changing terrain curvature well: as the terrain levels off from a concave, the avalanche risk increases steeply until approaching the linear region, where there is no significance change in the risk level; and as the terrain curvature moves away from this region into convex, the avalanche risk increases again steeply. The curvature value  $Cur$  can be easily computed by taking the second derivative of (4.1) with respect to the aspect angle (calculated by  $\theta = \text{atan}(\frac{H}{G})$ ) in the fitted neighbourhood [29, Ch. 6.2.2].

### 4.2.4 Modelling avalanche risk from terrain roughness based on aspect and slope

While slope angle and curvature have a direct influence on the risk of avalanche release [15, p. 1311], aspect angle does not hold a significance in avalanche risk itself, as it only indicates the direction in horizontal plane the terrain is facing. Instead, aspect angle is used in conjunction with slope angle to calculate the roughness of the terrain, which measures the mechanical resistance of the terrain on the

## 4 Theories and Methodologies

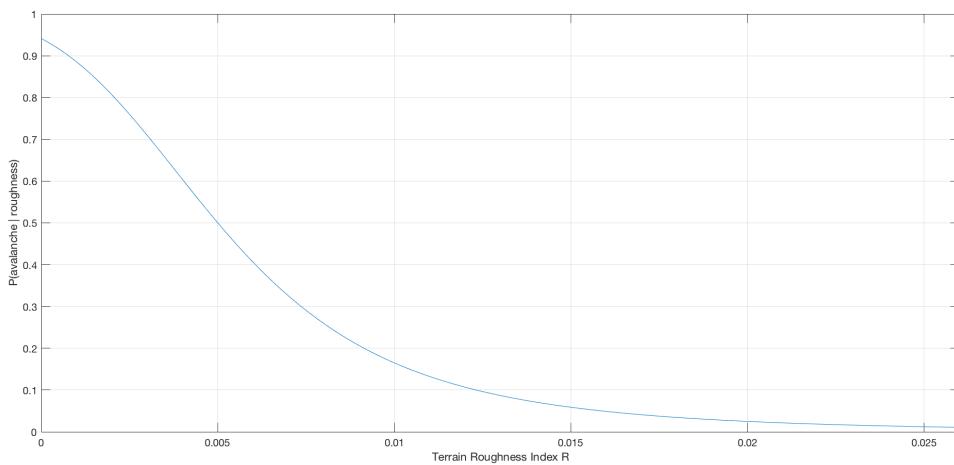
snow cover. The higher the mechanical resistance, the harder it is for a large scale avalanche to be released [42, p. 313].

There are various methods for calculating a terrain roughness index, some of which involves the consideration of landcover, which may change between seasons. Other factors such as changes in vegetation due to planting and logging may also affect the accuracy of a roughness index calculated from landcover data. Therefore, an alternative method based on elevation can be used. A notable method is based on vector dispersion, originally developed by Sappington *et al.* [72] and improved by Veitinger and Sovilla [39, p. 2216].

The method works by decomposing the fitted neighbourhood of nine points into three components ( $X$ ,  $Y$  and  $Z$ ) based on the values of slope angle ( $\alpha$ ) and aspect ( $\beta$ ) at each point. Then for each of the three components, we sum up the neighbourhood. A resulting vector  $|r|$  can then be calculated for the center point by further summing up the squares of the three neighbourhood sums and taking the square root. Finally, the roughness index  $R$  can be determined by normalising the vector with the number of points in the neighbourhood (that is, 9) and subtract it from 1. The process is shown as followed:

$$\begin{aligned} d_{xy} &= 1 \cdot \sin(\alpha) \\ \Rightarrow Z &= 1 \cdot \cos(\alpha), \quad X = d_{xy} \cdot \cos(\beta), \quad Y = d_{xy} \cdot \sin(\beta) \\ |r| &= \sqrt{(\sum X)^2 + (\sum Y)^2 + (\sum Z)^2} \\ R &= 1 - \frac{|r|}{9} \end{aligned} \tag{4.9}$$

From (4.9), it is easily observed that the value of  $R$  is always positive. The tested model by Veitinger and Sovilla [39, p. 2216] believes that avalanche risk decreases sharply between 0 and 0.01, and further decreases slowly between 0.01 and 0.02. At a roughness level of above 0.02, an avalanche release will be unlikely. Therefore, a similar bell shaped curve of a probability density function is used as shown in Figure 4.4, albeit only using the relevant section where  $R > 0$ .



$$P(\text{avalanche} | S_r) = \frac{1}{\left(1 + \left(\frac{R+0.005}{0.01}\right)^4\right)} \cdot \frac{1}{C_r} \tag{4.10}$$

Figure 4.4: A bell-shaped curve for modelling avalanche risk of terrain roughness index, normalised to PDF with integral  $C_r$  between  $[0, \infty]$  (valid roughness index values).

#### 4.2.5 Combining static risk models

With three different static risk factors calculated: slope angle, curvature and roughness, a method of uniforming the three into a single per-point risk value is needed. While it is possible to apply a Support Vector Machine (SVM) [25] as described in Section 2.2 or a fuzzy logic operator [39, p. 2218], because all three static risk factors modelled to be probability density functions (PDFs), the most straightforward method of combining them is to normalise each curve and multiply the resulting probabilities at each point. This also has the advantage of avoiding potential biases caused by the choice of weighting factors in a SVM or fuzzy operator. Values of each risk factor at each point is divided by its normalising constant as described earlier, and then multiplied together at each point. The process is shown as followed:

$$P(\text{avalanche}|S_s, S_c, S_r) = \frac{P(\text{avalanche}|S_s)}{C_s} \cdot \frac{P(\text{avalanche}|S_c)}{C_c} \cdot \frac{P(\text{avalanche}|S_r)}{C_r} \quad (4.11)$$

### 4.3 Colour visualisation of risk

For the essential framework completed in **Stage 1**, a colour scheme identical to the colour coding used by the SAIS (as shown in Figure 2.3) was used for the risk levels (which at **Stage 1** relies only on the dynamic risk model). However, as described in Section 2.4, past researches have found this colour coding scheme to be ineffective and has the potential to mislead the user. Therefore, a new colour scheme was implemented, as shown in Figure 4.5.



Figure 4.5: The new colour scheme implemented in **Stage 2**.

Recommendations made by Conger [47] was partially applied: green has been eliminated from the colour scheme to avoid giving the user a false perception of safety; however, the recommendation to eliminate orange was not applied, due to inconsistency of findings on orange between past researches, and orange's wide usage in risk colour codings.

With the elimination of green, yellow is instead used for representing the low risk level, while a new risk colour with a hue value between yellow and orange, but lower lightness than both in the HSL scale is used for representing moderate risk. The hue values for other risk colourings were also slightly adjusted to produce a good distinction between different colours, the effect of which can be seen in Figure 1.2 (right).

### 4.4 Combining the static risk model and the dynamic risk model

As shown in Figure 2.3, the dynamic risk model produced by the SAIS has five distinct bands of risk levels (six if the grey data-not-available is included). In contrast, as shown in Section 4.2, the static risk model contains continuous data. This requires an appropriate method to combine the two models for visualisation purposes in the application. While it is possible to merge the two models numerically again through either a Support Vector Machine (SVM) or a fuzzy logic operator, due to the lack of an established precedence between the influence of the static model and the dynamic model used in this project, the use of different representations for the two models was instead preferred.

As described in Section 2.4, the two main schemes for colour representation of risk are saturation-based and hue-based. This provides an opportunity for implementing a mixed approach, in which

## 4 Theories and Methodologies

two different representations can be used: one with hue and the other with saturation.

In particular, the dynamic risk level will determine the value of hue at a point, thus determining the "colour" of the point in the traditional sense, which is in turn determined by its altitude and terrain aspect as intended by the SAIS. The static risk level will then determine the saturation at that point. This means that the less likely static risk analysis believes that an avalanche release could occur at a point, the fainter the colour of that point will be. Conversely, a strong colour implies a high likelihood of avalanche release at a point based on static risk analysis.

In practise, thresholding will be applied to all static risk levels in the raster for the scaling of static risk value to saturation. To reduce the influence of extreme values on the visualisation of static risk, the static risk values are scaled between the 1st percentile and the 99th percentile rather than between the minimum and maximum risk values.

The resulting risk colour scheme, used in **Stage 2**, can be seen in Figure 4.6. The effects of static risk controlling saturation can also be observed in Figure 1.2 (right).

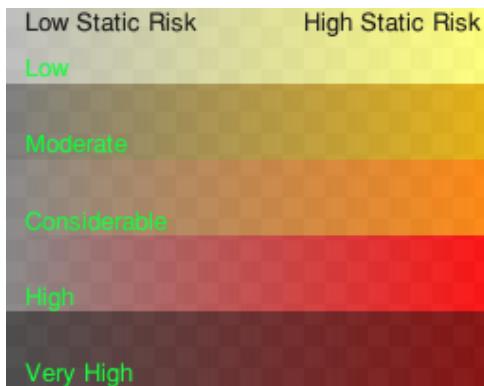


Figure 4.6: The new risk colour scheme implemented in **Stage 2**. Each row represents a dynamic risk level.

