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

Corso di Laurea in Informatica

Final Thesis

Analysis and implementation of a particle-based model for skiing traffic

Relatore interno:
Prof. Alberto Montresor
Relatore esterno:
Dott. Cesare Furlanello

Laureando:
Matteo Poletti

Anno accademico 2012-2013

Contents

Introduction

Chapter 1

The model

The model implemented was proposed by Holleczek and Troster [2]. It can be classified as a two-dimensional microscopic-driven many-particle system with the constraint that skiers are exposed to gravity and centripetal forces.

The skiers are modeled as particles with a specific mass, m, that are exposed to forces. The model distinguishes between two kind of forces: the physical forces are those forces that regulate the skier motion determining the acceleration while the social forces determine the skiers behavior. In the following, the position of a particular skier at time t is represented by the vector $r(t)$, $\dot{r}(t) = \frac{d}{dt}r(t)$ is his or her speed and $e_{\dot{r}}(t) = \dot{r}(t)/\|\dot{r}(t)\|$ the direction of motion.

1.1 Social forces

The social forces model the decisions made by a skier that is descending a slope. They are used to determine whether the skier should perform a turn, however they do not act on the acceleration. The social forces are dimensionless, the superposition of all the forces, F_{social}, gives the desired direction of the skier $e_{social}(t) = F_{social}/\|F_{social}\|$. If the desired direction $e_{social}(t)$ diverges from the skier direction $e_{\dot{r}}(t)$ more than an angle δ the skier start turning to adjust his or her direction (see Fig.1.1).The social forces are used to model the repulsion from the edges and the obstacles, the repulsion from other skiers and to attract the skiers towards the destination chosen.

To describe the social force that attracts the skier towards the destination chosen the model assumes that each skier, during the descent, selects several waypoints $x_a^1....x_a^n$ 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

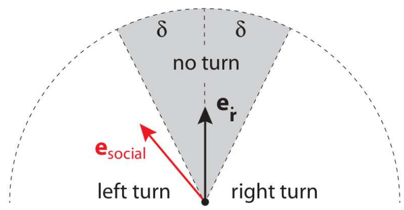

Figure 1.1: (From [2]) When the angle between the current direction of motion $e_{\dot{r}}$ and the desired direction e_{social} is bigger than δ the skier performs a turn to adjust his or her direction

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

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

$$F_D(r_a) = A_0 e_a(t) \tag{1.2}$$

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 more stronger when the skier is closer to the edge. At each position r_a the skier a is subjected to a repulsion force from the left edge and to a repulsion force from the right edge. 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

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

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

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

The model takes into account also the natural 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 skier repulsion force that a skier b imposes on the skier a can be expressed as

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

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

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

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

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

In general, the repulsion social forces act on the 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

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

To summarize, the social forces that applies on a skier a are

$$F_D(r_a) = A_0 e_a(t), \tag{1.9}$$
$$F_L(\dot{r_a}, r_{aL}) = w(\dot{r_a}, -r_{aL}) F_L(r_{aL}), \tag{1.10}$$
$$F_R(\dot{r_a}, r_{aR}) = w(\dot{r_a}, -r_{aR}) F_R(r_{aR}), \tag{1.11}$$
$$F_A(\dot{r_a}, r_{ab}) = w(\dot{r_a}, -r_{ab}) F_A(r_{ab}), \tag{1.12}$$
$$F_O(\dot{r_a}, r_{ao}) = w(\dot{r_a}, -r_{ao}) F_O(r_{ao}) \tag{1.13}$$

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

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

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

1.2 Physical forces

As described in [2], there are two major techniques of performing turns: *skidding* and *carving*. During carving turns the direction of motion is exclusively

