

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

Summary

The goal of this thesis work is to implement a computational model of skiers at high spatial resolution based on the agent-based approach. The model simulates the trajectories of skiers descending a ski slope, with different applications of interest for skiing safety analysis as well as skiers behavior profiling. In particular the first applicative goal of this thesis is to contribute to a general predictive model for skiing accidents prevention by providing estimates of traffic indicators along the ski slopes. Specifically, the computational model aims to provide average speed and density of skiers.

The thesis work was developed within the context of the SicurSkiWeb project, which is a new project of the Bruno Kessler Foundation in collaboration with the Polizia di Stato and the Association of skiing resorts of Trentino. SicurSkiWeb offers a GeoICT management platform collecting all data about rescue interventions by ski patrols, with the goal of providing a predictive platform to increase safe skiing.

The model implemented is a particle-based model proposed by Holleczeck and Troster [5]. Skiers are modeled as particles with a mass subjected to physical and social forces, which define their riding behavior during the descents on ski slopes. Physical forces determine the acceleration of skiers. Social forces are used to individuate the skiers desired direction, namely the direction in which a skier is inclined to travel, that is used to decide whether skiers need to adjust their direction performing a turn. The model presented by Holleczeck and Troster was limited to experienced skiers only. The major change introduced in this thesis is the modeling of other classes of skiers, such as beginners and intermediates. Other changes are related to the implementation of the social forces: skiers choices related to their trajectories were made dynamically depend on the status of the skiers and on the condition of the slope.

An adaptation of the model was additionally realized in order to simulate ski lessons. In these patterns agents have strongly correlated trajectories: the first agent simulates the instructor and decides the trajectory autonomously, the following agents are forced to emulate the instructor trajectory.

The implementation was realized in C++ using the GRASS GIS C-API for the spatial computations. GRASS GIS is an Open Source Geographic Information System (GIS) software used for geospatial data management and analysis. A new GRASS module has been developed.

The model has been tested by using a set of real data collected in February and March 2013. The mobile application SkiLogger, developed inside the SicurSkiWeb project, was used to collect data of real skiers. SkiLogger can be used by skiers to track their trajectories by GPS sensors, now usually available also on common smartphones. The application allowed to track a total of 171 Km of descents on ski slopes by 18 distinct skiers. An initial calibration of the model was based on the data collected. A comparison between real data and simulated data was performed in terms of speed values. The average error for the tracks considered ranges between 4.28 m/s and 6.44 m/s, while recorded speeds range from 0 to 30 m/s. The model has been then modified to include possible stops along the ski slope. A new experimental stop parameter was introduced, obtaining a 55% reduction of the average error made for a specific track.

Figure 1 shows the recorded speeds of a track on the ski slope “Uno di Tognola” of San Martino di Castrozza in Trentino (Italy) and compares them to the speeds predicted by the simulation.

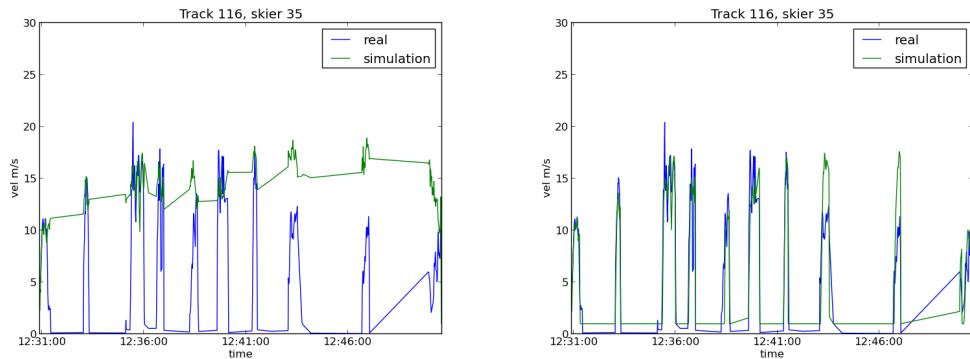


Figure 1: Comparison between speed values recorded for a track and average simulated speeds. Figure on the left shows simulated speed values without stops. Figure on the right shows speed values with forced stops.

Chapter 1 describes the model and defines the physical and social forces. Chapter 2 presents the technologies used and gives details related to the model implementations and the parameters selection. Chapter 3 presents the data collected and the comparisons with the simulated data.

Acknowledgements

This thesis work was developed in the MPBA unit of the Bruno Kessler Foundation. I would like to thank Prof. Alberto Montresor, who has supervised me during this work. This thesis would not have been possible without the help and support of all the MPBA unit, in particular I would like to thank Cesare Furlanello, for having guided and advised me during this period, and Riccardo De Filippi, who helped me in writing the thesis. A big thank goes to Andrea Gobbi, for his patience, his competence and for the many useful discussion we had during this months. Finally, I thank my family for supporting me throughout all my studies at University.

Contents

Introduction	5
1 The model	7
1.1 Social forces	7
1.2 Physical forces	10
1.3 Limitations	14
1.4 Differences from the original model	14
2 Implementation	19
2.1 Technology choice	19
2.2 Software structure	21
2.3 Input data	22
2.4 Output data	25
2.5 Parameters Selection	26
2.6 Optimization	27
3 Steps to validate the model	29
3.1 Collecting data	29
3.2 Simulated vs. real data	32
Conclusions	35

Introduction

Alpine skiing is a winter sport widely practiced in the Alps and it is a key resource for the tourism sector. More than 80.000.000 of

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 [5].

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 δ , 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

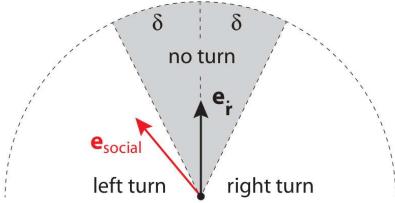


Figure 1.1: (From [5]) When the angle between the current direction of motion e_r and the desired direction e_{social} is bigger than δ the skier performs a turn to adjust his or her direction.

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 repulsion forces from the left and right edges 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 φ of the skier direction. 2φ 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 acting on 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.

