

University of Trento
Departement of Information Engineering and Computer Science



Degree Course in Computer Science

Final Thesis

An open-source implementation of a particle-based
model for skiing dynamics

Supervisor:

Prof. Alberto Montresor

Co-supervisor:

Dr. Cesare Furlanello

Graduant:

Matteo Poletti

Accademic Year 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, enabling 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 [7]. 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. A second group of changes is related to the implementation of the social forces: skier choices related to their trajectories were made dynamically depend on the position and velocity of the skiers and on the condition of the slope, such as the shape of the trail and the terrain 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. An analysis of the code performances suggested optimizing the implementation with caching mechanisms that reduce the global computation time.

The model has been tested by using a set of real data collected in February and March 2013. The mobile application SkiLogger, developed within 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 SkiLogger dataset. A comparison between real data and simulated data was performed in terms of speed values. The average error for the considered tracks ranges between 4.28m/s and 6.44m/s, while recorded speeds range from 0 to 30m/s. The model was 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.

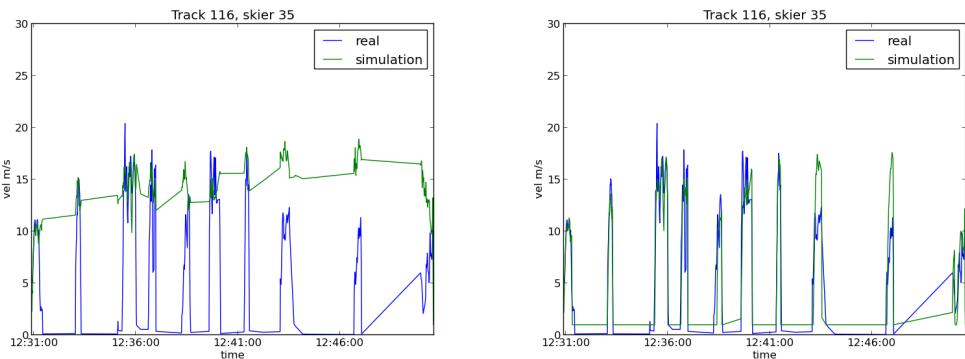


Figure 1: Comparison between speed values recorded for a track and average simulated speeds. Figure on the left shows simulated speed values without stops. Figure on the right shows speed values with forced stops.

Chapter 2 describes the model and defines the physical and social forces. Chapter 3 presents the technologies employed and gives details related to the

model implementation and the parameters selection. Chapter 4 presents the data collected and the comparisons with the simulated data. A summary of results and a concluding remark are given in Chapter 5.

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 discussions we had during these months. I would like to thank also Prof. Nicola Pugno, for his suggestions concerning the physical model. Finally, I thank my family for supporting me throughout all my studies.

Contents

1	Introduction	5
2	The model	7
2.1	Social forces	7
2.2	Physical forces	10
2.3	Limitations	14
2.4	Innovation from the original model	14
2.4.1	Extension 1	14
2.4.2	Extension 2	15
2.4.3	Other possible extensions	18
3	Implementation	19
3.1	Technologies used	19
3.2	Software structure	21
3.3	Input data	23
3.4	Output data	26
3.5	Parameters Selection	26
3.6	Optimization	29
4	Steps to validate the model	31
4.1	Field data sampling	31
4.2	Simulated vs. real data	34
4.3	Simulated ski lessons	37
5	Conclusions and future work	39
A	Simulations summary values varying parameters	43

Chapter 1

Introduction

Alpine skiing is a winter sport widely practiced in the Alps and it is a key resource for the tourism sector. Each year more than 80.000.000 single ski lifts passages are counted in Trentino's ski resorts where the ski accidents risk ratio on single passages is about 1:10.000.

SicurSkiWeb is a project of the Bruno Kessler Foundation and Polizia di Stato aiming to monitor and improve safety on ski slopes, in collaboration with the Trentino Skiing Resorts Managers Association and the Trentino Ski Instructor Association. The main output of the project is a GeoICT platform for the collection and the analysis of all data collected during the rescue interventions by ski patrols. Up to today the platform geodatabase includes records for 2571 interventions: the 70% of accidents involves skiers while 20% are snowboarders; moreover 80% of the interventions are due to accidental causes while 11% are caused for collisions with other persons, and 4% are due to illnesses.

The GeoITC infrastructure is composed of three modules:

1. An Android application for data collection synchronized with the main platform geodatabase.
2. A WebGIS platform to manage or visualize field data and to constantly monitor the evolution of the risk on the slopes.
3. A set of analysis tools for understanding, studying and mapping the risk of accidents and its relationship with environmental and social factors.

This thesis work aims to provide an analysis framework to describe the behavior of skiers while performing their descents on ski slopes. The output of this analysis will be used as a useful component of the analysis module of the SicurSkiWeb platform, in particular for speed and traffic density estimates.

Traffic simulations have been widely used to predict motions of objects such as vehicles and pedestrians using models from statistical physics. Holleczek and Troster proposed this methodology to predict skiing traffic. Following their approach, skiing dynamics has been simulated using a particle-based model.

The problem faced in this thesis work involves patterns that can be recognize also modeling other kind of systems. In sports, the behavior of a player is influenced and determined by the other players actions, by their distribution on the field and by the phase of the game. The players behaviors depends also on their role and their interpretation of the game. Dynamics of the players on the field has already been investigated for sports as basketball [1], tennis [13] and squash [9]. For team sports and specifically rugby, a model of interpersonal dynamics in terms of multiagent system was proposed [14]. Multiagent system are often simulated also in videogames. However, this kind of simulations do not aim to simulate real agents but try to reproduce credible scenarios allowing to use heuristics and predetermined sets of rules derived by specific knowledge of the domain. Nevertheless, the physics implemented by skiing simulator videogames is closely related to the one implemented in this thesis.

Empirical data plays a key role modeling this kind of system: they are essential to calibrate the model parameters and to investigate the agents behavior. In the collection of data, wearable sensors offer important possibilities for the capturing of motions. Pressure sensors and gyroscope were already successfully employed to track snowboarders activities [6], a similar workflow can be followed for skiers.

Chapter 2

The model

The model implemented in this thesis 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 [7].

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.

2.1 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.2.1). 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 chosen destination. The repulsion from obstacles can be thought as the repulsion from physical

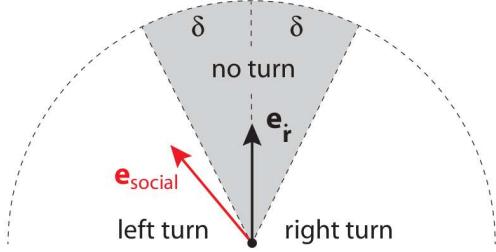


Figure 2.1: (From [7]) When the angle between the current direction of motion e_r and the desired direction e_{social} is bigger than δ the skier performs a turn to adjust his or her direction.

obstacles, such as snow guns or piles of ski lifts, but also as the repulsion from areas of the ski slope perceived as dangerous, such as sheets of ice.

To describe the social force that attracts the skier towards the chosen destination, the model assumes that each skier, while descending, selects 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 $x_a(t) = x_a^k$ for some k . The direction toward the current waypoint is expressed by

$$e_a(t) = \frac{x_a^k - r_a(t)}{\|x_a^k - r_a(t)\|}, \quad (2.1)$$

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

$$F_D(r_a) = A_0 e_a(t), \quad (2.2)$$

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

$$F_L(r_{aL}) = -\nabla_{r_{aL}} U(\|r_{aL}\|), \quad (2.3)$$

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

$$F_R(r_{aR}) = -\nabla_{r_{aR}} U(\|r_{aR}\|). \quad (2.4)$$

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

$$F_S(r_{ab}) = -\nabla_{r_{ab}} V(s(r_{ab})), \quad (2.5)$$

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

$$s(r_{ab}) = \frac{\sqrt{(\|r_{ab}\| + \|r_{ab} - v_b \Delta t e_b\|)^2 - (v_b \Delta t)^2}}{2}. \quad (2.6)$$

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

$$F_O(r_{ao}) = -\nabla_{r_{ao}} W(\|r_{ao}\|). \quad (2.7)$$

In general, repulsion social forces act on skiers only if they are 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

$$w(u, v) = \begin{cases} 1 & \text{if } (u/\|u\|) \cdot (v/\|v\|) \geq \cos \varphi \\ 0 & \text{otherwise} \end{cases} \quad (2.8)$$

