RWTH AACHEN UNIVERSITY

Towards a Wearable Snowboarding Assistant

Diploma Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Christian Guggenmos

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr.-Ing. Klaus Wehrle

Registration date: June 14th, 2007
Submission date: Dec 20th, 2007

Contents

List of Figures

List of Tables

Abstract

Acquiring sport skills can be difficult, time-consuming, and frustrating, especially for novices. We have initiated a project to investigate how wearable computing can support snowboarders in their learning process. Due to spatial separation during riding exercises, a snowboard instructor usually cannot give feedback to his students on their mistakes immediately. Feedback is only possible when instructor and student are close to each other.

Body-worn sensors on the students could detect wrong movements in real-time and give direct feedback. This might increase the students' awareness of their mistakes and thus decrease learning time.

This thesis is the initial step to develop a *Wearable Snowboarding Assistant*. By interviewing snowboard instructors and reviewing instructional literature we identified four mistakes common for beginners. We selected suitable hardware components and developed approaches to detect these mistakes. In an iterative design process, we have developed a mobile wearable prototype robust enough to be taken on the slope. To show the feasibility of automatic mistake detection, we have conducted a user study with snowboard beginners and evaluated their sensors recordings.

Überblick

Das Erlernen einer Sportart kann mühsam, zeitaufwändig und frustrierend sein. Im Rahmen eines größeren Projektes untersuchen wir, wie mit Hilfe von Wearable Computing der Lernprozess von Snowboardern unterstützt werden kann.
Aufgrund der räumlichen Distanz zwischen Snowboardlehrer und Schüler, bekommen Schüler während ihrer Fahrübungen häufig kein unmittelbares Feedback von ihrem Lehrer. Dies ist nur zwischen den Übungen möglich, wenn sich Lehrer und Schüler in Reichweite voneinander befinden.
Sensoren die am Körper der Schüler angebracht sind, könnten fehlerhafte Bewegungsabläufe dagegen in Echtzeit erkennen und sofortiges Feedback geben. Wir sind der Ansicht, dass der Schüler dadurch fehlerhafte Bewegungen eher zur Kenntnis nimmt und effektiver lernt.
Diese Arbeit ist der erste Schritt in unserem Projekt zur Realisierung eines *Wearable Snowboarding Assistant*. Aufgrund von Interviews mit Snowboardlehrern und der Recherche von entsprechender Literatur konnten vier Fehler identifiziert werden, die als typische Anfängerfehler gelten. Um diese Fehler mit Hilfe von Sensoren zu erkennen, mussten geeignete Hardware und Verfahren zur Fehlererkennung entwickelt werden.
In einem iterativen Entwicklungsprozess entstand ein für die Piste geeigneter mobiler und robuster Prototyp. Die Realisierbarkeit einer automatischen Fehlererkennung wurde durch einen Benutzertest mit Snowboardanfängern und den daraus gewonnen Sensordaten gezeigt.

Acknowledgements

A lot of people contributed to this thesis and should not be forgotten here:

First of all, Prof. Dr. Jan Borchers for the opportunity to write this thesis at his chair. I really appreciated the good working atmosphere.

Daniel Spelmezan who offered me this interesting topic and had an open door whenever I needed advice.

Professor Wehrle, my second examiner, for offering his help on sensor related topics.

Julie Wagner, Max Möllers and Jonathan Diel who were patient test users even on the freezing cold slopes of SnowWorld.

Elaine Huang, Thorsten Karrer and Christian Mattar who found the time to proof-read parts of my thesis regardless of their tight time schedule.

Tim Stappen for his great comments on my initial drafts and for lending his name to the scenario persona.

Elisabeth Bak and Marianne Mokosch for suggesting idiomatic English expressions I never even heard of.

. . . and everyone else at the chair, for the pleasant time. Thank you!

Finally, I would like to thank my girlfriend Birgit Noack for her constant emotional and professional support during the ups and downs of my thesis.
You helped me a lot more than I can repay, thank you!

Last but not least, a big thank you to my parents Marianne and Gerhard Guggenmos.

Conventions

Throughout this thesis the following conventions will be used:

The plural “we” will be used throughout this thesis instead of the singular “I”, even when referring to work that was primarily done by the author.

The whole thesis is written in American English.

Chapter 1

Introduction

"We learn by example and by direct experience because there are real limits to the adequacy of verbal instruction."

—Malcolm Gladwell, Blink: The Power of Thinking Without Thinking, 2005

Learning sports in cooperation with an instructor

For almost every kind of sport it is essential to first learn the very basics. This needs to be done properly so that further improvement is possible. Therefore, it is common to seek the assistance of a professional instructor. In several lessons the instructor teaches important aspects of the specific sport, theoretically and practically. One of the most valuable and crucial attributes of an instructor is the supervision of his trainee and constructive feedback on his performance. The trainee's progress highly depends on the quality of the instructor's feedback.

Immediate feedback on tennis strokes

A tennis instructor, for instance, observes the strokes of his trainee and immediately after each stroke the instructor can tell him how to improve his technique. The instructor might even guide the trainee's hand to demonstrate the movement (see Figure 1.1). In other sport areas direct feedback might not be possible because of spatial separation of trainer and trainee. A sprinter, for example, will not get any feedback on his performance during his training run. The trainer will give advice afterwards.

Figure 1.1: Tennis lesson.

Snowboard instructor and students are spatially separated

When learning how to snowboard a similar spatial separation of instructor and student exists. The snowboard instructor usually explains how to perform a movement and demonstrates it. During demonstration he will move down the slope, away from his students. Thereafter the students try to repeat the demonstrated movement as an exercise. The instructor observes every student from the distance. He can only give advice after the exercise, when the students are close enough to talk to him. Thus, the instructor's feedback is not given immediately on the students' mistakes.

1.1 A Wearable Snowboarding Assistant

The wearable *Snowboarding Assistant*

Out of our experience with snowboarding we have initiated a project at the Media Computing Group[1] (RWTH Aachen University) to investigate whether immediate feedback during the exercises can help snowboard beginners in their learning process. A wearable system with sensors on the body, woven into the clothing or attached to the snowboard, could detect wrong movements and give real-time feedback. We believe that making the snowboarders aware of their mistakes and giving hints on how to correct them

[1] http://hci.rwth-aachen.de

during an exercise might increase learning speed. Throughout the whole thesis we will refer to such a system as the *Snowboarding Assistant*.

Based on our research in the application domain (see Chapter 4—"The Snowboarding Domain") we provide two scenarios. The first scenario illustrates the practices of today's typical snowboarding lessons. The second one envisions how lessons could be improved with the *Snowboarding Assistant*.

1.1.1 Scenario of Typical Snowboarding Lessons

Family on their two week winter vacation

Every year the dutch family van Stappen comes to Tyrol for their two week winter vacation. Their fourteen-year-old son Tim is very curious about trying out new things. This time he wants to try snowboarding. Although he has no experience with board sports, like skateboarding or surfing, he is confident that it will be fun. The van Stappens register their son for snowboarding lessons, and Tim cannot wait to get on the slope.

Three days of lessons

After having received their snowboard equipment Tim and eight other snowboard beginners meet their instructor Florian. He tells them that they will have lessons for two hours in the morning for the next three days. After the course they will be able to accomplish simple turns on the snowboard.

First day: the snowboard basics

First the students need to get familiar with the snowboard basics. They learn to strap on their bindings, how to fall down and stand up and how to move with their snowboard on flat ground.

The basic stance on a snowboard

Before continuing with further exercises, Florian explains and demonstrates his students the basic stance (see 4.1) on the snowboard. The knees should be bent to be able to compensate bumps on the slope. The upper body should be almost upright and the shoulders parallel to hip and snowboard. The weight should be distributed equally on right and left foot.

Tim's motivation decreases

After one hour the beginners learn to slide down a flat hill.

Tim falls down a couple of times but his ambition to learn snowboarding like he saw it on TV during the X-Games[2] keeps him motivated. Florian teaches his students how to slide down the hill on the snowboard's heel-side and toe-side edge. He also explains that the snowboard always slides and turns on the side with the greatest pressure. Tim has problems keeping his upper body upright. He feels a bit embarrassed in front of the other students because Florian repeatedly reminds him to straighten up his body. Tim's motivation is decreasing, but nevertheless he wants to give his best.

Second Day: students are not able to realize instructions immediately

On the next day the students learn how to traverse a hill on the edge, i.e., without letting the snowboard drift away. They continue with several exercises which prepare them for turns. At the end of the lesson, Florian explains and demonstrates how to do turns. His students try to follow his example but they are not able to realize Florian's instructions immediately and sometimes struggle to make a complete turn. Florian tells his group that they are going to improve doing turns in the next lesson.

Doing turns and resolving mistakes

On the last day the students revise what they have learnt so far. When they try to do turns, Florian observes everyone individually at a time and gives advice afterwards. Tim performs quite well, although sometimes he cannot accomplish a turn. While his back is facing downhill he is afraid to shift his weight on the front foot because this accelerates his snowboard. With his weight on the back foot, however, the snowboard does not turn easily. After every run Florian reminds Tim to move his weight on the front foot, but while riding, Tim often forgets this. He is too focused to keep his knees bent and his back upright and therefore does the same mistake again and again. Florian tells Tim that he is doing quite well. Yet, for improving his technique in the future he should try to resolve this last mistake.

[2] http:// expn.go.com/expn/index

1.1.2 Scenario of Snowboarding Lessons with the *Snowboarding Assistant*

Like in the previous scenario the van Stappens come to Tyrol for their winter vacation.

Tim's parents register him for snowboarding lessons. Florian, the snowboard instructor, asks Tim if he likes to rent the ordinary equipment or the new sensor-enhanced clothing that will analyze his movements and help him recognize his mistakes more easily. Tim is very interested in new technology, so he decides to give it a try. The clothing and boots, however, do not look special and Tim is a bit disappointed because he expected a futuristic suit. In addition to his clothing, Tim receives knee pads and a helmet, which also contain sensors and moreover help to prevent injuries.

Sensor enhanced clothing

Thereafter, the snowboarding lessons begin and the students Florian teaches his students the basics of snowboarding. At first Tim feels a little uncomfortable on the snowboard. When he practices to slide down the slope on the heel-side edge he cannot control his speed. Florian tells Tim to bend his knees further to gain more control over his snowboard. Tim *does* bend his knees, but not sufficiently.

Therefore, Florian uses his mobile phone to enable the sensor system attached to Tim's clothing, boots, helmet and knee pads. Florian tells Tim, that as long as his knees are not bent enough he will notice a vibration at his knees. After that Tim practices again and notices the vibration from his knee pads. At first Tim is surprised and does not know what to do. In the next run, however, he remembers what Florian has told him and bends his knees further until the vibration stops. Every time Tim does not bend his knees enough the vibration starts again. After a while Tim remembers to always bend his knees and Florian turns off the sensors.

Instant feedback reminds beginner of mistakes

The second day passed without Florian making use of the *Snowboarding Assistant*.

On the third day the beginners improve their turns. Tim manages to accomplish turns but Florian is not satisfied with his technique. He tells Tim that he is doing a good job but that he should focus on improving his turning tech-

Improving turns with the *Snowboarding Assistant* nique. To initiate turns on steeper slopes Tim would need to shift his weight more towards the front foot. Florian enables the sensors in Tim's boots to detect whether his weight is too much on his back foot. When initiating a turn, Tim's back foot now vibrates if his weight distribution is not optimal. Tim notices the vibration immediately and leans more on his front foot. In fact, turning becomes easier.

System can be used without instructor

Florian is satisfied with his student and lends Tim the mobile phone for the rest of the afternoon, so Tim can continue to practice even without an instructor. The settings on the phone are already adjusted and Tim only needs to choose on which of his mistakes he would like to focus.

1.1.3 Goals

As illustrated in the second scenario we envision two main aspects of snowboarding lessons to be improved by the *Snowboarding Assistant*.

1. Helping snowboarders to be aware of their mistakes immediately when they occur. This should decrease learning time and thereby reduce frustration.

   perceive their mistakes more easily *during* the exercise. This should increase learning speed, thereby reducing frustration.

2. Allowing beginners to exercise even without the presence of an instructor by providing a system that ensures the correct performance of movements. This should not substitute a human instructor but support beginners beyond the lessons.

To reach these goals several steps towards a wearable *Snowboarding Assistant* are necessary:

1. Building a wearable hardware platform with appropriate sensors which operates on the slope.

2. Identifying and detecting common mistakes in snowboarding.

3. Providing an adjustable interface to control the *Snowboarding Assistant* .

4. Giving appropriate real-time feedback to the students, e.g., audio or tactile feedback.

5. Showing the benefit of real-time feedback for the learning process.

1.1.4 Requirements

This thesis initiates the development of the *Snowboarding Assistant* and focuses on the first two steps. Therefore, the following requirements need to be fulfilled:

Exploring the Application Domain. We need a deep insight into the basic terms and techniques of snowboarding and teaching methods. This serves as a starting point for further development and provides the basic knowledge to people working on the project.

Opportunities for Change. We have to identify disadvantages in the communication between instructor and student. The focus should lie on beginners problems.

Robust Hardware. A hardware platform that withstands the conditions on a slope needs to be assembled. The hardware should not restrict the user's freedom of movement. Thus, we need to select robust and unobtrusive sensors to detect mistakes.

Algorithms. To give feedback to the student we must recognize his mistakes via sensors. Therefore, we need to find appropriate algorithms to process the sensor data.

1.2 Structure of the Thesis

According to the identified goals and requirements the remainder of the thesis is structured as follows:

Chapter 2—"Sensor Technology" gives an overview of different sensor types used in the development of the *Snowboarding Assistant*.

Chapter 3—"Related Work" discusses projects which deal with context awareness, health care and sports.

Chapter 4—"The Snowboarding Domain" provides an overview of the basic terms, concepts and techniques of snowboarding. Moreover, we identify common beginner mistakes and discuss further aspects of the teaching process as results of interviews with snowboard instructors.

Chapter 5—"A Lab Prototype" describes the necessary steps to build a first prototype. Based on the results of the previous chapter, we have developed a wired lab prototype to stimulate further ideas. The chapter discusses choice and placement of sensors and first algorithms to detect mistakes.

Chapter 6—"A Mobile Prototype for the Slope" documents the building of a wireless prototype. The chapter discusses the hardware setup as well as problems we faced on the slope.

Chapter 7—"User Study and Data Analysis" takes a closer look at the sensor data recorded on the slope. After conducting self-tests we recorded snowboard beginners' sensor data and discuss how the second prototype could detect common mistakes.

Chapter 8—"Summary and Future Work" sums up the results of the previous chapters. As this thesis is only the initial step in the *Snowboarding Assistant* project, the chapter outlines the following steps.

Appendix A—"Interview Guideline" contains the original German version of the interview guideline we followed during our interviews with snowboard instructors. We provide an English translation as well.

Appendix B—"MAX/MSP Patches" includes screenshots of the software we developed for the first prototype in Chapter 6—"A Mobile Prototype for the Slope".

Appendix C—"Smoothing Filters" provides the formulae for the smoothing filters we used during the evaluation of the sensor data.