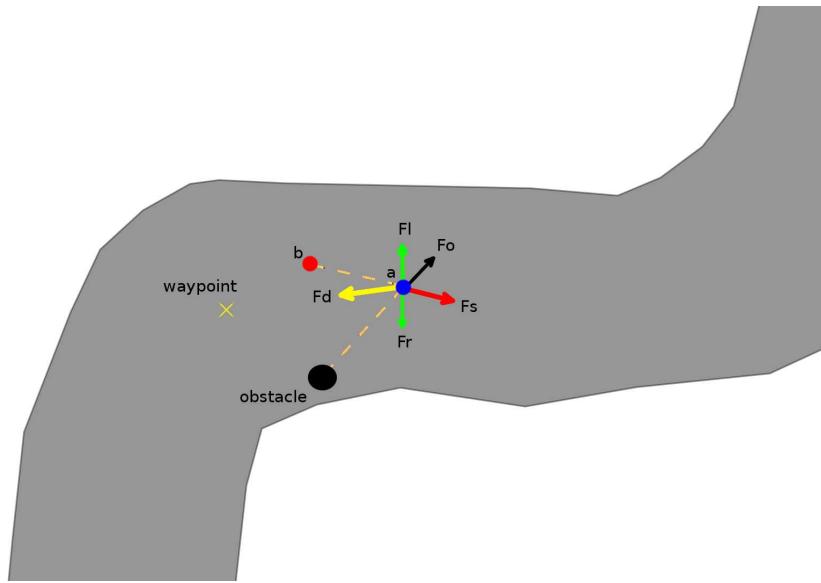


Figure 1.2: Diagram of the social forces. The forces F_R and F_L keep the skier a away from the edges, the force F_O repels the skier from the obstacle, the force F_S repels a from the skier b and the force F_D attracts the skier towards the waypoint.

1.2 Physical forces

As described in [5], 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 non-experienced skiers tend to perform skidded turns.

In [5] skiers are supposed to perform turns with a radius corresponding to the *sidecut radius* of their skis. Although some studies [6] [3] 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_r . Consider a skier at position r with speed \dot{r} and direction of motion e_r and let n denote the surface normal on the ski slope at r . At r , the slope has an inclination angle of

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

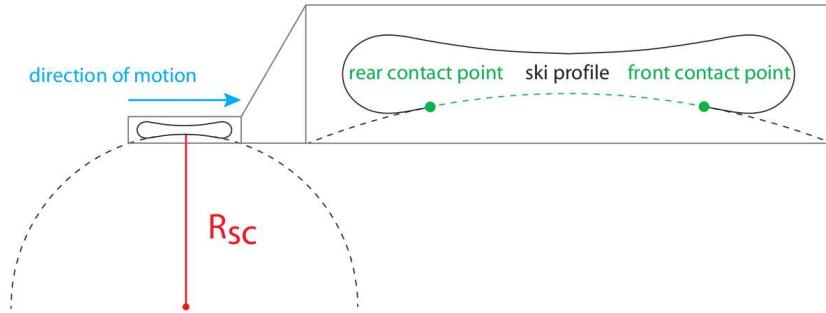


Figure 1.3: (from [5]) Profile of a carving skiing with the sidecut radius and the turning radius evidenced.

and the inclination angle γ of the current trajectory e_r is

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

where β is the angle between e_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_r and into the lateral force F_{lat} acting perpendicularly to the current trajectory (see. Fig1.4).

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

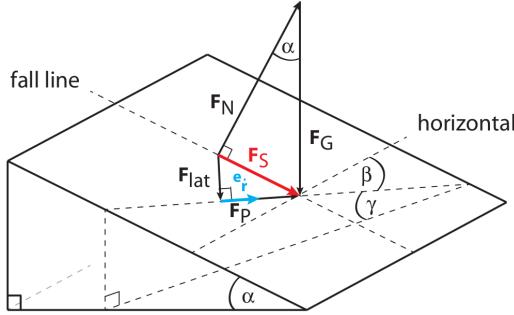


Figure 1.4: (from [5]) The downhill force F_S can be decomposed into the downhill force F_P , acting parallel to the current trajectory, and into the lateral force F_{lat} acting perpendicularly to the direction of travel.

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 γ 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

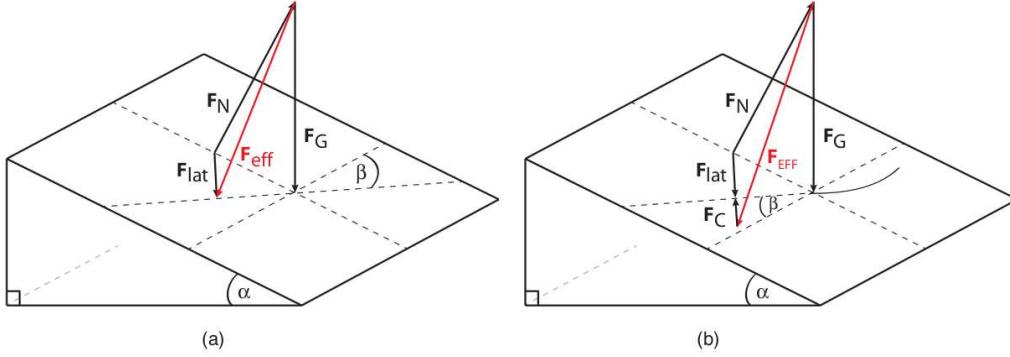


Figure 1.5: (from [5]) In (a) effective force during the descent on a straight line ($F_{eff} = F_{lat} - F_N$). In (b) effective force during a carved turn ($F_{EFF} = F_{lat} - F_N - F_C$).

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}).

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. μ 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, ρ 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 main change introduced in the original model was modeling also non-experienced skiers. Beginner and intermediate skiers perform skidded turns. In a skidded turn the direction of motion is parallel to the skis, however there is an additional component of the motion directed to the side of the turn due to a slippage of the skis on the snow. This results in a significant loss in speed due to the “plowing” action of the skis on the snow which generates a high frictional resistance. To simulate skidded turns, only a fraction of the force F_P , the component of the gravitational force parallel to the direction of motion, was considered to accelerate the skiers. In fact, a fraction of this force is lost in the slippage and do not produce an increase of the speed. Therefore, for non-experienced skiers the net force while turning (defined in 1.30), F_{NET} , should be redefined as

$$F_{NET} = (1 - \eta)F_P + F_{AIR} + F_{GROUND} + F_C \quad (1.31)$$

where η takes value between 0 and 1 and is a coefficient describing the entity of the slippage in the turns. For expert skiers it is set equal to 1.