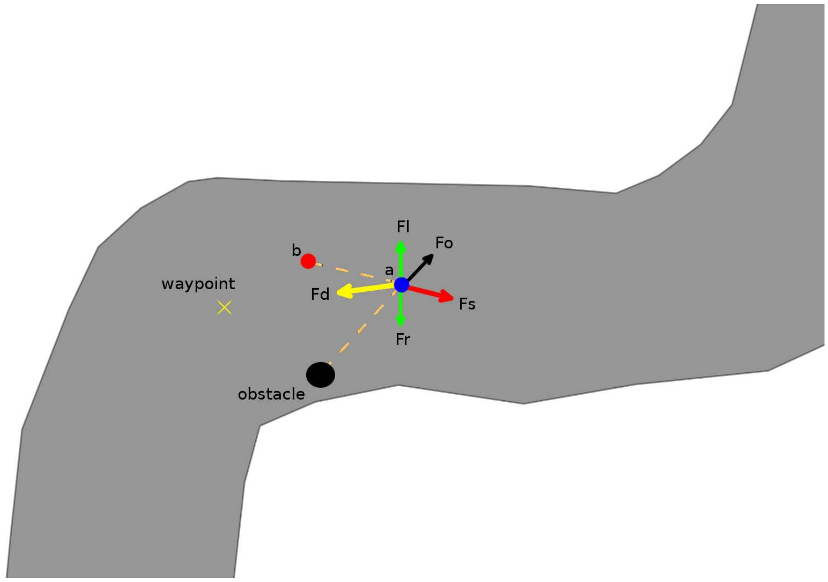

Figure 1.2: Diagram of the social forces. The forces F_R and F_L keep the skier a away from the edges, the force F_O repels the skier from the obstacle, the force F_S repels a from the skier b and the force F_D attracts the skier towards the waypoint.

parallel to the skis while in skidded turns there is an additional slippage to the side. Carved turns are usually performed by expert skiers, while beginners and not experienced skiers tend to perform skidded turns.

In [2] skiers are supposed to perform turns with a radius corresponding to the *sidecut radius* of their skis. Although some studies [3] [1] have proposed a more realistic model of carving turns, investigating more deeply the effects of the penetration of the skis in the snow and of the skier tilt angle, for 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 $e_{\dot{r}}$. Consider a skier at position r with speed $\dot{r}$ and direction of motion $e_{\dot{r}}$ and let n denote the surface normal on the ski slope at r. At r, the slope has an inclination angle of

$$\alpha = \arccos([0,0,1] \cdot n) \tag{1.14}$$

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

$$\gamma = \arcsin[(\sin\alpha)(\sin\beta)] \tag{1.15}$$

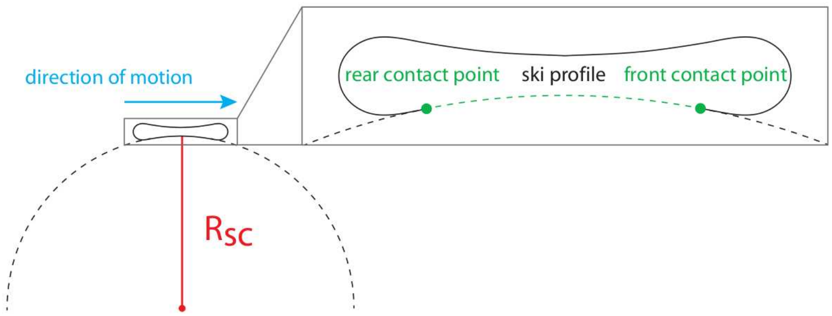

Figure 1.3: (from [2]) Profile of a carving sking with the sidecut radius and the turning radius evidenced.

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

To compute the force accelerating the skier the gravitational force, the friction forces and the centripetal forces should be investigated. First the gravitational force F_G is considered: it can be expressed as

$$F_G = mg \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \tag{1.16}$$

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

$$F_G = F_S - F_N \tag{1.17}$$

The normal force F_N can be expressed as

$$F_N = mg(\cos\alpha)n \tag{1.18}$$

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

$$F_S = F_P + F_{lat} \tag{1.19}$$

The downhill force F_p can be written as

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

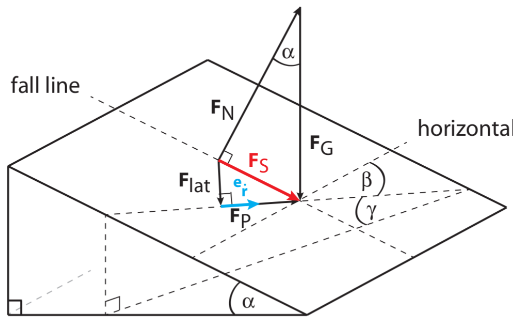

Figure 1.4: (from [2]) The downhill force F_S can be decomposed into the downhill force F_P, acting parallel to the current trajectory, and into the lateral force F_{lat} acting perpendicularly to the direction of travel.