Chapter 2

Sensor Technology

"A sensor is a device that receives a stimulus and responds with an electrical signal."

—[Fraden, 2003, p. 2]

In this chapter we give an overview of relevant sensors used in wearable computing[1]. This overview is important for the projects discussed in Chapter 3—"Related Work" and for the decisions we have made for our own project.

For every sensor we discuss its measurand, working principle, and any special characteristics that need to be considered for its application.

2.1 Accelerometer

Accelerometers measure rate of change of velocity

Accelerometers measure acceleration, i.e., the rate of change of velocity, along a designated axis. 2-D accelerometers combine two single axis accelerometers to measure acceleration on two orthogonal axes. Analogously, 3-D accelerometers combine three orthogonally arranged single axis accelerometers.

[1] More in depth information on sensors can be found in [Fraden, 2003].

Acceleration measured through forces on a tiny proof mass

An accelerometer can be imagined as a ball in a tube which is fixed with a spring at each end (see Figure 2.1) [van Laerhoven et al., 2003]. When the tube is accelerated along its longitudinal axis, the ball will lag behind the movement of the tube due to inertia. This causes the ball to change its position relative to the tube. The change of position is proportional to the acceleration and can be thought of as the sensor's output. Many of today's accelerometers are built using MEMS[2] (Micro-Electro-Mechanical Systems) technology, which allows the 'ball' to be a tiny proof mass of less than 0.1 micrograms [Riedel, 1993].

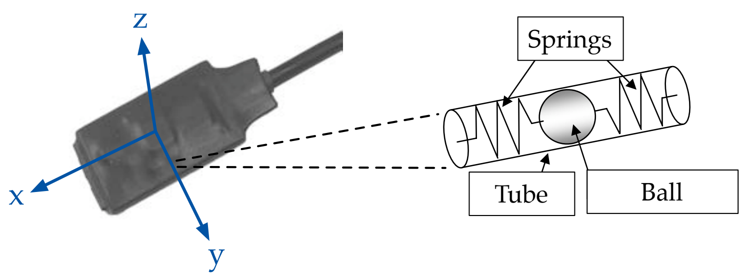


Figure 2.1: 3-D accelerometer Infusion Systems[a] and working principle. Based on [van Laerhoven et al., 2003].

[a] http://infusionsystems.com

Measurement is sensitive to gravity

An accelerometer is sensitive to gravity. Thus, its output is the sum of dynamic and static acceleration, i.e., acceleration due to movement and due to gravity.[3] When the sensor is at rest it measures exclusively the gravitational acceleration and can be used as a tilt sensor. The angle between the sensor's axis and the gravity vector can be computed with basic geometry (Figure 2.2). Many wearable computing projects use accelerometers to measure movements because of their small composition and low price (see Chapter 3—"Related Work").

[2] For further information see http://www.memsnet.org.
[3] This is about $9.81\frac{m}{s^2}$ in Europe at sea level.

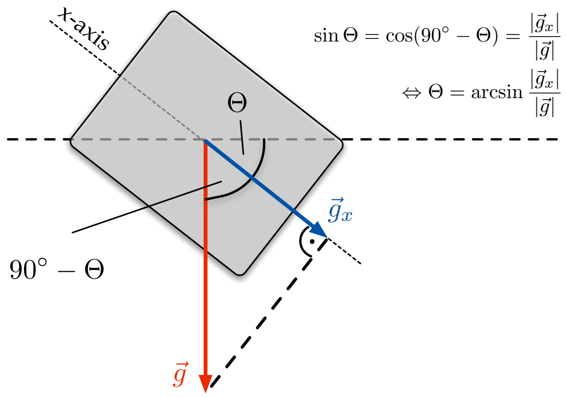


Figure 2.2: Gravitational force on a tilted accelerometer. The accelerometer measures only the projection $\vec{g}_x$ of the gravity vector $\vec{g}$ on the x-axis. Hence, with $|\vec{g}| = 9.81\frac{m}{s^2}$ the angle Θ can be computed.

2.2 Gyroscope

Gyroscopes are sensitive to rotational speed, i.e., they measure angular velocity relative to a designated axis. Similar to accelerometers, 2-D and 3-D gyroscopes combines orthogonally arranged single axis gyroscopes.

Gyroscopes measure angular velocity

Today's MEMS gyroscopes measure the 'Coriolis acceleration', which can be explained as follows: Consider a person standing at point ① in Figure 2.3 on a rotating platform with tangential velocity v_1 relative to the (non-rotating) ground. If this person walks to point ②, away from the center of rotation, its tangential velocity will increase to v_2. The acceleration that causes this increase is the Coriolis acceleration. It is proportional to the angular velocity of the rotation [Geen and Krakauer, 2003].

Coriolis acceleration occurs during angular movements

To measure the Coriolis acceleration, gyroscopes contain a tiny mass which vibrates up and down in a fixed frame. When the mass is moving up (away from the center), it will be accelerated towards the right. This will exert a force on the frame to the left (Figure 2.4 (a)) as the mass is fixed within the frame. Vice versa when moving down (towards

Measuring the Coriolis acceleration with a tiny mass

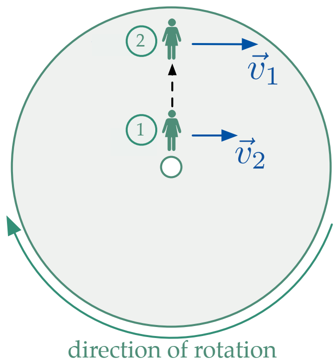


Figure 2.3: A person on a rotating platform. If she moves from point ① to point ② she will notice a tangential acceleration, the Coriolis acceleration. Taken from [Geen and Krakauer, 2003]

the center), the mass will exert a force on the frame to the right (Figure 2.4 (b)). This force is measured to indicate angular velocity as they are proportional to each other.

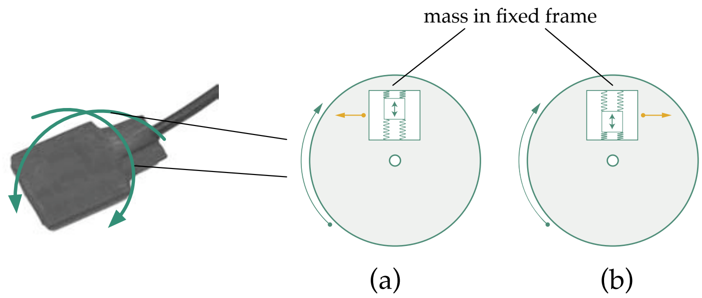


Figure 2.4: 2-D gyroscope 'Spin2D' (I-CubeX) and the working principle of a MEMS gyroscope. A mass in a fixed frame is vibrating up and down. Because of the Coriolis acceleration the mass exerts a force on the frame (orange vector) proportional to the angular velocity of the rotating plane.

Gyroscopes are not sensitive to gravity

Coriolis acceleration only occurs during rotational movements. Thus, gyroscopes are insensitive to linear accelerations and movements. In contrast to accelerometers, they are not affected by gravity.

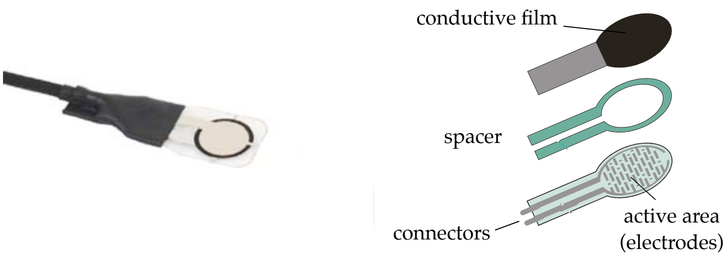
Use in car safety systems

Gyroscopes can be used to monitor rotation of machines, airplanes, or vehicles. They are often used in vehicle safety systems.[4]

2.3 Force Sensitive Resistor (FSR)

FSRs measure pressure

Despite its name, *force* sensitive resistor, an FSR's electrical resistance drops proportional to the amount of *pressure* applied to its surface. Hence, its outcome depends on the amount of force applied as well as the area covered by the force. FSRs are very thin (ca. 0.2 mm) and consist of different layers as depicted in Figure 2.5.



(a) (b)

Figure 2.5: (a) I-CubeX[a] TouchMicro. (b) Different layers of an FSR.[b]

[a] http://infusionsystems.com
[b] http://www.interlinkelectronics.com

Higher pressure results in lower resistance

The connectors on the bottom layer lead to an 'active area' that consists of interdigitating electrodes printed on a flexible substrate. A conductive film is printed on the top layer separated from the bottom layer by a plastic spacer. If no pressure is applied to the sensor's surface, top and bottom layers are not in contact. This results in a high resistance of the active area, as its interdigitating electrodes are not connected. The more pressure is applied the more of the active area is pushed against the conductive film which leads to

[4] For example, the Electronic Stability Control (ESC) to prevent skidding of cars (http://www.chooseesc.eu/)

a connection of the electrodes. Thus, the resistance of the sensor drops.[5]

FSRs need to be mounted on flat surfaces

The distance between the top and bottom layer can also be decreased by flexing the sensor. Therefore, it should be mounted on a flat surface to eliminate mistakes through deformation.
FSRs are used in a wide range of application areas where force or pressure needs to be measured, e.g., in the automobile industry for measuring a tire's pressure footprint[6] or in the medical industry to analyze a patient's force distribution under the feet.[7]

2.4 Bend Sensor

Bend sensors, also known as flex or flexion sensors, are long, thin (ca. 0.1 mm, Figure 2.6(a)) sensors and change their electrical resistance proportionally to their flexion.

Resistance drops according to the flexion

Most bend sensors consist of a conductive ink printed on a flexible substrate. The ink is very brittle, hence flexion of the sensor results in micro gaps within the ink (Figure 2.6 (d),(e)). Higher flexion causes greater gaps and decreases the conductance of the ink resulting in a higher resistance.[8]

Sensor reading depends on bend angle *and* radius

Gloves for virtual reality applications

The outcome of a bend sensor depends on both the flexion *angle* as well as the flexion *radius* — a smaller flexion radius will cause greater gaps (Figure 2.6(d)). A bend sensor of this type only responds to one bending direction. Bending it in the opposite direction does not cause any gaps and thus the sensor's resistance remains unchanged (Figure 2.6(c)). Bend sensors are often used in input gloves for virtual reality environments to determine the finger's flexion, e.g., the CyberGlove® II[9] .

[5] http://www.electrade.com/html/produkte/sensorik_fsr.htm
[6] http://www.tekscan.com/industrial/tirescan-system.html
[7] http://www.tekscan.com/medical/systems.html
[8] http://www.flexpoint.com/technicalDataSheets/mechanicalDesignGuide.pdf
[9] http://www.immersion.com

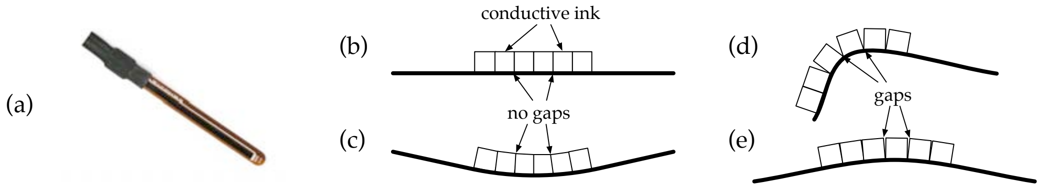


Figure 2.6: Different flexion states of a bend sensor.

2.5 Inertial Measurement Unit (IMU)

IMUs combine accelerometers and gyroscopes to calculate absolute orientation angles

Inertial Measurement Units (IMU) measure orientation in 3-D space based on an initial state, e.g., in Euler angles. IMUs usually consist of a 3-D accelerometer and a 3-D gyroscope with their axes aligned parallel. As long as the IMU is not moving, its orientation can be inferred with the 3-D accelerometer relative to the gravity vector. Yet, when the IMUs is moving, the measured acceleration is the sum of dynamic and static acceleration, which is difficult to separate. As gyroscopes are not sensitive to linear accelerations, they are used to keep track of changes in the orientation. Dedicated sensor fusion algorithms, which are usually implemented on the IMU, combine the readings of the accelerometer and the gyroscope to infer the absolute orientation relative to the initial state [Bachmann, 2004].

The magnetic field of the earth serves as reference

The absolute orientation in terms of 'world coordinates' cannot be derived with accelerometers and gyroscopes alone. Moreover, as the calculation of the current orientation is always based on the previous one, any error in the sensor readings is accumulated through the whole calculation. To account for these shortcomings, several IMUs incorporate a 3-D magnetometer to measure earth's magnetic field as a static reference. Thereby global coordinates can be calculated.

There are several manufacturers that offer IMUs with built-in processing capabilities, thus minimizing errors with specialized data fusion algorithms. Examples are the *MTx* from XSens,[10] the *InertiaCube3* from InterSense,[11] and the

[10] http://www.xsens.com
[11] http://www.isense.com

SHAKE SK6.[12] Figure 2.7 shows the SHAKE SK6 and the angles it calculates relative to a fixed coordinate system. The SHAKE SK6 can also be used as a digital compass which returns values between 0 to 360° in the x–y plane independent of the SHAKE's orientation.

Figure 2.7: SHAKE SK6 IMU and its axes. It calculates orientation with respect to a fixed coordinate system.

[12]http://www.samh-engineering.com

Chapter 3

Related Work

"Not to know what has been transacted in former times is to be always a child. If no use is made of the labors of past ages, the world must remain always in the infancy of knowledge."

—Cicero (106 BC–43 BC)

The research projects presented in this chapter deal with body-worn sensors to measure human movements. They are structured by application domain:
First we present projects in the broad field of context awareness, i.e., being aware of the user's surroundings. Thereafter we discuss work done in the health care sector, which is another promising application area for wearable technology. To finish the discussion of related work we present research projects in the application domain of sports.

3.1 Context-Awareness

3.1.1 Definition and Examples

Context is not clearly defined

Body-worn sensors are often used to identify the wearer's context. Context has been defined differently by several authors. Abowd et al. define context as

> *"[...] any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."*
>
> [Abowd et al., 1999, pp. 3–4]

This broad definition includes the user's physical, social, emotional or informational state.[1] Systems that take context into account when providing services to the user are called 'context-aware'.

Context aware mobile phone

For instance, Schmidt et al. [1999a] have built a hardware platform that incorporates several sensors, among others, a 2-D accelerometer, a light sensor, and a temperature sensors. With a combination of the sensor readings they infer different contexts of a mobile phone, e.g., whether the phone is in the user's hand or in a bag. Accordingly the ring tone profiles are automatically adjusted.

Activity recognition with wearable sensors

In recent years several projects have focused on 'activity recognition', i.e., recognizing the user's activity, which is an important part of the user's context.[2] Laerhoven and Cakmakci [2000] and Ravi et al. [2005] both try to recognize different everyday activities of the user, like sitting, standing or walking, with only one accelerometer. Similarly Lester et al. [2005] try to identify basic activities with one sensor node that incorporates different sensor types.