Other changes in the original model are related to the social forces. In particular, the destination force F_D (defined by 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

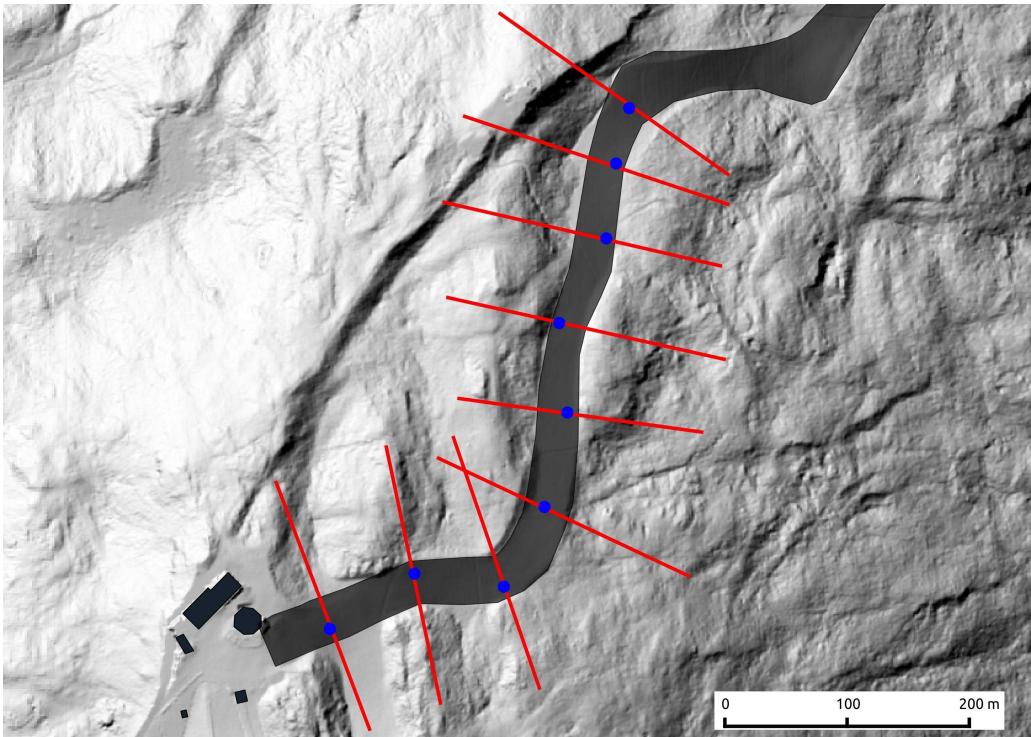


Figure 1.6: In [5] waypoints were selected randomly every 50m following a uniform distribution on the line from the left to the right edges of the slope.

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_a . The vectors given the directions toward the edges are $e_{aR} = (r_a^R - r_a) / \|r_a^R - r_a\|$ and $e_{aL} = (r_a^L - r_a) / \|r_a^L - r_a\|$. Let α be the angle between e_{aR} and e_{aL} defined as

$$\alpha = \begin{cases} \arccos(e_{aR} \cdot e_{aL}) & \text{if } (e_{aR} \times e_{aL}) \cdot n \geq 0 \\ 2\pi - \arccos(e_{aR} \cdot e_{aL}) & \text{if } (e_{aR} \times e_{aL}) \cdot n < 0 \end{cases} \quad (1.32)$$

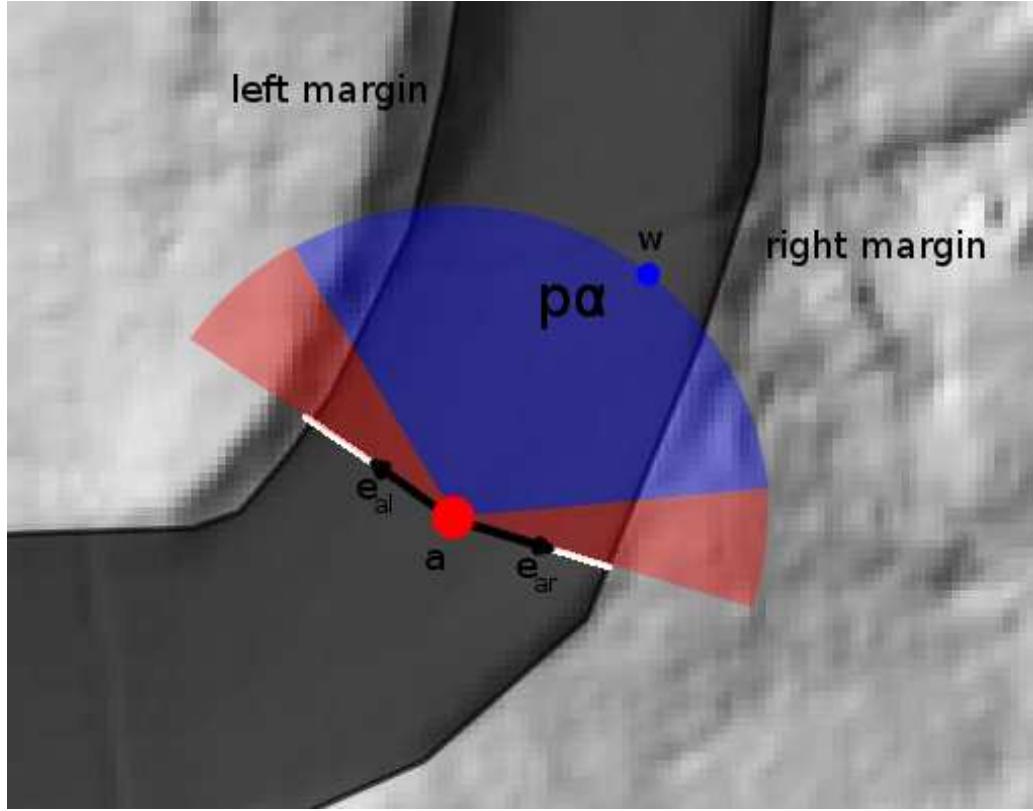


Figure 1.7: Selection of a waypoint: the skier a selects the new waypoint w choosing with uniform distribution on the angle pa , a fraction of α . The angle α is the angle between e_{aR} , the vector representing the direction towards the right edge, and e_{aL} , the vector towards the left edge.