To summarize, social forces acting on every skier a are

$$F_D(r_a) = A_0 e_a(t), \quad (2.9)$$

$$F_L(\dot{r}_a, r_{aL}) = w(\dot{r}_a, -r_{aL}) F_L(r_{aL}), \quad (2.10)$$

$$F_R(\dot{r}_a, r_{aR}) = w(\dot{r}_a, -r_{aR}) F_R(r_{aR}), \quad (2.11)$$

$$F_A(\dot{r}_a, r_{ab}) = w(\dot{r}_a, -r_{ab}) F_A(r_{ab}), \quad (2.12)$$

$$F_O(\dot{r}_a, r_{ao}) = w(\dot{r}_a, -r_{ao}) F_O(r_{ao}). \quad (2.13)$$

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

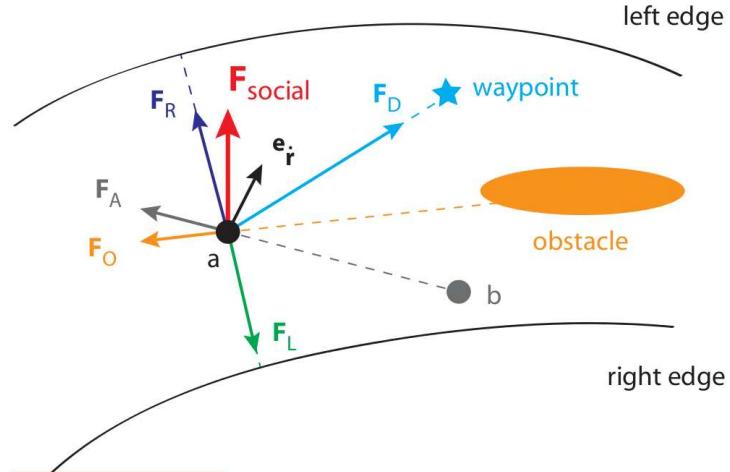


Figure 2.2: (From [7]) Diagram of the social forces. The forces F_R and F_L keep the skier a away from the edges, the force F_O repels the skier from the obstacle, the force F_A repels a from the skier b and the force F_D attracts the skier towards the waypoint. The vector e_r is the direction of motion.

$$F_{social}^a = F_D(r_a) + F_L(\dot{r}_a, r_{aL}) + F_R(\dot{r}_a, r_{aR}) + \sum_{b \in S} F_A(\dot{r}_a, r_{ab}) + \sum_{o \in O} F_O(\dot{r}_a, r_{ao}),$$

where S is the set of skiers and O the set of obstacles in the current time frame and visibility. Figure 2.2 shows a diagram of the social forces described above.

2.2 Physical forces

As described in [7], 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 [7] skiers are supposed to perform turns with a radius corresponding to the *sidecut radius* of their skis. Although some studies [8] [4] 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 was

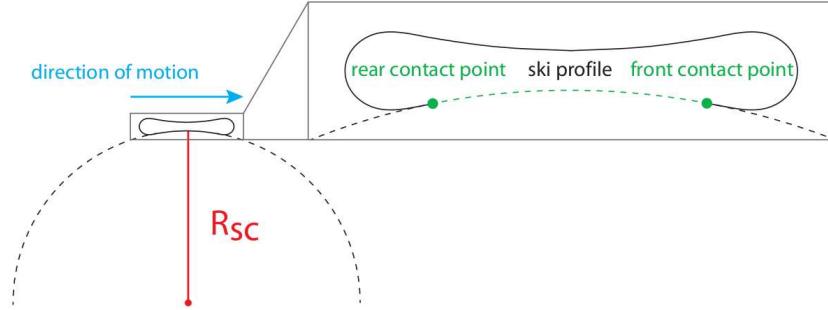


Figure 2.3: (From [7]) Profile of a carving ski with the sidecut radius R_{SC} evidenced.

considered acceptable. Figure 2.3 shows the relation between sidecut radius and turning radius.

Gravitational, centripetal and friction forces determine the skier's 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 ski slope has an inclination angle of

$$\alpha = \arccos([0, 0, 1] \cdot n) \quad (2.14)$$

and the inclination angle γ of the current trajectory e_r is

$$\gamma = \arcsin[(\sin \alpha)(\sin \beta)], \quad (2.15)$$

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

$$F_G = mg \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \quad (2.16)$$

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.

$$F_G = F_S - F_N. \quad (2.17)$$

The normal force F_N can be expressed as

$$F_N = mg(\cos \alpha)n. \quad (2.18)$$

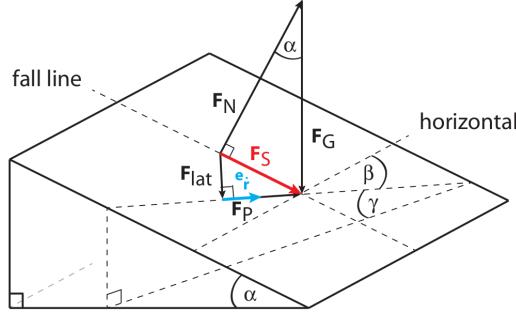


Figure 2.4: (From [7]) The downhill force F_S can be decomposed into the downhill force F_P , acting parallel to the current trajectory, and into the lateral force F_{lat} acting perpendicularly to the direction of travel.

The downhill force F_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 Fig.2.4).

$$F_S = F_P + F_{lat}. \quad (2.19)$$

The downhill force F_P can be written as

$$F_P = mg(\sin \gamma) e_r = mg(\sin \alpha)(\sin \beta) e_r. \quad (2.20)$$

where γ is the inclination angle of e_r defined as in 2.15.

Remembering 2.17, the downhill force F_S can be computed as

$$F_S = F_G + F_N. \quad (2.21)$$

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

$$F_{lat} = F_S - F_P = F_G + F_N - F_P. \quad (2.22)$$

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

$$F_C = \frac{m}{R_{SC}} \|\dot{r}\|^2 \frac{F_{lat}}{\|F_{lat}\|} \times \begin{cases} (+1) & \text{before crossing the fall line} \\ (-1) & \text{after crossing the fall line} \end{cases}, \quad (2.23)$$

where m is the mass of the skier and R_{SC} the sidecut radius of the skis. The force 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.

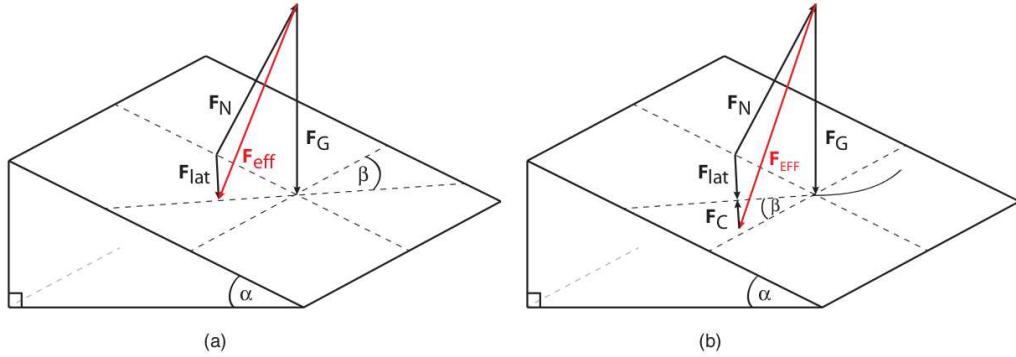


Figure 2.5: (From [7]) In (a) effective force during the descent on a straight line ($F_{eff} = F_{lat} - F_N$). In (b) effective force during a carved turn ($F_{EFF} = F_{lat} - F_N - F_C$).

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

$$F_{eff} = F_{lat} - F_N. \quad (2.24)$$

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

$$F_{EFF} = F_{lat} - F_C - F_N. \quad (2.25)$$

Figure 2.5 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

$$F_{ground} = -\mu \|F_{eff}\| e_r, \quad (2.26)$$

when descending on a straight line. The parameter μ is the kinetic friction coefficient of the skis on the snow. In case of a turn

$$F_{GROUND} = -\mu \|F_{EFF}\| e_r. \quad (2.27)$$

The air drag force F_{air} is antiparallel to the direction of motion e_r and is defined as

$$F_{air} = -\frac{1}{2}C_d\rho A\|\dot{r}\|^2 e_r, \quad (2.28)$$

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

$$F_{net} = F_P + F_{air} + F_{ground}, \quad (2.29)$$

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

$$F_{NET} = F_P + F_{AIR} + F_{GROUND} + F_C. \quad (2.30)$$

2.3 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. Non-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 neither to jump nor to exit the ski slope: if skiers collide with an edge of the slope they are reflected back with an angle equals to the angle at which they have collided.