## 4.5 A\* pathfinding in a risk-adjusted environment

As described in Section 2.5, A\* search is a well-studied algorithm for pathfinding in a graph, especially in challenging conditions such as large datasets and strict or real-time timing constraints. It has been chosen as the basis for the pathfinding feature in **Stage 3** over a simpler implementation of Dijkstra's Algorithm, due to the convenience of introducing avalanche risk into the former's heuristic function, as well as former's usually superior performance [73] when only an approximate answer is required.

There are nevertheless a number of challenges when adapting A\* search for the specific purpose of risk-adjusted pathfinding in a 3D terrain, including calculating the vertical distance between two points, balancing costs in distance and risk as required, and devising a suitable heuristic function for the problem. These will be addressed in the next few subsections, followed by an overview of the adapted algorithm.

### 4.5.1 Vertical distance between two points

As terrain rasters arranges data in a grid with uniform resolution, horizontal distances between neighbouring nodes are consistent. Therefore, the only distinction of distance between two neighbouring nodes is the vertical distance. While it is possible to calculate the linear distance between

two points based on trigonometry, this is both computationally expensive for a large data grid and inaccurate: it is unlikely that the terrain surface is smooth and straight between two neighbouring points. As an alternative, it is possible to estimate this distance based on observations and average cases: the Naismith's Rule, as introduced in 2.5.

The Naismith's Rule estimates the amount of equivalent distance walked by a mountaineer to ascend or descend each unit of vertical distance, called the Naismith distance. In average case on a natural terrain, it has been estimated that walking on a levelled terrain for 3 miles is equivalent to ascending or descending 2000 feet vertically, which is equivalent to 7.92 meters of walking distance for each meter of vertical distance, as summarised by Scarf [74]. This allows a simple conversion process to be established as followed:

$$\begin{aligned} \text{Naismith\_Distance}(\text{NodeA}, \text{NodeB}) &= \text{Raster\_Resolution}(\text{NodeA}, \text{NodeB}) \\ &\quad + 7.92 * |\text{NodeA.height} - \text{NodeB.height}| \end{aligned} \quad (4.12)$$

where Raster\_Resolution returns the raster resolution in the direction of PointA to PointB, which is either straight or diagonal.

#### 4.5.2 Balancing costs in distance and risk

A requirement of the project requires the algorithm to accept a user-configurable weighting of risk against distance, allowing the user to receive several possible routes between two points based on the importance they give to the risk enroute and the overall distance. After the user has requested a path with such a risk weighting, the application needs to calculate the overall cost of moving between two nodes accordingly. The risk weighting is between 0 and 1, representing the percentage of overall cost attributed to avalanche risk.

While it is simple to multiply the separately calculated distance cost and avalanche risk cost by their defined weights, it is also important that the two costs are normalised into the same scale before combining them. This requires the maximum and minimum distance cost and avalanche risk cost to be calculated beforehand. The full process is shown as followed:

$$\begin{aligned} \text{normalised\_distance} &= \frac{\text{Naismith\_Distance}(\text{NodeA}, \text{NodeB}) - \text{min\_distance}}{\text{max\_distance} - \text{min\_distance}} \\ \text{normalised\_risk} &= \frac{\text{NodeB.risk} - \text{min\_risk}}{\text{max\_risk} - \text{min\_risk}} \\ \text{overall\_movement\_cost} &= \text{normalised\_risk} \cdot \text{risk\_weighting} \\ &\quad + \text{normalised\_distance} \cdot (1 - \text{risk\_weighting}) \end{aligned} \quad (4.13)$$

It is worth noting that the avalanche risk of moving between two nodes is dependent on the target node only, as calculating a potentially negative difference in risk levels between two nodes could affect the admissibility of the heuristic function.

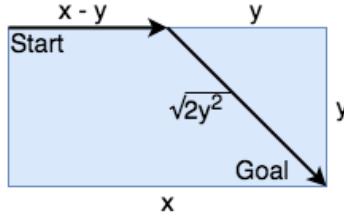
#### 4.5.3 Heuristic function for risk-adjusted A\* search

The heuristic function used to determine the priority under which each node is explored considers not only the Naismith distance between the candidate node and the goal node, but also the immediate static risk of moving to the candidate node from the current node. The latter creates a risk-averse behaviour for the algorithm, which helps the pathfinding to avoid localised avalanche hotspots.

Due to the fact that the search will be conducted on a rectangular section of the data grid whose position is determined by the location of the start node and the goal node, and that it is possible

## 4 Theories and Methodologies

to move diagonally in the real world, the commonly used goal distance estimation based on the Manhattan Distance must be altered, as shown in Figure 4.7.



$$\begin{aligned}
 \text{estimated\_distance} &= (x - y) + \sqrt{2 \cdot y^2} \\
 &\quad + 7.92 * |\text{NextNode.height} - \text{GoalNode.height}| \\
 \text{heuristic\_cost} &= \text{estimated\_distance} \cdot \text{NextNode.risk} \\
 \text{scaled\_heuristic} &= \frac{\text{heuristic\_cost} - \text{min\_naismith}}{\text{max\_naismith} - \text{min\_naismith}}
 \end{aligned} \tag{4.14}$$

Figure 4.7: Calculating the scaled heuristic cost, where  $x$  represents the longer side of the rectangle.

The heuristic cost of moving to a node can then be calculated by multiplying the risk level of the next node with the estimated Naismith distance to the goal node. The resulting heuristic cost is then scaled to be between the maximum and minimum actual Naismith distances calculated. This combines with the actual cost of moving to the node, as shown in (4.14).

### 4.5.4 Overview of the adapted A\* algorithm

---

**Algorithm 1** Pseudocode for the adapted A\* search algorithm.

---

```

1: search_queue  $\leftarrow$  PriorityQueue()
2: search_queue.push(priority  $\leftarrow$  0, initial_node)
3: source_index  $\leftarrow$  {}
4: costs_dict  $\leftarrow$  {}
5: source_index[initial_node]  $\leftarrow$  None
6: costs_dict[initial_node]  $\leftarrow$  0
7: while search_queue  $\neq$  empty do
8:   current_node  $\leftarrow$  search_queue.pop();
9:   if current_node = goal_node then ▷ Goal found, terminate.
10:    break
11:   end if
12:   neighbours  $\leftarrow$  get_neighbours(current_node)
13:   for all neighbour_node in neighbours do
14:     cost  $\leftarrow$  naismith_distance(current_node, neighbour_node) * (1 - risk_weighting) +
        neighbour_node.risk * risk_weighting;
15:     new_cost  $\leftarrow$  costs_dict[current_node] + cost;
16:     if neighbour_node not in costs_dict or new_cost  $<$  costs_dict[neighbour_node] then
17:       costs_dict[neighbour_node]  $\leftarrow$  new_cost ▷ A shorter path.
18:       neighbour_priority  $\leftarrow$  new_cost + heuristic(neighbour_node, goal_node)
19:       search_queue.push(priority  $\leftarrow$  neighbour_priority, neighbour_node)
20:       source_index[neighbour_node]  $\leftarrow$  current_node
21:     end if
22:   end for
23: end while
24: reconstruct_path(goal_node, source_index);

```

---

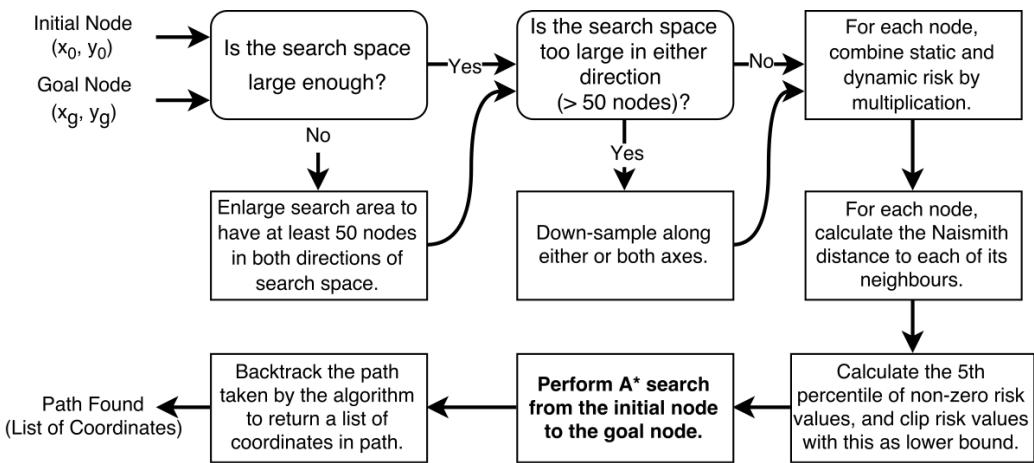


Figure 4.8: The operations of the pathfinder implementation.

As shown in Figure 4.8, a few pre-processing steps are required to ensure that the algorithm can finish quickly with a meaningful result. If the bounding box created by the initial and final node coordinates is too small (occurs when the line between two nodes is almost in parallel with a longitude or latitude line), the search area is enlarged to allow the algorithm more choices. If the search space is too large, linear down-sampling is applied, picking the maximum height and risk level in the sample (for conservative estimations of Naismith distance and risk), and the mean aspect angle in the sample.

Thereafter, the Naismith distances from each node to its neighbours are calculated by applying (4.12). A clipping of risk values to between the 5th percentile of non-zero risk values and the maximum risk is also conducted, as preliminary testing of the implementation showed that blocks of zero risk values significantly reduce the performance of the algorithm.