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 α . More precisely, let p be the fraction of angle that should be considered, then the new waypoint w_a is chosen as

$$w_a = r_a + \rho F \left(\frac{e_{aR} + e_{aL}}{\|e_{aR} + e_{aL}\|}, \mathcal{U} \left(-\frac{p\alpha}{2}, \frac{p\alpha}{2} \right) \right) \quad (1.33)$$

where ρ is the distance at which waypoints are chosen, $F(v, \beta)$ rotates the vector v of an angle β 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

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 p of the equation 1.33. When p 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 p 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_{high} the angle should have maximum width. Therefore, p can be set to

$$p = \frac{v}{v_{high}} \quad (1.34)$$

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 1.34 then we can write p as

$$p = \begin{cases} \frac{v}{v_{high}} \frac{s}{s_{lim}} & \text{if } s < s_{lim} \\ \frac{v}{v_{high}} & \text{otherwise} \end{cases} \quad (1.35)$$

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 p 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

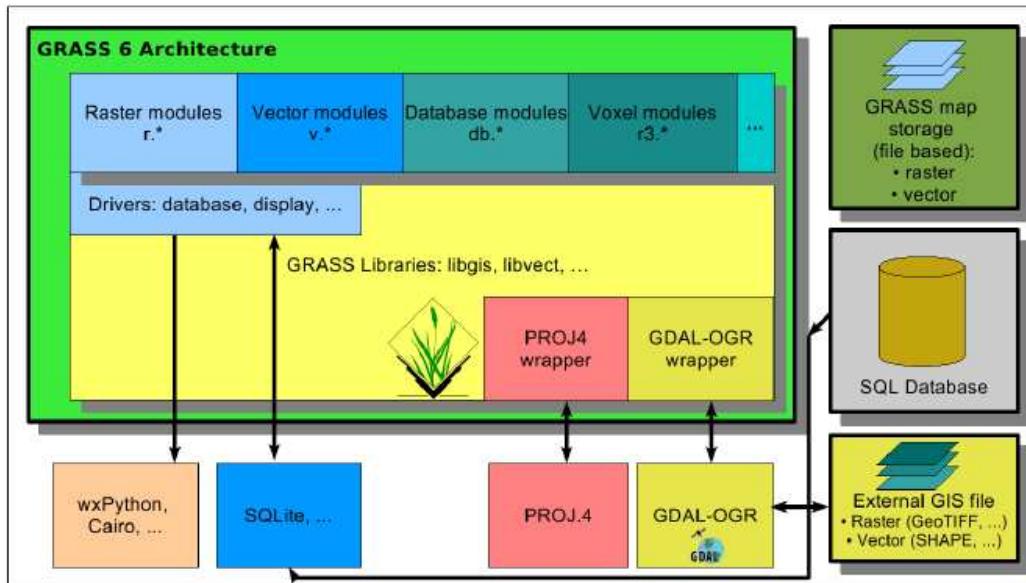


Figure 2.1: (from <http://grass.osgeo.org>) GRASS software architecture.

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.

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 [9].

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,

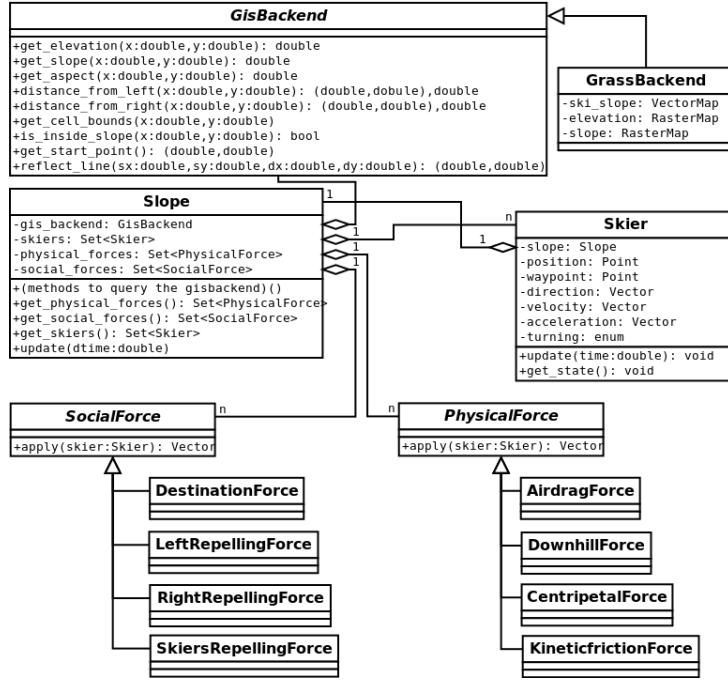


Figure 2.2: Class diagram.

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. It requires the vector map with the polygon of the ski slope for which the model should be run. The ski slope (see Fig.2.3 for an example) is extracted from a vector map containing the polygons of slopes of the skiing

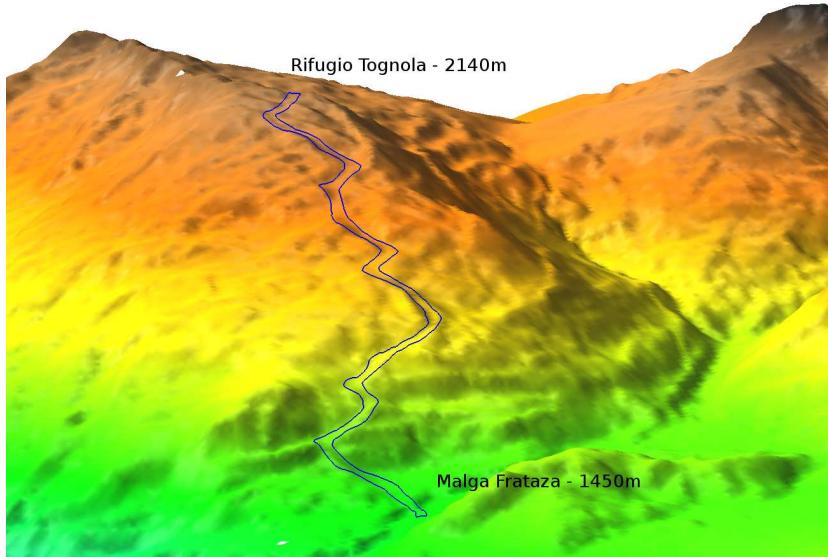


Figure 2.3: 3d visualization of the ski slope “Uno di Tognola” polygon (in blue) in San Martino di Castrozza.