2.4 Innovation from the original model

2.4.1 Extension 1

The main extension introduced in the new implemented 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 F_P component of the gravitational force parallel to the direction of motion is considered to accelerate the skiers. In fact, a fraction of this force is lost in the slippage and does not produce a speed increase. Therefore, for non-experienced skiers the net force while turning (defined in

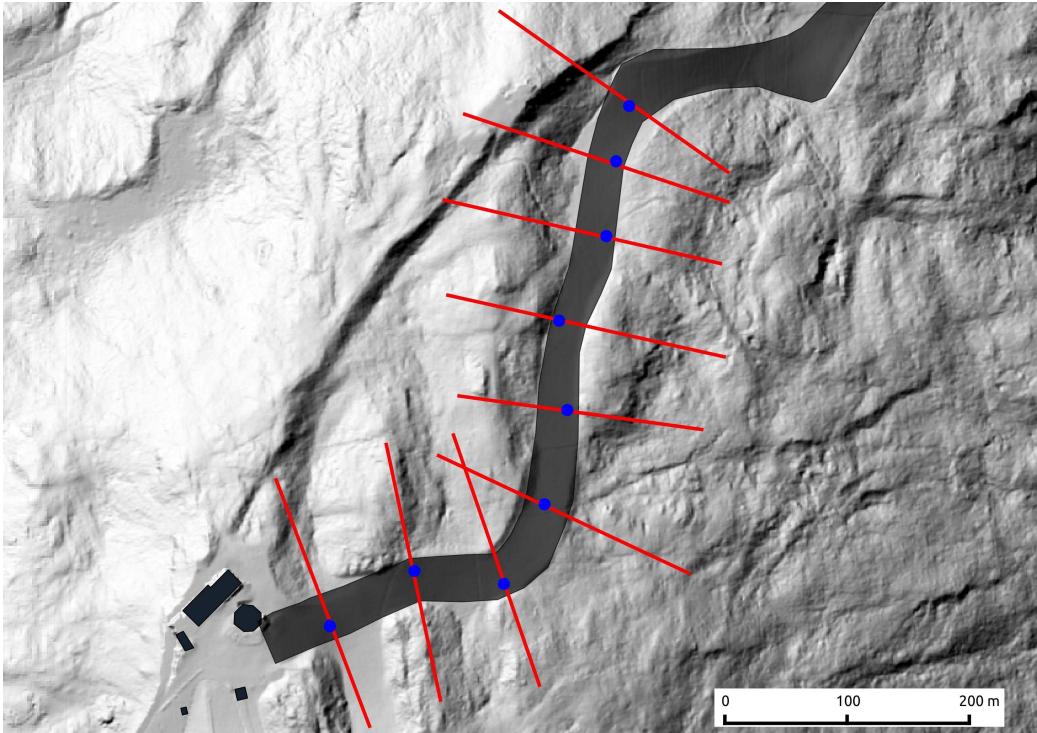


Figure 2.6: In [7] waypoints were selected randomly every 50m following a uniform distribution on the line from the left to the right edges of the slope.

2.30), F_{NET} , should be redefined as

$$F_{NET} = (1 - \eta)F_P + F_{AIR} + F_{GROUND} + F_C, \quad (2.31)$$

where η takes value between 0 and 1 and is a coefficient describing the entity of the slippage in the turns. For expert skiers η is set equal to 0. It should be noted that in the case a skier is not turning or stops the net force remains defined as in 2.29.

2.4.2 Extension 2

Other changes in the original model are related to the social forces. In particular, the destination force F_D (defined by 2.2) 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.2.6).

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 allows 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 (see Fig.2.7). 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 $e_{aR} = (r_a^L - r_a) / \|r_a^L - r_a\|$ and $e_{aL} = (r_a^R - r_a) / \|r_a^R - r_a\|$. Let α be the angle between e_{aR} and e_{aL} defined as

$$\alpha = \begin{cases} \arccos(e_{aR} \cdot e_{aL}) & \text{if } (e_{aR} \times e_{aL}) \cdot n \geq 0 \\ 2\pi - \arccos(e_{aR} \cdot e_{aL}) & \text{if } (e_{aR} \times e_{aL}) \cdot n < 0 \end{cases}, \quad (2.32)$$

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

$$w_a = r_a + \rho F \left(\frac{e_{aR} + e_{aL}}{\|e_{aR} + e_{aL}\|}, \mathcal{U} \left(-\frac{p\alpha}{2}, \frac{p\alpha}{2} \right) \right), \quad (2.33)$$

where ρ is the distance at which waypoint are chosen, the function $F(v, \beta)$ 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.2.7).

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 according to the following constraints:

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 skier's 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 skiers will face in their range of action: approaching flat areas skiers tend to go straight to increase velocity.

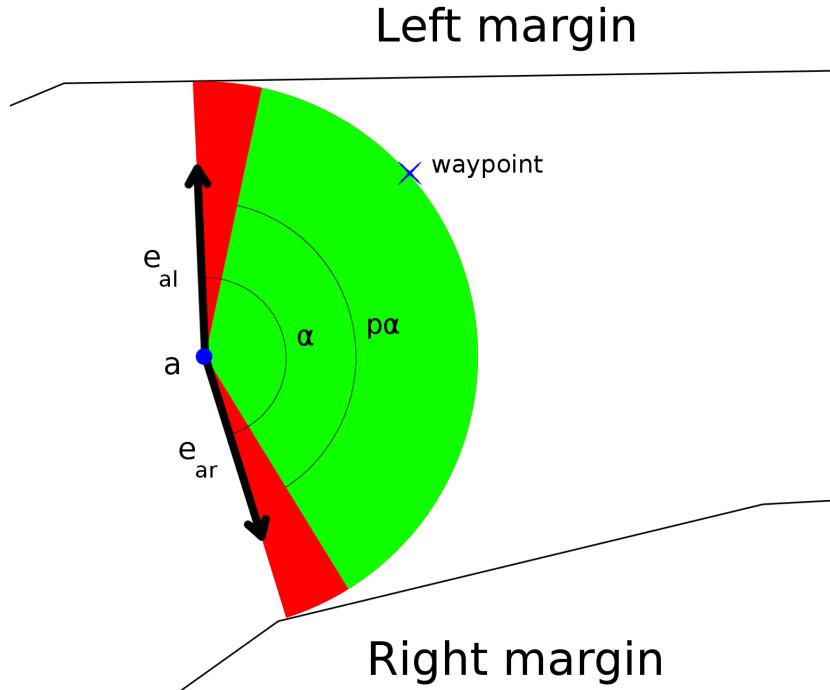


Figure 2.7: Selection of waypoints: the skier a selects the new waypoint w choosing with uniform distribution on the angle $p\alpha$, a fraction of α . The angle α is the angle between e_{aR} , the vector representing the direction towards the right edge, and e_{aL} , the vector towards the left edge.

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 2.33. 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

$$p = \frac{v}{v_{high}}. \quad (2.34)$$

v_{high} is a speed value that is considered high for the skier modeled. Approaching this value the skier enlarges the possible angle of waypoints selection and

is more likely to perform turns decreasing their speed.

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 skiers are faster they choose 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 2.34 then we can write p as

$$p = \begin{cases} \frac{v}{v_{high}} \frac{s}{s_{lim}} & \text{if } s < s_{lim} \\ \frac{v}{v_{high}} & \text{otherwise} \end{cases}. \quad (2.35)$$

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 does 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.

2.4.3 Other possible extensions

The simulation can include a more advanced model of the obstacles repulsion. The repulsion from physical obstacles can be easily implemented when the GIS database will provide the information needed (e.g. the position of the piles of the ski lift). In addition to this, other kind of obstacles can be included in the model to represent areas of the slope considered dangerous by the skiers. For example obstacles can simulate sheets of ice (that can change position depending on the time of the day) or areas with high slope (depending on skiers' skill). This extension was not implemented in this thesis.

When skiing in groups it may be appropriate to make the skiers trajectories influencing each other. During ski lessons for example, skiers trajectories can be forced to emulate the instructor trajectory. This approach is investigated in more detail in section 4.3.

Chapter 3

Implementation

The model was implemented using C++ and GRASS GIS. GRASS 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.

3.1 Technologies used

Different technologies were explored and used to implement the GIS platform of 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 provides 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 has 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

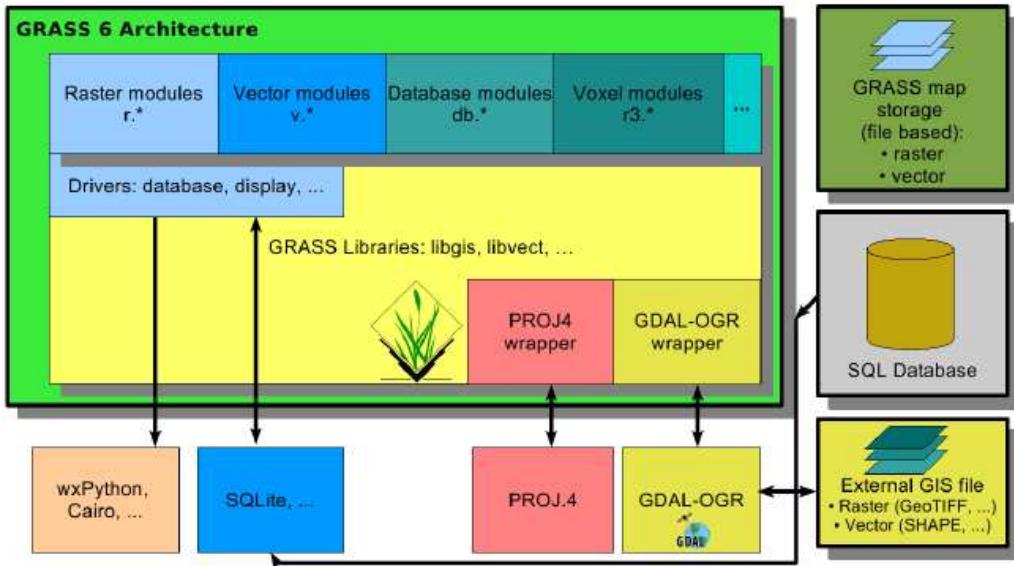


Figure 3.1: (From <http://grass.osgeo.org>) GRASS software architecture.