Finally, the A\* search algorithm is performed on the pre-processed search space, as shown in Algorithm 4.5.4. A path in the form of a list of coordinates can then be constructed through backtracking the search space from the goal node.

## 4.6 Statistical evaluation of forecast accuracy

In order to determine the quality of the avalanche risk models used by the project, it is essential to conduct some statistical tests on the model. An ordinary approach is to conduct an inferential statistical test such as the Mann-Whitney U test [75] to determine whether our models accurately determine the locations of actual avalanche releases. However, as there is no prior static risk model applied to the Scottish Highlands that we are aware of, this is not possible.

As described in Section 4.4, our static risk model is combined with the dynamic model through multiplication at runtime, as the dynamic factors change too frequently to efficiently store a combined model. This presents a major difficulty in evaluating the combined model, in addition to historic SAIS forecasts being incomplete (compass rose only available since 2009 with large gaps), as well as the computational difficulties in storing 5-6 large dynamic risk rasters for each of the thousands of days. Therefore, our statistical evaluation will focus on the static risk model only, as described in Section 6.2.

# 5 Design and Implementation of the Application

This chapter provides details on the implementation of the software system. Starting with an overview on the architecture design in Section 5.1, different levels of the implementation will be covered: retrieval and processing of various types of data (5.2), the API servers serving as an intermediary between data and the user interface (5.3), and the user interface itself (5.4).

## 5.1 Architecture of the system

As shown in Figure 5.1, the system is structured across four levels: Data Store, Data Interface, API Servers and Front-end. The first two are collectively described in detail in Section 5.2, and the last two in 5.3 and 5.4 respectively. Each level abstracts away the implementation details of the previous level by providing interfaces of interaction for the next level, allowing maintenance and extensions to be conducted easily on the codebase. During live operations, a user request is handed down each level from the Front-end to the Data Store, and returned in the reverse order, forming the operation flows.

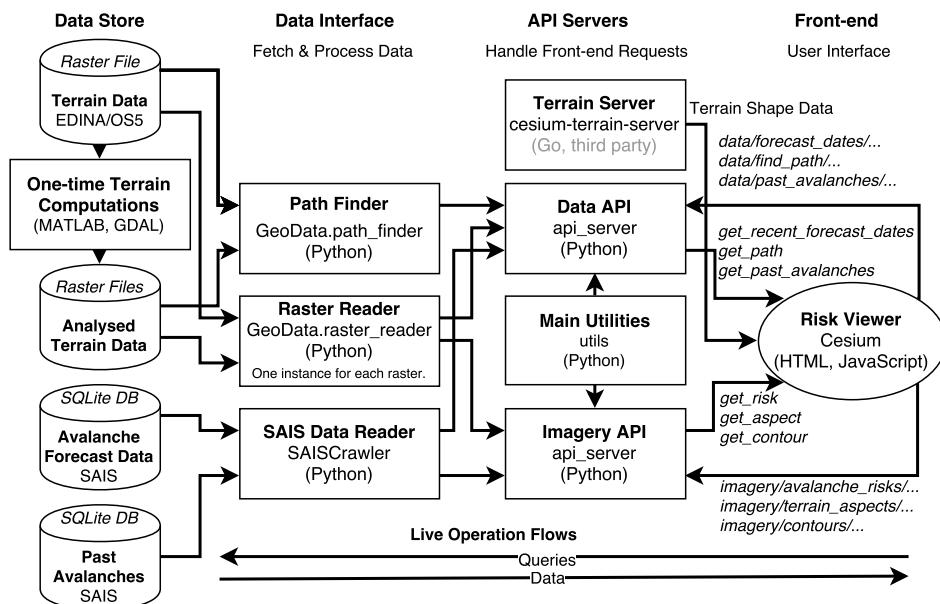


Figure 5.1: The architecture of the system. All components in Data Interface, API Servers and Front-end are implemented from scratch during this project, except the Terrain Server.

The Front-end takes the form of a Cesium application running in the user's web browser, served by Nginx running on the application server. Nginx also terminates the TLS connection from the user to ensure secure and tamper-resistant data transmission. As the user interacts with the virtual 3D scene in Cesium, it will automatically request the necessary data from the API servers on the next level – either overlay images through the Imagery API or route and past avalanche data through the Data API.

The intermediate level, API Servers, contains the two Python APIs implemented with the Flask

## 5.2 Data retrieval and processing

library: Imagery API and Data API (covered in 5.3). Upon receiving data requests, the APIs will first determine whether the requests are of valid form, and scrutinise the parameters. Once a request is verified, the handling API will make function calls to both the utility library and the modules of the next level – Data Interface. In addition to serving the Front-end, Nginx also provides reverse proxy for the API Servers, passing requests and returned data in between them and Front-end.

Data Interface is the level where specialised functionalities are implemented, and modules interact with stored data. They are called upon by specific APIs for specific purposes, such as pathfinding or retrieving past avalanche records. For operational safety, input sanitisation is conducted again at this level.

Data Store is the lowest level, where various types of data required for the operation are stored. Some types of data such as terrain models are stored as static files once computed, while others are stored in a SQLite database for dynamic update and retrieval.

## 5.2 Data retrieval and processing

The Data Store of the application stores various form of data, some of which require pre-processing before usage (static risk data, 5.2.2), while others are fetched and stored directly live (5.2.1 and 5.2.3). This section covers both their sourcing and pre-processing (if applicable).

### 5.2.1 Dynamic risk data

The Scottish Avalanche Information Service (SAIS)[1] provides forecasts and information of the avalanche risks in Scottish Mountains several days a week during the winter sports season. The most prominent scale of risk is the hazard compass rose [46, p. 4], as shown in Figure 5.2. Although not completely representing the risk assessment by itself, it provides an excellent source of information in correlating altitude and aspect to the approximate level of avalanche risk, ranging from 0 (Risk Not Applicable) to 5 (Very High Risk), which can be used as part of a data model in estimating avalanche risk.

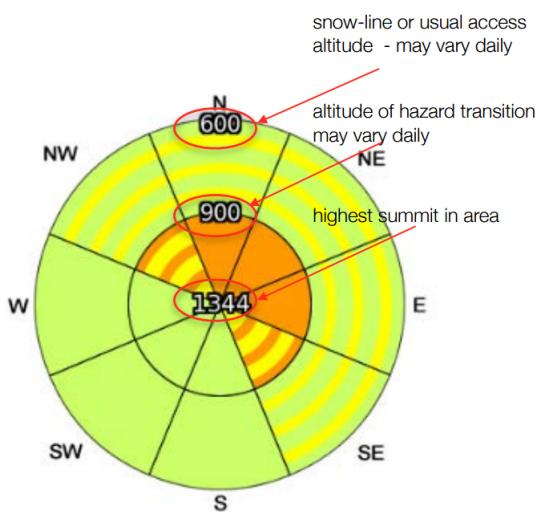


Figure 5.2: An example compass rose produced by the SAIS.[46, p. 4]

Unfortunately, the SAIS does not provide an API with which the risk levels making up the compass rose could be retrieved easily. Therefore, it was necessary to build a web crawler (*SAISCrawler*)

## *5 Design and Implementation of the Application*

with the Selenium library to automatically search for and download avalanche forecasts from the SAIS.

Vertically, the forecast data is divided by altitude into two sectors, each of which is further divided by terrain aspect into eight sections, each representing a  $30^\circ$  circular segment, as depicted in Figure 5.2. Each segment contains two risk levels: the primary level and the secondary level. Therefore, a total of 32 risk level values are extracted from each compass rose.

In conformity with the entity-relationship model, eight forecast records are created for the eight circular segments, each containing the primary and secondary risk levels for the two altitude sectors in that segment, along with the date of forecast and the transition altitudes. These records are stored in a SQLite database by the crawler. An example set of forecast records is depicted in Figure 5.3.

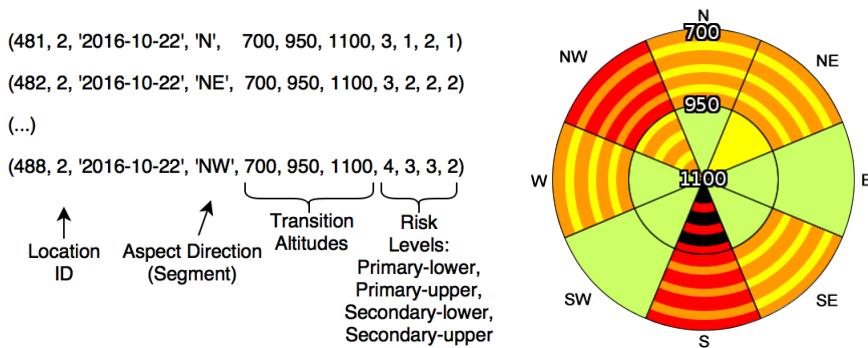


Figure 5.3: An example set of forecast records (left) extracted from a compass rose (right) [1].

As a conservative assessment of dynamic risk, the higher of the primary and the secondary risk levels in each section is used as the dynamic risk of that section in processing.

### **5.2.2 Altitude, aspect and static risk data**

As static risk data originate from analyses of the terrain model, it is necessary to obtain a full Digital Elevation Model (DEM) for the areas of interest (Scottish Highlands). There are a number of available sources for DEM data of the United Kingdom, most notably the EU-DEM Data [76] provided by the European Commission (25-30m resolution, open license), and the UK Government's Ordnance Survey Data (OS5) [2] (5m resolution, limited license, mentioned in 2.3).