3.1.2 Multi-Sensor Activity Context Detection for Wearable Computing

Activity recognition for real-world applications

Kern and Schiele [2003] have built their own sensor hardware on top of the Smart-Its hardware platform [Beigl et al., 2003] for activity recognition. Instead of building a lab prototype they have designed their system to be used in a real-world setting. As potential applications they envision the

[1] For a more discrete classification of context see [Schmidt et al., 1999b].

[2] Other aspects of context are, e.g., the user's location, or his heartrate, depending on the application.

domains of sports and manual work. Targeting these applications they draw requirements on their hardware platform which apply to the *Snowboarding Assistant* as well. The most important requirements are robust hardware and proper fixation of the sensors at the desired location, as this heavily influences the quality of the sensor data. Furthermore, the user's freedom of movement should not be restricted.

Robust hardware is essential

To fulfill these requirements, Kern and Schiele cover their sensors with shrink wrap and attach them with velcro straps for tight fixation (Figure 3.1(c)). To be able to move around freely, Kern and Schiele put the laptop that is connected to the sensor hardware into a backpack (Figure 3.1(a)). A PDA connected to the laptop starts and stops sensor recordings and is used for online data annotations. Data is later analyzed off-line to extract information.

Hardware setup

(a) (b) (c)

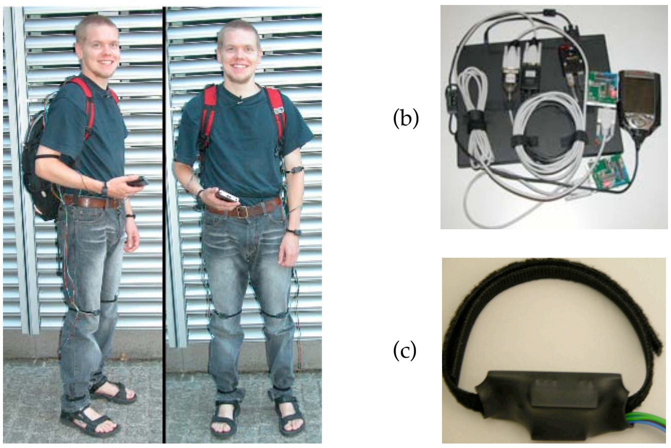


Figure 3.1: The wearable system introduced in [Kern and Schiele, 2003]: (a) Person wearing the sensors and holding the PDA, (b) The system and its components, (c) Shrink-wrapped sensor with velcro strap

Unlike the previously mentioned projects that try to infer the user's activity with only one sensor [Ravi et al., 2005, Lester et al., 2005], Kern and Schiele have decided to attach several sensors at different locations specific for the intended activity.

Sensor placement according to the application

In a first experiment, they aim at identifying activities such as sitting, standing, walking and hand-shaking. Based on

Initial experiment

these target activities they attach 3-D accelerometers to major joints of the human body at the following locations: ankle, knee, hip, wrist, elbow and shoulder.

For complex activities several sensors are needed

The experiment shows that simple activities, like standing or walking, can be recognized using only one sensor on the leg. However, for more complex activities, like walking downstairs, the combination of sensors at different locations improves the recognition rate.

3.2 Health Care

Monitoring the patient's health condition

The application of wearable computing technology in the health care sector has been explored in several projects. A wide range of research projects focus on monitoring the patient's health condition [Anliker et al., Dec. 2004, Oliver and Flores-Mangas, 2006]. Most of them raise alarm in case of dangerous changes in the measured parameters, especially for elderly patients [Najafi et al., 2003, Degen et al., 2003].
The following projects, however, analyze movements and postures. We present them because they are more related to the concept of the *Snowboarding Assistant*.

3.2.1 GaitShoe

Two methods for gait analysis

Usually gait can be analyzed using two different methods: Either in a motion laboratory with computer-based methods like optical tracking or in an office with a clinician observing the patient. The first approach results in highly accurate data but is expensive. The second method, although being less expensive, yields highly subjective data depending on the clinician.

GaitShoe fills gap between traditional methods

Bamberg et al. [2007] propose the *GaitShoe*, a wearable system that falls in between the two methods and combines their benefits. It provides accurate data and can be used in the patient's natural environment. The *GaitShoe* can be attached to any shoe to analyze the wearer's gait. To measure gait-relevant parameters Bamberg et al. have incorporated several types of sensors. An overview is shown in Figure 3.2.

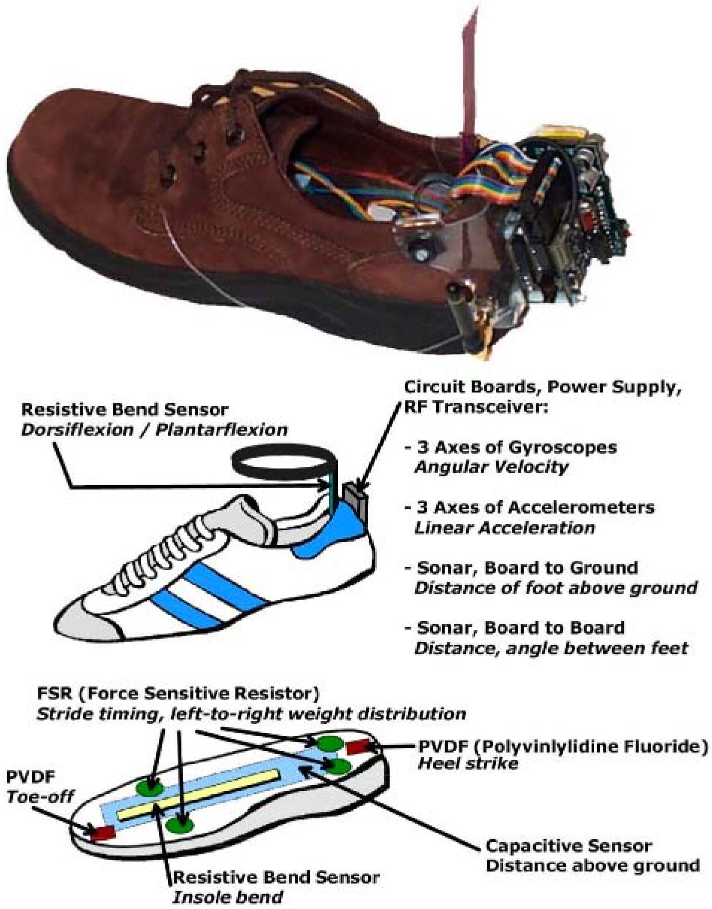


Figure 3.2: Overview of the *GaitShoe* and used sensors. Taken from [Paradiso et al., 2004].

The FSRs, PVDFs[3] and bends sensors are collocated on one insole. The gyroscopes, accelerometers as well as the microcontroller, the power supply, and the antenna to transmit data to a base station are placed on the back of the shoe (Figure 3.2).

Comparison with traditional method

The *GaitShoe* has been used simultaneously with a traditional gait analysis data acquisition system for compari-

[3] Polyvinylidene fluoride (PVDF) strips — sensors that react on dynamic pressure.

son. The *GaitShoe* has been proven successful in distinguishing the gait patterns of healthy persons and subjects with Parkinson's disease. In addition, determining the heel-strike and toe-off timing, i.e., when the foot touches or leaves the ground, was highly successful using the FSRs and PVDFs. For the stride length and the pitch of the foot the *GaitShoe* integrates the values of the gyroscopes. Due to the imprecisions of the gyroscopes and the compounded effect in the integration only fair result could be provided.

Auditory feedback for therapeutic purposes

Gait is analyzed in real-time and gait analysis has been explored to be used for therapeutic purposes through auditory feedback [Paradiso et al., 2004]. To provide rhythmic cues on how to walk, ambient music is played. Whenever a gait defect is detected the music becomes less melodic, encouraging the subject to return to a steady pace.

3.2.2 Biofeedback Wireless Wearable System

Farella et al. have contributed several projects to the wearable computing community. In their recent research they have developed wireless sensor nodes to track human gestures [Barbieri et al., 2004] and detect human body postures with a body area sensor network [Farella et al., 2006].

Optimizing balance through audio feedback

Based on their previous work, they have introduced the 'Biofeedback Wireless Wearable System' (*Bio-WWS*). This system detects a human's posture and gives audio feedback to help optimizing balance, e.g., to support the rehabilitation of patients that have lost their sense of balance [Brunelli et al., 2006].

Rehabilitation with cumbersome machines

Similar to gait analysis, current rehabilitation practices for balance monitoring are carried out with cumbersome and expensive machines. These devices need to be controlled by an expert and cannot be operated by the patient alone. The *Bio-WWS*, however, is designed for autonomous and unobtrusive usage.

Hardware setup: PDA, headset and sensor nodes

The current setup consists of a PDA, a Bluetooth headset, three sensor nodes each with a 3-D accelerometer, and a gateway. The sensor nodes attached to trunk, thigh and calf measure acceleration and forward the values wirelessly to the gateway. The gateway collects the data of the different sensor nodes and sends them to the PDA via Bluetooth. The software for creating auditory feedback resides on the PDA and sends the created audio stream wirelessly to the headphones via Bluetooth.

'Target Region' and 'Limit Region'

The range of acceleration values where the patient is in good balance is called the 'Target Region' (*TR*) (Figure 3.3). The *TR* is subject specific and needs to be calibrated at the beginning of each monitoring session. Therefore the patient nees to stand still for 10 seconds while the system samples the acceleration values from the attached sensors. Based on experiments, Farella et al. set the *TR* to 1.5 times the standard deviation of the samples collected during the calibration process. Similarly they set a so-called 'Limit Region' (*LR*) to 10 times the standard deviation of these samples.

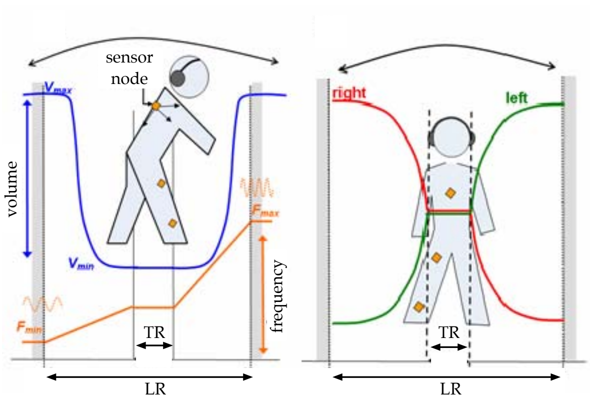


Figure 3.3: The *Bio-WWS*: As soon as the patient leaves the Target Region (*TR*) the audio feedback is modulated to guide the patient back to the *TR*. The maximum modulation is achieved at the borders of the 'Limit Region' (*LR*) [Brunelli et al., 2006].

Working principle of audio feedback

Within the *TR* the audio stream is not modulated at all. The maximum modulation is achieved at the borders of the *LR*. The more the patient leaves the *TR* the more the sound gets modulated: volume and frequency modulation for forward vs. backward leanings, left–right audio balance modulation for left vs. right leanings (Figure 3.3). This should guide the patient back to the *TR*.

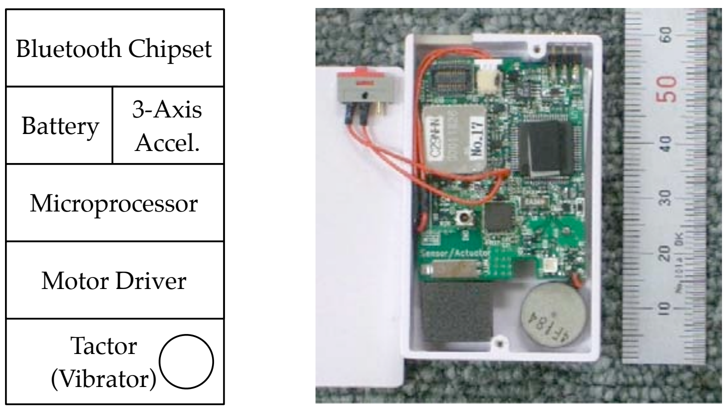
Farella et al. have conducted an evaluation of their system with healthy subjects who were required to close their eyes. With the audio feedback, the subjects left the *TR* less often than without feedback.

3.2.3 TactaPacks

Rehabilitation for joint replacement patients

Lindeman et al. [2006] aim at supporting physical therapy for joint replacement patients. They try to decrease injury risk by monitoring and warning patients when they are doing harmful motions that could result in injury.

For this purpose, Lindeman et al. have developed *TactaPacks*, small sized wearable boxes, consisting of a microprocessor with a Bluetooth unit for communicating to a host computer, a 3-D accelerometer for sensing and a vibrator for giving feedback (Figure 3.4).



(a) (b)

Figure 3.4: *TactaPack*: (a) outline of its components, (b) picture of the interior. Taken from [Lindeman et al., 2006].

Vibration as feedback

During a training session the patient attaches several *TactaPacks* to the limbs around the replaced joint. Each of the *TactaPacks* autonomously measures the momentary tilt of its accelerometer relative to the gravity vector (cp. Figure 2.2).

Like the *Bio-WWS*, the *TactaPacks* need to be calibrated to store the 'safe region' of accelerometer values. If the patient leaves this region the boxes begin to vibrate.

Vibration patterns and intensity as well as the delay, after which vibration starts, can be adjusted via a graphical user interface on the computer. The sensor data, however, is processed by the microprocessor on the *TactaPacks* to prevent communication delays.

Vibration intensity can be adjusted

3.3 Sports

The following projects cover a promising application of wearable computing in the area of sport: monitoring motions of athletes for objective analysis and to enhance training practices.

3.3.1 Wireless Force Sensing Body Protectors for Martial Arts

Judging in Taekwondo competitions is a subjective task. The judges cannot always tell if a punch or a kick was executed powerful enough or if it hit the right body part to be considered a valid score. To provide a more objective approach, Chi et al. [2004] have built protectors with built-in force sensors, which are worn on the taekwondo competitor's torso. The force sensors measure the impact of a punch or a kick. They send the readings in real-time over a wireless connection to a base station, which is connected to a laptop.

Wearable device to judge Taekwondo competitions objectively

Chi et al. have conducted experiments with experienced Taekwondo competitors to gather sensor data for different kicks and punches. From the experiments they have derived thresholds for the force readings to determine automatically whether a punch or a kick is valid. Chi et al. envision their system to be used together with human judges ensuring more objective results of Taekwondo competitions.

Automatic scoring is based on force measurements

3.3.2 Towards Recognizing Tai Chi

Kunze et al. [2006] have conducted a feasibility study to explore the potential of body-worn sensors to automatically recognize Tai Chi movements. As video analysis for such movements is tedious, time-consuming, expensive and error prone, they argue in favor of a wearable solution to analyze trainees.

Eight sensor units mounted on body

Kunze et al. use eight MT9[4] Inertial Measurement Units (cp. 2.5—"Inertial Measurement Unit (IMU)") sensor units on different parts of the body. Discussions with Tai Chi experts yield the following attachment locations for the sensor units: above the elbow, above the feet, above the knee (two sensors in each case), and one on neck and hip.