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.3.1). 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 first approach is to use script programming to call the high-level GRASS modules. A more advanced approach is to access the low-level functionalities 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 chapter 2 this is not feasible. Even if the second approach forces to use spatial 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 [12].

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 having an object oriented allows to better represent the models concepts, reducing the complexity and making the code structure clearer and keeping the code structure modular and flexible.

3.2 Software structure

The main entities modeled are the ski slope, the skiers and the forces that drive them. In Figure 3.2 the software's class diagram is described:

1. Class Slope models the ski slope, it has a set of skiers, a set of physical forces that act on the slope and a set of social forces that skiers are subjected to.
2. Skier class models the skier's behavior: it has a position, a velocity, an acceleration and a status describing if they are turning.
3. Forces skiers are subjected to have been modeled using two different classes: 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 social and physical forces are conceptually different, making easier for future development of the model to change them using separated interfaces.

The simulation has to perform some specific spatial computations. 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 model should be independent by the technology chosen to implement the GIS Backend. For this reason, an abstract class called GisBackend was used. The class declares the interface that the GIS backend used to run the simulation must implement. This interface includes methods to get elevation, slope and aspect for given

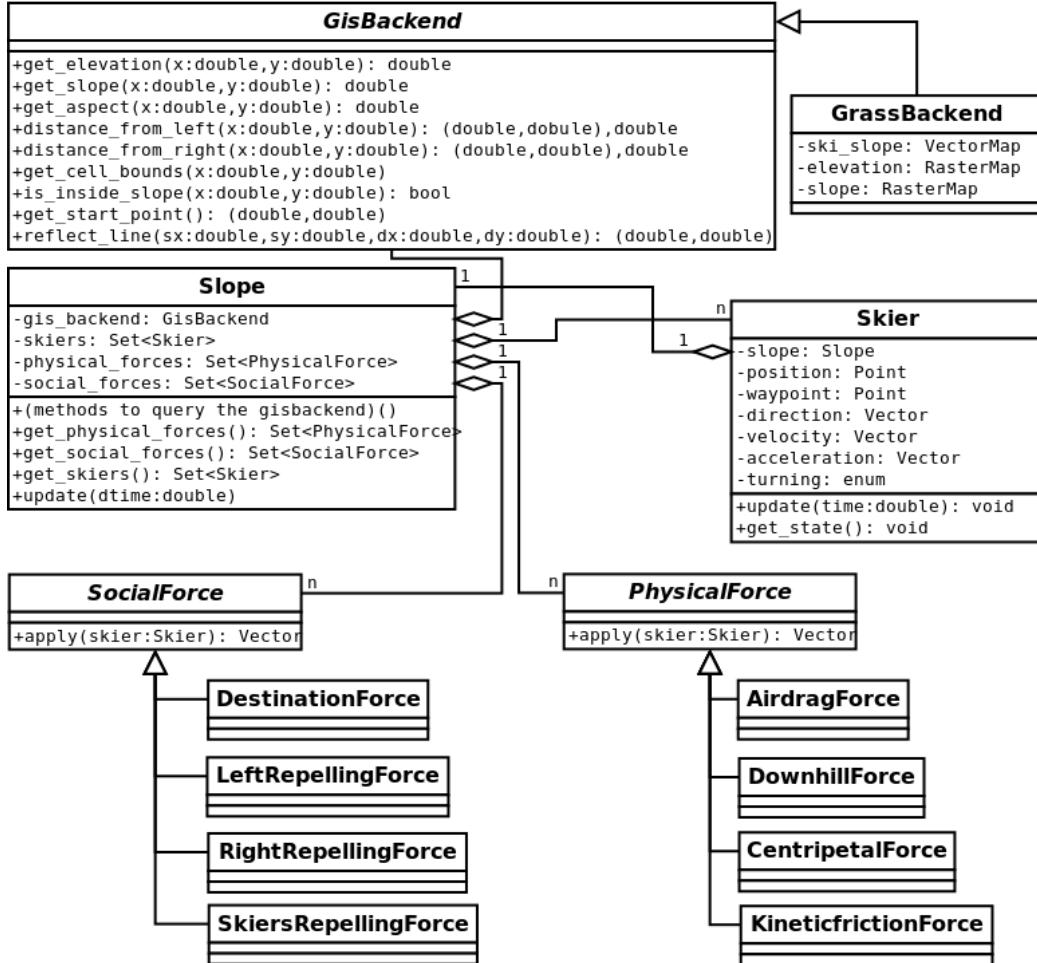


Figure 3.2: Class diagram.

coordinates, methods to get the distance of a point from the edges of the slope, methods to give random starting 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 uses the GRASS GIS Library. It checks to have available a DEM (Digital Elevation Model) raster map for the elevation, the slope and the aspect computation, a polygon vector map describing the ski slope to determine if a point is inside the slope, a line vector map describing the right and the 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.

3.3 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.3.3 for an example) is extracted from a vector map containing all polygons of the ski slopes of Trentino's ski resort. This data come from the SicurSkiWeb database and were derived starting from orthophotos and adjusted with field measurements. From the polygon of the ski slope the left and right edges, the start and stop areas are derived.

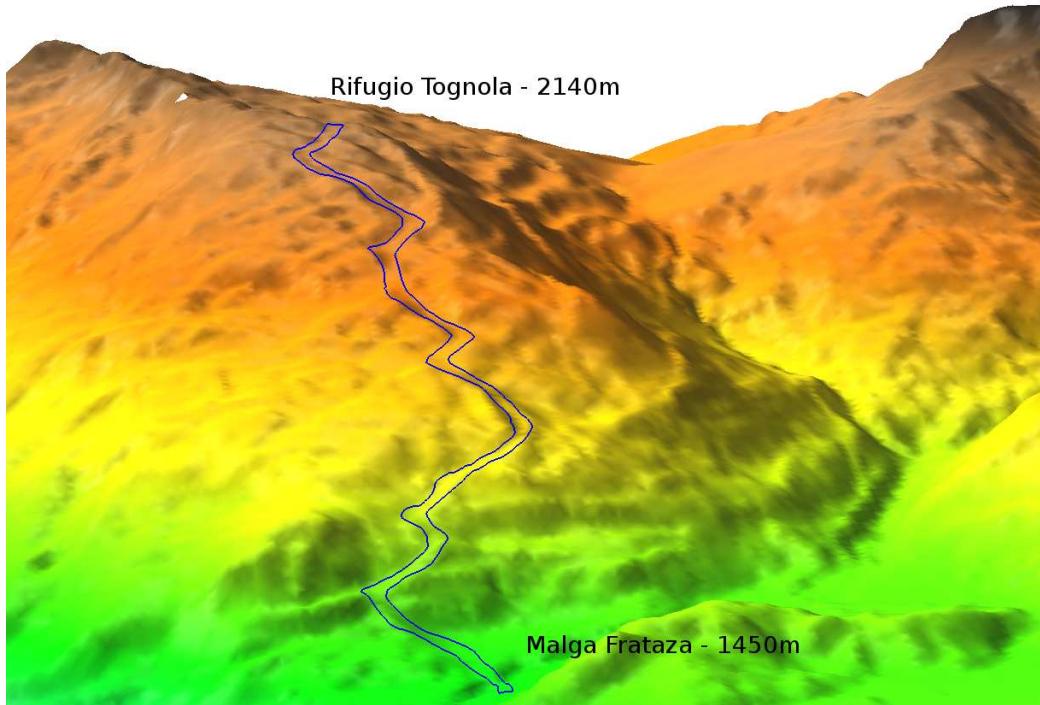


Figure 3.3: 3d visualization of the ski slope “Uno di Tognola” polygon (in blue) in San Martino di Castrozza.

As second set of inputs, the simulation model requires some features extracted from a Digital Elevation Model. In this case the Digital Terrain Model (DTM) of the Autonomous Province of Trento was used. The DTM was computed using the LIDAR flight data of 2006 and 2007, it has a spatial

resolution that variates between 1 meters and 2 meters, depending on the location.

The measurements for the DTM were taken during the summers season without the presence of snow. As a consequence the surface on which real skiers move is different from the DTM obtained from the Lidar data.

Two strategy was thought to simulate the effect of the snow on the DTM:

1. A simple model was implemented 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 precipitations accumulate 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. The actual profile after the precipitation can be described as

$$Y(x) = (1 - l)y_S(x) + ly_L, \quad (3.1)$$

where l is a parameter describing how much snow has a liquid-like behavior. From 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 [11], used to compute the profile y_L .

2. The second strategy starts from the consideration that there are three main factors determining 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 this 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 this points into a new elevation map using regularized spline with tension [10].

