

Università degli Studi di Trento  
Facoltà di Scienze Matematiche, Fisiche e Naturali



Corso di Laurea in Informatica

---

Final Thesis

An Open Source implementation of a particle-based  
model for skiing dynamics

Relatore interno:

**Prof. Alberto Montresor**

Relatore esterno:

**Dott. Cesare Furlanello**

Laureando:

**Matteo Poletti**

Anno accademico 2012-2013



# Contents

<b>Introduction</b>	<b>5</b>
<b>1 The model</b>	<b>7</b>
1.1 Social forces . . . . .	7
1.2 Physical forces . . . . .	10
1.3 Limitations . . . . .	14
1.4 Differences from the original model . . . . .	14
<b>2 Implementation</b>	<b>19</b>
2.1 Technology choice . . . . .	19
2.2 Software structure . . . . .	21
2.3 Input data . . . . .	23
2.4 Output data . . . . .	25
2.5 Parameters Selection . . . . .	26
2.6 Optimization . . . . .	28



# Introduction



# Chapter 1

## The model

The model implemented can be classified as a two-dimensional microscopic-driven many-particle system with the constraint that skiers are exposed to gravity and centripetal forces. It was published by Holleczeck and Troster [4].

Skiers are modeled as particles with a specific mass,  $m$ , that are exposed to two class of forces:

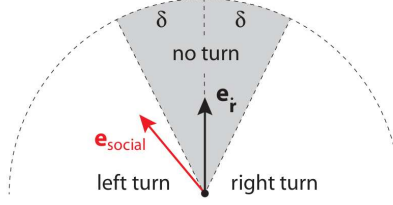
1. Social forces, that determine skiers behavior.
2. Physical forces, that regulate the skier motion determining the acceleration.

In the following, the position of a skier at time  $t$  is represented by the vector  $r(t)$ ,  $\dot{r}(t) = \frac{d}{dt}r(t)$  is the vector speed, and  $e_{\dot{r}}(t) = \dot{r}(t)/\|\dot{r}(t)\|$  the direction of motion.

### 1.1 Social forces

The social forces implemented in the model describe the decisions taken by a skier while descending a slope. Social forces are dimensionless and are used to determine whether the skier should take a turn or not, however they do not act on acceleration. The superposition of all social forces,  $F_{social}$ , gives skier's desired direction  $e_{social}(t) = F_{social}/\|F_{social}\|$ . If the desired direction  $e_{social}(t)$  diverges from the actual skier direction of motion  $e_{\dot{r}}(t)$  more than an angle  $\delta$ , the skier starts to turn adjusting the direction (see Fig.1.1). Social forces are used to model the repulsion of the skier from slope edges, potential obstacles and from other skiers and to attract skiers towards the destination chosen.

To describe the social force that attracts the skier towards the chosen destination, the model assumes that each skier, while descending, selects



**Figure 1.1:** (From [4]) When the angle between the current direction of motion  $e_r$  and the desired direction  $e_{social}$  is bigger than  $\delta$  the skier performs a turn to adjust his or her direction

several waypoints  $x_a^1, \dots, x_a^n$  as temporary destinations. Thus, at each time  $t$  the skier  $a$  wants to reach a waypoint  $x_a^k$ . The direction toward the current waypoint is expressed by

$$e_a(t) = \frac{x_a^k - r_a(t)}{\|x_a^k - r_a(t)\|} \quad (1.1)$$

where, as defined above,  $r_a(t)$  is the position of  $a$  at time  $t$ . The destination social force drives the skier toward the waypoint and is defined as

$$F_D(r_a) = A_0 e_a(t) \quad (1.2)$$

where  $A_0$  is a scaling constant that represents the strength of the destination force.

The attitude of skiers to keep a minimum distance from the edges is modeled with repulsion forces that are stronger when the skier gets closer to the edge of the slope. At each position  $r_a$  the skier  $a$  is subjected to a repulsion force from the left edge of the slope and to a repulsion force from the right edge of the slope. Let  $r_a^L$  be the closest location to  $r_a$  on the left edge, then the distance between the skier and the edge can be expressed as  $r_{aL} = r_a - r_A^L$ . The repulsion force from the left edge is defined as

$$F_L(r_{aL}) = -\nabla_{r_{aL}} U(\|r_{aL}\|) \quad (1.3)$$

where  $U(\|r_{aL}\|)$  is a monotonically decreasing potential. In a symmetric way the repulsion force from the right edge can be defined as

$$F_R(r_{aR}) = -\nabla_{r_{aR}} U(\|r_{aR}\|) \quad (1.4)$$

The model takes into account also the natural human behavior of avoiding collisions with other skiers. This is described by a repulsion force, referred



as skier repulsion force, that each skier imposes on the other skiers. The repulsion force that skier  $b$  imposes on the skier  $a$  can be expressed as

$$F_S(r_{ab}) = -\nabla_{r_{ab}} V(s(r_{ab})) \quad (1.5)$$

where  $r_{ab} = r_a - r_b$  is the distance vector between the two skiers,  $V(s(r_{ab}))$  is a monotonically decreasing potential with equipotential lines shaped as ellipses directed into the direction of motion and  $s$  represents the semiminor axis of this ellipse and is defined as

$$s(r_{ab}) = \frac{\sqrt{(\|r_{ab}\| + \|r_{ab} - v_b \Delta t e_b\|)^2 - (v_b \Delta t)^2}}{2} \quad (1.6)$$

Finally a repulsion force from the obstacles on the slope is considered. The force that an obstacle  $o$  imposes on the skier  $a$  is defined as

$$F_O(r_{ao}) = -\nabla_{r_{ao}} W(\|r_{ao}\|) \quad (1.7)$$

In general, repulsion social forces act on skiers only if he or she is capable of perceiving what triggers the force. The model assumes that objects are perceived only within a certain range  $\varphi$  of the skier direction.  $2\varphi$  can be considered as the angle of view. This is modeled by the weight

$$w(u, v) = \begin{cases} 1 & \text{if } (u/\|u\|) \cdot (v/\|v\|) \geq \cos\varphi \\ 0 & \text{otherwise} \end{cases} \quad (1.8)$$