Amateurs and experts can be distinguished

Kunze et al. have conducted an experiment with two Tai Chi amateurs and two Tai Chi experts. Collecting sensor data over a sample window of 100 and calculating various features, e.g., root mean square, Kunze et al. try to recognize different Tai Chi movements. After having trained a K-Nearest-Neightbor clustering algorithm, they are able to distinguish three types of expertise with 76% accuracy and two different Tai Chi movements with 85%. These results show that recognizing Tai Chi movements automatically is feasible.

3.3.3 Audiofeedback for Karate Training

Movements hard to explain with words

Takahata et al. [2004] try to improve a trainee's understanding of how to perform a certain karate punch. They argue that instructors can only vaguely deliver movements by means of expressions and explanations.

Audio feedback on punches

To deliver feedback on the trainee's performance they provide real-time audio feedback. 2-D accelerometers on the wrists measure twists and a microprocessor maps the accelerometers' data to sound. For well performed punches the trainee gets clear sounds as feedback and therefore can

[4] This is the predecessor of the MTx mentioned in Section 2.5—"Inertial Measurement Unit (IMU)" from XSens.

check his performance on his own. In their tests Takahata et al. show that audio feedback increases the trainees' motivation. However, feedback on several aspects of a punch should be avoided as the trainees can only focus on one.

3.3.4 Combining Body Sensors and Visual Sensors for Motion Training

Similarly to the previous project, Kwon and Gross [2005] propose a system to improve traditional training methods in motion training, especially in martial arts. During a training session, a trainer usually demonstrates a certain movement and the trainees try to follow his demonstrations.

Body movements captured with camera and body-worn accelerometers

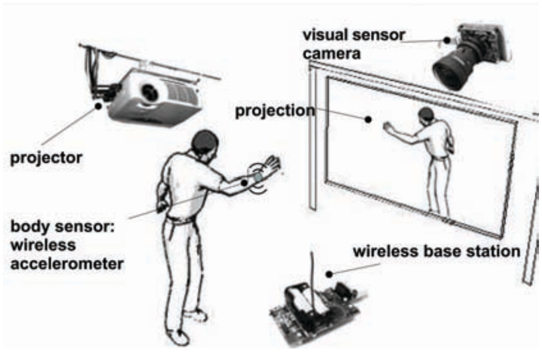
In the new training method the participants' movements are captured via a camera and body-worn accelerometers (Figure 3.5(a)) to create a motion data model in real-time. This data model has two purposes:

1. The trainer's motion data is used to automatically create an instructive training video enriched with non-visible information, e.g., a circle around the hand changes its diameter according to the magnitude of the acceleration (Figure 3.5(b)).

2. The trainee's data on the other hand is evaluated by the system based on the trainer's reference data using Hidden Markov Models.[5] Therefore, a trainee can study the trainer's movements in detail with the instructional video and gets feedback on the quality of his own movements.

When practicing a basic movement in martial arts, a trainee first performs a certain posture, then executes the motion, e.g., one punch, and ends up in a posture again. Hence, Kwon and Gross divide the sensor data into postures and

[5] Hidden Markov Model (HMM) are used in speech analysis to recognize words and sentences and distinguish speakers. For more information see [Rabiner, 1990].



(a)

(b)

Figure 3.5: (a) The system and its components. (b) Sequence of the visual feedback. The circle's diameter in each picture reflect the current acceleration of the hand.

gestures, i.e., static and dynamic chunks of a movement. A single motion thus consists of three chunks: static–dynamic–static. Based on these chunks the trainee's motion is evaluated with respect to the trainer's reference data. The system calculates a score for every motion chunk enabling the trainee to systematically improve postures and gestures.

The system has been tested in an experimental Taekwondo training with one trainer and six trainees who had no experience with Taekwondo. The results have shown that the system helps beginners to learn simple postures and gestures.

3.3.5 Wearable Sensing System for Professional Downhill Skiing

A project closely related to the *Snowboarding Assistant* is described in [Michahelles et al., 2005]. This project aims to support and enhance training practices of professional skiing athletes Usually the training runs of a skier are recorded with a video camera during the day. Afterwards the skier and his coach analyze the videos to identify mistakes and opportunities for improvement.

Video analysis is a common practice

Like [Kwon and Gross, 2005] in the previous section, Michahelles et al. argue that making non-visible information (e.g., acceleration forces on the skier) visible through body-worn sensors might improve video analysis. Out of their own experience and from literature reviews they have identified ski-relevant features and according sensors:

Enriching video material with sensor data

- 3-D accelerometers at thigh, lower leg and torso to measure movements of the skier
- 3-D gyroscopes on the skies to measure rotation
- three FSRs under each foot to measure the weight balance of the skier
- distance sensors attached at the boots to measure the edging angle of the skis relative to the slope
- a radar unit for measuring velocity (this sensor was dropped for the final prototype due to its insufficient accuracy)

Figure 3.6 shows a skier wearing the sensors. A laptop in the backpack records the sensor data. The sensors on ski and boots are fixated with adhesive tape. The accelerometers on the body are attached via velcro straps. They shrink-wrapped the sensors to protect them from snow.

Hardware setup

To analyze a skier's run Michahelles et al. have written a dedicated software to view the sensor and video data of the run synchronously. Sensor values appear in different visualizations. Apart from raw data plots, they combine the

Dedicated software to visualize and analyze sensor data

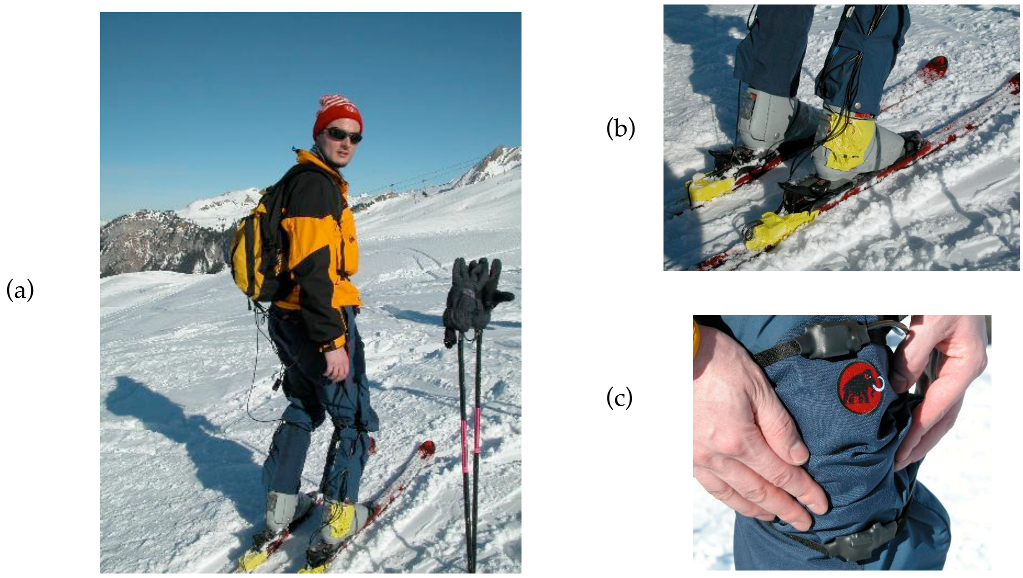


Figure 3.6: (a) A Skier with the attached sensors. (b) Sensors fixed on the boots with adhesive tape. (c) Accelerometers on the knee fixed with velcro straps [Michahelles et al., 2005].

readings of the FSRs under the feet to estimate the center of gravity. This is visualized as depicted in Figure 3.7.

Prototype stimulates trainers' ideas

Following their own prototype-centered development-framework for wearable computing [Michahelles, 2004], they used their first prototype as a starting point for discussions with professional skiing trainers. The prototype should stimulate further ideas. In fact, the trainers were open to new technologies and brought in their own ideas for sensor placements. Moreover, they could imagine incorporating wearable sensors into their training practices.

3.4 Summary and Discussion

Some aspects of the *Snowboarding Assistant* overlap with the presented work

The *Snowboarding Assistant* will be used in an outdoor environment to analyze the movements of snowboarders and to detect mistakes. The objective is to improve the teaching process. In some of these aspects, the projects presented in this chapter overlap with the *Snowboarding Assistant*. In the

Figure 3.7: Screenshot of the analysis software developed by Michahelles et al. [2005]. Video and sensor data can be viewed synchronously. The 'foot balance' view shows the three FSRs under each foot. White points illustrate the center of gravity of each foot and between both feet.

following we will summarize these similarities as well as differences that distinguish the *Snowboarding Assistant* from the related projects.

3.4.1 Systems for Outdoor Use

Off-line analysis of sensor data collected outdoors

Both Kern and Schiele [2003] (sec. 3.1.2) and Michahelles et al. [2005] (sec. 3.3.5) develop their hardware specifically to be used in an outdoor environment. Their objective is to collect sensor data for off-line analysis. Kern and Schiele try to automatically recognize several activities. The objective is 'activity recognition' in general. In contrast, Michahelles et al. [2005] aim at the specific goal to support professional skiing training. They do not, however, automatically extract relevant features from the sensor data.

Same environment but different goals

These projects provide useful information on different types of sensors and their locations for use in an environment that is not restricted to the lab. Nevertheless, both of them do not process data in real-time to support the wearer.

3.4.2 Training Systems

Feedback for improving single motions

The goal of the martial arts projects (sec. 3.3.3, 3.3.4) is to improve the teaching process in motion training. This process suffers from the fact that trainers cannot explain motions by explanation or demonstration adequately. To overcome this problem, the mentioned projects provide real-time feedback for the trainees so they can judge their own performances. The projects concentrate on distinct motions such as single punches and compare them to those of an expert. As training is usually performed in a gym, Kwon and Gross [2005] make use of a visual feedback system installed in the environment.

Sequence of motions is important in snowboarding

The common goal of these projects and the *Snowboarding Assistant* is to improve teaching methods for beginners. While focusing on *single* motions, which are executed precisely, is important for martial arts, the overall *sequence* of motions and proper body posture is essential for snowboarding.
Furthermore, snowboarding is performed outside on a slope, making it unfeasible to follow an approach as proposed by Kwon and Gross, which relies on the static environment of a gym.

3.4.3 Monitoring Systems

Monitoring patients restricted on body posture

The projects in Section 3.2—"Health Care" monitor the patient's posture with wearable sensors to give feedback, if the patient leaves a 'safe area'. Both the *TactaPacks* (sec. 3.2.3) and the *Bio-WWS* (sec. 3.2.2) utilize accelerometers for posture detection. They determine tilt from the sensor readings (see Figure 2.2, p. 13). This works only if the gravitational contribution to the sensor value is dominant, i.e., the wearer of the sensors is not moving or at least moving slowly.

The idea to “nudge” ([Lindeman et al., 2006]) the user back to a good body posture is certainly useful for the *Snowboarding Assistant*. Nonetheless, in snowboarding we will have to consider sequences of motions as well, as the rider moves constantly while descending the slope.

GaitShoe for rehabilitation purposes

The *GaitShoe* is not restricted to monitoring the patient's body posture. It analyzes movements. The project is closely related to the *Snowboarding Assistant*, however, the target application differs greatly. The *GaitShoe* focuses on the rehabilitation of patients, whereas the *Snowboarding Assistant* will support the learning process of snowboarders.

3.4.4 Comparison

None of the systems supports trainees outdoors

Table 3.1 shows an overview of the relevant projects. None of these projects focus on analyzing human movements to support the learning process of trainees in an outdoor environment. [Michahelles et al., 2005] lacks real-time analysis and [Kwon and Gross, 2005] is exclusively designed for applications in controlled environments, prohibiting an outdoor use. As mentioned before, the *GaitShoe* is similar to the *Snowboarding Assistant*, but their application domains are different.

Project	Real-Time Analysis	Outdoor Application	Purpose of Feedback
[Kern and Schiele, 2003] (3.1.2)	–	$\surd$	[no feedback]
[Michahelles et al., 2005] (3.3.5)	–	$\surd$	[no feedback]
Bio-WWS (3.2.2)	$\surd$	–	rehabilitation
TactaPacks (3.2.3)	$\surd$	–	rehabilitation
GaitShoe (3.2.1)	$\surd$	($\surd$)	rehabilitation
[Takahata et al., 2004] (3.3.3)	$\surd$	–	teaching
[Kwon and Gross, 2005] (3.3.4)	$\surd$	–	teaching
Snowboarding Assistant	$\surd$	$\surd$	teaching

Table 3.1: Comparison of related work ($\surd$ = Yes, – = No, ($\surd$) = Perhaps).

Chapter 4

The Snowboarding Domain

"Snowboarding is an activity that is very popular with people who do not feel that regular skiing is lethal enough."

—Dave Barry

An in-depth examination of the snowboarding environment and a detailed understanding of beginners' and instructors' problems are important when designing a system like the *Snowboarding Assistant*. As criticized by Michahelles, wearable computing projects often emphasize the technology side while the target application domain is not taken into consideration sufficiently. This is one reason why only few systems are successfully employed in real world settings [Michahelles, 2004, p. 4].

This chapter is an introduction to the snowboarding domain. First we explain elementary snowboarding terms and techniques, which are important, primarily for those readers not familiar with this environment, to follow our design decisions. Thereafter, we discuss the most important results that we have gained from researching snowboarding literature and conducting interviews with snowboard instructors, which led to the design of our first prototype described in Chapter 5—"A Lab Prototype".

4.1 Snowboarding Terms and Techniques

Nose and Tail. On a snowboard the rider stands sideways. The leading end of the snowboard is called 'nose' and the rear end is called 'tail'. A sketch of important terms is given in Figure 4.2.

Regular: left foot in front. Goofy: right foot in front.

Regular and Goofy. Regular and goofy are the two possible riding stances. Standing with the left foot in front is called regular footed (or 'regular') and standing with the right foot in front is called goofy footed (or 'goofy').[1] This distinction is important due to the fact that the movements and motion sequences of regular and goofy footed riders are reversed, as illustrated in Figure 4.1.

regular goofy

Figure 4.1: The snowboarders' postures are reversed due to their different stances. Taken from [Reil et al., 2003, pp. 12, 70].

Terms like'left' and 'right' are replaced by 'front' and 'back'

Frontside and Backside. In the snowboarding jargon the terms 'left' and 'right' are generally avoided because depending on the rider's stance the terms have a different meaning and therefore might lead to confusion. For instance, the rider's left foot can be either the leading or the

[1] The rider's stance, like being left- or right-handed, is determined by nature.

rear foot depending on the stance. In order to resolve this confusion, snowboarders speak of their front (leading) and back (rear) foot. Likewise the term 'frontside' refers to what is in front of the rider and what is to his back is called 'backside'. The snowboard's edges are named frontside (toe-side) and backside (heel-side) edge correspondingly (see Figure 4.2).