With considerations on the nature of the project and the potential uses of the application, the Ordnance Survey Data is used for the course of the project, as an elevation model with higher accuracy is beneficial for accurate computations and evaluations. However, should an expansion of coverage or usage be required, the EU-DEM Data can be processed and used with the same procedures.

The Ordnance Survey Data is supplied on a tile-by-tile basis. To improve processing efficiency, the first part of the processing procedures is to merge the data into a single GeoTIFF [77] raster with the GDAL tool:

```
find ./OS5/ -name '*tif' | xargs gdal_merge.py -of GTiff -o OS5_Merged.tif
```

The merged DEM data is then reprojected by GDAL into the WGS84 geodetic system [78] that is used by the majority of mapping and navigation systems:

```
gdal_translate -a_srs EPSG:4326 -of GTiff OS5_Merged.tif WGS84.tif
```

### 5.3 The freshly-implemented API servers

The reprojected elevation (height-map) raster `WGS84.tif` can now be used to look up altitude for each point in the geographical area covered by the DEM Data. However, we must also analyse the elevation raster to infer a few other sets of raster data: a raster storing terrain aspect at each point (`WGS84_Aspects.tif`) is required for both dynamic (4.1) and static risk analysis (4.2.4); a raster storing terrain slope angle (`WGS84_Slope.tif`) is part of the static risk model (4.2.2); as well as a raster (`WGS84_Curvature.tif`) storing terrain curvature (4.2.3). The aspect and slope angle rasters are computed with `gdaldem` as shown:

```
gdaldem aspect WGS84.tif WGS84_Aspects.tif -of GTiff  
gdaldem slope WGS84.tif WGS84_Slope.tif -of GTiff
```

The curvature raster is computed with the `Slope`, `Aspect`, `Curvature` tool in the GRASS library.

The final raster, `WGS_StaticRisk.tif`, storing the static risk level of each point in the static risk model is one-time computed with a MATLAB script, using data from aspect, slope angle, and curvature rasters. The methodology behind this script has been covered in detail in Section 4.2.

For the convenience of the user, a map raster directly downloaded from The Ordnance Survey is also stored, providing overlay images of roads, location names and contours lines.

The terrain shape files used by Cesium to display 3D terrain models can also be built and served with third-party tools developed by the University of Southampton<sup>1</sup>, which provides an alternative to the default terrain shape files served by the maintainers of the Cesium library.

#### 5.2.3 Past avalanches

As reflected in the requirements, requests from a potential user include the ability to label the location of past avalanches on the map. The SAIS provides an Avalanche Map which labels the time, approximate location and additional descriptions of each past avalanche in the monitored area, as reported by both the SAIS' field forecasters and users of the website. This information is not only useful to the user assessing the avalanche risk of their route, but also a good source of reference data for statistical evaluations of the risk model. However, it is worth noting that the existence, the date, and the location of each avalanche are not exact, as they were not always recorded scientifically.

A crawler similar to the one used for retrieving SAIS avalanche forecasts (5.2.1) was written for retrieving past avalanche records. The data is stored in the format shown in Figure 5.4.

Field	Type	Description
<code>avalanche_internal_id</code>	INT	Our ID of avalanche record, primary key.
<code>avalanche_id</code>	INT	The SAIS' ID of the avalanche for update checking.
<code>easting</code>	INT	The horizontal coordinate value of the location in OSGB36 format.
<code>norting</code>	INT	The vertical coordinate value of the location in OSGB36 format.
<code>avalanche_time</code>	TEXT	The date and time of the recorded avalanche.
<code>avalanche_comment</code>	TEXT	Additional descriptions of the avalanche supplied by the SAIS.

Figure 5.4: The format of a stored past avalanche record, stored in SQLite.

### 5.3 The freshly-implemented API servers

Serving as the intermediary between the user and the data processors, the API Servers level provide two categories of APIs required for implementing the actions of the application: Data API and Imagery API, both of which will be covered in this section.

<sup>1</sup>cesium-terrain-builder: <https://github.com/geo-data/cesium-terrain-builder>

## 5 Design and Implementation of the Application

### 5.3.1 Imagery APIs

APIs returning an image on valid calls are categorised as Imagery APIs. This includes: `avalanche_risks`, `terrain_aspects`, and `contours`. The images returned are PNG images varying levels transparency for optimised interlacing.

**avalanche\_risks:** this is the main imagery API, responsible for returning per-point risk level images for either dynamic-only or static-and-dynamic risk models (4.4), as determined by an optional parameter. It fetches data for an area defined by input parameters representing two points in the raster, as shown in Figure 5.5. Height and aspect rasters are required for dynamic-only mode, while the computed static risk raster is also required for the combined mode.

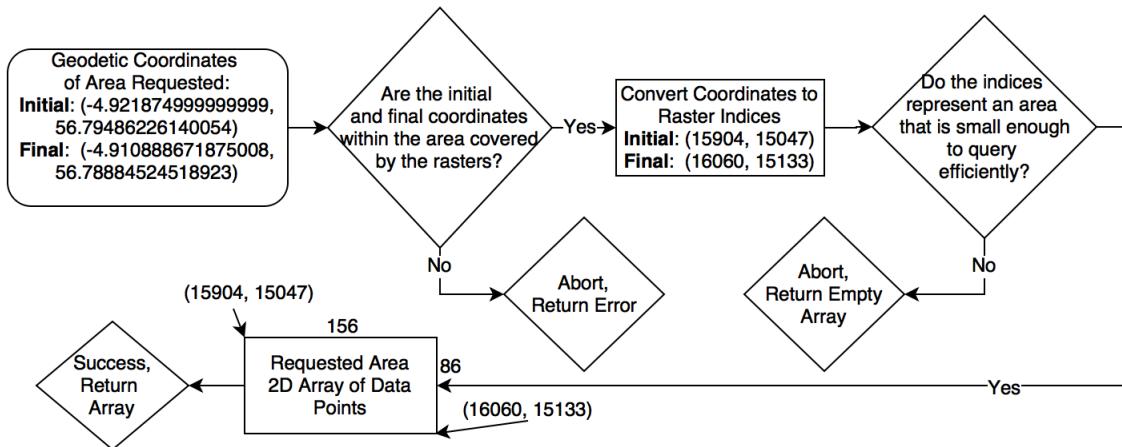


Figure 5.5: A diagram illustrating the retrieval of any raster's data with geodetic coordinates. In order to ensure the continuity of operations, APIs return empty content and silently log errors when an error is encountered, instead of breaking out.

```
/imagery/api/v1.0/avalanche_risks/<string:longitude_initial>/<string:latitude_initial>/<string:longitude_final>/<string:latitude_final>/<string:forecast_date>?  
showStaticRisk=[0,1]
```

Required: `longitude_initial`, `latitude_initial`, `longitude_final`, `latitude_final`.  
Optional: `forecast_date` (default: today's date), `showStaticRisk` (default: 0).

Figure 5.6: The API call format for `avalanche_risks`.

The API call format is shown in Figure 5.6. Note that the parameter format of coordinate components separated by slashes are designed specifically for Cesium's `UrlTemplateImageryProvider` [79] class to automatically retrieve risk level images as tiles.

**terrain\_aspects:** this API works in a similar way as `avalanche_risks`, but instead of retrieving risk level images, it retrieves terrain aspect images encoded by the full spectrum of RGB colours. It was used for debugging aspect fitting and can be potentially used to provide the user with a terrain facing map.

**contours:** this also works in a similar way as `terrain_aspects` and as shown in Figure 5.5. However, it returns general map images as described in Section 5.2.2, rather than risk level images.

The same call format for both `terrain_aspects` and `contours` is shown in Figure 5.7.

```
/imagery/api/v1.0/[terrain_aspects,contours] /<string:longitude_initial>/<string:  
latitude_initial>/<string:longitude_final>/<string:latitude_final>
```

Required: longitude\_initial, latitude\_initial, longitude\_final, latitude\_final.

Figure 5.7: The API call format for both terrain\_aspects and contours.

### 5.3.2 Data APIs

APIs returning JSON datasets on valid calls are categorised as Data APIs. This includes: forecast\_dates, find\_path, and past\_avalanches.

**forecast\_dates:** This simple API returns up to 50 most recent dates when an avalanche forecast has been conducted by the SAIS in an area, defined by a single geodetic point. This is to allow the user to adjust the forecast\_date parameter of the avalanche\_risks API on the UI, in order to view risk model of past days. Its call format is shown in Figure 5.8.

```
/data/api/v1.0/forecast_dates/<string:longitude>/<string:latitude>
```

Required: longitude, latitude.

Figure 5.8: The API call format for forecast\_dates.

**find\_path:** This API interfaces the Path Finder module, which implements risk-adjusted pathfinding as required in **Stage 3**. The methodology and algorithms behind the implementation have been discussed in detail in Section 4.5. The start and end locations for pathfinding are supplied as geodetic points, along with a weighting factor to for risk-distance adjustment. The call format is shown in Figure 5.9.

```
/data/api/v1.0/find_path/<string:longitude_initial>/<string:latitude_initial>/<string:  
longitude_final>/<string:latitude_final>/<string:risk_weighting>/<string:  
forecast_date>
```

Required: all parameters except forecast\_date, defaulting today's date.

Figure 5.9: The API call format for find\_path.