To summarize, social forces modeled for every skier  $a$  are

$$F_D(r_a) = A_0 e_a(t), \quad (1.9)$$

$$F_L(\dot{r}_a, r_{aL}) = w(\dot{r}_a, -r_{aL}) F_L(r_{aL}), \quad (1.10)$$

$$F_R(\dot{r}_a, r_{aR}) = w(\dot{r}_a, -r_{aR}) F_R(r_{aR}), \quad (1.11)$$

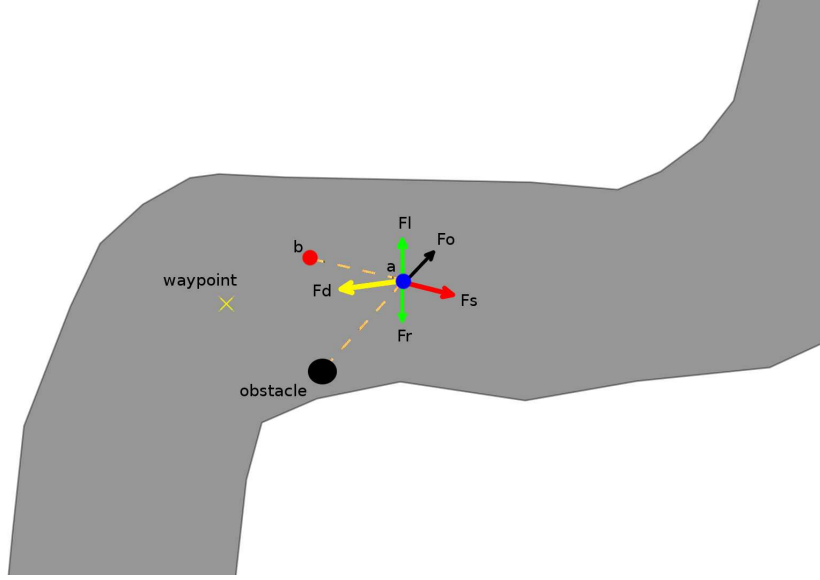
$$F_A(\dot{r}_a, r_{ab}) = w(\dot{r}_a, -r_{ab}) F_A(r_{ab}), \quad (1.12)$$

$$F_O(\dot{r}_a, r_{ao}) = w(\dot{r}_a, -r_{ao}) F_O(r_{ao}) \quad (1.13)$$

The resultant social force  $F_{social}^a$  is the superposition of all social forces that apply on skier  $a$ :

$$F_{social}^a = F_D(r_a) + F_L(\dot{r}_a, r_{aL}) + F_R(\dot{r}_a, r_{aR}) + \sum_b F_A(\dot{r}_a, r_{ab}) + \sum_o F_O(\dot{r}_a, r_{ao})$$

Figure 1.2 shows a diagram of the social forces described above.



**Figure 1.2:** Diagram of the social forces. The forces  $F_R$  and  $F_L$  keep the skier  $a$  away from the edges, the force  $F_O$  repels the skier from the obstacle, the force  $F_S$  repels  $a$  from the skier  $b$  and the force  $F_D$  attracts the skier towards the waypoint.

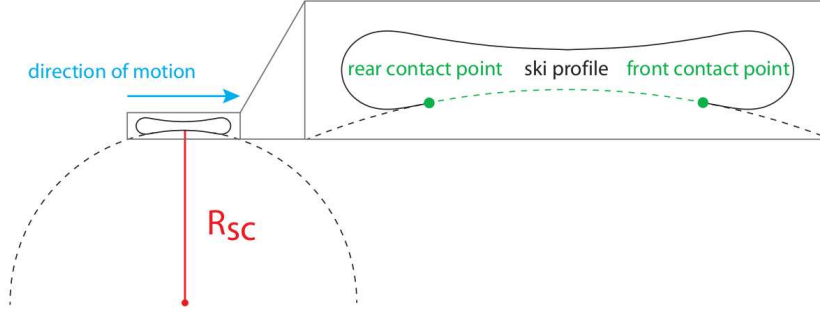
## 1.2 Physical forces

As described in [4], there are two major techniques used to curve while skiing: *skidding* and *carving*. When carving, the direction of motion is exclusively parallel to the skis while in skidding there is an additional slippage to the side. Carved turns are usually performed by expert skiers, while beginners and not experienced skiers tend to perform skidded turns.

In [4] skiers are supposed to perform turns with a radius corresponding to the *sidecut radius* of their skis. Although some studies [5] [2] have proposed a more realistic model of carving turns, deeply investigating the effects of ski penetration in the snow and of the skier tilt angle, as a first version of the model the approximation of the turning radius to the sidecut radius has been considered acceptable. Figure 1.3 shows the relation between sidecut radius and turning radius.

Gravitational, centripetal and friction forces determine the skier acceleration according to their direction  $e_{\hat{r}}$ . Consider a skier at position  $r$  with speed  $\dot{r}$  and direction of motion  $e_{\hat{r}}$  and let  $n$  denote the surface normal on the ski slope at  $r$ . At  $r$ , the slope has an inclination angle of

$$\alpha = \arccos([0, 0, 1] \cdot n) \quad (1.14)$$



**Figure 1.3:** (from [4]) Profile of a carving sking with the sidecut radius and the turning radius evidenced.

and the inclination angle  $\gamma$  of the current trajectory  $e_{\dot{r}}$  is

$$\gamma = \arcsin[(\sin \alpha)(\sin \beta)] \quad (1.15)$$

where  $\beta$  is the angle between  $e_{\dot{r}}$  and the horizontal of the slope.

To compute the force accelerating the skier the gravitational force, the friction forces and the centripetal forces should be investigated. As first the gravitational force  $F_G$  is considered. It can be expressed as

$$F_G = mg \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad (1.16)$$

where  $g$  is the gravitational acceleration and  $m$  the mass of the skier. The gravitational force can be subdivided into normal force  $F_N$ , acting parallel to the surface normal  $n$ , and into downhill force  $F_S$ , acting parallel to the fall line.

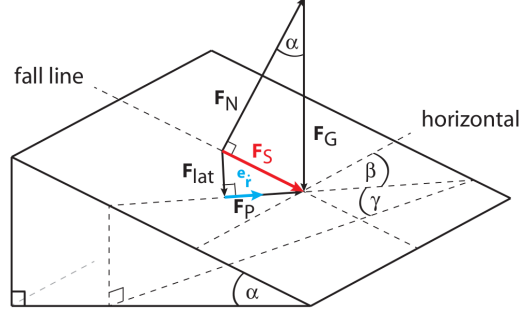
$$F_G = F_S - F_N \quad (1.17)$$

The normal force  $F_N$  can be expressed as

$$F_N = mg(\cos \alpha)n \quad (1.18)$$

The downhill force  $F_S$  itself can be subdivided into  $F_P$  which is acting parallel to the current trajectory  $e_{\dot{r}}$  and into the lateral force  $F_{lat}$  acting perpendicularly to the current trajectory (see. Fig1.4).

$$F_S = F_P + F_{lat} \quad (1.19)$$



**Figure 1.4:** (from [4]) The downhill force  $F_S$  can be decomposed into the downhill force  $F_P$ , acting parallel to the current trajectory, and into the lateral force  $F_{lat}$  acting perpendicularly to the direction of travel.

The downhill force  $F_P$  can be written as

$$F_P = mg(\sin \gamma)e_{\dot{r}} = mg(\sin \alpha)(\sin \beta)e_{\dot{r}} \quad (1.20)$$

where  $\gamma$  is the inclination angle of  $e_{\dot{r}}$ .