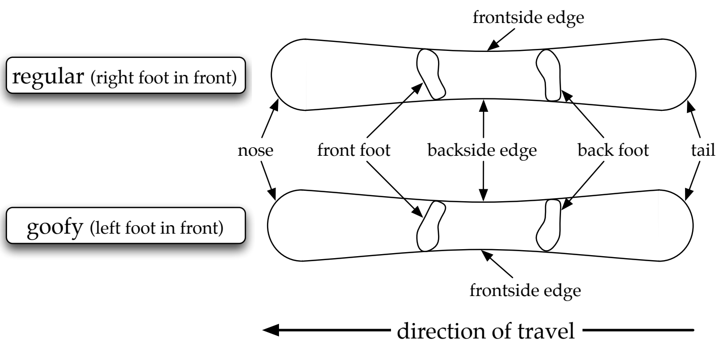


Figure 4.2: Sketch of important terms in snowboarding.

Basic Stance. The 'basic stance' is an extremely important body position on the snowboard because it is the starting point and foundation for any further movement. For example, in between turns the rider returns to the basic stance [Reil et al., 2003, p. 22]. Figure 4.3 illustrates the main characteristics of the basic stance:

The basic stance is the starting point for any further movement

- The rider's weight is distributed evenly on both feet, in order to maintain balance.

Balanced weight distribution

- The joints (including ankles, knees, hip, spine) are in a central position, i.e., neither entirely straight nor completely bent. Even though the body is relaxed, a certain tension remains to allow quick reactions.

Joint in a central position

- The shoulders and hip are parallel to the snowboard.
- The direction of movement is solely tracked by the head. Upper and lower body remain aligned.

Figure 4.3: Basic stance on a snowboard. Taken from [Reil et al., 2003, p. 23].

Two different types of turns: frontside and backside

Turns. There are two different kind of turns: the frontside turn and the backside turn.
When starting a frontside turn, the rider is on his backside edge facing downhill. After the turn he is on his frontside edge facing uphill. For a backside turn the snowboarder is riding on his frontside edge facing uphill. After performing the turn he is on his backside edge facing downhill.

The instructions of how to perform a basic frontside turn can be specified into the following four points:[2][3]

1. In basic stance position the rider approaches on the backside edge (Figure 4.4 ①).

2. To initiate the turn, the rider shifts his weight onto the front foot and pre-rotates his upper body towards turning direction. This is important because otherwise the snowboard does not turn easily. (Figure 4.4 ②).

[2] This works vice versa for backside turns.
[3] Even though various other techniques to perform turns exists [Reil et al., 2003, pp. 48, 52, 72], a basic frontside and backside turn is what beginners learn first.

3. To pivot the snowboard from the backside edge to the frontside edge, the rider shifts his weight from the heels to the toes (Figure 4.4 ③).

4. The rider shifts his weight back onto his back foot to finish the turn. Thereafter he traverses the slope on the frontside edge (Figure 4.4 ④). After finishing the frontside turn the rider returns to basic stance and prepares to initiate a backside turn.

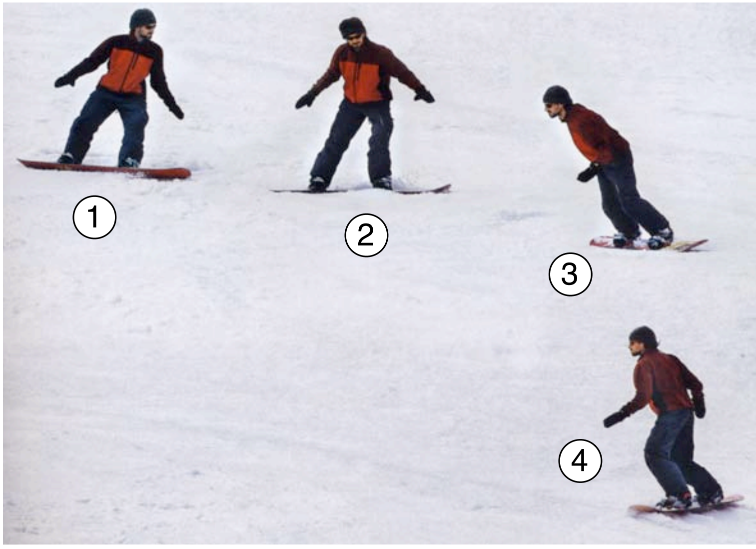


Figure 4.4: Stages of a frontside turn. Taken from from [Reil et al., 2003, p. 35].

4.2 Literature and Interview Findings

Our findings are based on instructional literature and interviews with snowboard instructors. We refer to [Reil et al., 2003] as the only literature source since this book is officially used by the German Ski Instructor Association (Deutscher Verband für das Skilehrwesen e.V.). Nevertheless, the content of the book is consistent with various handouts, websites and movies of different associations for snowboard instructor training.[4]

[4] For example, the Canadian Association of Snowboard Instructors or the Tiroler Skilehrerverband.

Interviews with snowboard instructors yields a deeper understanding of practices in snowboarding

We conducted guided interviews with four snowboard instructors (see A—"Interview Guideline"). The goal of these interviews was to collect qualitative data in order to attain a better understanding of the future users and the context of snowboarding lessons. Furthermore, we wanted to know, if the instructors have experience with typical mistakes that arise when learning how to snowboard and how they deal with these mistakes. We wanted to substantiate what we had read in the instructional literature. At the end of each interview, we presented our idea of the *Snowboarding Assistant* to specifically obtain feedback and possibly new ideas for the development of the first prototype.[5] Our interviewees had at least several weeks of teaching experience with snowboard beginners (see Table 4.1).

	Instructor 1	Instructor 2	Instructor 3	Instructor 4
Profession	student (geography)	research assistant (metallurgy)	physio-therapist	physio-therapist
Age	24	27	26	27
Years on snowb.	12	12	11	9
Years as instr.	4	5	1.5	5
Type of instr.	basic	basic	basic	basic/advanced
Experience:				
Outdoors	1 season	>7 weeks	1 week	several years
Indoors		several weeks	several weeks	several years

Table 4.1: Overview of the snowboard instructors we have interviewed.

In the following sections we summarize the most outstanding interview findings.

4.2.1 Common Beginner Mistakes

The instructors agreed that common beginner mistakes exist. We were able to identify four main mistakes beginners are likely to make.

[5] Without question, interviews with snowboard beginners as the target user group of our system are equally important. Nonetheless, at this early stage we focused on the instructors who are capable of judging the situation and the problems of beginners.

Straight Knees

Balance on the snowboard can be maintained by lowering the center of gravity. Therefore, the snowboarder needs to adopt a relaxed stance by slightly bending the major joints (see 4.1—"Basic Stance"). From the interviews we have learnt that bending the joints, especially the knees too little, is one of the most typical and frequently occurring beginner mistake. As mentioned by the instructors, one reason is the awkward and unfamiliar situation. In addition, beginners often lack a correct perception of their own bodies, which makes them believe that their knees are bent when actually they are too straight.

Beginners do not bend their knees sufficiently

Wrong Upper Body Posture

An almost unbowed position ensures that the center of gravity is above the snowboard and balance can be maintained (see 4.1—"Basic Stance"). Nevertheless, beginners often bend their upper body and look down to the snowboard and to their feet (see Figure 4.5). This posture should be avoided because by this it is easy to lose balance and fall. Moreover, it is important that the snowboarder sees what is in front of him and therefore needs to look in the direction he is going.

Beginners tend to bow their upper body

Figure 4.5: A snowboarder with incorrect body posture. His knees are straight and the upper body is bowed. Presumably, his backside edge cuts into the snow and he falls on his back. Taken from [Reil et al., 2003].

Often in combination with straight knees

As mentioned by the instructors, a wrong upper body posture often occurs in combination with straight knees. Once more, the reason is the lack of the own body perception: The beginners know they need to be closer to the ground so they bow their bodies instead of bending their knees.

Wrong Weight Distribution

Beginners tend to put their weight on their back foot

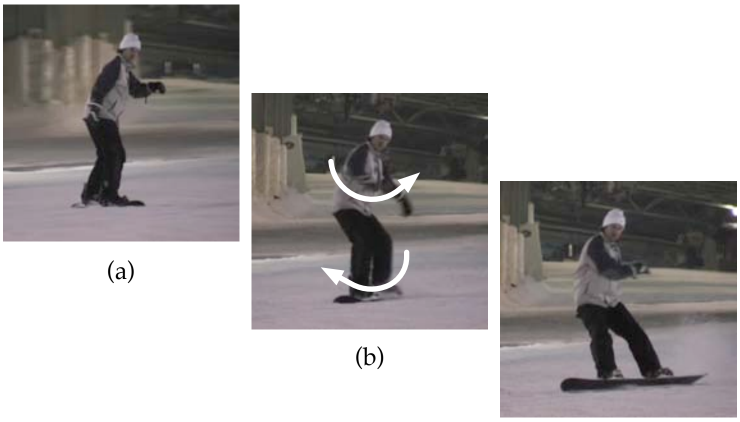
Distributing one's weight correctly is essential as it controls the snowboard's sliding behavior. In between turns, weight is distributed equally between the front and the back foot. In order to perform turns the weight needs to be shifted towards the front foot (see 4.1). As a result the board turns downhill into the fall line, the line of greatest slope. As this accelerates the snowboard, beginners are often afraid to lean onto their front foot. As a counter-intuitive reaction they tend to lean back because they think this slows down the snowboard. Instead with weight on the back foot, the snowboard does not turn easily. As a result the rider is not able to leave the fall line, accelerates and loses control (Figure 4.6).

Figure 4.6: A snowboarder falling down because of too much weight on his back foot. Taken from [Reil et al., 2003, p. 41]

Counter-Rotation

Turning the upper body against the lower body to turn the snowboard

When the upper body rotates contrary to the turning direction this is called counter-rotation. For example, after a frontside turn, beginners are likely to keep their upper body pointed downhill instead of returning to the basic stance [Reil et al., 2003, p. 43]. This results in a twisted posture as illustrated in Figure 4.7(a). If the next backside turn is initiated by unwinding the twisted body, we speak of counter-rotation. This ‘technique’ is physically exhausting and should not be used for frontside and backside turns. Nevertheless, it is necessary to stop the snowboard [Reil et al., 2003, p. 28].



(a) (b) (c)

Figure 4.7: Sequence of a snowboarder performing a backside turn with counter-rotation. Taken from our user study.

4.2.2 Further Aspects

Instructors could imagine the system to be helpful during snowboarding lessons

Altogether the interviewees’ opinions on our project were positive. They could imagine the *Snowboarding Assistant* improving the learning process. They stated that beginners have problems to put into practice what the instructor tells and demonstrates them. Feedback on their performances is crucial as one interviewee stated:

> “I think it is important to get feedback as fast as possible. Feedback whether something was right or wrong. At the beginning we observe everyone individually and give feedback afterwards.” Instructor 4

Snowboarding Assistant increases awareness of body posture

The interviewees felt that the *Snowboarding Assistant* could improve the beginners' awareness of their body postures. This could prevent mistakes related to the insufficient perception of their own bodies, e.g., knees too straight (see sec. 4.2.1). Another advantage is that the *Snowboarding Assistant* can monitor the beginner even when the instructor is not present. When practicing without supervision mistakes are likely to creep in.

Besides of a positive opinion on the *Snowboarding Assistant*, the instructors mentioned other important aspects we had not thought about before

Target User Group

Not helpful for first-time riders, but for advanced beginners

According to the interviewees, first-time riders are too overwhelmed by the unfamiliar situation on a snowboard and feedback from the *Snowboarding Assistant* could overstimulate them. Therefore, the target user group of our system should be beginners who already know how to perform basic turns but still make mistakes.

Advanced snowboarders as possible user group

As another user group the interviewees suggested advanced riders who would like to improve their techniques e.g., when learning to carve.[6] This user group might even benefit from the *Snowboarding Assistant's* feedback to a greater extent than beginners because riding on a snowboard does not overwhelm them. Therefore, they would be able to precisely focus on the feedback of *Snowboarding Assistant*. This user group has to be kept in mind for future developments, but for the first version we focus on beginner mistakes.

Supporting the Instructor

Even though the *Snowboarding Assistant* is supposed to support students, the instructors imagined certain situations in which the *Snowboarding Assistant* could support them.

[6] With this technique turns are performed on the edges of the snowboard and not by drifting [Reil et al., 2003, p. 14].

For instance when students wear wide pants it is not possible to see their knee bending. Moreover, if mistakes recur frequently it is exhausting to repeat the same advices over and over again and it might be embarrassing from the student's point of view. Repeatedly getting unobtrusive feedback from a device, however, might feel less embarrassing in front of others.

Focus on One Mistake

One mistake at a time

The instructors further mentioned that the *Snowboarding Assistant* should only detect one mistake at a time. Feedback on several mistakes would distract beginners. Moreover, according to the instructors' experience, beginners usually maintain one mistake after they have learnt how to do turns. The *Snowboarding Assistant* could help to specifically eliminate this mistake.

Chapter 5

A Lab Prototype

"An idea not coupled with action will never get any bigger than the brain cell it occupied."

—Arnold H. Glasgow

During the development process we followed a DIA cycle (Design–Implement–Analyze). Every iteration resulted in a prototype that was evaluated. We made three iterations, starting with the lab prototype, followed by a first wireless prototype, which was refined to the final prototype used in the user study.

We designed the first prototype for a lab environment in order to discover how mistakes can be detected using body-worn sensors. This lab prototype is able to detect several common beginner mistakes and give simple real-time feedback.

This chapter describes the hardware setup, our approach to determine the mistakes, how it was implemented, as well as initial tests and findings which fed into the design of the second prototype described in Chapter 6—"A Mobile Prototype for the Slope".

5.1 Hardware Setup

Targeted at the common beginner mistakes (see sec.4.2.1), we selected several sensors. To connect the sensors to the computer, we first needed to chose an appropriate sensor interface.

5.1.1 Sensor Interface

Many of the projects in Chapter 3—"Related Work" built their own hardware platforms, but for our first prototype an off-the-shelf sensor interface seemed more appropriate. We considered several alternatives:

Phidgets. The PhidgetInterfaceKit 8/8/8[1] sensor interface has eight analog and eight digital inputs, as well as eight outputs. Phidgets provides several sensors which can simply be plugged into the interface. The interface connects to the computer via USB. There is no wireless version available.

MakingThings. The Make Controller Kit[2] sensor interface board is similar to the Phidgets board. It offers eight analog inputs and eight outputs. MakingThings provides custom sensors for their interface. The sensor interface can be connected to the computer either via USB or Ethernet. No wireless version is available.

Smart-Its. The Smart-Its hardware platform [Beigl et al., 2003] emerged from a collaborative research project from different universities.[3] Smart-Its do not hide the hardware layer as much as the above sensor interfaces do. This results in a higher threshold for users not familiar with physical computing.
In contrast to the previous interfaces, they incorporate wireless sensors into their platform.

[1]http://www.phidgets.com

[2]http://www.makingthings.com

[3] In particular the Lancaster University, the ETH Zurich, and the University of Karlsruhe. See http://www.smart-its.org for more information.

I-CubeX. The I-CubeX[4] sensor system sold by Infusion Systems provides different sensor interfaces. Two of them connect to the computer via MIDI cables and one can be operated wirelessly over Bluetooth. Infusion Systems provides a wide range of different sensors which can be plugged into their sensor interfaces.