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.

The simulation requires a Digital Elevation Model. For this purpose a Digital Terrain Model (DTM) of the area of Trentino was used. The DTM was obtained using the LIDAR technology, a remote sensing sensing technology that uses laser light to measures distances. The measurements were done in the years 2006 and 2007. It is at a resolution of $1 \times 1\text{m}$ except for areas at high elevation where resolution is of $2 \times 2\text{m}$.

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 strategy was 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 were 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 [8]. 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 was represented by a point. Then the GRASS module v.surf.rst was used to interpolate these points into a new elevation map using regularized spline with tension [7].

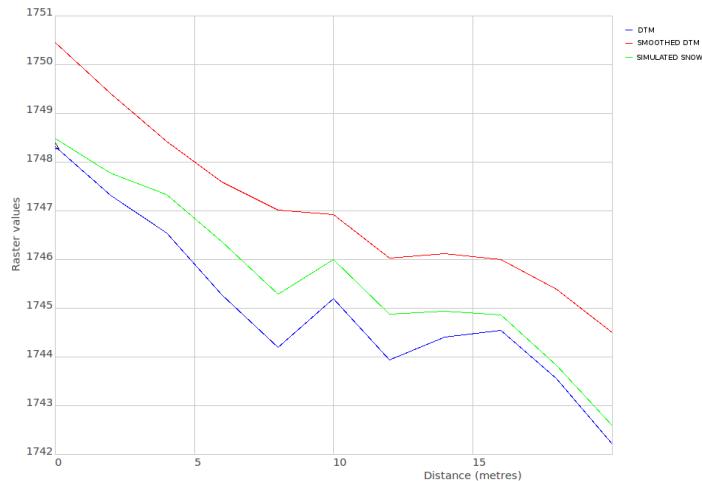


Figure 2.4: Profiles of the dtm and of the simulated dtm after rain fall.

Figure 2.4 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.

Starting from the smoothed 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 [2]. The direction the plan faces is the aspect.

2.4 Output data

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

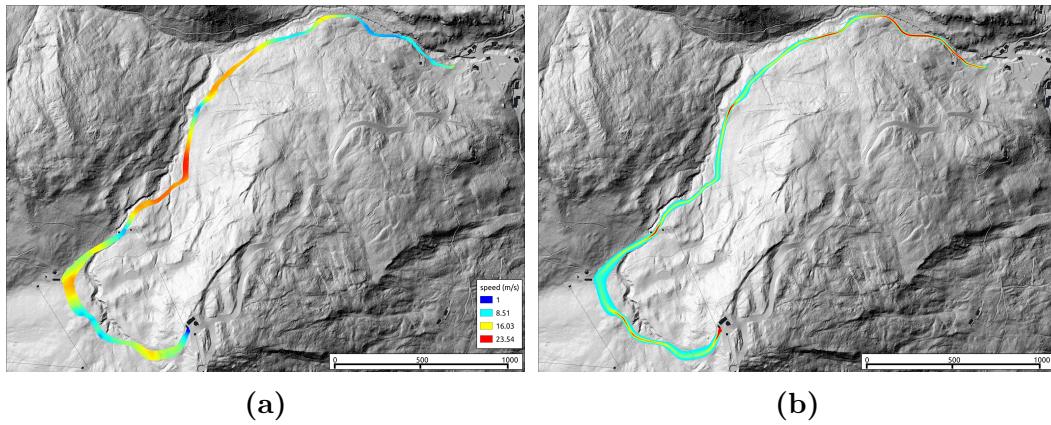


Figure 2.5: Average simulated speed map (a) and skiers density map (b) for the ski slope “Rododendro”.

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.

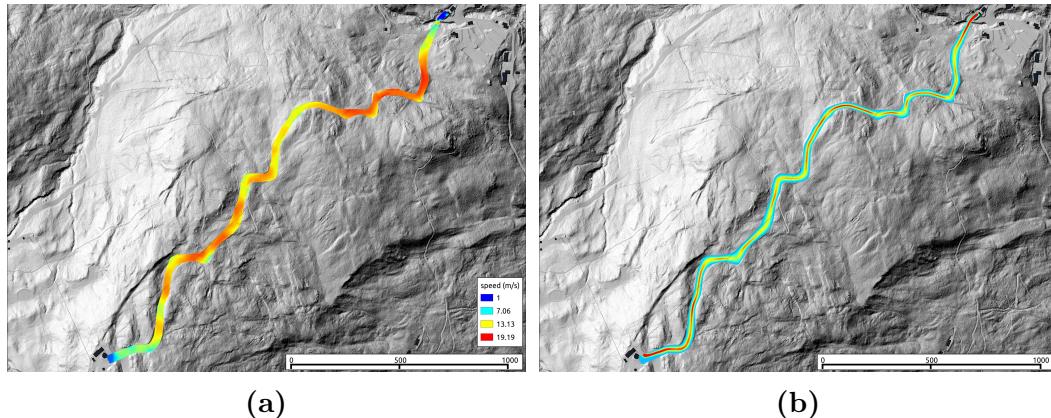


Figure 2.6: Average simulated speed map (a) and skiers density map (b) for the ski slope “Rododendro”.

The map is obtained interpolating the speed values simulated by the simulation (see Fig.2.6a and Fig.2.5a). 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.2.6b and Fig.2.5b).