Figure 3.4 compares the profiles of the original dtm, of the dtm obtained from the first strategy and of the one obtained from the second strategy

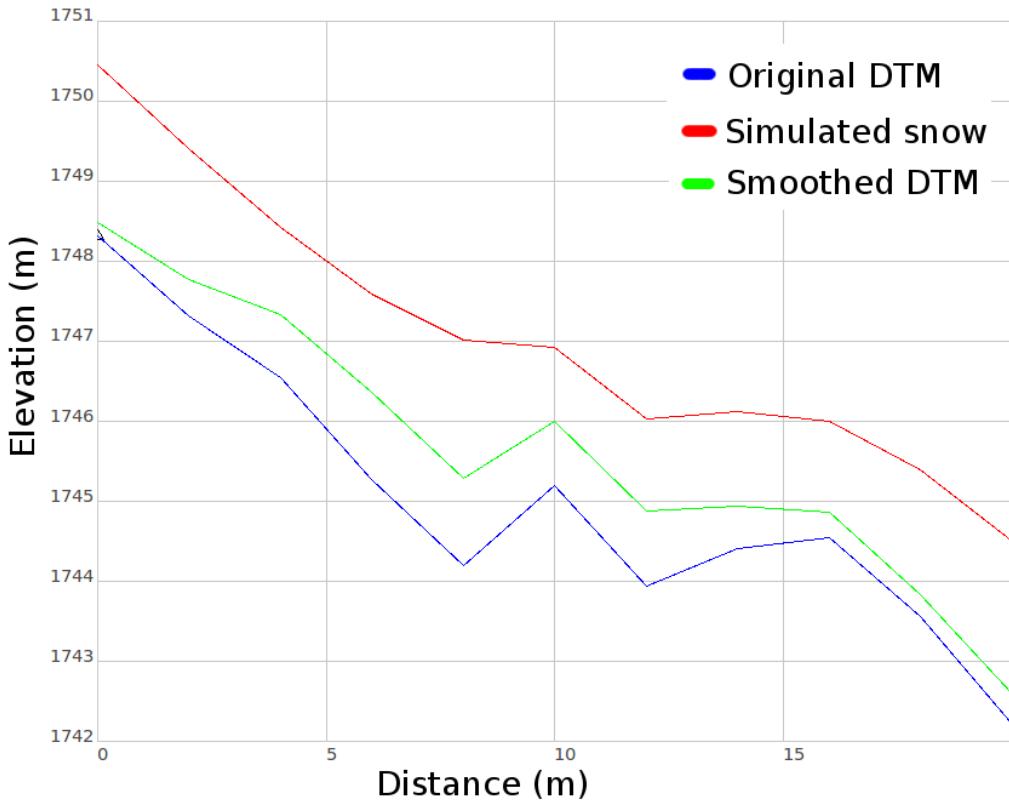


Figure 3.4: Profile of the DTM compared to profile of the DTM smoothed and of the DTM covered with simulated snow.

for a particularly critical point on a ski slope. Following the first strategy, the results of the module `r.sim.water` did not return a feasible profile for a liquid-like behavior. This can be linked to a misconfiguration of the module parameters or to very high level of precision demanded. The best result was given by the second strategy, therefore it was the chosen one for this purpose.

Starting from the smoothed DTM, slope raster map and aspect raster map were computed. To compute slope raster map and aspect raster map the GRASS module `r.slope.aspect` was used. The slope map contains the slope of every single raster cell, defined as the degrees 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 [3]. The orientation of the maximum slope direction toward the east defines the aspect.

3.4 Output data

The output produced by the simulation model is the status (position and speed) 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 .

Starting from this data the skiers speed average and the skiers density on the considered ski slopes were computed. Interpolating the speed values obtained by the simulation (see Fig.3.6a and Fig.3.5a) a raster map of the skiers speed is obtained. This map can be then used to analyze the skiers speed on every location of the ski slope surface. 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.3.6b and Fig.3.5b).

3.5 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 [7] and in [5], the repulsive potentials defining the social forces are exponential and defined as

$$U(\|r_{aL}\|) = U_0 \exp(-\|r_{aL}\|/R_L), \quad (3.2)$$

$$U(\|r_{aR}\|) = U_0 \exp(-\|r_{aR}\|/R_R), \quad (3.3)$$

$$V(s) = V_0 \exp(-s/R_A), \quad (3.4)$$

where R_L , R_R and R_A describe 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 $R_A = 8$. These choices have the practical implication that the magnitude of

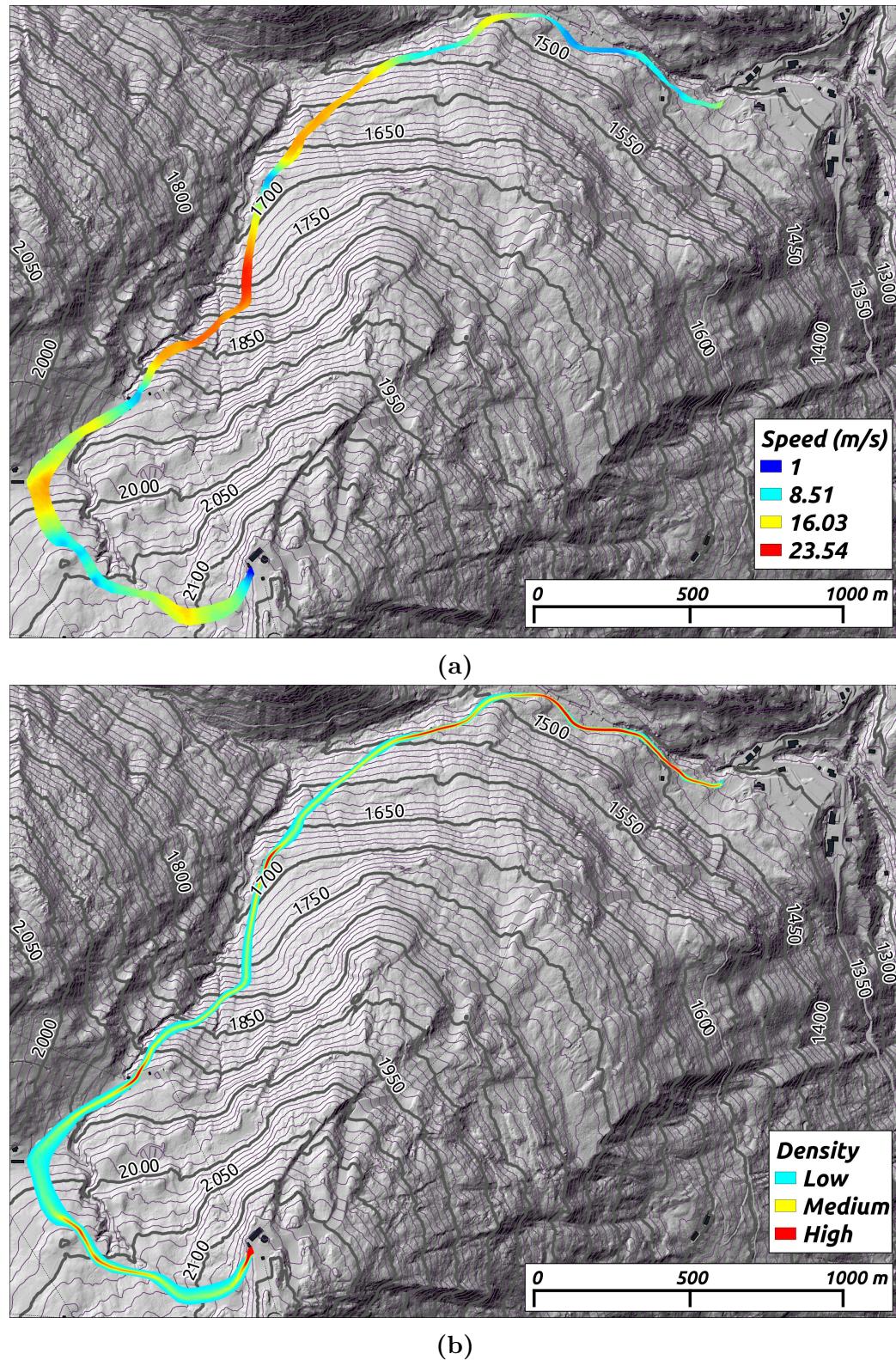


Figure 3.5: Average simulated speed map (a) and skiers density map (b) for the ski slope “Rododendro”.