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

Remembering 1.17, the downhill force F_S can be computed as

$$F_S = F_G + F_N \tag{1.21}$$

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

$$F_{lat} = F_S - F_P \tag{1.22}$$

The centripetal force F_C a skier is exposed to during turns can be written as

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

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, the effective force should be defined. The effective force is the force that has 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

$$F_{eff} = F_{lat} - F_N \tag{1.24}$$

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

$$F_{EFF} = F_{lat} - F_C - F_N \tag{1.25}$$

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

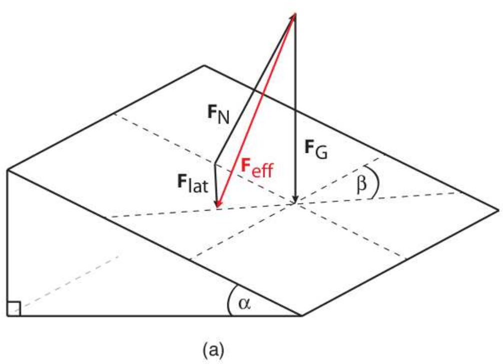

Figure 1.5: (from [2]) In (a) effective force during the descent on a straight line ($F_{eff} = F_{lat} - F_N$). In (b) effective force during a carved turn ($F_{EFF} = F_{lat} - F_N - F_C$)

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

$$F_{ground} = -\mu \|F_{eff}\| e_{\dot{r}} \tag{1.26}$$

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

$$F_{GROUND} = -\mu \|F_{EFF}\| e_{\dot{r}} \tag{1.27}$$

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

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

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

$$F_{net} = -F_P + F_{air} + F_{ground} \tag{1.29}$$

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

$$F_{NET} = -F_P + F_{AIR} + F_{GROUND} + F_C \tag{1.30}$$

1.3 Limitations

The model proposed describes the motion of an expert skier that is performing perfect carved turns. The turning radio has been 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 allow skiers to stop their descent. Moreover skiers are not allowed to jump nor to exit the ski slope. If a skier collides with an edge of the slope it is reflected back with an angle equals to the angle at which he or she has collided.

1.4 Differences from the original model

The physical model of the skiers motion has been keep equals to the one exposed in [2]. Between the social forces, the destination force F_D (see. 1.2) is of critical importance. Its action depends on the selection of the waypoints. In the original paper the waypoints were selected randomly every 50m, using a uniform distribution on the corresponding line from the left to the right edge of the slope (see Fig.1.6).

A more dynamical approach in the selection of the waypoints has been considered to better model the choices that a skier takes descending a slope. The new strategy allow each skier to dynamically choose waypoints during the descent, basing on the position of the skier, on the shape of the trail, on the slope of the terrain and on the skier velocity.

In the following 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. Then the vectors given the direction toward the edges are $e_{aR} = \left(r_a^L - r_a\right)/\|r_a^L - r_a\|$ and $e_{aL} = \left(r_a^R - r_a\right)/\|r_a^R - r_a\|$. Let α be the angle between e_{aR} and e_{aL} defined as

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

where n is the normal of the plane containing e_{aR} and e_{aL}. The skier a chooses the new waypoint with uniform distribution on a fraction of the angle