**past\_avalanches:** Another simple API. Given a start and an end date, it returns a list of recorded avalanches at any location between the dates. The user interface will provide the user with a set of quick date range choices, such as the past month or the past year. The returned data for each avalanche is identical to that of Figure 5.4. The call format of this API is shown in Figure 5.10.

## 5.4 The user interface

The user interface of the application is implemented with the Cesium WebGL library, which runs in most modern web browsers, as shown in Figure 5.11. The vast majority of the interface area are left to the virtual 3D scene that the user can navigate freely in, while the main features of the application are accessed via UI elements on the top-left corner of the interface.

From top to bottom, there are five elements for feature access:

**Terrain Picker** - A dropdown menu for switching between different terrain shape sources provided by the self-hosted terrain server.

## 5 Design and Implementation of the Application

/data/api/v1.0/past\_avalanches/<string:start\_date>/<string:end\_date>

Required: start\_date, end\_date (must be in ascending order).

Figure 5.10: The API call format for past\_avalanches.

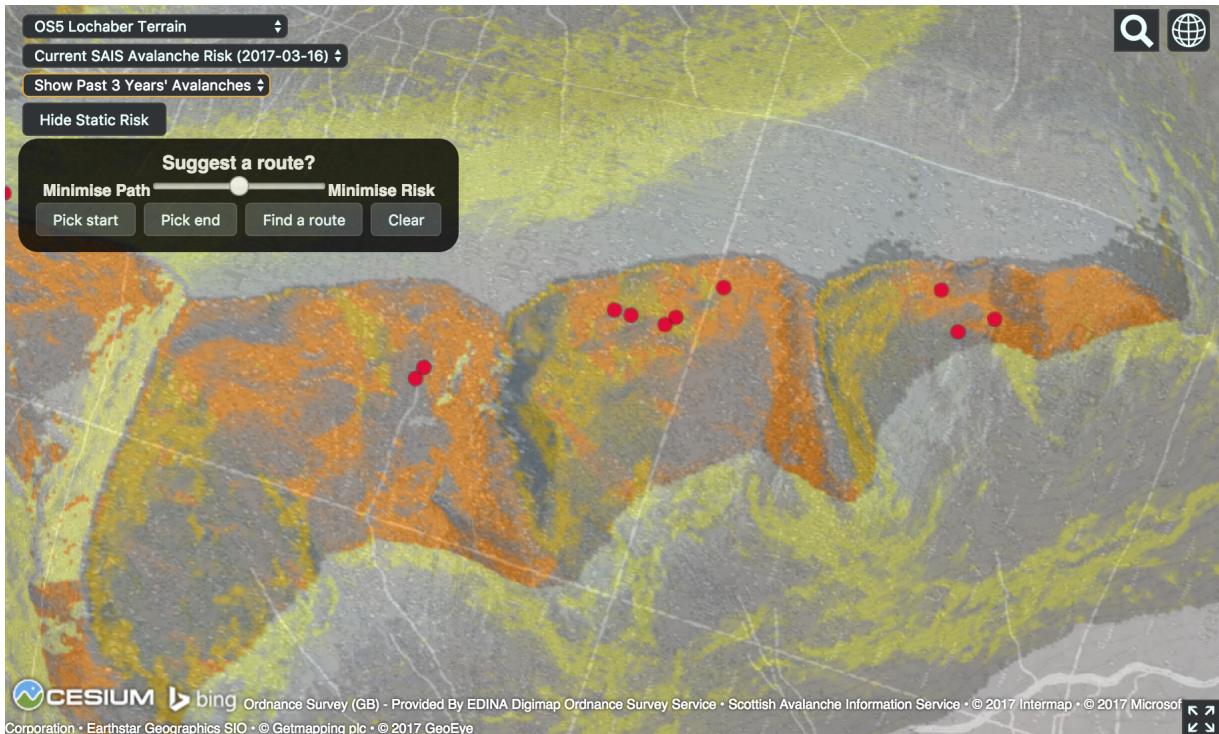


Figure 5.11: The user interface of the final product. The red-dots indicate the locations of past avalanches.

**Past Forecast Picker** - A dropdown menu with up to 50 dates of past avalanche forecasts. The user is able to view risk level images for any of the past 50 forecasts with this feature.

**Past Avalanche Picker** - A dropdown menu allowing the user to choose the date range of past avalanches labeled on the map. The options are: past month, past year, past three years, all recorded avalanches, and show none. The default is to show avalanches recorded in the past month. A dynamic panel displays the time and description of an avalanche when the user clicks on the red-dot labels.

**Risk Model Switcher** - A button allowing the user to switch between the two risk models: the dynamic risk-only model (**Stage 1**), and the comprehensive dynamic and static risk model (**Stage 2**). The text of the button changes to the opposite model when the model is switched.

**Route Selector** - A panel for pathfinding capabilities introduced in **Stage 3**. It contains a slider for adjusting the weighting between minimising risk and distance, two buttons for picking the start and end location, a button for starting route search, and a clearing button.

The interface is programmed in JavaScript, interfacing the APIs provided by Cesium. Extensive use of asynchronous XHTML requests allows seemingless transmission of dropdown options, user inputs and returned data, as well as necessary interface changes and map labelings required by the returned data.

The user interface is designed in conformance with the principles described in Section 2.6. While it is necessary to use a mouse or a touchscreen to interact with the application, the mouse proficiency

#### *5.4 The user interface*

required on the user [62] is no greater than what's required by a standard map application such as Google Maps, as no new types of interactions are programmed. The Cesium library also implements the style of touchscreen interactions recommended by [63]. The application also responded well to rough touchscreen inputs [65] during test uses.

# 6 Evaluation and Testing of the Application

The evaluations conducted on the application attempt two answer two questions: how usable is the application to the user; and how accurate our devised risk model is. Section 6.1 attempts to answer the first question, while Section 6.2 attempts to answer the second.

## 6.1 Evaluation and testing of usability

In usability testing, another experienced mountaineer was invited to use both the traditional map-and-forecast method and the new application to plan routes in the Lochaber area, as if climbing on a day with recorded avalanches. Without knowledge of the recorded avalanches, the user successfully avoided planning a route in the avalanche area with both methods. The user's comments and any associated changes we made are shown in Figure 6.2.



Figure 6.1: Rob Stone, an experienced mountaineer, testing the traditional map-and-forecast method before test-using the application of this project.

## 6.2 Evaluation of the risk model

In order to evaluate the accuracy of the static risk model as described in Section 4.2, a Python script was produced to read in data from modules in Data Interface, and compare the risk model with past avalanche records (nearly 4,000 since 1991) to measure the accuracy of the model.

As our risk model essentially computes the probability of each point being an avalanche release area, a common evaluation strategy for release area estimation models can be used: calculate the accuracy of release area identification at different percentages of areas of study, which produces two important

User Feedback	Our Action (if applicable)
<b>Data and Data Models</b>	
More dynamic data layers will be useful, such as snowfall levels and weather.	Data will have to be specially sourced and updated. For future work due to insufficient time.
Risk of cornice avalanche near ridges not sufficiently considered.	Can be taken into account in the static risk model, difficulties in determining ridges require future work.
Use of crowdsource avalanche records, and risk area labelings by experts could help enhance the risk model.	This is possible if this application can be commercialised.
The data model does not consider vertical effects of risk, e.g. a high release risk area would increase the risk of areas below it.	With a decay-by-distance function, this should be easily computable, but requires adjustment of the decay parameters require repeated computations, which is beyond our current computational capabilities.
<b>The Pathfinder</b>	
Setting the weighting factor to be 100% minimising risk would cause impractical routes to be returned.	The limit on the weighting factor has been adjusted to a maximum of 90% risk in the UI.
Multiple subsequent routes returned should have different colours or labels for easy identification.	This has been implemented, subsequent routes will loop between six possible colours.
Walking on the ridges are usually the least risky routes, but the pathfinder's behaviour is not consistent in this.	Adjustments to the heuristic function and Naismith's Rule factor require significant future work.
The Pathfinder may not be an appropriate feature, as any inappropriate routes returned by the pathfinder would affect the user's confidence in the entire application. Users should probably be encouraged to work out the route based on the risk map of the application and their own experience. However, this functionality can be a very useful training tool.	

Figure 6.2: Comments and suggestions made by the potential user during usability testing, and any associated action we took to improve the application.

metrics: the accuracy itself, and the percentage of area identified as at risk – if this is too large then the accuracy becomes meaningless. Since our risk model produces a continuous value for each point, risk thresholds can be set at different percentiles of all risk values to create definite percentages of identified areas.

Unfortunately, as discussed in Section 5.2.3, the avalanche locations pin-pointed by the SAIS's Avalanche Map are far from exact. Therefore, it was necessary for us to search in a square box around the pined location of each avalanche, with varying extents (starting from 25m).

A histogram of all risk values is first produced to help determine the risk thresholds, as shown in Figure 6.3. Based on the distribution of risk values, the thresholds were selected as each whole number between 70% and 99%, as well as 99.5% and 99.9% to test the nature of extreme risk values. The value of each threshold is calculated, and a point is identified as a potential release area under that threshold if its value is above the threshold's risk value. The results of this evaluation is shown in Figure 6.4 and plotted in Figure 6.5.