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 ski slope at a constant rate. Let λ be the persons per hour descending a ski slope, then the time between the start of two consecutive skiers is $T = \mathcal{U}(2\frac{3600}{\lambda})$, 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 [5] and in [4], the repulsive potentials defining the social forces are exponential defined as

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

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

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

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.

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	2φ	180
Directional deviation	δ	10

Table 2.1: Summary table for parameters regulating social forces.

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 $R_A = 8$. These choices have the practical implication that the magnitude of the destination force is always equals to 1 and the repulsion forces tend to 1 when the skier approximates the edges or other skiers. The angle of view is set to $2\varphi = 180$. As presented in chapter 1, skiers adjust their direction performing turns when the angle between their direction of motion and the desired direction exceed the angle δ (see Fig.1.1). This angle was set to $\delta = 10$. The values selected for the social parameters are summarized in table 2.1.

The initialization of the parameters regulating the physical forces was based on values known from the literature. The main reference for choosing the values of the parameters was [5]. The density of air was set to $\rho_{air} = 1.3163 \text{ kg/m}^{-3}$, the mass of the skiers, including clothes and equipment, to $m = 85 \text{ kg}$. The gravitational acceleration was taken equals to $g = 9.81 \text{ m/s}$. The sidecut radius of the skis was set to $R_{SC} = 10 \text{ m}$ for expert skiers. This values was increased simulating non-experienced skiers, that usually need more space to complete the turns. According to the values suggested in [1], the kinetic friction coefficient was set to $\mu = 0.05$. The air drag coefficient $C_d = 1.0$ was chosen and the frontal area was set to $A = 0.6 \text{ m}^2$. These values are summarized in table 2.2.

2.6 Optimization

An analysis of the performances of the software was performed in order to improve the computation time. The section of the software more critical were those sections accessing the files containing the spatial data. These operations, performed by the GrassBackend, were needed to access the data

Parameter	Symbol	Value
Air density	ρ_{air}	$1.3163kg/m^{-3}$
Skier mass	m	$85Kg$
Gravitational acceleration	g	$9.81m/s^2$
Sidecut radius	R_{SC}	$10m$
Kinetic friction coefficient	μ	0.05
Air drag coefficient	C_d	1.0
Frontal area	A	$0.6m^2$

Table 2.2: Summary table for parameters regulating physical forces.

contained in the GRASS map storage and are very time consuming. To speed up the calls to the GrassBackend a caching mechanism was implemented. For the queries requesting raster map values, once a cell value has been fetched it is cached in memory, so that the next queries will access the data quicker. The implementation of the cache was done using the map class available inside the C++ standard library. The map class is an implementation of an associative containers that was used to efficiently store and retrieve values (the raster map values) using keys (the combination of rows and columns). The speed-up reached thanks to the caching mechanism can be seen in Figure 2.7.

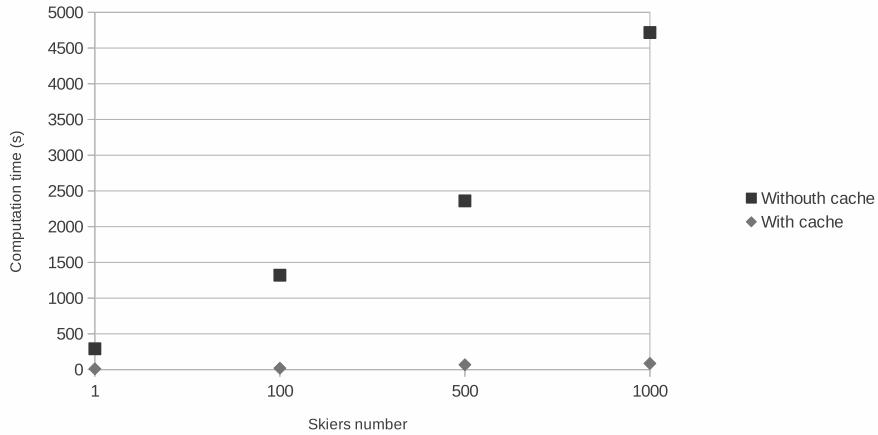


Figure 2.7: Computation time comparison between simulations with cache and simulation without it.

Chapter 3

Steps to validate the model

3.1 Collecting data

The mobile application SkiLogger was used to collect real data describing the skiers trajectories. SkiLogger is an Android application developed inside the MPBA unit of the Bruno Kessler Foundation. It allows the skiers to track their position using GPS technology while skiing.

SkiLogger was released for beta testing starting from February 2013. At this moment the application does not offer any additional functionality, but for the next seasons it is planned to implement new features such as real time information report about ski slope conditions and traffic, routing through the ski slope network and friends position tracking. The goal is to incentive more users to use the application and collect a large dataset of real data.

The data collected by the application needs to be filtered: the gps does not always give an accurate position and the application logs every moment including when using the ski lifts. Automated procedures have been developed to individuate and filter the single downhill tracks. As a first elaboration the speed of each point recorded has been computed. In order to minimize the effects of oscillation in the gps measurements, for the recorded position x_i at the time t_i , the speed was computed considering the previous and following position as follow

$$v_i = \frac{\|x_i - x_{i-1}\| + \|x_{i+1} - x_i\|}{t_{i+1} - t_{i-1}} \quad (3.1)$$

The second step processing the data is to filter the position tracked when skiers are on the ski lifts. For this purpose, a sliding window that considers sequences of ten points at a time was used. A sequence of point is considered valid if the slope between the first and the last point is downhill and it is greater than a minimum threshold or if the skier keeps a speed greater

than a fixed minimum speed. The idea is to exclude sequences of point tracking movements in flat areas or moving uphill, avoiding at the same time to eliminate those sequences tracking movement in area of ski slopes that are counter slope and where skiers should usually keep a minimum speed to avoid stopping.

Finally, the last step in the process of cleaning the data is to isolate the single downhill tracks. This is done in two steps: first the start of the tracks are individuated, then each point is assigned to the right track (if any exists). The start of a track is defined as a sequence of at least forty points of the same skier, that are logged within a temporal range and that cover more than a fixed minimum distance. This is done to eliminate the noise data produced by the gps when skiers stop. In fact, when a skier does not move, the gps tends to jump from different positions generating false movement. This is particularly true when the gps does not have a good precision. Once the start of a track is found all the following sequences of forty points that are within a time range and that represent a downhill movement are considered to be part of this start track. This last step individuates the single tracks and eliminates the points that are noise due to the inaccuracy of the gps.