I-CubeX as platform for the first prototype

We dismissed the Smart-Its platform because the threshold to built our own hardware on top of this platform seemed to high for a first prototype. The other systems provide easier access to their sensing capabilities. We also dismissed the PhidgetInterfaceKit and the Make Controller Kit because they both lack wireless versions. Consequently we chose the I-CubeX sensor system and selected the I-CubeX Digitizer for the first prototype. We made the decision based on the following advantages:

Simple connection. The system can be easily connected to the computer. Only a MIDI interface and the I-CubeX configuration software are required. Moreover, with the MIDI standard as communication protocol, every programming language or software that supports MIDI can process data from the Digitizer.

Variety of robust sensors. The variety of sensors offered for the I-CubeX system exceeds those provided by Phidgets and MakingThings. This was the main advantage of the rather expensive system. Additionally, the sensors are already assembled in robust casing and can be plugged into the Digitizer without soldering.

24 inputs and 8 outputs The Digitizer offers 32 sensor connections, more than enough for our purposes. Moreover, eight of them work as outputs as well. This was considered for providing feedback in later stages of the development.

Limited processing capabilities on the Digitizer

Although the setup was appropriate for a lab prototype it was obvious that we would not take this system on the

[4]http://infusionsystems.com

slope. The automatic sensor processing capability of the Digitizer is limited to simple preprocessing like averaging or setting thresholds for every sensor. Although we wanted to work with thresholds, *fixed* threshold were too restricted. We needed to change them for every user. Moreover, data of several sensors cannot be combined to define output with the I-CubeX configuration software.

Wired system will not work on the slope

Hence, we needed to connect the Digitizer via two MIDI cables to a computer to allow further processing. We considered taking a laptop in a backpack on the slope, as Michahelles et al. [2005] did for their initial test runs, but we dismissed the idea. The risk of damaging the laptop was high, as beginners often fall. In addition, the Digitizer was sensitive to movements when operated via battery. Sometimes the battery lost contact and turned off the Digitizer.

Setup is appropriate for a rapid prototype

The limitations of the system, however, were acceptable for a rapid prototype. We deliberately chose a wired system for our first trials with the sensors in the lab. Data processing on the Digitizer was not necessary as it was connected via cables to a host computer and introduced hardly any latency. Moreover, for a wireless prototype the wired Digitizer could later be replaced with the wireless I-CubeX Wi-microDig.

5.1.2 Sensor Types and Locations

Among the vast range of sensors offered by I-CubeX we selected the following sensors and corresponding sensor locations to identify common mistakes.

Force Sensitive Resistors (Weight Distribution)

FSRs to measure weight distribution

We selected four FSRs to measure the rider's weight distribution. Similar to the related projects described in Sections 3.3.5 and 3.2.1, we planned to put the FSRs directly under the feet: one FSR under the heel and another under the ball of the foot for each foot. By this we wanted to detect forward and backward (frontside vs. backside) leaning, as well as left and right (front foot vs. back foot) leaning.

Small sized FSRs can be placed under insole

FSRs provided by I-CubeX differ in size, shape and range of measurement. The round shaped TouchMicro-10 fitted our needs. Its dimensions of 30 mm width, 14 mm depth and most of all its thin construction of 0.2 mm allowed us to attach it unobtrusively under a shoe insole (Figure 5.1). The TouchMicro-10 measures force up to 200 N distributed evenly on its active area. Its absolute maximum is at approximately 4.5 kN. This means even though it cannot measure forces above 200 N, the sensor endures much higher forces.

Figure 5.1: Two FSRs taped on the bottom side of a shoe insole.

Range of up to 200 N is sufficient

The range of up to 200 N, i.e., about 20 kg, might seem too restricted for measuring weight distributions of humans. We have to consider, however, that a human's weight is distributed on the area under his feet. The small-sized FSRs measure only a fraction of the total foot area and thus are exposed to only a fraction of the total force. Even though the absolute force on the feet cannot be measured, the relations between different FSR readings reveal the weight distribution.

Bend Sensors (Knee Bending)

Bend sensors to reveal knee bending

Knee flexion is an important aspect of a snowboarder's posture (cp. 4.2.1—"Straight Knees"). Hence, we decided to attach two bend sensors at the back of the knees with velcro straps (Figure 5.2). Even though a bend sensor's output depends on the flexion angle *and* the flexion radius, i.e., the exact angle cannot be measured,[5] they should reveal coarse

[5] See [Morris, 2004, pp. 144] regarding problems with bend sensors.

levels of knee bending. For detecting straight knees of beginners this seemed appropriate.

Figure 5.2: Bend sensors attached to the back of the knee via velcro straps.

We chose the I-CubeX BendShort v1.1 because of its thin composition of 0.1 mm and its appropriate length of 87 mm. The sensor can measure flexion up to 180°— enough to measure knee flexion.

Accelerometer (Upper Body Posture)

Accelerometer to measure tilt of the upper body

As accelerometers can measure inclination (see. sec.2.1), we attached a 3-D accelerometer on the upper body. The I-CubeX GForce3D-3 v1.0 measures acceleration in three orthogonal axes in the range of $-3G$[6] to $+3G$. We did not expect higher acceleration for snowboard beginners. For advanced snowboarders, however, this range might be exceeded.

Gyroscope (Counter-Rotation)

Gyroscope do not provide accurate information

To detect rotations of the upper and the lower body we intended to use gyroscopes. We experimented with the

[6] $G = 9.81\frac{m}{s^2}$, the acceleration due to gravity.

I-CubeX Spin2D-500, a 2-D gyroscope, but could not infer upper versus lower body twists. As gyroscopes only measure angular velocity the absolute position could only be calculated through integration. Yet, the findings of other projects show that this does not lead to good results (cp. [Morris, 2004, pp. 116]). Sensor readings when the sensor is at rest usually drift over time. These mistakes are multiplied by the integration and yield poor results.
Hence, we postponed detecting counter-rotation and planned to acquire inertial measurement units (see Section 2.5—"Inertial Measurement Unit (IMU)") which are able to measure absolute orientation.

5.1.3 Additional Setup

Digitizer is connected to a laptop running Max/MSP

We connected the I-CubeX Digitizer to an Apple PowerBook G4 1.25 GHz via a M-Audio MIDISPORT 2x2 interface. The MIDI data from the Digitizer was processed using Max/MSP,[7] a graphical programming environment intended mainly for creating musical applications. We chose this software because it enables rapid prototyping. Implementation details are described in Section 5.3—"Implementation". Additionally, we displayed a picture of a snowboarder with different postures on a screen as visual feedback. An overview of the whole hardware setup is shown in Figure 5.3.

5.2 Design for Mistake Detection

After experimenting with the sensor values, we came up with the following ideas:

Basic stance servers as reference to compare sensor values

- The basic stance (4.1—"Basic Stance"), as most important posture in snowboarding, serves as reference for comparing any further values. We need to capture the value of each sensor while the snowboarder is in basic stance as calibration data.

[7] http://www.cycling74.com/

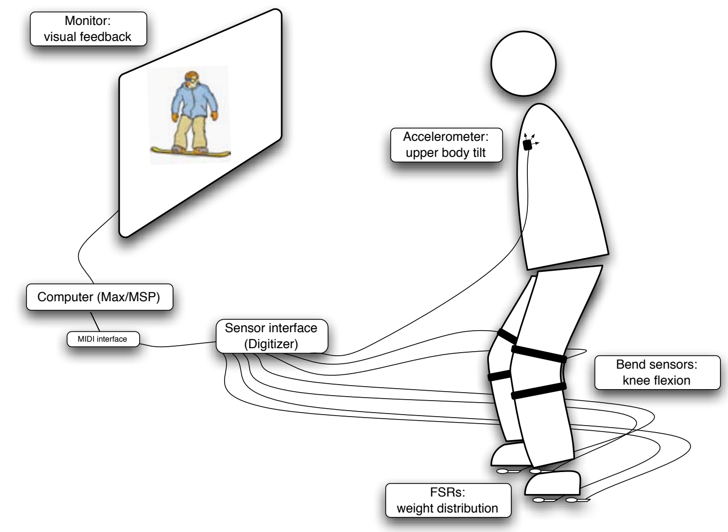


Figure 5.3: Overview of the setup for the prototype. The FSRs were attached under two insoles. Two velcro straps fixed the bend sensors to the users knees. The accelerometer was taped to the upper body. A screen showed pictures of a snowboard as visual feedback.

- A certain range above and below the values of the basic stance should also be considered a good posture. We call this region *tolerance range*, similar to the *Target Region* in Section 3.2.2—"Biofeedback Wireless Wearable System" and define it by a *tolerance value*:

  tolerance range = *calibration value* ± *tolerance value*

5.2.1 Knee Bending

Bend sensors directly measure knee flexion

As the readings of the bend sensors indicate the knee bending we compare the sensor values to those of the basic stance within a given tolerance range. In our case the values of the bend sensors drop when the knees are bent more. Accordingly, when the readings $value(bend)$ exceed the calibration value $value^{cal}(bend)$ captured in basic stance plus a fixed tolerance value T_{bend} we consider the knee straight. We measure knee flexion for both knees independently as follows:

if $value(bend) > (value^{cal}(bend) + T_{bend})$ **then**
knee is straight
else
knee is bent

5.2.2 Upper Body Posture

As depicted in Figure 2.2 an accelerometer only measures acceleration in distinct directions. If the gravity vector is not oriented parallel to one of the axes of the accelerometer only its respective projection is measured. In particular, if one axis of an accelerometer is aligned parallel to the ground, the projection of the gravity vector on this axis is the zero vector, i.e., the reading is 0G (Figure 5.4 (a)). The sensor readings rise when the axis is tilted towards the ground (Figure 5.4 (b), (c)).

Accelerometer can be used to directly measure tilt

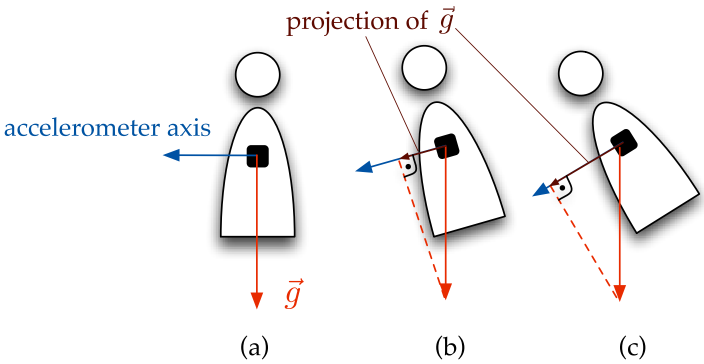


Figure 5.4: Person with an accelerometer attached to the upper body. The axis of the accelerometer points towards the person's heading. In (a) the projection of the gravity vector $\vec{g}$ on the accelerometer's axis is $\vec{0}$. If the upper body is tilted the projections grows, resulting in a higher accelerometer reading ((b), (c)).

As depicted in Figure 5.4, we align one of the axes of the 3-D accelerometer perpendicular to the vertical body axis pointing forwards. If the upper body is tilted the sensor readings rise. To analyze the accelerometer data $value(accel)$, we proceed analogously to knee bending:

One axis of the accelerometer is aligned perpendicular to the body axis

if $value(accel) > (value^{cal}(accel) + T_{accel})$ **then**
    upper body is bent over
else
    upper body is upright

5.2.3 Weight Distribution

Position within a turn is important

The basic stance is not always appropriate and mistakes might depend on the stage of a turn. Thus, we also need to infer this information from the sensor readings. Based on the different stages of a turn (see 4.1—"Turns") we identified according weight distributions under the feet of the snowboarder that are illustrated in Figure 5.5. When the snowboarder is traversing the hill on the frontside edge his weight is mostly on the toe-side of his feet (Figure 5.5 ⑤). Vice versa, while on the backside edge, his weight is on the heel-side (Figure 5.5 ①). Among the front and the back foot weight is distributed equally. When initiating a turn, however, weight is shifted towards the front foot (e.g., Figure 5.5②). We thus consider two independent aspects of the weight distribution and distinguish between three levels:

Two independent aspects of weight distribution

Toe–heel distribution

1. weight distribution between toe-side and heel-side of each foot (in the following referred to as *toe–heel distribution*):
   (a) weight is mostly on the toe-side
   (b) weight is mostly on the heel-side
   (c) weight is distributed equally between toe and heel-side

Front–back distribution

2. weight distribution between the front foot and the back foot (in the following referred to as *front–back distribution*):
   (a) weight is mostly on the front foot
   (b) weight is mostly on the back foot
   (c) weight is distributed equally on both feet

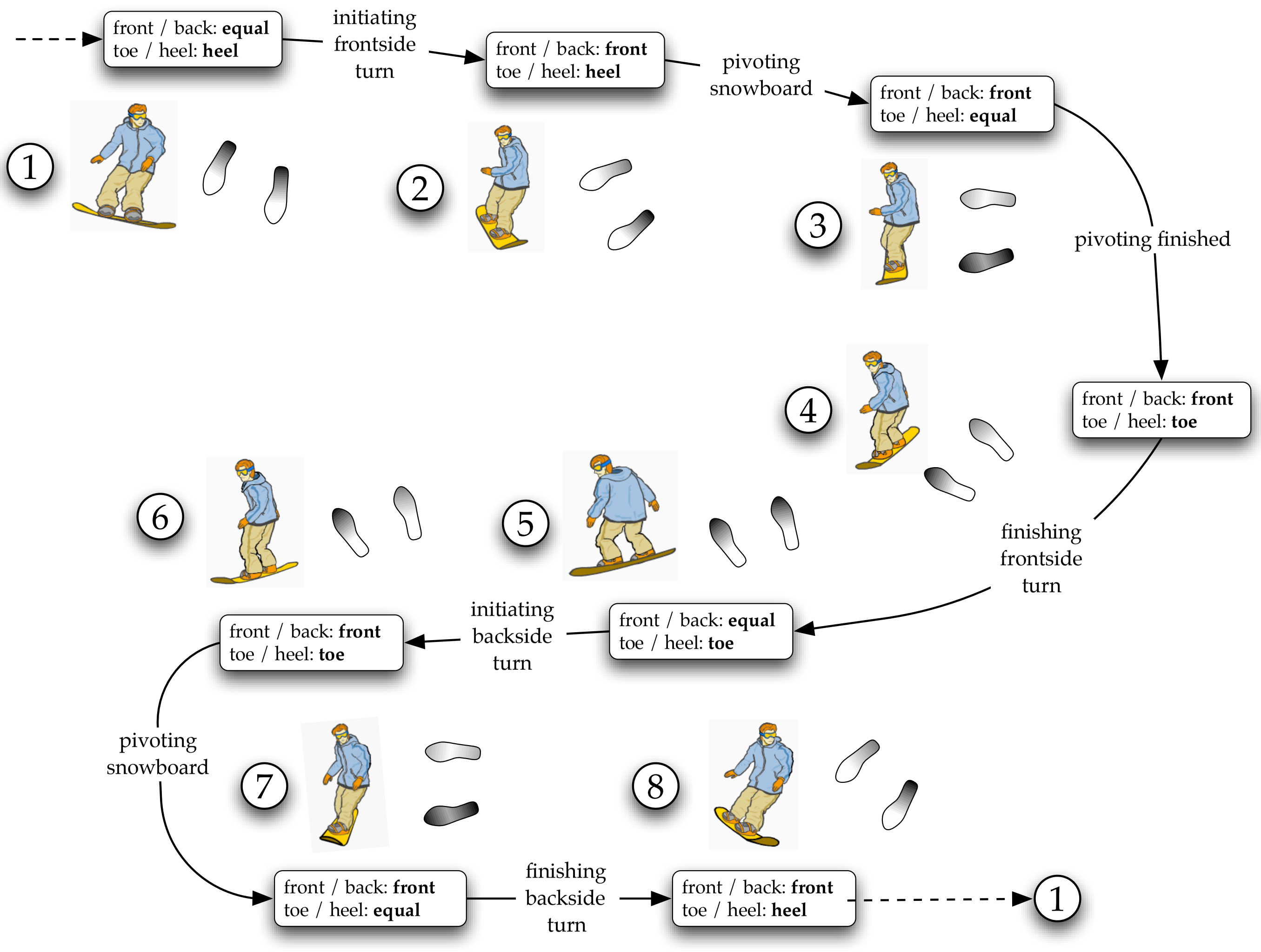


Figure 5.5: Sequence of different stages during a frontside turn followed by a backside turn. The weight distribution in every step is illustrated as pairs of foot steps from the rider's point of view. The color gradients show the magnitude: the darker the color the higher the force.