Remembering 1.17, the downhill force  $F_S$  can be computed as

$$F_S = F_G + F_N \quad (1.21)$$

The lateral force  $F_{lat}$  can therefore be computed as

$$F_{lat} = F_S - F_P \quad (1.22)$$

The centripetal force  $F_C$  a skier is exposed while turning can be written as

$$F_C = \frac{m}{R_{SC}} \|\dot{r}\|^2 \frac{F_{lat}}{\|F_{lat}\|} \times \begin{cases} (+1) & \text{before crossing the fall line} \\ (-1) & \text{after crossing the fall line} \end{cases} \quad (1.23)$$

where  $m$  is the mass of the skier and  $R_{SC}$  the sidecut radius of the skis.  $F_C$  is parallel to  $F_{lat}$  before the skier crosses the fall line and antiparallel to  $F_{lat}$  after having crossed the fall line.

Before defining the kinetic friction of skis on snow, a definition of effective force should be given. The effective force is the force that needs to be compensated by the snow. Its formulation depends on whether the skier is performing a turn or is descending on a straight line. In the following, when the definition of a force changes depending on whether the skier is turning, the index is written lowercase in the case of a straight line and uppercase

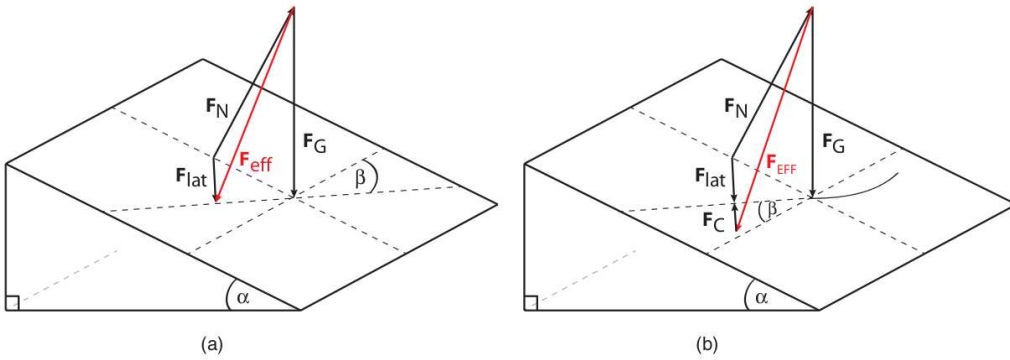
in the case of a turn. So  $F_{eff}$  is the effective force acting on a skier that is descending on a straight line and is defined as

$$F_{eff} = F_{lat} - F_N \quad (1.24)$$

If the skier is performing a carved turns then the effective force  $F_{EFF}$  can be written as

$$F_{EFF} = F_{lat} - F_C - F_N \quad (1.25)$$

Figure 1.5 shows the effective force ( $F_{eff}$  and  $F_{EFF}$ ).



**Figure 1.5:** (from [4]) In (a) effective force during the descent on a straight line ( $F_{eff} = F_{lat} - F_N$ ). In (b) effective force during a carved turn ( $F_{EFF} = F_{lat} - F_N - F_C$ )

The kinetic friction force  $F_{ground}$  can be expressed in terms of the skier's effective force as

$$F_{ground} = -\mu \|F_{eff}\| e_{\dot{r}} \quad (1.26)$$

when descending on a straight line.  $\mu$  is the kinetic friction coefficient of the skis on the snow. In case of a turn

$$F_{GROUND} = -\mu \|F_{EFF}\| e_{\dot{r}} \quad (1.27)$$

The air drag force  $F_{air}$  is antiparallel to the direction of motion  $e_{\dot{r}}$  and is defined as

$$F_{air} = -\frac{1}{2} C_d \rho A \|\dot{r}\|^2 e_{\dot{r}} \quad (1.28)$$

where  $C_d$  is the drag coefficient,  $\rho$  the air density and  $A$  the projected frontal area of the skier perpendicular to the direction of motion.

Finally, the net force  $F_{net}$  accelerating the skier can be defined as

$$F_{net} = -F_P + F_{air} + F_{ground} \quad (1.29)$$

if the skier is not turning. Otherwise the force is defined as

$$F_{NET} = -F_P + F_{AIR} + F_{GROUND} + F_C \quad (1.30)$$

### 1.3 Limitations

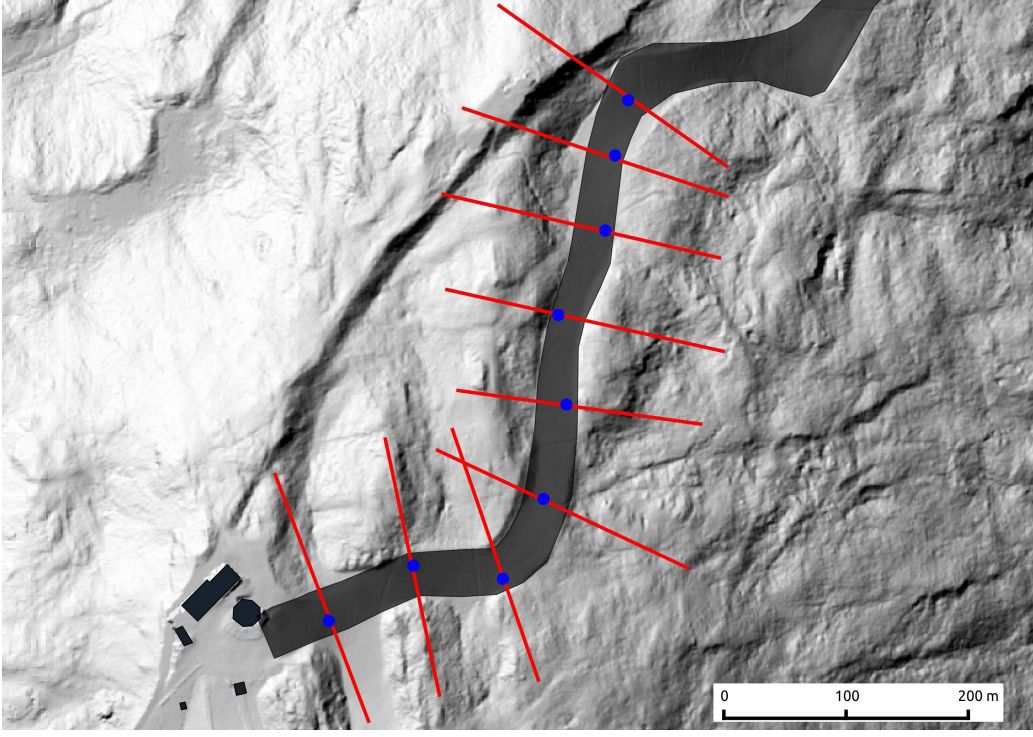
The model proposed describes the motion of an expert skier that skies performing perfect carved turns and the turning radio is taken constant and equals to the sidecut radius of the skis. Not experienced skiers and other snow-sport athletes are not considered. An important limitation of the model is that it does not consider that skiers could stop while descending a slope. Moreover skiers are not allowed to jump nor to exit the ski slope. If a skier collides with an edge of the slope it is reflected back with an angle equals to the angle at which he or she has collided.

### 1.4 Differences from the original model

The physical model of the skiers motion has been keep equals to the one exposed in [4]. Within the social forces, the destination force  $F_D$  (see. 1.2) is of critical importance. The effectiveness of this destination force is strongly related to the waypoints selection. In the original paper the waypoints were selected randomly every 50m, using a uniform distribution on the corresponding line from the left to the right edge of the slope (see Fig.1.6).