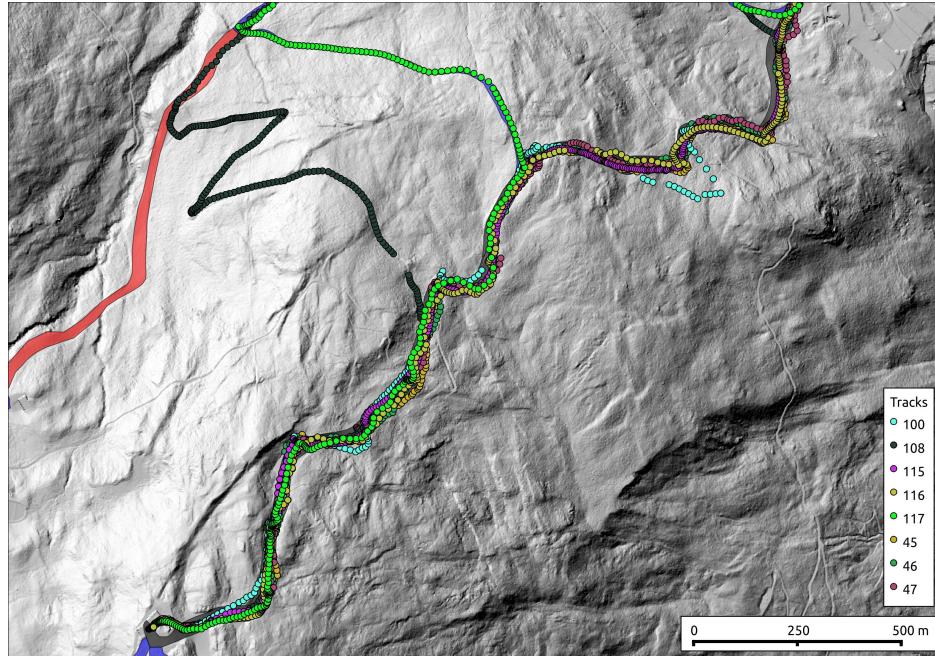


Figure 3.1: Real skiers data logged with the mobile application SkiLogger on the ski slope “Uno di Tognola” in San Martino di Castrozza.

Figure 3.1 shows an example of the data recorded with the tracks high-

lighted. In some sections of the slope gps inaccuracies can be noted. Sometimes discrepancies can be due also to inaccuracies in the trail of the ski slope.

For the season 2012/2013 a total of 178502 meters of descent has been tracked divided in 115 separated tracks. The average length of a track is of 1552 meters. 100 tracks are longer than 500 meters (Table 3.1).

Total number of tracks	115
Total meters tracked	178502m
Average length for track	1552m
Tracks longer than 500m	100

Table 3.1: Summary table for tracks data logged.

The total number of skiers that has used the application is 18. Each skier was request to provide some information before starting to use the application. The data asked included a self-assessment of the skiing skill, the kind of tool used (skis, snowboard, telemark...), the dimension of the group and the age group of the skier. Only two skiers chose the beginner level and tracked 6 tracks for a total of 906 meters. Three skiers were intermediate, with 6 tracks and 6264 meters tracked. Thirteen skiers were experts, they logged 103 tracks and 171331 meters (Table 3.2).

Level	Skiers number	Tracks	Total meters
Beginner	2	6	906m
Intermediate	3	6	6264m
Expert	13	103	171331m

Table 3.2: Summary table for skiers that have used SkiLogger.

The main issue of the data collected is that they are not statistically relevant. Of the total 115 tracks, 52 were tracked by one skier. Another problem is that they are spread on ten different skiing resort in Trentino. For this reason we have very few ski slopes that were tracked by different skiers. In particular there are only two slopes with 5 or 4 different skiers, 8 slopes with 3 skiers and 62 slopes with 1 or 2 skiers (Table 3.3).

The data collected does not allow to do a realistic tuning of the parameters and to validate the model for ski slopes in Trentino. However some comparisons between the data collected and the simulated data has been performed.

Different skiers	Slopes number
5	1
4	1
3	8
2	18
1	44

Table 3.3: Table summarizing the number of different skiers for the ski slopes.

3.2 Simulated vs. real data

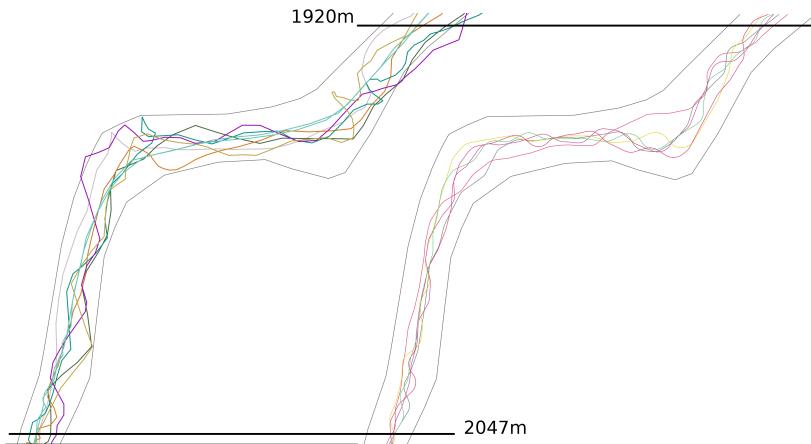


Figure 3.2: Simulated and recorded trajectories on the ski slope “Uno di Tognola”.

The data collected are too few to analyze the map of skiers density produced. Instead, this part of the analysis focuses on the difference between the speed estimated by the simulation and the speed recorded by the SkiLogger application. Starting from the map of simulated speed (see Fig.2.6a), produced as described in section 2.4, the values of speed computed for each point tracked by the application were compared to the values of speed of the map. Figure 3.3 shows the comparison between real and simulated values of speed for the track 47 of the skier 15 recorded on the ski slope “Uno di Tognola”: while the simulated skiers do not stop during the descent the real skier stopped at least three times.