Toe–Heel Distribution

Detecting *toe–heel distribution* with the FSRs should be straight forward as there is one FSR on the toe-side and one on the heel-side of each foot. We refer to the FSRs according to their placement as illustrated in Figure 5.6.

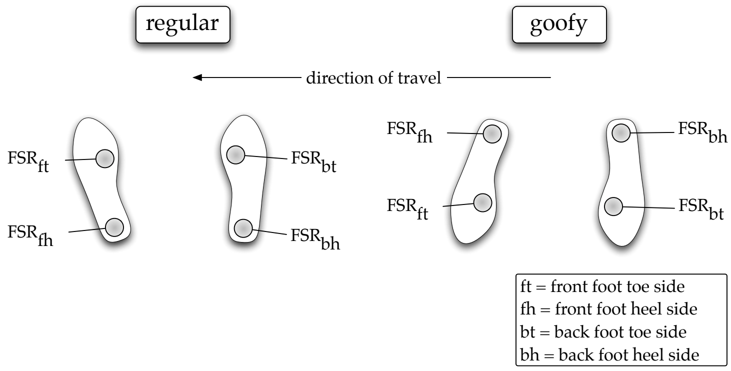


Figure 5.6: Placement and denomination of the FSRs under the feet.

For the three levels (1.(a), 1.(b), 1.(c)) of *toe–heel distribution* the relations between the FSRs are as follows:

1.(a) weight is mostly on toe-side:

front foot: $value(FSR_{ft}) \gg value(FSR_{fh})$
back foot: $value(FSR_{bt}) \gg value(FSR_{bh})$

1.(b) weight is mostly on heel-side:

front foot: $value(FSR_{ft}) \ll value(FSR_{fh})$
back foot: $value(FSR_{bt}) \ll value(FSR_{bh})$

1.(c) weight is distributed equally:

front foot: $value(FSR_{ft}) \approx value(FSR_{fh})$
back foot: $value(FSR_{bt}) \approx value(FSR_{bh})$

Even in basic stance values are not equal

When the snowboarder is in basic stance one might expect the sensor readings of all FSRs to be almost equal (as in 1.(c)) because the weight is distributed evenly on his feet.

This will, however, not be the case because the FSRs cannot be placed at precisely the same locations under the feet. Moreover, different FSRs differ slightly in their readings even if they measure the same force.

However, this does not impose a problem as we are interested in the relations between the FSR readings. An approach to compensate the different readings of the FSRs is to take their difference during the basic stance as an offset.

Relation between FSR values is important

To measure *toe–heel distribution* we calculate the differences between toe- and heel-side FSRs on front and back foot. In a first calibration step, we save the FSRs' values in the basic stance as calibration data. Additionally, we save the differences between toe and heel values:

1. Capture values in basic stance:
   (a) front foot:
   $F_{ft}^{cal} := value(FSR_{ft})$
   $F_{fh}^{cal} := value(FSR_{fh})$
   (b) back foot:
   $F_{bt}^{cal} := value(FSR_{bt})$
   $F_{bh}^{cal} := value(FSR_{bh})$
2. Calculate and save differences:
   (a) front foot (index *ff*):
   $(\Delta F)_{ff}^{cal} := F_{ft}^{cal} - F_{fh}^{cal}$
   (b) back foot (index *bf*):
   $(\Delta F)_{bf}^{cal} := F_{bt}^{cal} - F_{bh}^{cal}$

$(\Delta F)_{ff}^{cal}$ and $(\Delta F)_{bf}^{cal}$ constitute the offsets due to the inaccuracies of the FSRs. Values above this offset indicate a shift towards the toe-side, values below a shift to the heel side. Thus, for any subsequent values we calculate the current differences of the toe- and heel-side FSRs and compare them with $(\Delta F)_{ff}^{cal}$ for the front foot, and with $(\Delta F)_{bf}^{cal}$ for the back foot:

Comparing with values to the basic stance

3. Read data from the FSRs:
   (a) front foot:
   $F_{ft} := value(FSR_{ft})$
   $F_{fh} := value(FSR_{fh})$

(b) back foot:
$F_{bt} := value(FSR_{bt})$
$F_{bh} := value(FSR_{bh})$

4. Calculate the difference:
   (a) front foot: $(\Delta F)_{ff} := F_{ft} - F_{fh}$
   (b) back foot: $(\Delta F)_{bf} := F_{bt} - F_{bh}$
5. Compare with basic stance values plus a tolerance value T_{foot} (shown for front foot, back foot works analogously):

if $(\Delta F)_{ff} > ((\Delta F)_{ff}^{cal} + T_{foot})$ **then**
  weight is on toe-side
else if $F_{ff} < (F_{ff}^{cal} - T_{foot})$ **then**
  weight is on heel-side
else
  weight is distributed equally on toe and heel

Front–Back Distribution

To estimate the *front–back distribution* we proceed similarly. Yet, this time we sum up the values of the FSRs for each foot to get an estimation of the weight on front (index *ff*) and back foot (index *bf*):
$(\sum F)_{ff} = value(FSR_{ft}) + value(FSR_{fh})$
$(\sum F)_{bf} = value(FSR_{bt}) + value(FSR_{bh})$

Analogously to *toe–heel distribution* for the three levels (2.(a), (b), (c)) of *front–back distribution* (sec. 5.2.3) the data will be:

2.(a) weight is mostly on front foot:
$(\sum F)_{ff} \gg (\sum F)_{bf}$

2.(b) weight is mostly on back foot:
$(\sum F)_{ff} \ll (\sum F)_{bf}$

2.(c) weight is distributed equally on front and back foot:
$(\sum F)_{ff} \approx (\sum F)_{bf}$

The rest of the procedure is analogous to *toe–heel distribution*:

First, calibration data is set in the basic stance:

1. Sum up the FSR values of each foot in basic stance and save their difference as calibration value:

   front foot: $(\sum F)^{cal}_{ff} = F^{cal}_{ft} + F^{cal}_{fh}$
   back foot: $(\sum F)^{cal}_{bf} = F^{cal}_{bt} + F^{cal}_{bh}$

   difference: $(\Delta F)^{cal}_{ff-bf} = (\sum F)^{cal}_{ff} - (\sum F)^{cal}_{bf}$

Again for further values we calculate the difference between front and back foot values and compare them with the calibration data:

2. For further values calculate the difference:
   $(\Delta F)_{ff} = F_{ft} + F_{fh}$
   $(\Delta F)_{bf} = F_{ft} + F_{fh}$
   $(\Delta F)_{ff-bf} = (\Delta F)_{ff} - (\Delta F)_{bf}$

3. Compare the difference with the calibration data and a tolerance value T_{ff-bf}:

   if $(\Delta F)_{ff-bf} > ((\Delta F)^{cal}_{ff-bf} + T_{ff-bf})$ **then**
   weight is on front foot
   else if $(\Delta F)_{ff-bf} < ((\Delta F)^{cal}_{ff-bf} - T_{ff-bf})$ **then**
   weight is on back foot
   else
   weight is distributed equally

5.2.4 Combining Basic Information to Derive More Complex Mistakes

Combining information to derive the stage of a turn

We can combine the basic information about *toe–heel* and *front–back distribution* to infer information about the stage of a turn or mistakes in the weight distribution, e.g., too much weight on the back foot.

For example to derive stage ⑥ of Figure 5.5 from the sensor readings, we combine three different results of the basic processing:

if *weight is distributed equally on front and back foot* **then**
  if *weight is on toe-side for the front foot* **then**
    if *weight is on toe-side for the back foot* **then**
      rider is traversing on the frontside edge

Stages with same weight distributions

Some stages have the same weight distributions, e.g., stages ③ and ⑦. We could keep track of the current stage by building a state machine according to the different stages. Transitions should be allowed between succeeding stages in both directions, so turns can be aborted.

We can detect mistakes in the weight distribution by looking at the *front–back distribution*:

if *weight is on back foot* **then**
  give feedback on the mistake

In this case, it is not important whether the weight is on the toes or on the heels, as shifting one's weight on the back foot is not needed (at least for beginners).[8]

We will give a last example considering knee bending:

if *front knee is straight* **then**
  if *back knee is straight* **then**
    give feedback on straight knees

Focussing on simple mistakes

Even though these mistakes are quite simple we could combine this basic information to detect more complex mistakes. For example, to detect counter-rotation we need to know the turning direction. This can be derived from the weight distribution. With the lab prototype we are not able to detect twists of upper and lower body. We will see how counter-rotation can be detected in Section 7.2.4—"Counter-Rotation". Moreover, we have divided the mistake detection into distinct levels. The snowboarder either

[8] For snowboarding in powder snow, e.g., the rider weight needs to be mostly on the back foot.

makes a mistake or not. To assess the severeness of the mistake we could introduce more levels. To prove the concept of automatic mistake detection, however, we focus on simple mistakes.

5.2.5 Providing Feedback

To give feedback on mistakes and the current position within the turn, we first decided to show pictures of the respective mistake and the position within the turn. Visual feedback might not be appropriate for the slope. It was meant to demonstrate that the detection is working correctly. This needs to be substituted in future developments

5.3 Implementation

Max/MSP as developing environment for rapid prototypes

As mentioned, for the first prototype data processing was done with Max/MSP. I-CubeX provides a plugin — the 'iCube object'— that controls the Digitizer from within the Max/MSP environment. Max/MSP allows direct manipulation of data values because the values are always visible even while editing. Data flow is visualized through connections that are graphically dragged from a sending object to a receiving object. With the 'iCube object', data values can simply be piped into further processing objects via dragging connections. Figure 5.7 shows the 'iCube object'. To illustrate the principle of Max/MSP, the first two sensor outputs are connected to an object that calculates their sum. The result is printed in a log window by the print object.

5.3.1 Max/MSP Patches

Layered architecture

The structure of our prototype was inspired by the layered architecture proposed in [Schmidt and Laerhoven, 2001]. We divided the software into three single programs, or 'patches' as they are called in Max/MSP, which represent

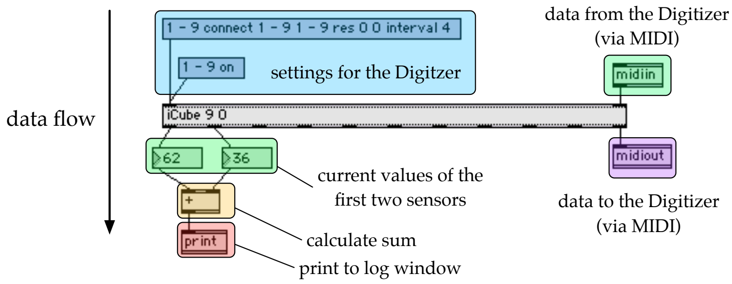


Figure 5.7: A Max/MSP patch with the iCube object. The iCube object receives sensor values from the Digitizer. As an example the first two sensor values are piped into a 'sum object'. The resulting sum is forwarded to a 'print object' that prints it into a log window.

the different layers. The single patches communicate via OSC[9] (Open Sound Control) messages. These are similar to MIDI messages but can contain arbitrary binary data. OSC messages are not bound to a specific communication protocol and can be sent over any networking mechanism, e.g., UDP (User Datagram Protocol). This enables loose coupling between the different patches. In particular we could also send messages to other applications that are not written in Max/MSP.

The three patches have the following functionalities:

Collecting sensor data

`MyCutePatch.pat` The first patch depends on the sensor system and the sensors used, i.e., the I-CubeX Digitizer and connected sensors. The patch contains the 'iCube object', collects samples from the Digitizer and evaluates them according to 5.2—"Design for Mistake Detection". The resulting OSC messages contain basic context information such as 'front knee straight' or 'weight on front foot'.

Combining basic information

`MyEventReceiver.pat` This patch receives the messages sent from the first patch and combines them as discussed in 5.2.4—"Combining Basic Information to Derive More Complex Mistakes". It contains the logic to make sense of the basic context information.

[9]http://www.cnmat.berkeley.edu/OpenSoundControl/

For instance, the messages 'front knee straight' and 'back knee straight' are combined to form the message 'knees too straight'. The results of the combinations are forwarded again via OSC messages to the next patch.

Giving appropriate feedback

MyFeedbackGenerator The last patch is responsible for giving feedback according to the received messages. For example, when the message 'knees too straight' arrives, the patch displays a picture of a snowboarder with straight knees.

In the following section, we describe the data flow within the different patches. Screenshots of the patches can be found in appendix B.

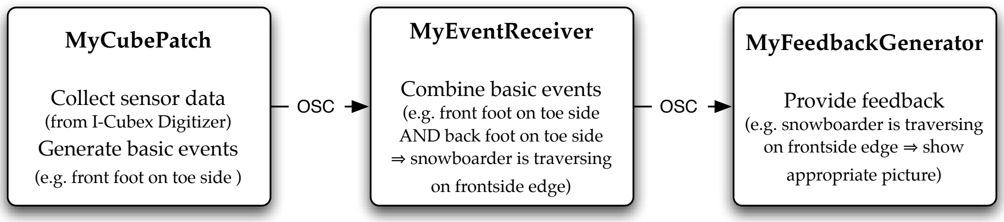


Figure 5.8: Outline of the three Max/MSP patches.

`MyCubePatch.pat`

Figure 5.9 shows the different modules of `MyCubePatch.pat`.

Setting calibration and tolerance values

Before using the prototype, the calibration data needs to be captured. This is done, when the user is in basic stance. A button in the calibration module captures the current sensor values of all attached sensors. Moreover, the tolerance values are set to realize the tolerance range as introduced in 5.2—"Design for Mistake Detection".