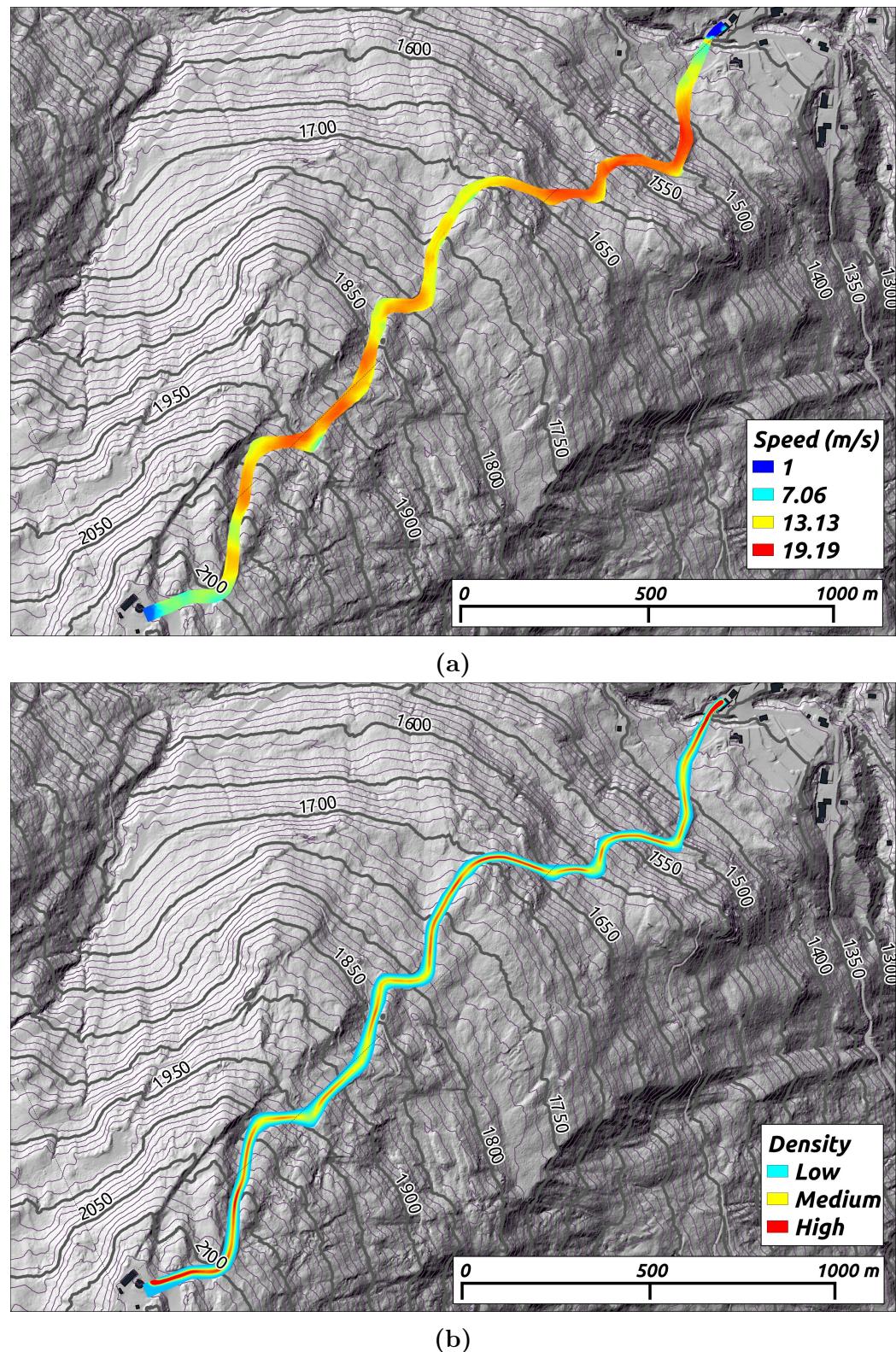


Figure 3.6: Average simulated speed map (a) and skiers density map (b) for the ski slope “Uno di Tognola”.

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 3.1: Summary table for parameters regulating social forces.

the destination force is always equals to 1 and the repulsion forces tend to 1 when the skier approximates the edges or other skiers. For non-experienced skiers the strength and the range of edge repulsion can be increased to model their attitude to keep a higher distance from the edges than experienced skiers.

The angle of view is set to $2\varphi = 180^\circ$. As presented in chapter 2, skiers adjust their direction performing turns when the angle between their direction of motion and the desired direction exceed the angle δ (see Fig.2.1). This angle was set to $\delta = 10^\circ$. The values selected for the social parameters are summarized in Table 3.1.

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 [7]. 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 [2], 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 3.2.

3.6 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 saved in the GRASS map storage and are very time consuming. To speed up

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 3.2: Summary table for parameters regulating physical forces.

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 3.7.

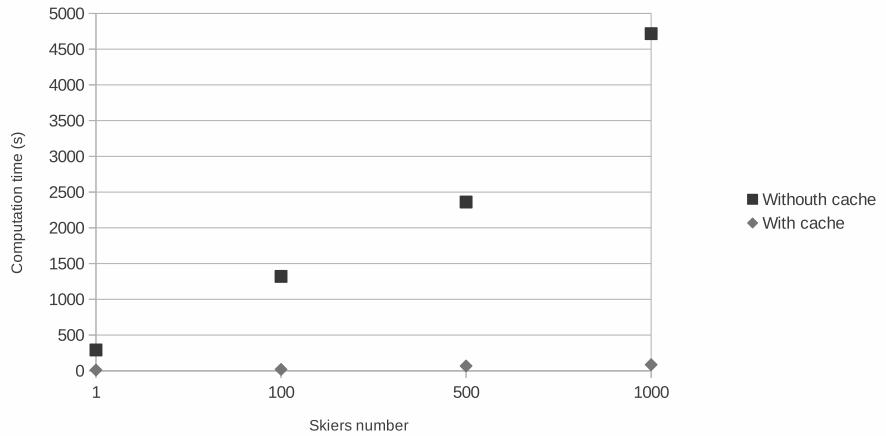


Figure 3.7: Computation time comparison between simulations with cache and simulation without it.

Chapter 4

Steps to validate the model

4.1 Field data sampling

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

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 other skiers position tracking. The goal is to incentive more users to use SkiLogger, collecting 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 at 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

$$v_i = \frac{\|x_i - x_{i-1}\| + \|x_{i+1} - x_i\|}{t_{i+1} - t_{i-1}}. \quad (4.1)$$

The second step preprocessing 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 preprocessing the data is to isolate the single down-hill 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):

1. 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 step is taken to eliminate the noise 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 signal.

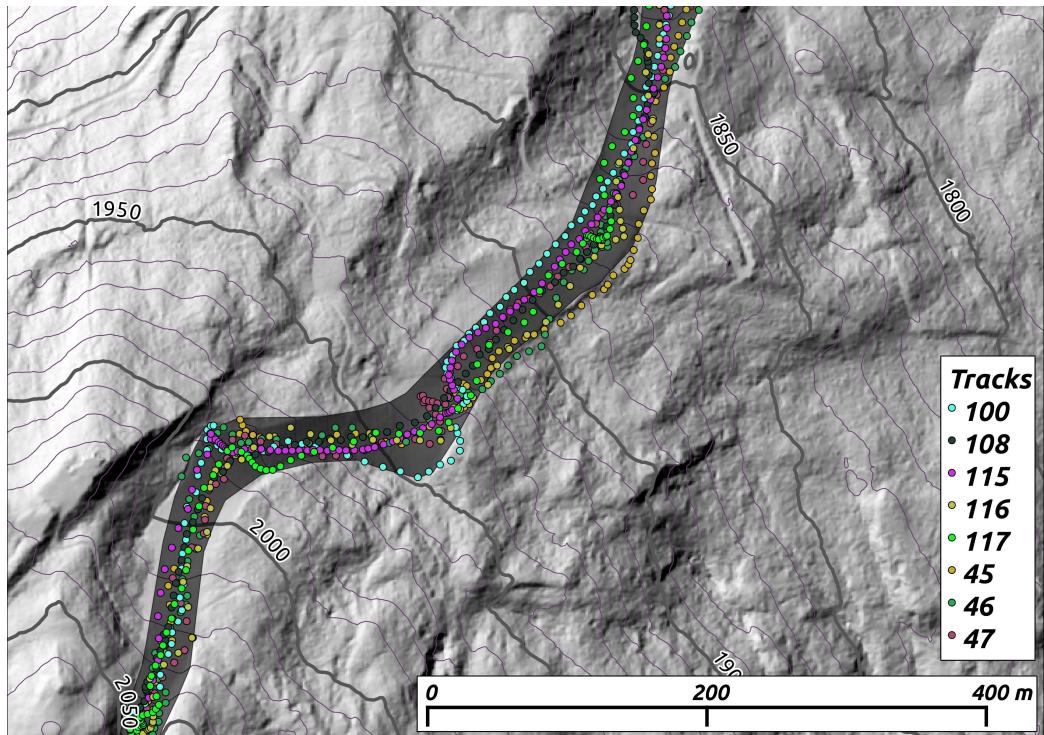


Figure 4.1: Real skiers data logged with the mobile application SkiLogger on the ski slope “Uno di Tognola” in San Martino di Castrozza.

2. 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 4.1 shows an example of the data recorded with the tracks highlighted. In some sections of the slope gps inaccuracies can be noticed. Sometimes discrepancies can also be due 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 4.1).