A more dynamic approach in the waypoints selection has been considered to better model the decisions taken from a skier while descending a slope. The new strategy allow each skier to dynamically choose waypoints during the descent, based on the position of the skier, on the shape of the trail, on the slope of the terrain and on the skier speed.

Hereafter the new mechanism for the waypoints selection is explained. Let  $a$  be a skier at position  $r_a$ , let  $r_a^L$  be the location on the left slope edge closest to  $r_a$  and  $r_a^R$  the location on the right slope edge closest to  $r$ . The vectors given the direction toward the edges are  $e_{aR} = (r_a^L - r_a) / \|r_a^L - r_a\|$  and  $e_{aL} = (r_a^R - r_a) / \|r_a^R - r_a\|$ . Let  $\alpha$  be the angle between  $e_{aR}$  and  $e_{aL}$  defined as



**Figure 1.6:** In [4] waypoints were selected randomly every 50m following a uniform distribution on the line from the left to the right edges of the slope.

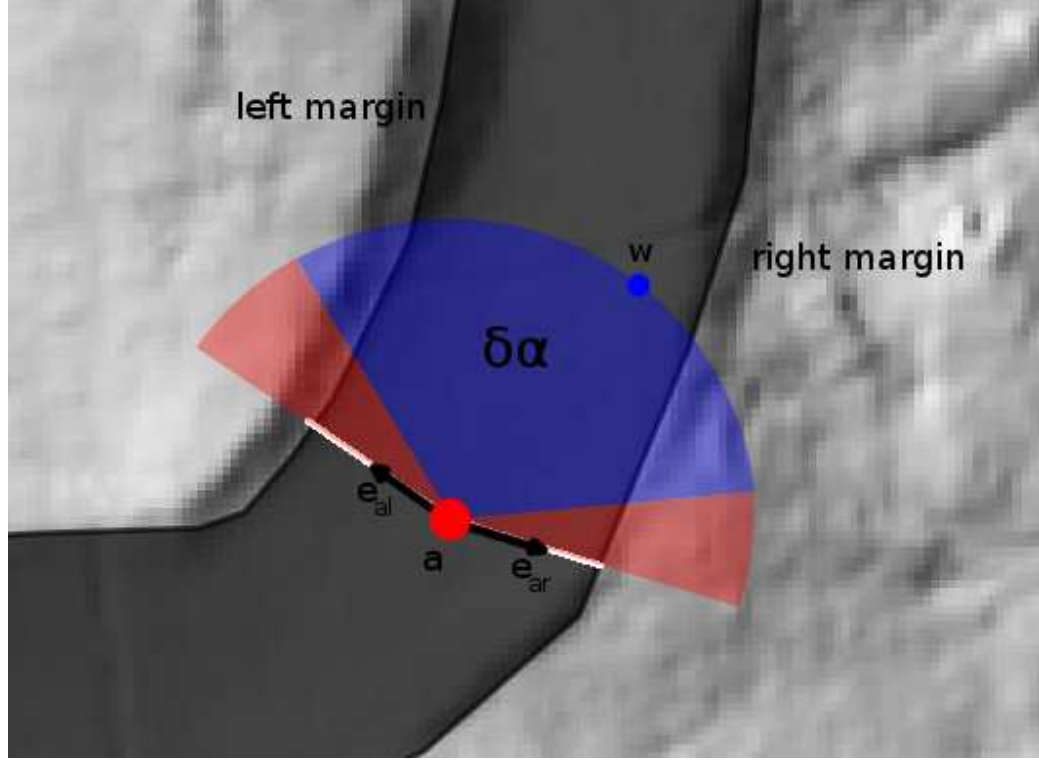
$$\alpha = \begin{cases} \arccos(e_{aR} \cdot e_{aL}) & \text{if } ((e_{aR} \times e_{aL}) \cdot n \geq 0) \\ 2\pi - \arccos(e_{aR} \cdot e_{aL}) & \text{if } ((e_{aR} \times e_{aL}) \cdot n < 0) \end{cases} \quad (1.31)$$

where  $n$  is the normal of the plane containing  $e_{aR}$  and  $e_{aL}$ . The skier  $a$  chooses the new waypoint with uniform distribution on a fraction of the angle  $\alpha$ . More precisely, let  $\delta$  be the fraction of angle that should be considered, then the new waypoint  $w_a$  is chosen as

$$w_a = r_a + \rho F\left(\frac{e_{aR} + e_{aL}}{\|e_{aR} + e_{aL}\|}, \mathcal{U}\left(-\frac{\alpha\delta}{2}, \frac{\alpha\delta}{2}\right)\right) \quad (1.32)$$

where  $\rho$  is the distance at which waypoint are chosen,  $F(v, \beta)$  rotates the vector  $v$  of an angle  $\beta$  and  $\mathcal{U}(a, b)$  returns a random number with uniform distribution on  $(a, b)$  (see Fig.1.7).

A new waypoint is selected when the old waypoint is no longer feasible, meaning that it is not in the interval that the skier would consider when choosing a new waypoint, or when the skier has traveled more than  $D$  meters



**Figure 1.7:** Selection of a waypoint: the skier  $a$  selects the new waypoint  $w$  choosing with uniform distribution on the angle  $\delta\alpha$ , a fraction of  $\alpha$ . The angle  $\alpha$  is the angle between  $e_{aR}$ , the vector representing the direction towards the right edge, and  $e_{aL}$ , the vector towards the left edge,

after having chosen the last waypoint.

The new approach for waypoints selection considers closely skier speed and the effect of the slope shape on skier's action:

1. When skiers are faster they tend to turn more often to decrease the speed.
2. Frequency of the selection of new waypoints should depend on the skiers speed, the more skiers are traveling fast, the more frequently they will choose new waypoints.
3. The selection of a new waypoint should depend on the slope that the skier will face in his range of action: before flat areas skiers tend to go straight to increase velocity.



4. Skiers usually avoid to choose a direction that would make them turn on the edge of the slope. Therefore, the new waypoint should be in a position that does not lead the skier to impact the edge of the slope.

To model the influences of skier's speed and shape of the slope it is possible to act on the parameter  $\delta$  of the equation 1.32. When  $\delta$  is increasing, the width of the angle in which new waypoints can be chosen becomes larger. As a consequence the probability of performing turns becomes higher.

Point 1 can be satisfied by making the parameter  $\delta$  linearly depend on skier's speed  $v$ . Moreover it is required that when the speed  $v$  is close to 0 the width of the angle should be also close to 0 and when  $v$  is close to a high value of speed  $v_{max}$  the angle should have maximum width. Therefore,  $\delta$  can be set to

$$\delta = \frac{v}{v_{max}} \quad (1.33)$$

The requirement 2 is already met by the mechanism described above. In fact, since a new waypoint is chosen each time the skier has traveled more than  $D$  meters, when a skier is faster he or she chooses waypoints more frequently.