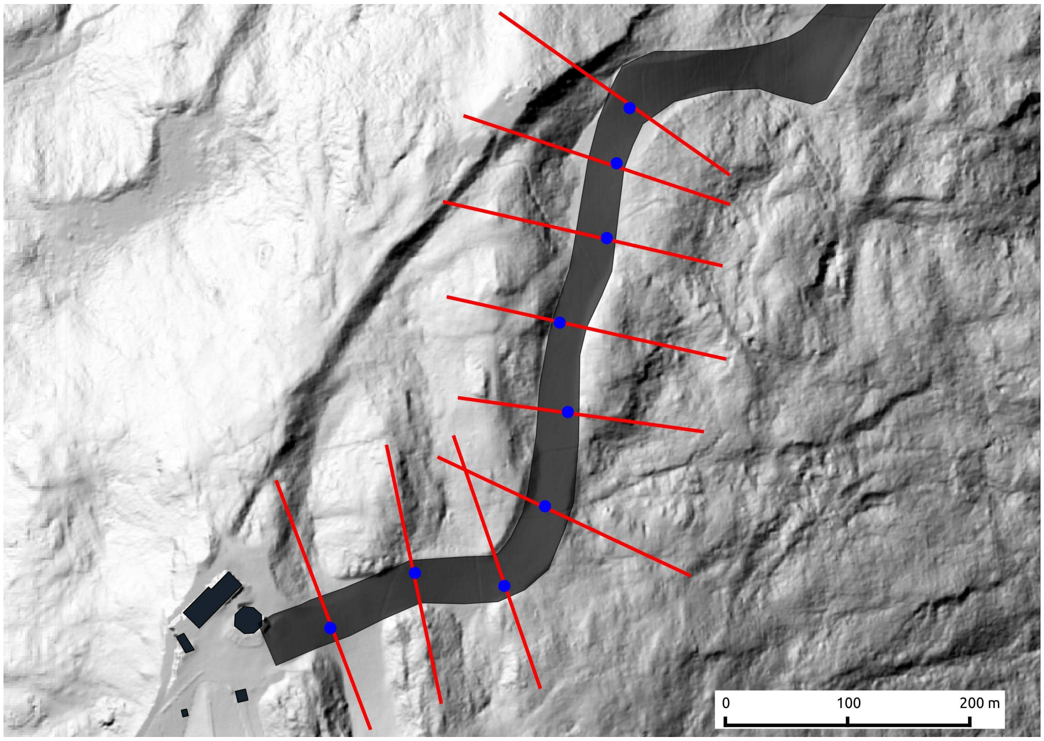

Figure 1.6: In [2] waypoints were selected randomly every 50m following a uniform distribution on the line from the left to the right edges of the slope.

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

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

where ρ is the distance at which waypoint 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 choosing a new waypoint, or when the skier has traveled more than D meters after having chosen the last waypoint.

Designing the new mechanism to the selection of waypoints some requirements it should satisfy has been individuated:

1. Speed should influence the choice of the new waypoint: when skiers are traveling fast they tend to perform more turns than when the speed is