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

Table 4.1: 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. SkiLogger application asked at the beginning a self-assessment of the skiing skill, the kind of tool used (skis, snowboard, telemark...), the dimension of the group and the age class 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 4.2).

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

Table 4.2: Summary table for skiers that have used SkiLogger.

The main issue with 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

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

Table 4.3: Table summarizing the number of different skiers for the ski slopes.

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 4.3).

The data collected does not allow to do a robust tuning of the parameters and to validate the model for ski slopes in Trentino. However a initial test of the data simulated by the model has been performed using the data collected.

4.2 Simulated vs. real data

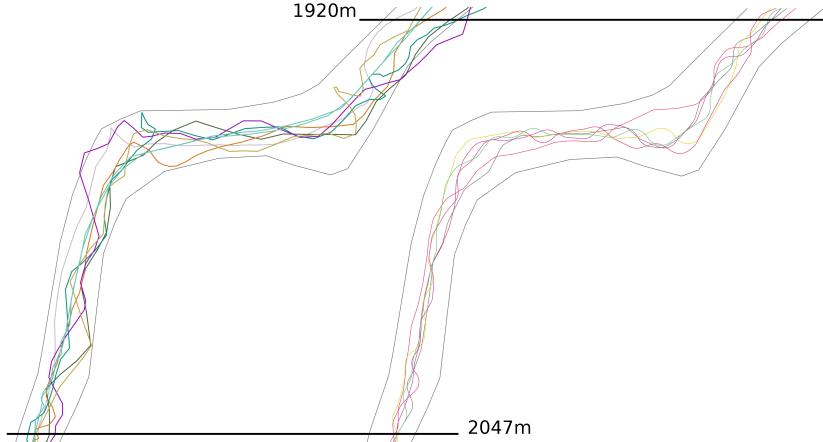


Figure 4.2: Simulated and recorded trajectories on the ski slope “Uno di Tognola”.

This part of the analysis focuses on the difference between the speed estimated by the simulation and the speed recorded by the SkiLogger application. The data collected are to few to analyze the map of skiers density produced. Starting from the map of simulated speed (see Fig.3.6a), produced as described in section 3.4, the values of speed computed for each point tracked by the application were compared to the values of speed of the map. Figure 4.3 shows the comparison between real and simulated values of speed for the

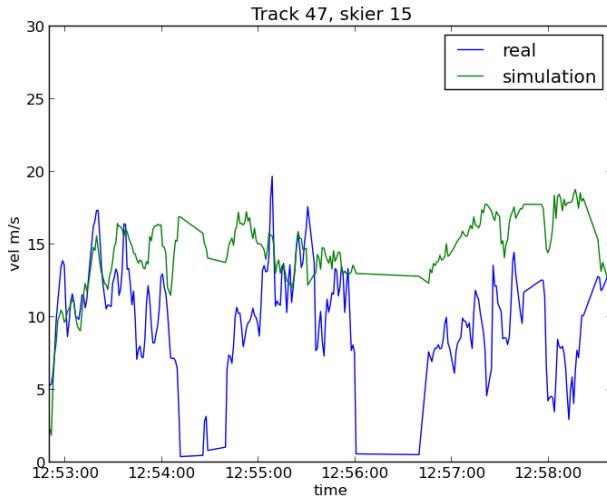


Figure 4.3: Comparison between simulated and real speed for the track 47 on the ski slope “Uno di Tognola”.

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. 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.

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 4.4 shows the map of the error made for each point of the tracks. The points in the map where the speed error is higher are where skiers stopped. It can be seen that the error produced by stops reverberates on the path sections following them.

To compare the speed values without the additional error due to 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 4.5 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. Overall, the average error decreased from 5.56 m/s to 2.52 m/s.

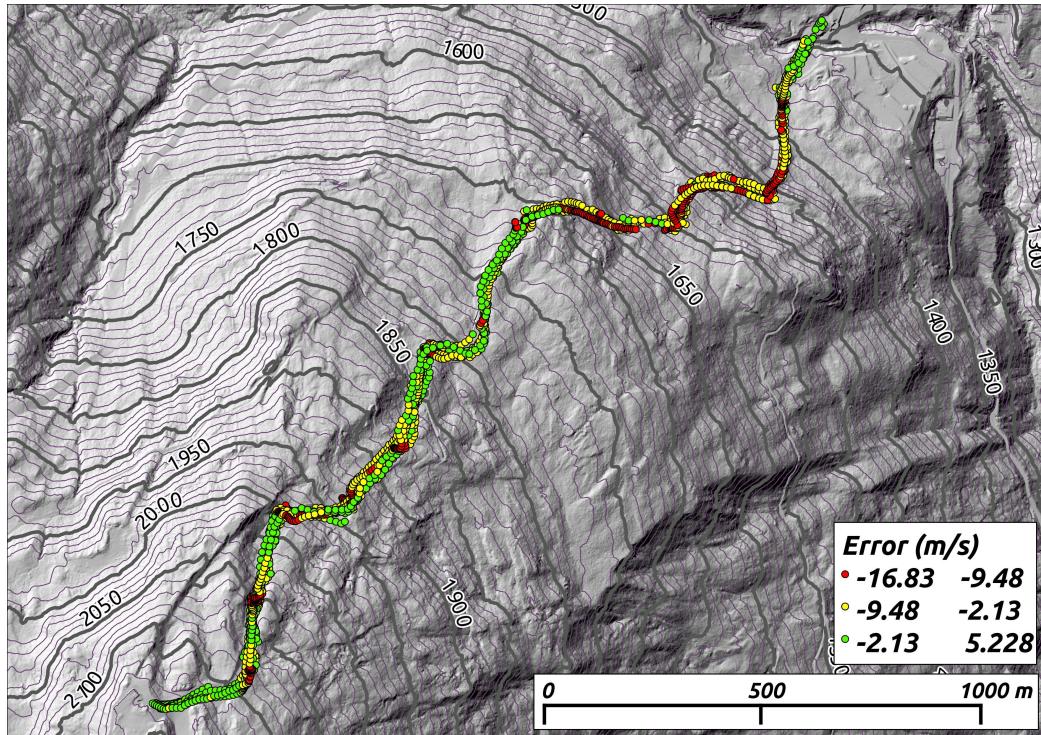


Figure 4.4: Map of the errors between the real and the simulated speed on the tracks recorded on the ski slope “Uno di Tognola”.

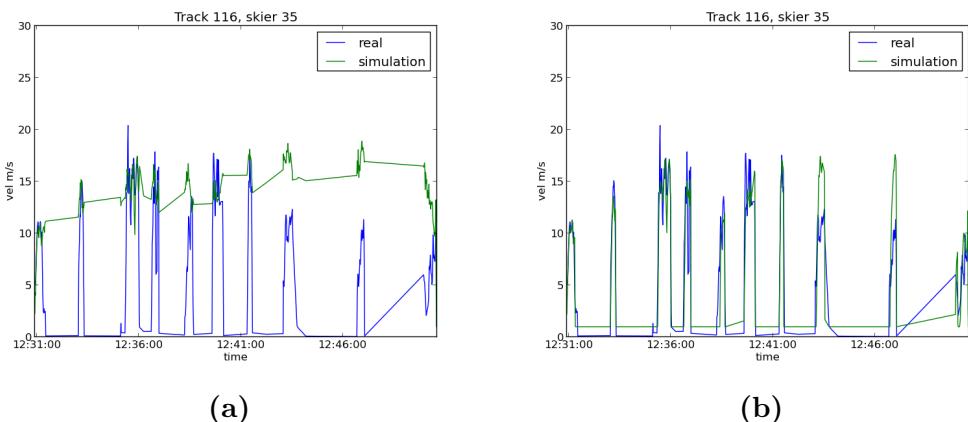


Figure 4.5: Simulated and real speed values for track 116 of “Uno di Tognola”. Figure 4.5a is without stops, in Figure 4.5b simulated skiers were forced to stop as the real skier.

4.3 Simulated ski lessons

In the data collected there were two recorded tracks related to a ski lessons. To make the model simulating this kind of situations an adaptation was realized. There are many kind of ski lessons. The first distinction is between individual and group lessons. In group lessons instructors usually choose the trajectory and is followed by the skiers that try to emulate their movement; alternatively the instructor performs some turns, stops and observes the skiers performing the turns one at a time. The ski lessons simulation focused on the group lessons in the case that skiers involved in the lesson form a queue behind the instructor.

The instructor was simulated as an expert skier, the skidding factor was set to $\eta = 0$ and the parameter v_{high} was kept low to force them making more turns. Additionally the time between the choice of two consecutive waypoints is higher than usual and also the turning radius is increased.