Before comparing the accuracy of our model against the performance of release area estimation of other models, it is important to note that percentages values cannot be compared directly due to the differences in size of study areas. Similar to the previous study in the Scottish Highlands by Purves *et al.* [22], our study area is significantly larger than many others, and the lack of precisely recorded avalanches added another layer of complications. Within an extent of 25 meters, covering 0.5% of study area, our model produces a comparable performance (70.12%) to others discussed in Section 2.3, as shown in Figure 6.6.

Despite the difficulties in comparing the identification accuracy of our model against other models directly, for the 0.5% of area of Scottish Highlands identified as most likely avalanche release areas by our model, the positive rate searching within 25 meters is above 70%. This increases to above 90% for the top 5% of areas.

As described in Section 4.4, the saturation of risk colour codings are determined by normalising the

## 6 Evaluation and Testing of the Application

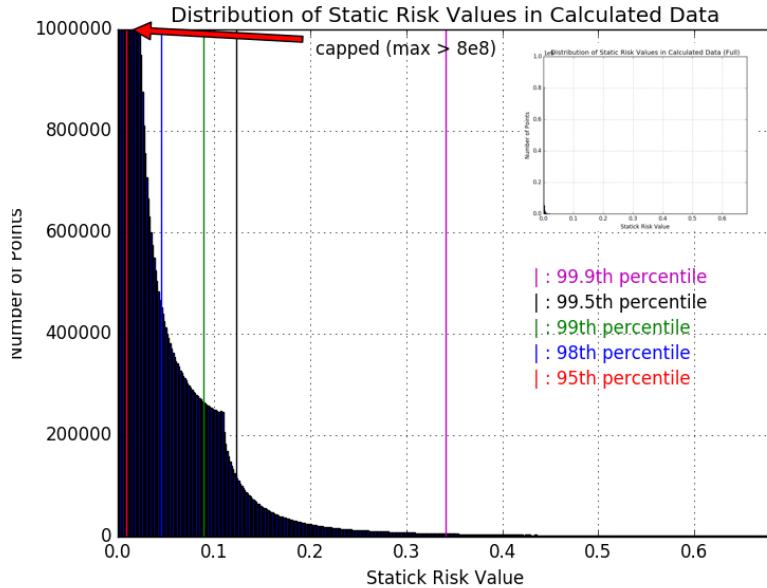


Figure 6.3: A histogram showing the distribution of risk values in the static risk model (1991 to March 16, 2017). The main histogram is capped at a count of 1,000,000 due to the distinct values of high risk points, as shown in the overview diagram at top-right. The total number of points in the full raster is just under 1,000,000,000.

Threshold %	% of study area	Search 25m	Search 50m	Search 125m	Search 250m
70%	30%	<b>98.69%</b>	98.77%	98.79%	98.79%
80%	20%	<b>97.92%</b>	98.67%	98.79%	98.79%
90%	10%	<b>96.20%</b>	98.28%	98.72%	98.79%
95%	5%	<b>93.48%</b>	97.07%	98.56%	98.77%
99.5%	0.5%	<b>70.12%</b>	88.17%	96.48%	98.28%
99.9%	0.1%	<b>27.41%</b>	54.44%	82.60%	94.56%

Figure 6.4: The identification rate of our model's release area estimation at varying search area extents. A distinct trade-off between identification rate and false positive rate can be observed: the lower the risk value threshold, the higher the identification rate, but also consequently higher the false-positive rate. Therefore high identification rates at lower risk thresholds are not necessarily “better”, but a balancing threshold needs to be chosen. This diagram presents a subset of values plotted in Figure 6.5.

risk levels to between the 1st and 99th percentiles of all risk values. Therefore we believe that as long as the proportion of high risk points remain small (Figure 6.3), the application will be effective at visualising high avalanche risk spots in the Scottish mountains, such as shown in Figure 6.7.

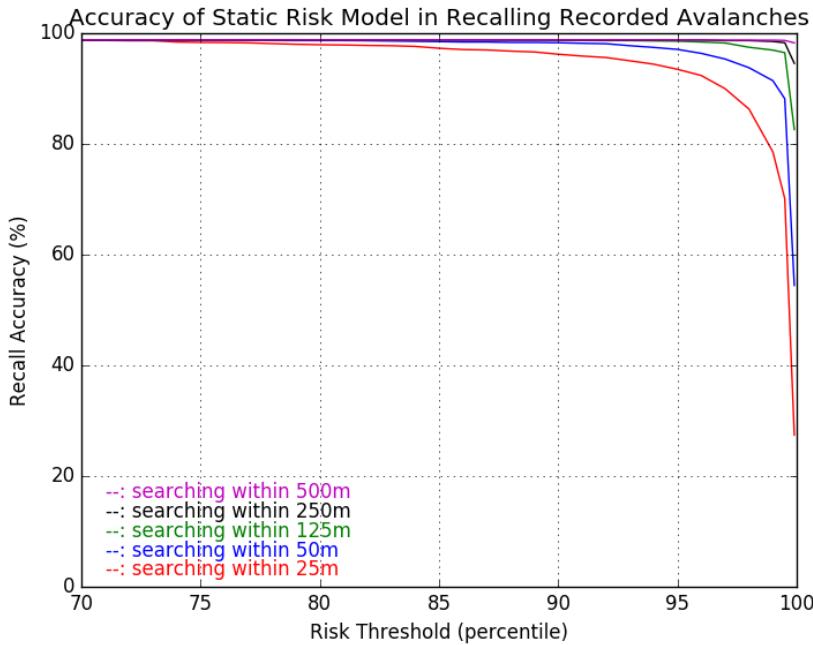


Figure 6.5: A diagram showing the accuracy of our model's release area estimation at different risk threshold percentiles. Note that % of area = 100% - threshold.

Study	Method	Location	Study area	% area identified as risky	Identification accuracy
Ghinoi and Chung [42]	slope, aspect, elevation, crest distance, roughness and curvature	Italian Dolomites	$\approx 230 \text{ km}^2$	7%-20%	67-82%
Bühler <i>et al.</i> [15]	slope, curvature, roughness, land cover	Manali, India	$126 \text{ km}^2$	36%	66.7%
Veitinger <i>et al.</i> [39]	slope, wind shelter, roughness	Zuoz, Switzerland	$\approx 50 \text{ km}^2$	20%	60-80%
Pistocchi and Notarnicola [38]	statistical methods and morphology	Italian Alps	$7,400 \text{ km}^2$	20%	$\approx 70\%$
Purves <i>et al.</i> [22]	statistical methods and dynamic observations	Scottish Highlands	$> 5,000 \text{ km}^2$	Not a static risk model, no localisation.	71%-82% (broad area only)
Our Model	slope, aspect, curvature, roughness and dynamic observations	Scottish Highlands	$> 5,000 \text{ km}^2$	0.5-30%	70.12%-98.69% (details in Figure 6.4)

Figure 6.6: Comparing the accuracy of our model against other release area estimation models, picking the most representative accuracy covering all avalanche releases in other studies. In most cases, the identification accuracy increases as the percentage of area allowed to be identified as risky increases.