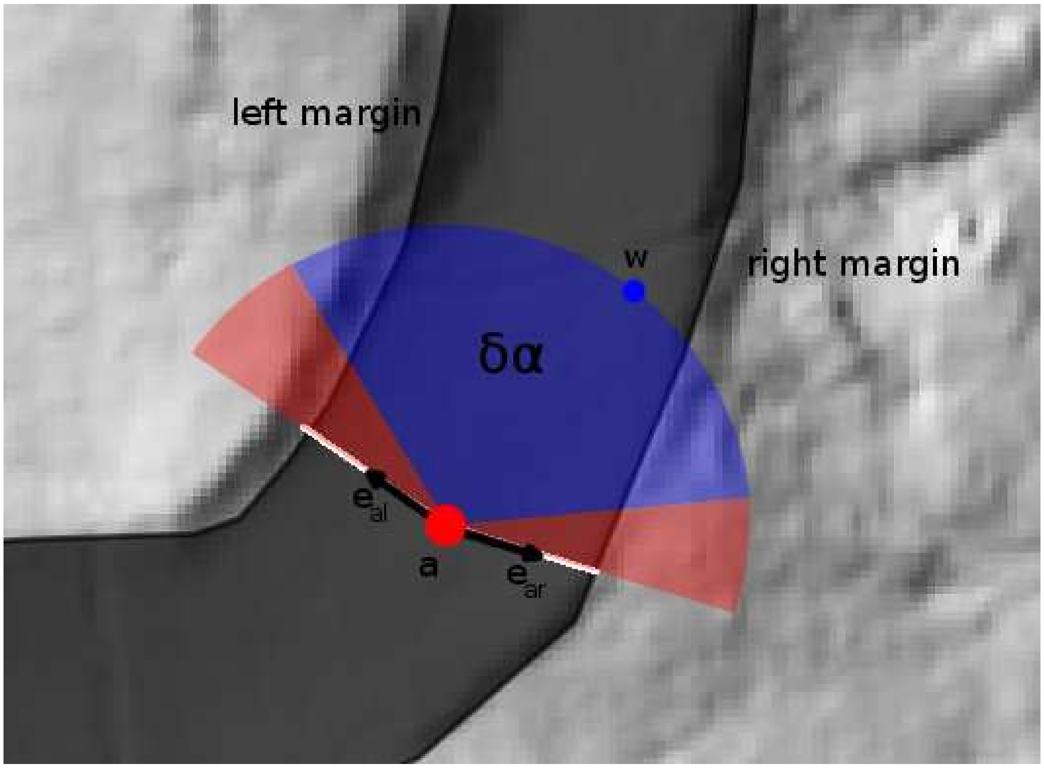

Figure 1.7: Selection of a waypoint: the skier a selects the new waypoint w choosing with uniform distribution on the angle $\delta\alpha$, a fraction of α. The angle α is the angle between e_{aR}, the vector representing the direction towards the right edge, and e_{aL}, the vector towards the left edge,

low.

2. The selection of the new waypoint should depend on the slope that the skier is going to encounter: before flat areas skiers tend to avoid turns to increase their velocity.

3. The 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.

4. Skiers usually avoid to choose a direction that would make them pass 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 fulfill this requirements it is possible to act on the parameter δ of the equation 1.32. When δ 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.

Requirement 1 can be satisfied by making the parameter δ depend linear on the speed v of the skier. Moreover it is required that when the speed v is near 0 then the width of the angle should be itself near to 0 and when v is near to a value of speed considered high v_{max} the angle should have maximum width. Therefore, δ can be set to

$$\delta = \frac{v}{v_{max}} \tag{1.33}$$

To fulfill the requirement 2 an additional factor depending on the slope should be considered. Let s be the slope that the skier is going to encounter and let s_{lim} a value of slope that is considered small enough to require an additional acceleration by the skier. Then if $s < s_{lim}$ the width of the angle in which to choose the new waypoint should be narrowed again. Taking into account 1.33 then we can write δ as

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

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

The requirement 3 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.

Finally, to satisfy the requirement 4 manipulating δ is not enough. The minimum turning radius of the skiers gives a bound to their capacity of avoiding a collision and of avoiding 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 because it exposes its GIS engine in form of libraries that were used 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. The main technologies considered were PostGIS, QGIS, GDAL/OGR and GRASS GIS.

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 problem with PostGIS is that it does not have dedicate 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 in PostGIS.

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. Unfortunately, the functionalities offered 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 are those for C++. OGR provides also slightly less complete API for C and Python API, although they are not really well documented. The problem using GDAL/OGR is especially 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 also contains 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 (both raster and vector) 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 they can be used from the GRASS GUI or from the command line.