To fulfill the third requirement an additional factor depending on the slope should be considered. Let  $s$  be the slope that the skier is going to encounter and let  $s_{lim}$  a value of slope that is considered small enough to require an additional acceleration by the skier. Then if  $s < s_{lim}$  the width of the angle in which to choose the new waypoint should be narrowed again. Taking into account 1.33 then we can write  $\delta$  as

$$\delta = \begin{cases} \frac{v}{v_{max}} \frac{s}{s_{lim}} & \text{if } (s < s_{lim}) \\ \frac{v}{v_{max}} & \text{otherwise} \end{cases} \quad (1.34)$$

If, despite this, skiers will come to a complete stop (maybe due to a counter slope), they will start walking at constant speed.

Finally, to satisfy point 4 manipulating  $\delta$  will not be enough. The minimum turning radius of skiers gives a bound to their ability of avoiding a collision or to hit the edges of the ski slope. Assuming that a skier do not choose a direction that will make them exit the ski slope, those angles indicating a direction along which the slope edge is reached in less than  $R_{SC}$  meters are not considered in the selection of the waypoint.



# Chapter 2

## Implementation

The model was implemented using C++ and GRASS GIS. GRASS GIS is a Geographic Information System (GIS) open source software used for geospatial data management and analysis. It was chosen to perform the geospatial computation needed by the model.

### 2.1 Technology choice

Different technologies were explored to implement the GIS Backend needed by the model. PostGIS, QGIS, GDAL/OGR and GRASS GIS were the main technologies considered.

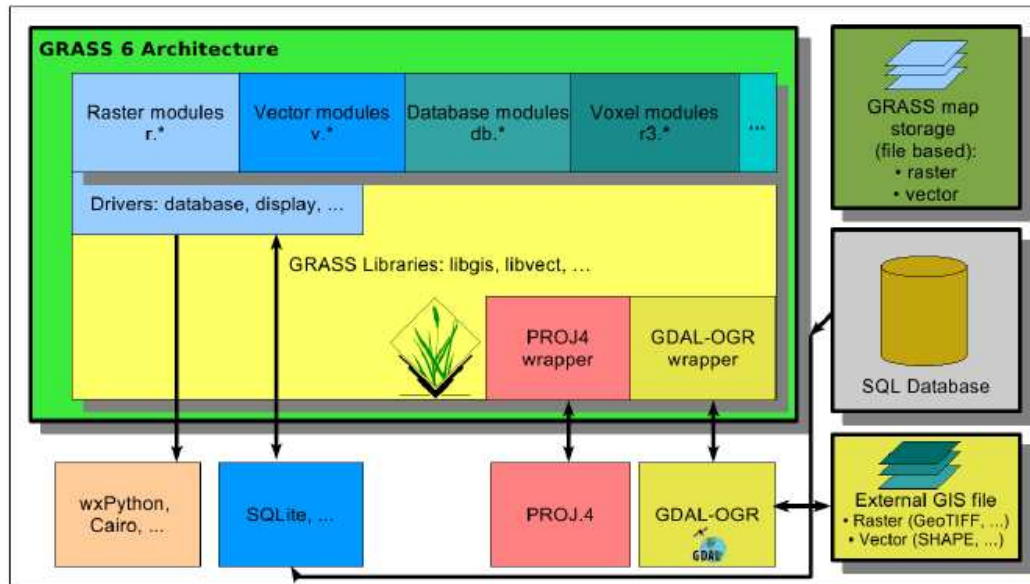
PostGIS is an open source spatial database extender for PostgreSQL database. It gives support for geographic objects manipulation allowing spatial queries to be run in SQL. The critical point using PostGIS is the absence of a dedicated API for any language. It requires to use a PostgreSQL adapter (such as psycopg for python) and to build dynamically the queries in the chosen language. Moreover the raster data support is still immature.

Quantum GIS Desktop, also known as QGIS, is an open source GIS desktop application. QGIS supports both vector and raster formats. It have a module, PyQGIS, that supports scripting using the Python language. However, the functionalities implemented are more focused on spatial data visualization and do not give enough support for the analysis operations.

GDAL, Geospatial Data Abstraction Library, is an open source library for geospatial data translation and processing. The related library OGR Simple Feature Library provides a similar capability for simple vector data processing. GDAL exposes API for C, C++ and Python while the more widely used API for OGR is the one for C++. OGR provides also slightly less complete API for C and Python, although they are not really well documented. The

main issue using GDAL/OGR is the lack of support for the analysis of vector data: OGR offers useful API to read and write vector data, but analysis such as point linestring distance calculation are not available.

GRASS GIS, commonly referred to as GRASS (Geographic Resource Analysis Support System) is an open source GIS software that gives support for geospatial data management and analysis. To understand how GRASS API works it is necessary to look at the GRASS software structure (see Fig.2.1. GRASS has a large GIS library, referred to as GRASS GIS Library, at the basis of the software stack. It is divided in two main library: the Raster File Processing library and the Vector File Processing library. It contains also many others library of less importance. The GRASS GIS Library is quite large and implements many basic GIS operation both for raster and vector format. The spatial data are stored in the GRASS map storage with the GRASS specific format. On the top of the GRASS GIS Library are built many modules for raster and vector processing. The modules perform analysis at a higher level, they are executables and can be used directly from the GRASS GUI or from the command line.



**Figure 2.1:** (from <http://grass.osgeo.org>)  
GRASS software architecture

GRASS allows two basic levels of programming. The most simple approach is to use script programming to call the high-level GRASS modules. A more advanced approach is to access the low-level functionalities of the GRASS GIS library through the C-API exposed. Since grass modules are

executables, the first approach requires to spawn a new process each time a spatial computation is needed. For the implementation of the model presented in 1 this is not feasible. Even if the second approach forces to use GIS operations at a lower lever it has many advantages: it gives support for database routines (GRASS file management), for projections, for raster and vector data management and it guarantees good performances [8].

GRASS GIS was chosen to implement the GIS Backend. Since the API is written in ANSI C, writing the code in C or in C++ has the advantage of accessing the API in a very simple manner. Moreover both C and C++ guarantee good performances and are often chosen to implement this kind of simulation. C++ was preferred to C because it is object oriented. An object oriented design allows to better represent the concepts presented in the model, reducing the complexity and making the code structure clearer. Moreover it keeps the code structure more modular and flexible.

## 2.2 Software structure

The software structure was designed to be the most flexible as possible. The main entities modeled inside the software are the slope, the skiers and the forces. In Figure 2.2 the class diagram of the software is described. The class Slope models the ski slope, it has a set of skiers, the skiers that are descending the slope, a set of physical forces, the physical forces that act on the slope and a set of social forces, the social forces skiers are subjected to. The Skier class models a skier: it has a position, a velocity, an acceleration and a status describing if he is turning. The forces have been modeled with two different classes: the SocialForce class and the PhysicalForce class are abstract classes that act as base class respectively for all the social forces and for all the physical forces. Although the interface declared by this two classes is the same, they were not merged together because the two type of forces, social and physical, are conceptually different and in future development it is possible that changes in the model will require to have separated interfaces.