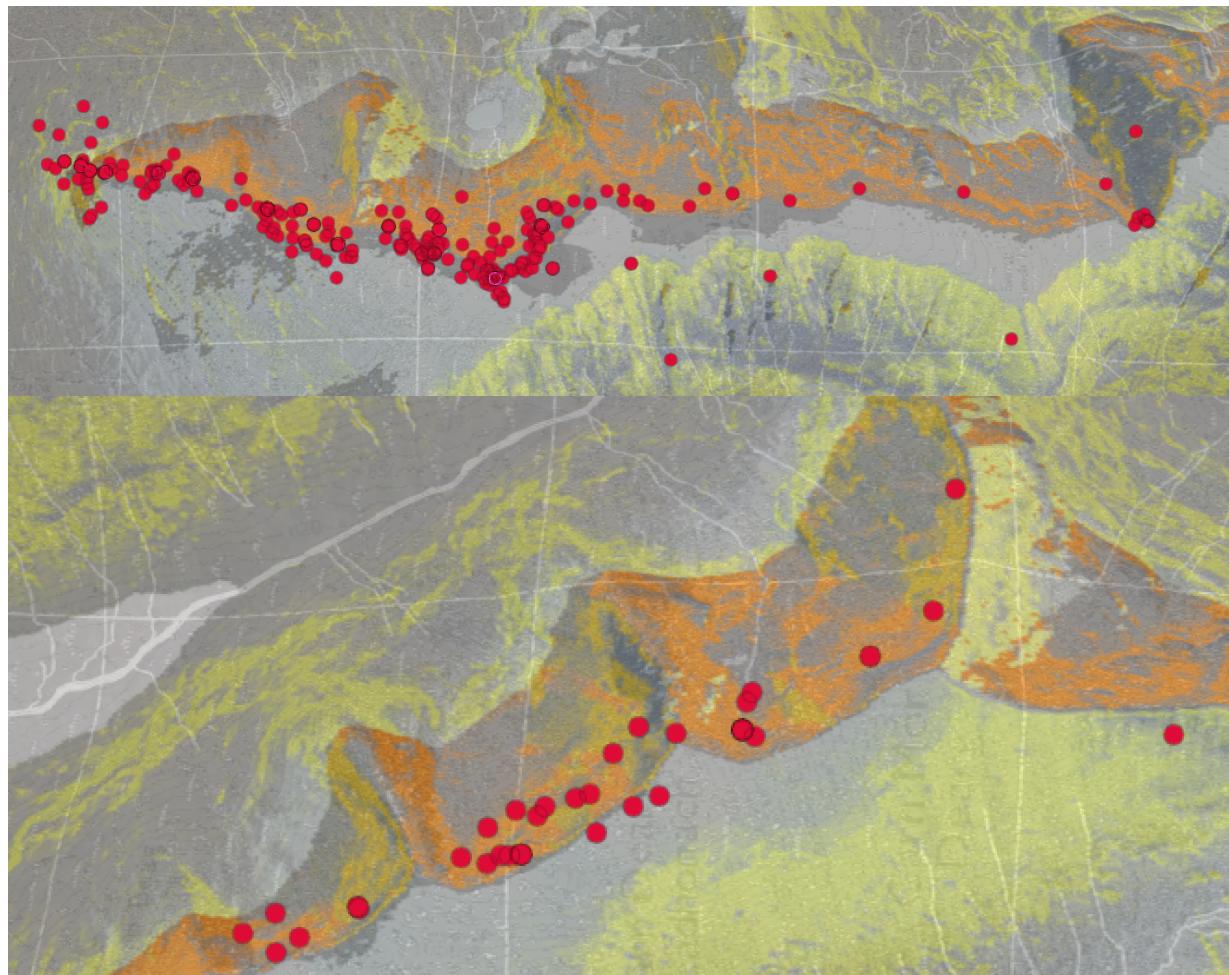


Figure 6.7: Combined risk map of two areas in Ben Nevis, Lochaber, overlaid with past avalanche records since 1991.

# 7 Conclusions

This chapter summarises what has been achieved by the project over its course, as well as potentials for related work to be conducted in the future.

## 7.1 Summary of achievements

This project implemented a new visualisation approach for avalanche hazards, providing a potential solution to the difficulties mountaineers face in correlating avalanche risk forecasts with mountainous terrains. The application follows usability considerations related to its potential uses.

While the dynamic risk model was based on data from the SAIS, a new static risk model was devised and evaluated, as well as methods of combining them in both visualisation and pathfinding. In addition to adaptations of existing topographical analysis techniques, new modelling techniques were devised together with a new risk colour coding. Evaluations demonstrated that our static risk model has better or comparable performance compared to past efforts [15][22][38][39][42] (Figure 6.6), and when combined with the dynamic risk model effectively and accurately visualises avalanche hazards. In the process, we also demonstrated that topographical analysis techniques used for avalanche forecasting in other regions can also be used in the Scottish Highlands with suitable adaptions such as what we made.

The A\* search algorithm has been adapted by this project for mountain terrains in a risk-adjusted environment, providing a method for safety-critical pathfinding in areas with avalanche risk.

In statistical evaluation of the risk model, a threshold-based method for evaluating the risk model against data with poor localisation and coverage was created, which can be adapted for evaluating similar models in lack of reliable test data.

The application was developed with Git version control, and in accordance with software engineering principles. The codebase is well-structured, documented and can be easily redeployed on most Linux environments. It will be made publicly available for future work.

## 7.2 Potential of future work

As mentioned in Section 6.2, the extreme computational requirement of storing thousands of combined risk models has made evaluating a combined static and dynamic risk model difficult. With greater access to computational resources, our risk model can be further evaluated.

Based on user feedback during usability testing, various image overlays useful to mountaineers can be added with appropriate data sources, such as localised weather and snowfall records. There are also potential additions to the risk model, including the vertical effects of high risk areas, and the risk of cornice avalanches.

During test uses, it was found that transferring and displaying large amount of data have slowed down the application, especially on devices with less computing power and slower network connections. In order to improve the usability of the application in the field, future developments could

## *7 Conclusions*

be conducted to migrate the user-end to native code and store pre-downloaded data to improve performance.

# Bibliography

- [1] "Scottish Avalanche Information Service.". [Online] <http://www.sais.gov.uk/> [Accessed Sep 27, 2016].
- [2] Ordnance Survey (GB), "OS Terrain 5 (ASC geospatial data), Scale 1:10000." EDINA Digimap Ordnance Survey Service, July 2016. [Online] <http://digimap.edina.ac.uk/> [Accessed Oct 23, 2016].
- [3] B. Barton and B. Wright, *A Chance in a Million?: Scottish Avalanches*. Scottish Mountaining Trust, 2000.
- [4] Scottish Avalanche Information Service, "Avalanche map.". [Online] [http://www.sais.gov.uk/avalanche\\_map/](http://www.sais.gov.uk/avalanche_map/) [Accessed Nov 16, 2016].
- [5] Scottish Avalanche Information Service, "Avalanche Report for Lochaber, Issued 06/01/2016," January 2016. [Online] [http://www.sais.gov.uk/lochaber/?report\\_id=4753](http://www.sais.gov.uk/lochaber/?report_id=4753) [Accessed Nov 18, 2016].
- [6] F. Rudolf-Miklau *et al.*, *The Technical Avalanche Protection Handbook*. Ernst & Sohn, 2015.
- [7] D. McClung and P. A. Schaefer, *The Avalanche Handbook*. The Mountaineers Books, 2006.
- [8] J. Schweizer and M. Lütschg, "Characteristics of human-triggered avalanches," *Cold Regions Science and Technology*, vol. 33, no. 2, pp. 147–162, 2001.
- [9] UN Education, Scientific and Cultural Organization, *Avalanche atlas: Illustrated international avalanche classification*. the United Nations, 1981.
- [10] C. E. Page *et al.*, "Avalanche deaths in the United States: a 45-year analysis," *Wilderness & Environmental Medicine*, vol. 10, no. 3, pp. 146 – 151, 1999.
- [11] S. Hudson, *Sport and adventure tourism*. Routledge, 2003.
- [12] D. M. McClung, "The elements of applied avalanche forecasting, part ii: The physical issues and the rules of applied avalanche forecasting," *Natural Hazards*, vol. 26, no. 2, pp. 131–146, 2002.
- [13] E. R. Lachapelle, "The fundamental processes in conventional avalanche forecasting," *Journal of Glaciology*, vol. 26, pp. 75–84, 1980.
- [14] Scottish Avalanche Information Service, "Scottish avalanche information service report for winter 2014/15," January 2016. [Online] [http://www.sais.gov.uk/wp-content/uploads/2016/01/SAIS-Winter-Report-2014\\_15.pdf](http://www.sais.gov.uk/wp-content/uploads/2016/01/SAIS-Winter-Report-2014_15.pdf) [Accessed Nov 16, 2016].
- [15] Y. Bühler *et al.*, "Automated identification of potential snow avalanche release areas based on digital elevation models," *Natural Hazards and Earth System Sciences*, vol. 13, no. 5, p. 1321, 2013.
- [16] D. M. McClung, "Computer assisted avalanche forecasting," in *Stochastic and Statistical Methods in Hydrology and Environmental Engineering*, pp. 347–358, Springer, 1994.
- [17] M. Lehning *et al.*, "Snowpack model calculations for avalanche warning based upon a new network of weather and snow stations model calculations for avalanche warning based upon a new network of weather and snow stations," *Cold Regions Science and Technology*, vol. 30, no. 1, pp. 145–157, 1999.
- [18] M. Lehning *et al.*, "An objective snow profile comparison method and its application to SNOWPACK," *Cold Regions Science and Technology*, vol. 33, no. 2–3, pp. 253 – 261, 2001.
- [19] P. Bartelt and M. Lehning, "A physical SNOWPACK model for the swiss avalanche warning: Part i: numerical model," *Cold Regions Science and Technology*, vol. 35, no. 3, pp. 123–145, 2002.
- [20] H. Hirashima *et al.*, "Avalanche forecasting in a heavy snowfall area using the snow model," *Cold Regions Science and Technology*, vol. 51, no. 2–3, pp. 191 – 203, 2008. International Snow Science Workshop (ISSW) 2006.
- [21] O. Buser, "Avalanche forecast with the method of nearest neighbours: an interactive approach," *Cold Regions Science and Technology*, vol. 8, no. 2, pp. 155–163, 1983.
- [22] R. Purves *et al.*, "Nearest neighbours for avalanche forecasting in scotland—development, verification and optimisation of a model," *Cold Regions Science and Technology*, vol. 37, no. 3, pp. 343 – 355, 2003. ISSW 2002: International Snow Science Workshop.
- [23] A. Singh and A. Ganju, "A supplement to nearest-neighbour method for avalanche forecasting," *Cold Regions Science and Technology*, vol. 39, no. 2–3, pp. 105 – 113, 2004.