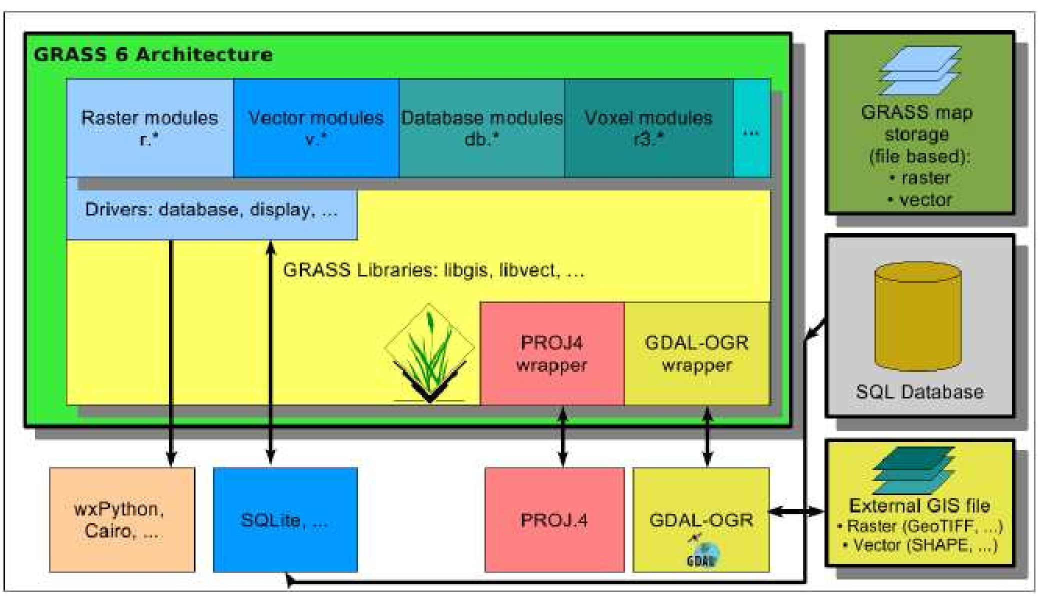

Figure 2.1: (from http://grass.osgeo.org
GRASS software architecture

GRASS allows two basic levels of programming. The most simple approach is to use script programming to call the high-level GRASS modules.

A more advanced approach is to access the low-level functionalities of the GRASS GIS library trough the C-API exposed. Since grass modules are executables, the first approach requires to spawn a new process each time a spatial computation is required. For the implementation of the model presented in 1 this is not feasible. Even if the second approach forces the implementation to use GIS operations at a lower lever it has many advantages: it gives support for database routines (GRASS file management), projections, raster data management, area, line and point vector data management [6] and it guarantees good performances.

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 choose 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, and to keep the code 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 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 needs to perform some specific GIS computation. The implementation wanted to avoid dependency between the code related to the simulation and the code written to access the GRASS GIS library. 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 implemented. This class declares