The simulation has to perform some specific GIS computation. The implementation realized wanted to avoid dependencies between the code related to the simulation and the code written to implement GIS specific operations. In other words the implementation of the simulation should be independent by the technology chosen to implement the GIS Backend. For this reason, an abstract class called GisBackend was used. This class declares the interface that a GIS backend used to run the simulation must implement. This interface includes methods to get elevation, slope and aspect for given coordinates, methods to get the distance of a point from the edges of the slope,

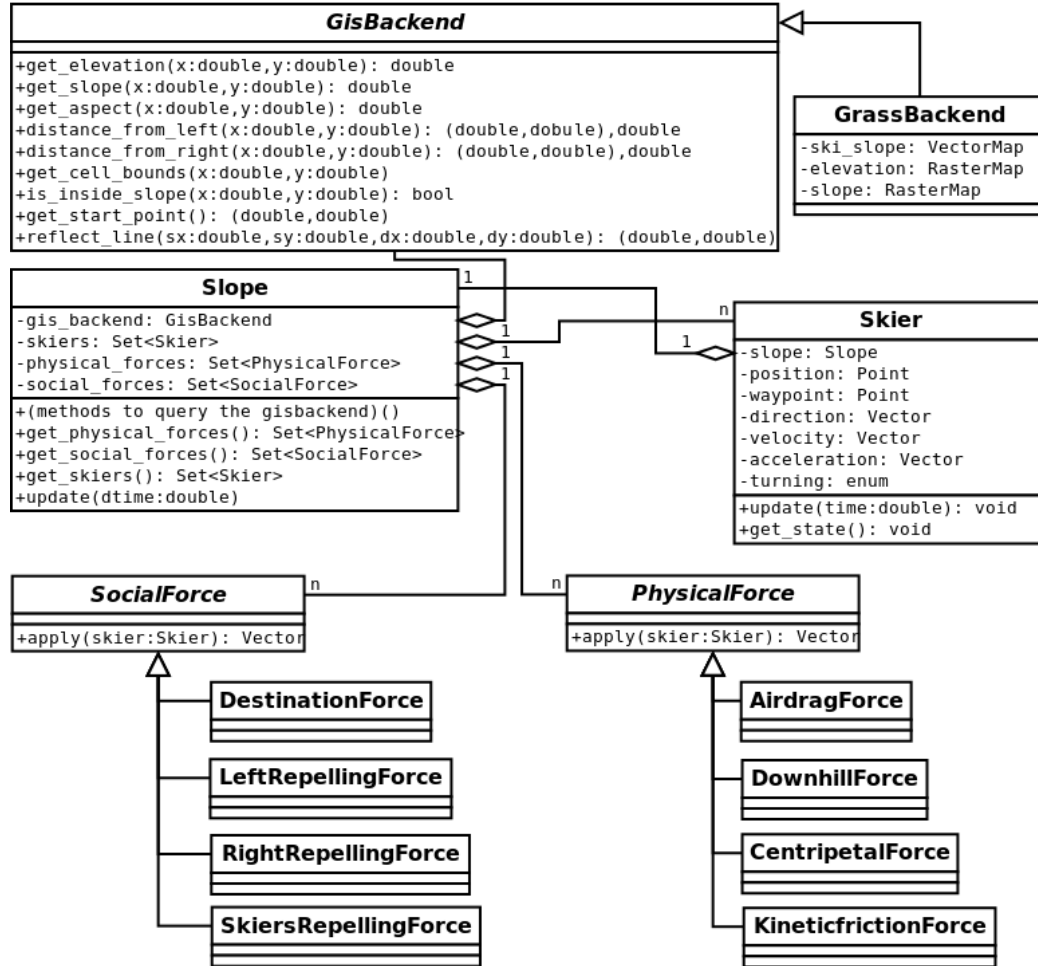


Figure 2.2: Class diagram

methods to give random start points at the top of the ski slope, to determine if a given point is inside the ski slope and a method to reflect a line colliding on a slope edge.

The class `GrassBackend` implements the interface `GisBackend` and is the class used in the code when spatial computations are needed. The class `GrassBackend` performs the operations needed using the GRASS GIS Library. It requires to have available a DEM (Digital Elevation Model) raster map for the elevation and the slope and aspect computation, a vector map with the polygon of the ski slope to determine if a point is inside the slope, a vector map with the lines of the right and left edges of the slope to find the distances from them and finally a vector map with the polygon of the start area and stop area to decide where skiers should be started and stopped.

## 2.3 Input data

The input data required by the simulation to run are the data needed by the GrassBackend. The Digital Elevation Model used was a Digital Terrain Model (DTM) of the area of Trentino. The DTM was obtained using the LIDAR technology, a remote sensing technology that uses laser light to measure distances. The measurements were done in the years 2006 and 2007. It is at a resolution of 1x1m except for areas at high elevation where resolution is of 2x2m.

The measurements for the DTM were done in seasons without snow. As a consequence the DTM on which real skiers move is different from the DTM obtained with the Lidar. Two strategies were thought to simulate the effect of the snow on the DTM.

The first strategy uses a simple model to simulate the distribution of the snow. The model considers two limit cases to describe the profile of the snow. Given a surface  $y(x)$  and a snowfall of  $h$  meters, the first case describes a new profile where the snow is supposed to adhere perfectly to the terrain:  $y_s(x) = y(x) + h$ . The second case considers the snow to behave like a liquid and creates a new profile  $y_L(x)$  where the new precipitation accumulates in the valleys, making the profile constant, and leaves  $y_L(x) = y(x)$  on the ridges. Neither the first nor the second case can be considered realistic, however it can be thought that the real behavior of the snow can be described as an average between the two cases. So the actual profile after the precipitation can be described as

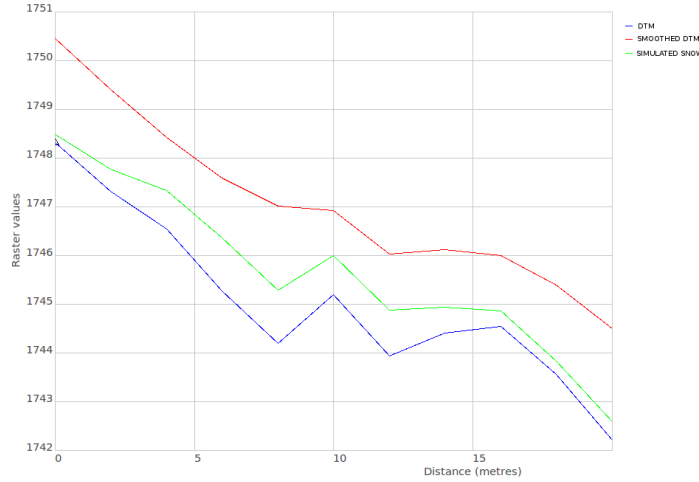
$$Y(x) = (1 - l)y_s(x) + ly_L \quad (2.1)$$

where  $l$  is a parameter describing how much the snow has a liquid-like behavior. By the point of view of the implementation, the critical point is to compute the profile  $y_L$ . For this purpose some GRASS modules were explored: `r.watershed`, `r.terraflow`, `r.sim.water`. The first two were excluded as they compute the flow accumulation rather than the water accumulation. The third module, `r.sim.water`, is an overland flow hydrologic simulation based on path sampling method [7]. This module was used to compute the profile  $y_L$ .