## Bibliography

- [24] A. Singh *et al.*, "Calibration of nearest neighbors model for avalanche forecasting," *Cold Regions Science and Technology*, vol. 109, pp. 33 – 42, 2015.
- [25] A. Pozdnoukhov *et al.*, "Spatio-temporal avalanche forecasting with Support Vector Machines," *Natural Hazards and Earth System Sciences*, vol. 11, no. 2, pp. 367–382, 2011.
- [26] C. Christophe *et al.*, "Spatio-temporal reconstruction of snow avalanche activity using tree rings: Pierres Jeanne avalanche talus, Massif de l'Oisans, France," *CATENA*, vol. 83, no. 2–3, pp. 107 – 118, 2010.
- [27] K. C. Clarke, "Advances in geographic information systems," *Computers, environment and urban systems*, vol. 10, no. 3-4, pp. 175–184, 1986.
- [28] B. K. P. Horn, "Hill shading and the reflectance map," *Proceedings of the IEEE*, vol. 69, pp. 14–47, Jan 1981.
- [29] M. J. De Smith *et al.*, *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*. Troubador Publishing Ltd, 2007.
- [30] M. F. Wilson *et al.*, "Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope," *Marine Geodesy*, vol. 30, no. 1-2, pp. 3–35, 2007.
- [31] Ordnance Survey (GB), *OS Terrain 5: User guide and technical specification*, v1.1 ed., 07 2013.
- [32] S. Rayburg *et al.*, "A comparison of digital elevation models generated from different data sources," *Geomorphology*, vol. 106, no. 3–4, pp. 261 – 270, 2009.
- [33] R. Ward, "Avalanche prediction in scotland: Ii. development of a predictive model," *Applied Geography*, vol. 4, no. 2, pp. 109 – 133, 1984.
- [34] Y. Durand *et al.*, "A meteorological estimation of relevant parameters for snow models," *Annals of Glaciology*, vol. 18, no. 1, pp. 65–71, 1993.
- [35] J. Veitinger *et al.*, "Slab avalanche release area estimation: a new GIS tool," in *Proceedings of International Snow Science Workshop 2014, Banff (Canada)*, 2014.
- [36] Scottish Avalanche Information Service, *Interpreting Snow Profiles*, 2014.
- [37] F. Cappabianca *et al.*, "Snow avalanche risk assessment and mapping: A new method based on a combination of statistical analysis, avalanche dynamics simulation and empirically-based vulnerability relations integrated in a GIS platform," *Cold Regions Science and Technology*, vol. 54, no. 3, pp. 193 – 205, 2008.
- [38] A. Pistocchi and C. Notarnicola, "Data-driven mapping of avalanche release areas: a case study in South Tyrol, Italy," *Natural Hazards*, vol. 65, no. 3, pp. 1313–1330, 2013.
- [39] J. Veitinger *et al.*, "Potential slab avalanche release area identification from estimated winter terrain: a multi-scale, fuzzy logic approach," *Natural Hazards and Earth System Sciences*, vol. 16, no. 10, p. 2211, 2016.
- [40] D. Straub and A. Grêt-Regamey, "A bayesian probabilistic framework for avalanche modelling based on observations," *Cold Regions Science and Technology*, vol. 46, no. 3, pp. 192 – 203, 2006.
- [41] A. Grêt-Regamey and D. Straub, "Spatially explicit avalanche risk assessment linking bayesian networks to a gis," *Natural Hazards and Earth System Science*, vol. 6, no. 6, pp. 911–926, 2006.
- [42] A. Ghinoi and C.-J. Chung, "STARTER: a statistical GIS-based model for the prediction of snow avalanche susceptibility using terrain features—application to Alta Val Badia, Italian Dolomites," *Geomorphology*, vol. 66, no. 1–4, pp. 305 – 325, 2005.
- [43] J. C. Aerts *et al.*, "Testing popular visualization techniques for representing model uncertainty," *Cartography and Geographic Information Science*, vol. 30, no. 3, pp. 249–261, 2003.
- [44] L. Cheong *et al.*, "Evaluating the impact of visualization of wildfire hazard upon decision-making under uncertainty," *International Journal of Geographical Information Science*, vol. 30, no. 7, pp. 1377–1404, 2016.
- [45] K. Spachinger *et al.*, "Flood risk and flood hazard maps—visualisation of hydrological risks," in *IOP Conference Series: Earth and Environmental Science*, vol. 4, p. 012043, IOP Publishing, 2008.
- [46] Scottish Avalanche Information Service, *Interpreting Avalanche Reports*, Nov. 2014.
- [47] S. Conger, "A review of colour cartography in avalanche danger visualization," in *Proceedings of the International Snow Science Workshop, Jackson Hole, Wyoming, USA*, vol. 477482, 2004.
- [48] C. C. Braun and N. C. Silver, "Interaction of signal word and colour on warning labels: differences in perceived hazard and behavioural compliance," *Ergonomics*, vol. 38, no. 11, pp. 2207–2220, 1995.
- [49] J. Gavaldà and I. Moner, "Mountaineeroriented local avalanche forecasting system in a graphical format in the Aran valley (Eastern Pyrenees)," in *International Snow Science Workshop, Whistler, Canada*, 2008.
- [50] T. Saito *et al.*, "Two-tone pseudo coloring: Compact visualization for one-dimensional data," in *IEEE Symposium on Information Visualization, 2005*, pp. 173–180, IEEE, 2005.

- [51] I. D. Moore *et al.*, "Digital terrain modelling: a review of hydrological, geomorphological, and biological applications," *Hydrological processes*, vol. 5, no. 1, pp. 3–30, 1991.
- [52] P. E. Hart *et al.*, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, July 1968.
- [53] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, pp. 269–271, Dec. 1959.
- [54] X. Cui and H. Shi, "A\*-based pathfinding in modern computer games," *International Journal of Computer Science and Network Security*, vol. 11, no. 1, pp. 125–130, 2011.
- [55] R. E. Korf, "Depth-first iterative-deepening: An optimal admissible tree search," *Artificial intelligence*, vol. 27, no. 1, pp. 97–109, 1985.
- [56] A. Botea *et al.*, "Near optimal hierarchical path-finding," *Journal of game development*, vol. 1, no. 1, pp. 7–28, 2004.
- [57] P. Tozour and I. Austin, "Building a near-optimal navigation mesh," *AI game programming wisdom*, vol. 1, pp. 298–304, 2002.
- [58] Z. Magyari-Sáska and Š. Dombay, "Determining minimum hiking time using DEM," *Geographica Napocensis Anul*, vol. 82, no. 4, pp. 124–129, 2012.
- [59] P. Yap, "Grid-based path-finding," in *Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 44–55, Springer, 2002.
- [60] G. M. Nielson and D. R. Olsen Jr, "Direct manipulation techniques for 3D objects using 2D locator devices," in *Proceedings of the 1986 workshop on Interactive 3D graphics*, pp. 175–182, ACM, 1987.
- [61] M. Haklay, *Interacting with geospatial technologies*. Wiley Online Library, 2010.
- [62] E. Dubois *et al.*, "Evaluating advanced interaction techniques for navigating google earth," in *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...But Not As We Know It - Volume 2*, BCS-HCI '07, (Swindon, UK), pp. 31–34, BCS Learning & Development Ltd., 2007.
- [63] C.-J. Ku and L.-C. Chen, "A study on the natural manipulation of multi-touch gestures for 3D object rotation using a large touch screen," in *Proceedings of the International Conference on Universal Design, UD 2014 (Lund, Sweden)*, vol. 35, p. 279, IOS Press, 2014.
- [64] J. Pascoe *et al.*, "Using while moving: HCI issues in fieldwork environments," *ACM Transactions on Computer-Human Interaction*, vol. 7, no. 3, pp. 417–437, 2000.
- [65] P.-A. Albinsson and S. Zhai, "High precision touch screen interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '03*, (New York, NY, USA), pp. 105–112, ACM, 2003.
- [66] Cesium Team, "Cesium API Reference.". [Online] <http://cesiumjs.org/> [Accessed Nov 16, 2016].
- [67] J. Heierli *et al.*, "Verification of nearest-neighbours interpretations in avalanche forecasting," *Annals of Glaciology*, vol. 38, no. 1, pp. 84–88, 2004.
- [68] L. W. Zevenbergen and C. R. Thorne, "Quantitative analysis of land surface topography," *Earth surface processes and landforms*, vol. 12, no. 1, pp. 47–56, 1987.
- [69] R. G. Heerdegen and M. A. Beran, "Quantifying source areas through land surface curvature and shape," *Journal of Hydrology*, vol. 57, no. 3-4, pp. 359–373, 1982.
- [70] I. Evans, M. Young, and J. Gill, "An integrated system of terrain analysis and slope mapping, final report," *Univ. of Durham, Durham, NC*, 1979.
- [71] A. Kimberling *et al.*, "Map use: reading, analysis, interpretation: Seventh edition," 2013.
- [72] J. M. Sappington *et al.*, "Quantifying landscape ruggedness for animal habitat analysis: a case study using bighorn sheep in the mojave desert," *Journal of Wildlife Management*, vol. 71, no. 5, pp. 1419–1426, 2007.
- [73] A. V. Goldberg and C. Harrelson, "Computing the shortest path: A search meets graph theory," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 156–165, Society for Industrial and Applied Mathematics, 2005.
- [74] P. Scarf, "A mathematical excursion in the isochronic hills," *Mathematics Today*, vol. 44, pp. 163–167, 2008.
- [75] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.
- [76] European Commission, "Digital Elevation Model over Europe (EU-DEM)," June 2015. [Online] <http://www.eea.europa.eu/data-and-maps/data/eu-dem> [Accessed Oct 23, 2016].
- [77] M. Ruth *et al.*, "Geotiff format specification," Dec. 2000.

## *Bibliography*

- [78] "WGS84: World Geodetic System 1984," June 2004.
- [79] Cesium Team, Analytical Graphics, Inc., "UrlTemplateImageryProvider - Cesium Documentation." [Online] <https://cesiumjs.org/Cesium/Build/Documentation/UrlTemplateImageryProvider.html> [Accessed Oct 23, 2016].