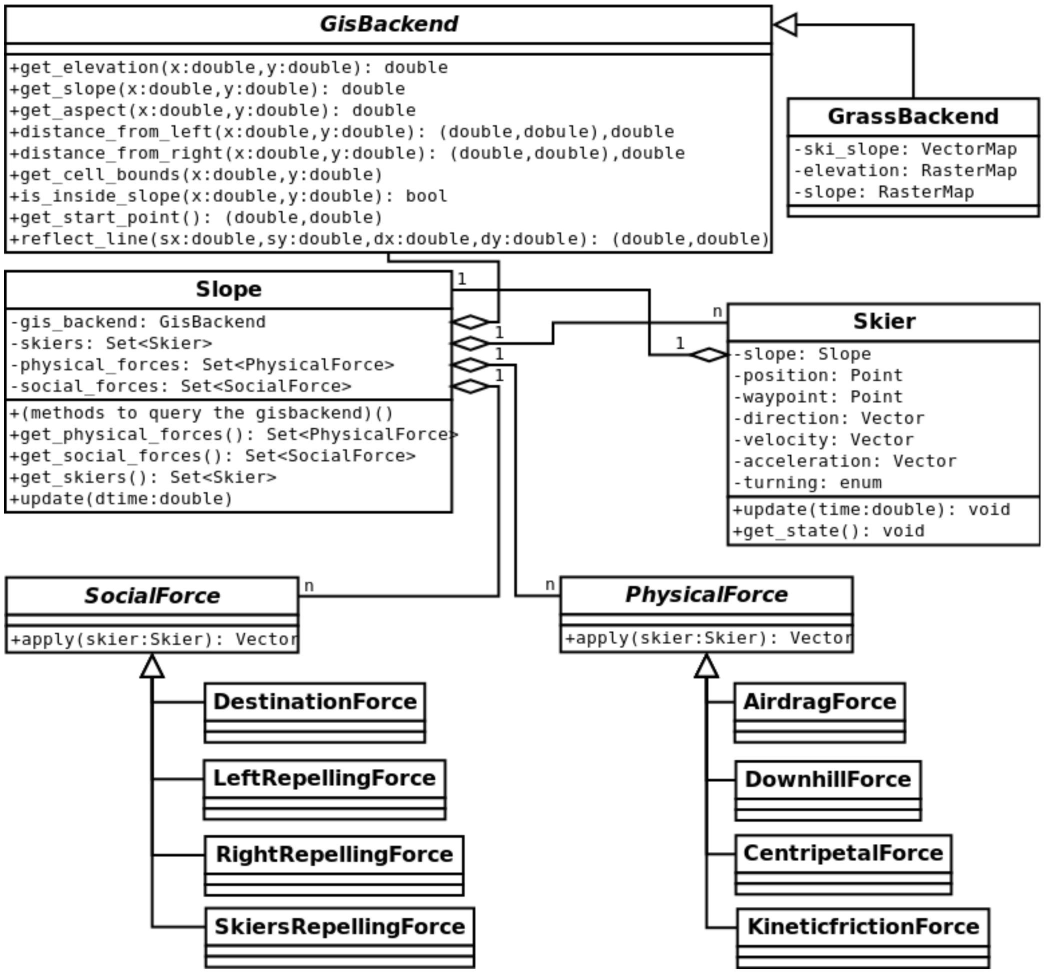

Figure 2.2: Class diagram

the interface that whatever GIS backend used to run the simulation should implement. This include methods to get elevation, slope and aspect for given coordinates, methods to get the distance of a point from the edges of the slope and the locations on the slope closer to the given point, methods to give random start points at the top of the ski slope and 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 actually used in the code when spatial computation are needed. The class GrassBackend perform the operations needed using the GRASS GIS Library. It requires to have available a DEM (Digital Elevation Model) raster map for the elevation, the slope and the 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. As Digital Elevation Model was a Digital Terrain Model of Trentino at resolution of 1x1m and 2x2m obtained used Lidar technology. TODO:specify better Lidar and DTM

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 consider two limit cases to describe the profile of the snow. Given a surface $y(x)$ and a snowfall of h meters. The first cases describes a new profile were the snow is supposed adhere perfectly to the terrain: $y_s(x) = y(x) + h$. The second cases consider the snow to behave like a liquid and create 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, but 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

$$Y(x) = (1 - l)y_s(x) + ly_L \tag{2.1}$$

where l is a parameter describing how much the snow has a behavior liquid-like. By the point of view of the implementation, the hard thing is to compute the profile y_L. For this purpose some GRASS modules was 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 modules, r.sim.water, is an overland flow hydrologic simulation based on path sampling method [5]. 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 by the consideration that on a ski slope there are three main factors that determine the distribution of the snow: 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 is approximated with a smoothed surface. Starting from the original DTM raster map a vector map was produced where each original value of elevation were represented by a point. Then the GRASS module v.surf.rst was used to interpolate a new elevation map using regularized spline with tension [4].

Figure 2.3 shows the profile 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 returns a feasible profile for a liquid-like behavior. This can be due to a misconfiguration of the parameters of the module or maybe to a too demanding level of precision. Finally, the second strategy was chosen.

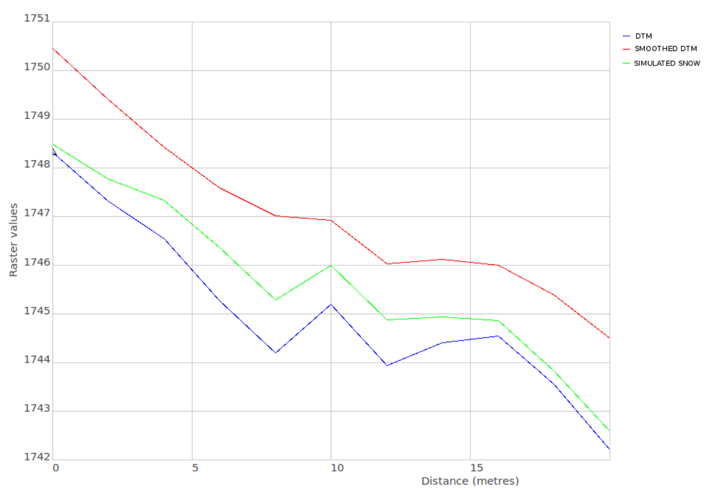

Figure 2.3: Profiles of the dtm and of the simulated dtm after rain fall

pista 3d

2.4 Output data

2.5 optimization

parameters selection

Bibliography

[1] 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.

[2] T. Holleczek and G. Tröster. Particle-based model for skiing traffic. *Physical Review E*, 85(5):056101, May 2012.

[3] U. D. Jentschura and F. Fahrbach. Physics of Skiing: The Ideal-Carving Equation and its Applications. *Canadian Journal of Physics*, 82:249–261, 2004.

[4] 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.

[5] 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.

[6] M. Neteler and H. Mitasova. *Open Source GIS: A GRASS GIS Approach.* Springer, 3rd edition, 2008.