The second strategy does not aim at building a realistic model for the snow precipitation but starts from the consideration that there are three main factors that determine the distribution of the snow on a ski slope: the snow precipitation, the snow produced by the snow cannons and the actions of the snow groomings. All these factors tend to smooth the original surface. To emulate this action the surface of the DTM was approximated with a smoothed surface. Starting from the original DTM raster map a vector map

was produced where each original value of elevation were represented by a point. Then the GRASS module `v.surf.rst` was used to interpolate a new elevation map using regularized spline with tension [6].

Figure 2.3 shows the profiles of the original dtm, of the dtm obtained from the first strategy and of the dtm obtained from the second for a particularly critical point on a ski slope. The best result is given by the second strategy. The problem following the first strategy is the imprecision of the results from the module `r.sim.water` which does not return a feasible profile for a liquid-like behavior. This can be due to a misconfiguration of the parameters of the module or maybe to a too demanding level of precision. Finally, the second strategy was chosen.



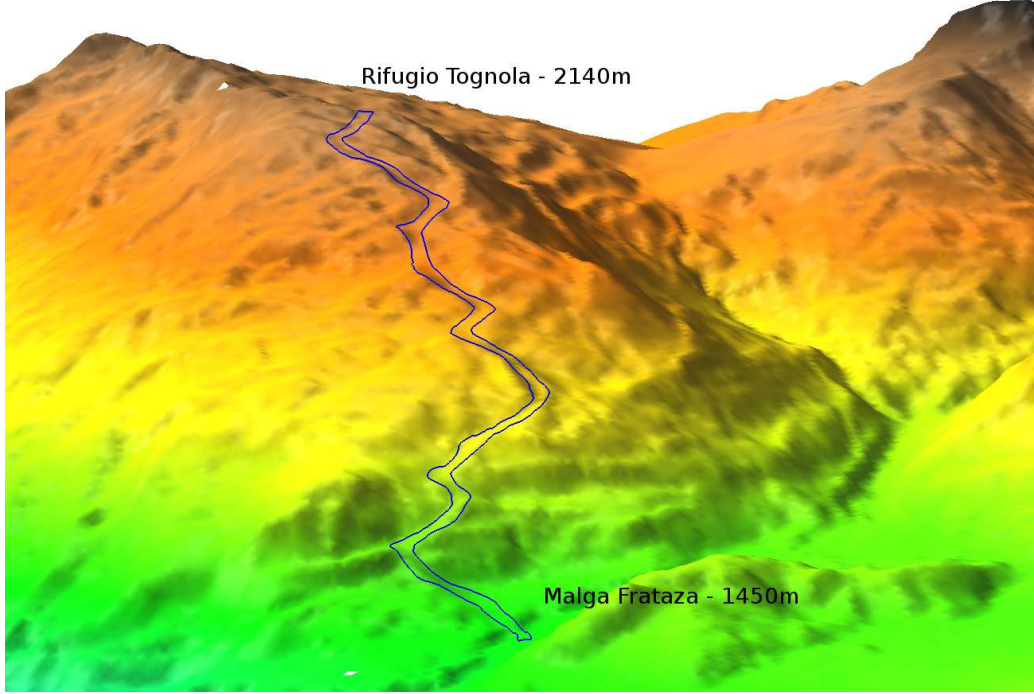
**Figure 2.3:** Profiles of the dtm and of the simulated dtm after rain fall

Starting from the DTM slope raster map and aspect raster map were computed. The GRASS module `r.slope.aspect` was used. The slope raster map contains the slope, stated in degree of inclination from the horizontal. The slope of a cell is the maximum rate of change in value from that cell to its neighbors. Conceptually, computing the slope is equivalent to fits a plane to the z-value of a 3x3 neighborhood around the center cell. The slope value of this plane is computed using the average maximum technique(see. [1]). The direction the plan faces is the aspect.

The simulation requires as input the vector map with the polygon of the ski slope for which the model should be run. The ski slope (see. Fig.2.4 for an example) is extracted from a vector map containing the polygons of slopes of the skiing resort in Trentino. This data are available inside the SicurSkiWeb project and were derived starting from orthophotos and tuned



with measurements done in place. From the polygon of the ski slope left and right edge, start and stop areas are derived.



**Figure 2.4:** 3d visualization of the ski slope “Uno di Tognola” polygon (in blue) in San Martino di Castrozza.

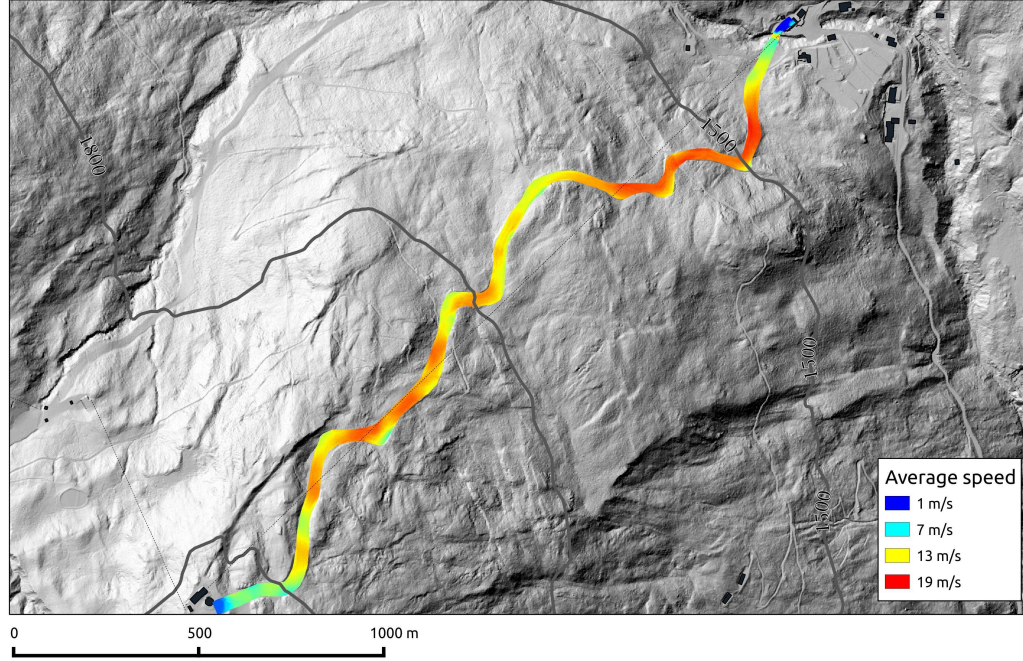
## 2.4 Output data

The output produced by the simulation is the status of each skier at each integration step. The information logged at a time  $t$  for a skier  $a$  are the position  $r_a$  and the velocity  $\dot{r}_a$ .