The other skiers were modeled as non-experienced skiers, with the skidding factor η assuming different values. These skiers have forced waypoints. Precisely, each time the instructor begins a turn a waypoint is set in this position for all the skiers. In this way each skier tries to reach each waypoint fixed by the instructor being at the same time subjected to the other social forces such as the repulsion from the other skiers. If a waypoint becomes

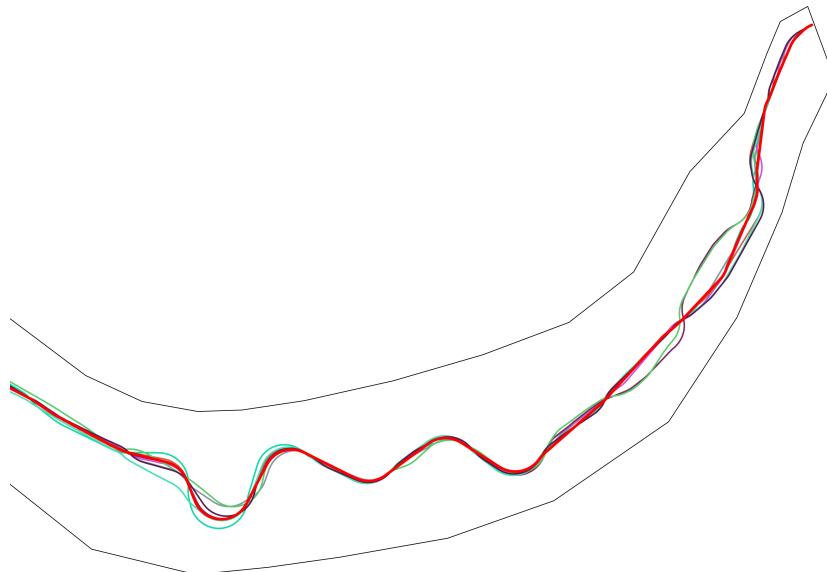


Figure 4.6: Trajectories of a simulated ski lesson on the ski slope “Rododendro” of San Martino di Castrozza. The red line is the instructor trajectory.

unreachable the skier passes to the following one. In the case that at a certain time no waypoints are available, the waypoint is set to the instructor position. To avoid being too distance from the other skiers the instructor reduces their speed according to the distance between the nearest skier and the farthest one. Figure 4.6 shows the simulated trajectories for a ski lesson.

Chapter 5

Conclusions and future work

A particle-based model to simulate the riding behavior of skiers descending ski slopes was implemented. The model was originally proposed by Holleczek and Troster [7], the following new features were introduced:

- Non-experienced skiers were included in the model. By calibrating a newly introduced parameter, the skidding parameter η , intermediate and beginner skiers can be simulated.
- The waypoints selection has been made dynamical, the new strategy implemented allows each skier to choose waypoints during the descent, based on their position, the shape of the trail, the slope of the terrain and the skier speed.

Additionally an adaptation of the model has been implemented to simulate ski lessons. In these simulations a variable number of skiers follow the instructor trying to reproduce his or her trajectory.

An Open Source GRASS module has been released. This module implements the model described using C++ and the GRASS GIS Library. The module is an executable that can be run in a GRASS environment. The code can be found at http://github.com/pollo/skiing_traffic.

The simulations were used to compute indicators about the skiing traffic on the slopes. In particular, the average speed and density of skiers have been investigated. Two raster maps were produced: one reporting the average speed of the simulated skiers for each pixel of the map and another one indicating the density of the skiers in every location of the ski slope.

The model has been tested on real data collected by the mobile application SkiLogger. For the ski slope “Uno di Tognola” in San Martino di Castrozza, 8 recorded tracks were available. A comparison between the simulated and the recorded speed showed an average error for single track that ranged between

4.28 m/s and 6.44 m/s, while recorded speeds ranged from 0 to 30 m/s. By introducing a new parameter that forced simulated skiers to stop in some predetermined areas, the average error for a specific track was decreased from 5.56 m/s to 2.52 m/s.

For the next ski season a new version of the SkiLogger application will be released offering new features to the skiers. Once a critical mass of data will be collected, a complete calibration and validation of the model for the Trentino ski slopes will be made. Moreover, the possibility of collecting more empirical data exploiting the potential of wearable-sensors will be investigated.

In future developments the repulsion from obstacles will be included in the simulation, by acquiring information about the physical obstacles on the ski slopes. Moreover, depending on the skiers ability, some areas of the ski slopes will be considered obstacles to make the skiers avoid trajectories perceived as dangerous (for example sheets of ice or high slope areas).

So far, there are no studies validating the prediction of skiers density on the ski slope made by this kind of simulations. If a large amount of data will be recorded in the next season, a validation of this part of the model will be performed. Alternatively, a different approach can be followed using video records of some selected ski slopes from which computer vision algorithms can extract the skiers trajectories [15].

To predict in a more precise way the skiers density, new aspects related to the skiers behavior will be investigated. So far, the model does not differentiate between different skiers attitudes. While riding downhill, the skiers choice of the trajectory is influenced by the perceptions they get from the environment. These are not limited to other skiers and edges repulsion but also deal with the personal attitude of the skier. For example a cautious skier will predict more carefully the possible trajectories of other skiers, will examine the condition of the slope and of the snow to choose the less risky trajectory possible. On the other hand, a more daring skier will choose a more challenging trajectory that gives him the possibility to test his or her skills. Lastly the behavior and the dynamics of skiers moving in groups will be explored. In the future, the collection of more empirical data will enable the investigation of the foundations of these conjectures.

Bibliography

- [1] J. Bourbousson, C. Seve, and T. McGarry. Space-time coordination dynamics in basketball: Part 2. the interaction between the two teams. *J Sports Sci*, 2010.
- [2] 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.
- [3] P. A. Burrough and R. A. Mcdonnell. *Principles of Geographical Information Systems*. Oxford University Press, USA, 2 edition, Mar. 1998.
- [4] 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.
- [5] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5):4282–4286, May 1995.
- [6] T. Holleczeck, H. Harms, and G. Tröster. Detection of snowboarding moves with sensorized bindings. In *Science and Skiing V*, pages 645–652. Meyer & Meyer Verlag, 2012.
- [7] T. Holleczeck and G. Tröster. Particle-based model for skiing traffic. *Physical Review E*, 85(5):056101, May 2012.
- [8] U. D. Jentschura and F. Fahrbach. Physics of Skiing: The Ideal-Carving Equation and its Applications. *Canadian Journal of Physics*, 82:249–261, 2004.
- [9] T. McGarry. Identifying patterns in squash contests using dynamical analysis and human perception. *International Journal of Performance Analysis in Sport*, 2006.

- [10] 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.
- [11] 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.
- [12] M. Neteler and H. Mitasova. *Open Source GIS: A GRASS GIS Approach*. Springer, 3rd edition, 2008.
- [13] J. Palut and P. Zanone. A dynamical analysis of tennis: concepts and data. *J Sports Sci*, 2005.
- [14] P. Passos, D. Araújo, K. W. Davids, L. Gouveia, S. Serpa, J. Milho, and S. Fonseca. Interpersonal pattern dynamics and adaptive behavior in multiagent neurobiological systems : conceptual model and data. *Journal of Motor Behavior*, 41(5):445–459, 2009.
- [15] K.-U. Schmitt, P. Niemetz, and M. Muser. Analyzing the riding behavior of recreational skiers and snowboarders. *ASTM*, 2010.

Appendix A

Simulations summary values varying parameters

Kinetic Friction	Travel time	Average speed	Maximum speed
0.05	166s	17.31m/s	23.23m/s
0.06	176s	16.38m/s	23.08m/s
0.07	194s	14.82m/s	22.45m/s
0.08	210s	13.63m/s	22.08m/s
0.09	224s	12.78m/s	20.59m/s
0.10	238s	12.04m/s	20.14m/s
0.11	247s	11.60m/s	20.32m/s
0.12	259s	11.04m/s	19.94m/s

Table A.1: Average travel time, average skiers speed and maximum skiers speed values simulating 20 skiers on the ski slope “Uno di Tognola” for different values of kinetic friction coefficient.

v_{high} parameter	Travel time	Average speed	Maximum speed
25m/s	237s	12.12m/s	20.97m/s
20m/s	240s	12.01m/s	20.44m/s
15m/s	242s	11.97m/s	20.44m/s
10m/s	244s	11.85m/s	20.30m/s

Table A.2: Average travel time, average skiers speed and maximum skiers speed values simulating 20 skiers on the ski slope “Uno di Tognola” for different values of the v_{high} parameter.

Mass	Travel time	Average speed	Maximum speed
110Kg	222s	13.01m/s	21.47m/s
100Kg	229s	12.56m/s	21.31m/s
90Kg	236s	12.17m/s	20.54m/s
80Kg	243s	11.87m/s	20.15m/s
70Kg	252s	11.42m/s	19.16m/s
60Kg	264s	10.87m/s	18.62m/s

Table A.3: Average travel time, average skiers speed and maximum skiers speed values simulating 20 skiers on the ski slope “Uno di Tognola” for different values of skiers mass.

Travel time	Average speed	Maximum speed
248s	11.61m/s	19.63m/s

Table A.4: Average travel time, average skiers speed and maximum skiers speed values simulating adolescents skiers on the ski slope “Uno di Tognola”. Mass was set to 50Kg and the frontal area A to $0.4m^2$.