To investigate the relation between simulate and real speeds, the spatial distribution of the error between them was considered. For each track recorded on the ski slope, the error between real and simulated values was computed as $e = v_R - v_S$, where v_R is the real speed recorded and v_S the

speed simulated by the model. Figure 3.4 show the map of the error made for each point of the tracks. The point in the map were the skiers error is higher is were the skiers stopped. It can be seen that the error produced by the stops reverberates on the following points.

To compare the speed values without the influence produced by the errors due to the stops, a new simulation was run where the simulated skiers were forced to stop according to the stops made by the skier 35 in the track 116. A new map of simulated speed was interpolated, Figure 3.5 shows the new comparison for the track 116. In the first part of the track the simulation shows a good correspondence with the real data. In the last two runs the simulation overestimates the speed.

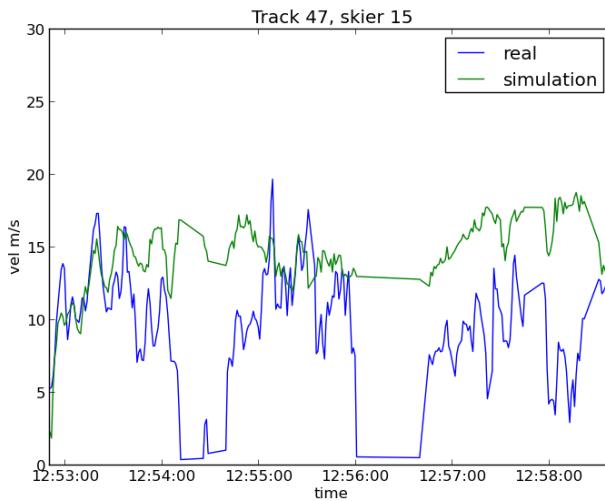


Figure 3.3: Comparison between simulated and real speed for the track 47 on the ski slope “Uno di Tognola”.

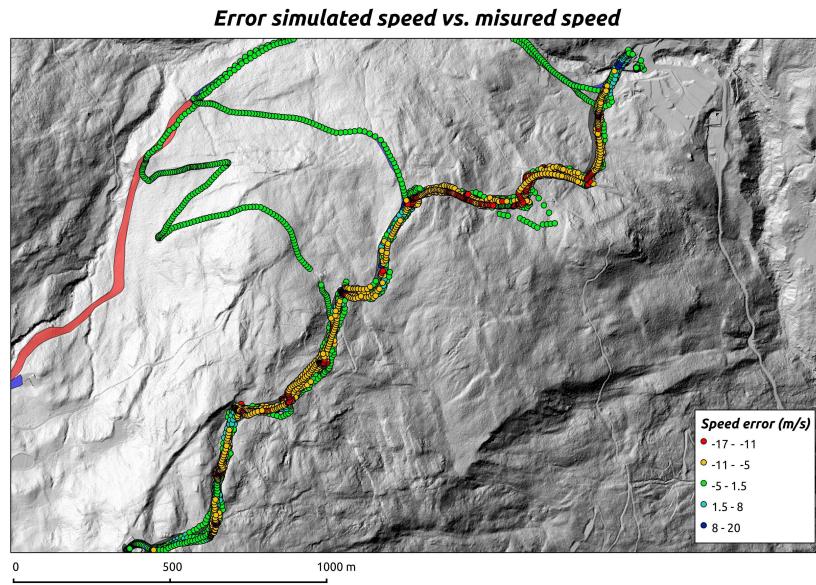


Figure 3.4: Map of the errors between the real and the simulated speed on the tracks recorded on the ski slope “Uno di Tognola”.

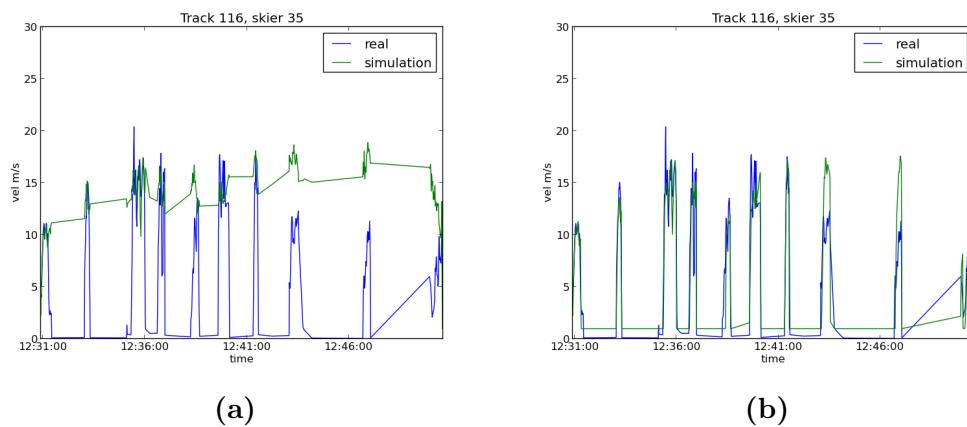


Figure 3.5: Simulated and real speed values for track 116 of “Uno di Tognola”. Figure 3.5a is without stops, in Figure 3.5b simulated skiers were forced to stop as the real skier.

Conclusions

Bibliography

- [1] D. Buhl, M. Fauve, and H. Rhyner. The kinetic friction of polyethylen on snow: the influence of the snow temperature and the load. *Cold Regions Science and Technology*, 33:133–140, 2001.
- [2] P. A. Burrough and R. A. McDonnell. *Principles of Geographical Information Systems*. Oxford University Press, USA, 2 edition, Mar. 1998.
- [3] 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.
- [4] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5):4282–4286, May 1995.
- [5] T. Hollecze and G. Tröster. Particle-based model for skiing traffic. *Physical Review E*, 85(5):056101, May 2012.
- [6] U. D. Jentschura and F. Fahrbach. Physics of Skiing: The Ideal-Carving Equation and its Applications. *Canadian Journal of Physics*, 82:249–261, 2004.
- [7] 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.
- [8] 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.
- [9] M. Neteler and H. Mitasova. *Open Source GIS: A GRASS GIS Approach*. Springer, 3rd edition, 2008.