Starting from this data two main analysis have been performed: speed average and average density of skiers computation. To analyze the speed of skiers along the ski slope a raster map of average speed has been produced. The map is obtained interpolating the speed values simulated by the simulation (see fig.2.5). The interpolation was done using the GRASS module `v.surf.idw`. The GRASS module `v.kernel` was used to compute the density

of skiers on the slope. The raster density map is computed using a moving kernel with gaussian distribution (see fig.2.6).

**Average simulated speed "Uno di Tognola", San Martino di Castrozza**



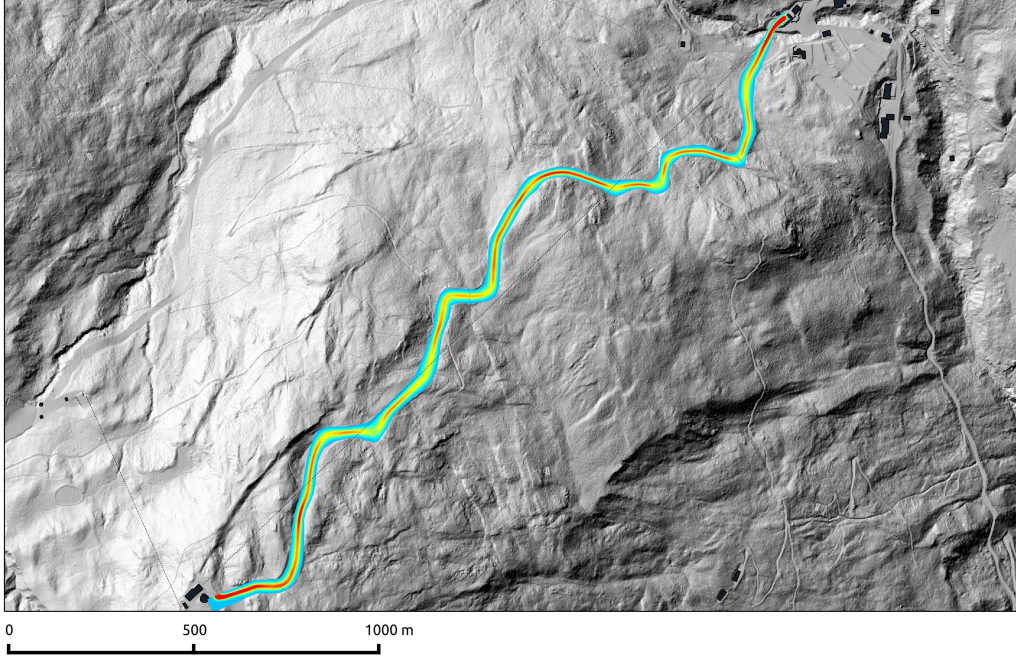
**Figure 2.5:** Average simulated speed map for the “Uno di Tognola” ski slope in San Martino di Castrozza.

## 2.5 Parameters Selection

The parameters of the model are of three types: parameters describing the ski slope, parameters regulating the social forces and parameters for the physical forces.

The model assumes that skiers arrive at the beginning of the ski slope at a constant rate. Let  $\lambda$  be the persons per hour, then the time between the start of two consecutive skiers is  $T = \mathcal{U}\left(\frac{3600}{\lambda}\right)$ , where  $\mathcal{U}(a)$  is the uniform distribution between 0 and  $a$ . For the ski slope “Uno di Tognola” an arrival rate of  $\lambda = 200 \text{ persons/h}$  was chosen.

As in [4] and in [3], the repulsive potentials defining the social forces are

**Skiers density map "Uno di Tognola", San Martino di Castrozza****Figure 2.6:** Simulated skiers density map for the “Uno di Tognola” ski slope in San Martino di Castrozza.

exponential

$$U(\|r_{aL}\|) = U_0 \exp(-\|r_{aL}\|/R_L) \quad (2.2)$$

$$U(\|r_{aR}\|) = U_0 \exp(-\|r_{aR}\|/R_R) \quad (2.3)$$

$$V(s) = V_0 \exp(-s/R_A) \quad (2.4)$$

where  $R_L$ ,  $R_R$  and  $R_A$  denote the ranges of the social repulsions, and  $U_0$  and  $V_0$  are scaling constants that regulate the interaction strength between skiers and the respective object.

The parameters regulating the social forces were set to have forces with equivalent strength and realistic interaction. For the destination force was set  $A_0 = 1$ , for the repulsion forces  $U_0 = 10$ ,  $V_0 = 8$ ,  $R_L = R_R = 10$  and  $R_A = 8$ . As a consequence the magnitude of the destination force is always equals to 1 and the edge repulsion force tend to 1 when the skier approximates the edges. The angle of view is set to  $2\varphi = 180$ . Skiers adjust their direction performing turns when the angle between their direction of motion and the desired direction exceed the angle  $\delta = 10$ , as described in Fig.1.1. The values selected for the social parameters are summarized in table 2.1.

Parameter	Symbol	Value
Strength of destination force	$A_0$	1
Strength of edge repulsion	$U_0$	10
Strength of athlete repulsion	$V_0$	8
Range of edge repulsion	$R_L = R_R$	10
Range of athlete repulsion	$R_A$	8
Angle of view	$2\varphi$	180
Directional deviation	$\delta$	10

**Table 2.1:** Summary table for parameters regulating social forces

## 2.6 Optimization

# Bibliography

- [1] P. A. Burrough and R. A. Mcdonnell. *Principles of Geographical Information Systems*. Oxford University Press, USA, 2 edition, Mar. 1998.
- [2] P. A. Federolf, M. Roos, A. Lüthi, and J. Dual. Finite Element Simulation of the Ski-Snow Interaction of an Alpine Ski in a Carved Turn. *Sports Engineering*, 12:123–133, 2010.
- [3] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5):4282–4286, May 1995.
- [4] T. Holleczeck and G. Tröster. Particle-based model for skiing traffic. *Physical Review E*, 85(5):056101, May 2012.
- [5] U. D. Jentschura and F. Fahrbach. Physics of Skiing: The Ideal-Carving Equation and its Applications. *Canadian Journal of Physics*, 82:249–261, 2004.
- [6] H. Mitasova, L. Mitas, and R. Harmon. Simultaneous spline approximation and topographic analysis for lidar elevation data in open-source gis. *Geoscience and Remote Sensing Letters, IEEE*, 2(4):375–379, 2005.
- [7] H. Mitasova, C. Thaxton, J. Hofierka, R. McLaughlin, A. Moore, and L. Mitas. Path sampling method for modeling overland water flow, sediment transport, and short term terrain evolution in open source {GIS}. In C. T. Miller and G. F. Pinder, editors, *Computational Methods in Water Resources: Volume 2 Proceedings of the XVth International Conference on Computational Methods in Water Resources*, volume 55, Part 2 of *Developments in Water Science*, pages 1479 – 1490. Elsevier, 2004.
- [8] M. Neteler and H. Mitasova. *Open Source GIS: A GRASS GIS Approach*. Springer, 3rd edition, 2008.