Data acquisition module

The data acquisition module is responsible for collecting sensor values from the Digitizer. We always connected the sensors to the I-CubeX Digitizer in the same order as shown

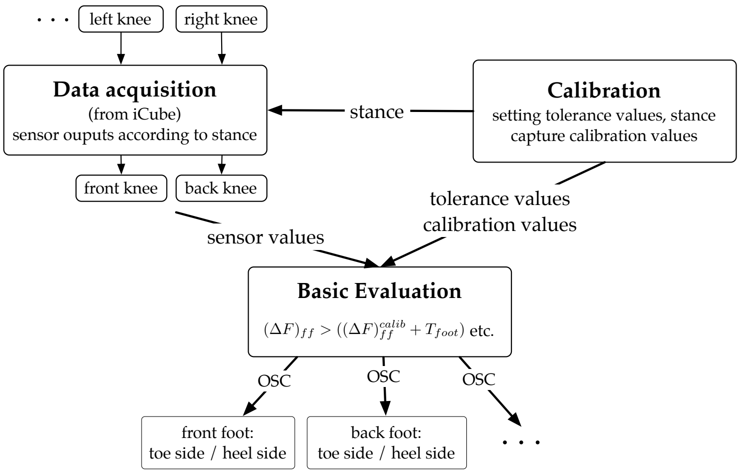


Figure 5.9: The different modules of `MyCubePatch.pat`

in Table 5.1. The data acquisition module on the other hand should output sensor data irrespective of the user's stance. Table 5.1 shows the connections of the sensors to the Digitizer as well as desired outputs of the data acquisition module.

I-CubeX Digitizer		Data Acquisition module	
Input	Connected sensor	Output	Sensor
0	FSR on **left** foot, toe-side	0	FSR on **front** foot, toe-side
1	FSR on **left** foot, heel-side	1	FSR on **front** foot, heel-side
2	FSR on **right** foot, toe-side	2	FSR on **back** foot, toe-side
3	FSR on **right** foot, heel-side	3	FSR on **back** foot, heel-side
4	Bend sensor on **left** knee	4	Bend sensor on **front** knee
5	Bend sensor on **right** knee	5	Bend sensor on **back** knee
6-8	3-D accelerometer on upper body	6-8	3-D accelerometer on upper body

Table 5.1: Sensor attachment on the I-CubeX Digitizer and desired outputs for the data acquisition module.

Sensor mapping based on the user's stance

We could have attached the sensors based on the user's stance. Nevertheless, an easier way was to map the inputs from the Digitizer within the data acquisition module onto its outputs as shown in (Figure 5.10).

Comparing values with calibration data

The basic evaluation module evaluates the sensor data based on the settings of the calibration and tolerance values as described in 5.2—"Design for Mistake Detection".

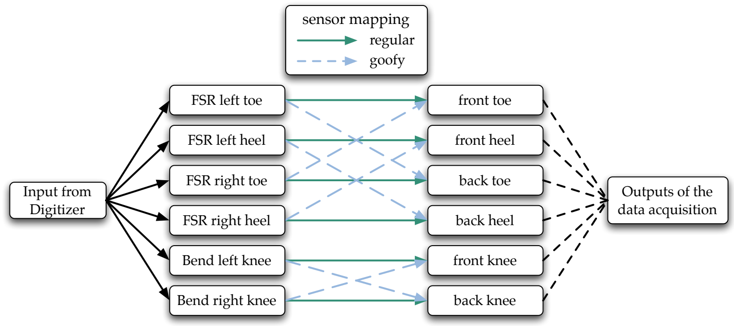


Figure 5.10: The mapping of the sensors according to the stance of the user.

It does not yet combine the different context information. This is the task of the next patch.

`MyEventReceiver.pat`

Combining information to infer mistakes

This patch combines basic context information of the snowboarder to detect mistakes and the position within a turn as outlines in Section 5.2.4—"Combining Basic Information to Derive More Complex Mistakes". For example, if it receives both messages 'front knee straight' and 'left knee straight' it creates a 'both knees straight' message and sends it to the following patch (Figure 5.11).

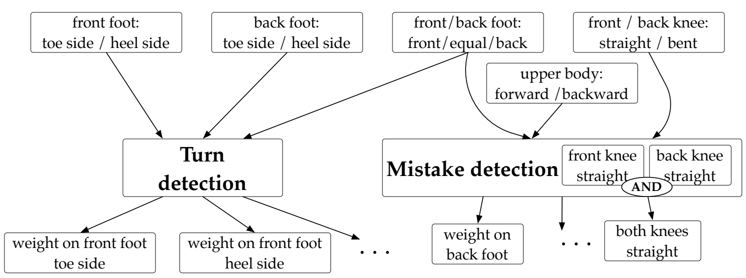


Figure 5.11: Outline of `MyEventReceiver.pat`, which combines basic information to infer mistakes and stages within a turn.

MyFeedbackGenerator.pat

The last patch gives feedback on the position within a turn and mistakes based on the messages it receives from `MyEventReceiver.pat` (Figure 5.12).

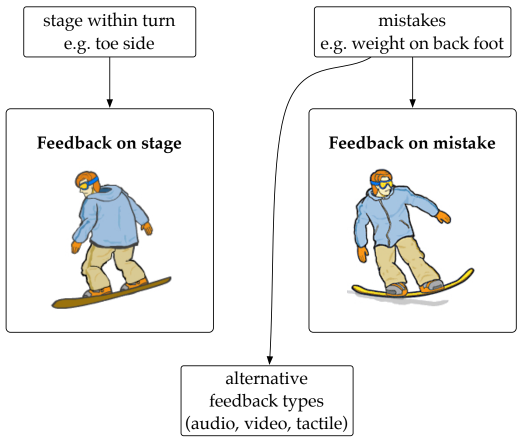


Figure 5.12: Outline of `MyFeedbackGenerator.pat`. For the lab prototype we, chose visual feedback. This could be substituted by other feedback types (picture used with permission of ABC-of-Snowboarding).[10]

Visual feedback mistakes and stage within a turn

In one window, the patch shows images of a snowboarder as shown in Figure 5.5 according to the current position within a turn (Figure 5.12). In another window, feedback on current mistakes is provided as depicted in Figure 5.13.

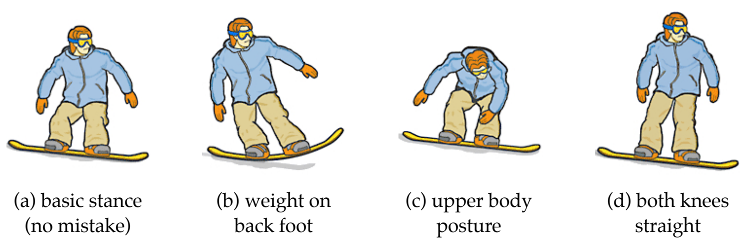


Figure 5.13: Pictures shown to give a hint on the current mistake (pictures used with permission of ABC-of-Snowboarding).[a]

[a]http://www.ABC-of-Snowboarding.com

Feedback is exchangeable

Visual feedback on a slope will not be appropriate as the snowboarders need to watch the slope. We chose visual feedback in the lab prototype to confirm that mistakes were detected correctly. Other types of feedback needs to be explored on the slope. As mentioned in 1.1.3—"Goals", this is not the focus of this thesis. If appropriate feedback has been found, this could simply be incorporated into `MyFeedbackGenerator.pat` to test it in the lab.

5.3.2 Discussion

The layered architecture of the prototype has several advantages:

- If the sensor system is exchanged, only the corresponding MAX object needs to be replaced — in our case the iCube object.
- If sensors are substituted, only the corresponding processing will have to be adjusted. For instance, if bend sensors were replaced by optical sensors, messages like 'knee straight' or 'knee bent' would remain the same.
- The logic to derive mistakes or the position within a turn from the basic messages is concentrated in one location (`MyEventReceiver.pat`). Different basic messages can be combined to detect more complicated mistakes.
- The last layer only deals with the type of feedback. It can easily be exchanged to test several modalities of feedback, such as video, audio or tactile feedback.

Sensor system should be exchangeable

In reality, a strict isolation of the different layers will most likely not be possible. It should, however, be made easy to exchange the sensor system and the sensors used as this will often necessary. Additionally, the type of feedback should be exchangeable to test several possibilities.

5.4 Test and Findings

We tested the prototype in the lab and simulated mistakes and turns. We simulated turns by shifting the weight according to Figure 5.5 and focused on mistakes that were related to a bad body posture as compared to the basic stance.

(a) (b) (c)

Figure 5.14: Testing the lab prototype. (a) in basic stance (no mistake), (b) weight on back foot, (c) both knees straight. The visual feedback indicates the mistakes.

The tests led to the following results:

Position within a turn. Shifting one's weight according to Figure 5.5, to simulate a turn, led to a correct sequence of the corresponding pictures in `MyFeedbackGenerator.pat`. Yet, sometimes the distinction between stage ② and ③, i.e., pivoting the snowboard, were not clearly separated. However, stages ① and ⑤ could be identified unambiguously.

Upper body posture. Detecting the upper body posture was possible with the accelerometer. Yet, we could not detect it as easily as the other mistakes.

Knee bending. The mistake 'both knees are too straight' was the most simple to detect. After adjusting the tolerance value straight knees could be detected reliably.However, as we will see in 6.3.1—"Problems with the Hardware" on the slope we were facing problems related to the bend sensors.

Weight too much on back foot. Comparing differences between FSRs worked well. The different levels of *front–back distribution* (sec. 5.2.3) could be identified. With an appropriate tolerance value, which depends on the user, too much weight on the back foot could be detected.

Focus on FSRs and bend sensors

The readings of the accelerometers were only partly satisfying. We did not dismiss the accelerometer completely for the next prototype but focused on the FSRs under the feet and the knee bending as they returned promising results.

Chapter 6

A Mobile Prototype for the Slope

> *"In theory, there is no difference between theory and practice. But, in practice, there is."*
>
> *—Jan L.A. van de Snepscheut*

Mobile prototype for the slope

For the next prototype, we build a wireless sensor system that can be taken on the slope. This chapter describes the hardware we used as well as the software to acquire and analyze sensor data. In addition, we conducted initial self-test runs with the system on a slope. Based on the experience from these tests we improved the prototype for a test with snowboard beginners.

The main goal of the second prototype is to collect sensor data in a real-world setting. Building a prototype for the slope imposes several requirements:

- The hardware needs to be robust enough to compensate impacts when the wearer falls.
- The use of a laptop, like in the first prototype, is not appropriate for the slope. We must
- switch to mobile devices.

- Sensor data should be logged for further analysis in the lab as conducting tests on the slope is expensive and time-consuming.

6.1 Hardware Setup

6.1.1 Sensor Interface

Arduino platform for future developments

Even though the I-CubeX Wi-microDig could have been useful for a wireless prototype, we dismissed it because it has no outputs for providing feedback[1] Furthermore, its on-board microprocessor cannot be programmed to support complex data processing. Instead we chose the Arduino[2] platform [Banzi et al., 2007] as our future development sensor interface. It offeres several advantages:

Programmable micro-controller

- Every Arduino board is equipped with a micro-controller which can be programmed in C. Thereby it allows advanced processing of sensor data in contrast to the I-CubeX sensor interfaces.

Low threshold for programming on hardware level

- Arduino's dedicated software environment allows easy programming of the micro-controller by hiding unnecessary hardware details. This is particularly useful for developing prototypes.

Communication via serial line

- Data is transferred over the serial line, which is supported by almost every programming language.
- Arduino offers different boards with outputs to connect motors and actuators for feedback.
- A wireless version of the Arduino, the Arduino BT[3] , with a built-in Bluetooth device is available.

Arduino allows customization of software and hardware

- The software as well as the hardware are open source. The schematics of the different Arduino boards can be found on the Arduino website, which allows customization of the hardware if needed.

[1] In the prototypes of this thesis feedback will not be implemented. However, the decision was made for future developments.

[2] http://www.arduino.cc

[3] http://www.arduino.cc/en/Main/ArduinoBoardBluetooth

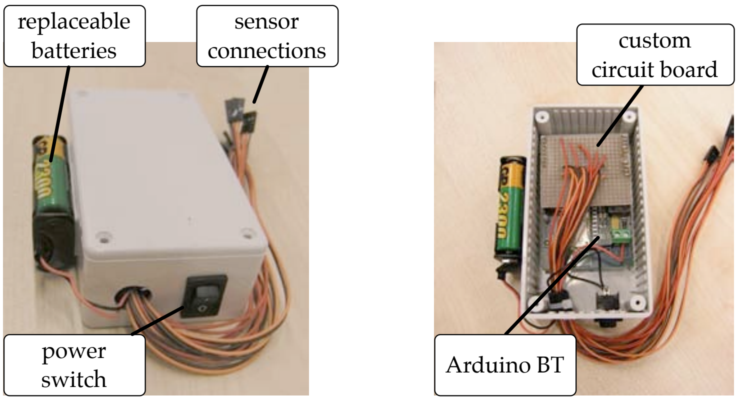
6.1.2 Robust Casing

Arduino does not provide any custom sensors for its platforms. As the I-CubeX sensors have proven to be useful, we integrated them into our wireless prototype. We soldered our own circuit board which allows us to connect six I-CubeX sensors to the analog inputs of the Arduino board.

Custom circuit board to connect I-CubeX sensors

To protect the board against impacts on the slope, we put it into a robust plastic box (Figure 6.1). The external batteries supply power for approximately ten hours and can be replaced without opening the box.

Robust box for Arduino BT



(a) (b)

Figure 6.1: Arduino casing. (a) Box closed, with external batteries and power switch. (b) Box opened, with an Arduino BT and the custom-built circuit board to connect to the I-CubeX sensors.

6.1.3 Sensor Attachment

In the lab the sensor attachment on the snowboarder was never considered a problem. But for tests in a real-world setting we had to ensure that sensors were attached properly. After evaluating several possibilities, we used flexible off-the-shelf knee pads to fix the bend sensors on the knees. The FSRs were taped on insoles like in the first prototype. The insoles were put into the snowboard boots. The sensor

Sensors need to be attached properly

cables ran through the pants to the Arduino box which was carried in a bag on the hips (Figure 6.9 on page 89 shows the whole setup).

Dismissed the accelerometer due to poor results on the slope

In the lab, we used an accelerometer to measure the upper body posture. However, on the slope it did not provide useful data with respect to body leanings. Due to the constant movement of the snowboarder, the dynamic acceleration is high in contrast to the gravitational acceleration and does not allow to infer the upper body posture easily. Hence, we decided to dismiss the accelerometer. We give an outlook on alternatives for measuring the upper body posture in Section 7.2.3—"Upper Body Posture".

6.2 Software

This section gives an overview of the software we have developed for this prototype at this point. We will not give detailed information of the implementation since the software is still under development.

6.2.1 Wireless Communication

Nokia N70 with Python script communicates with Arduino BT

We needed to find a mobile device to connect to the Arduino BT via Bluetooth to adjust its settings and to receive sensor data. Based on the chair's experience from a previous project,[4] we selected the Nokia N70 smartphone with the mobile operating system Symbian OS[5] which allows to run programs written in Java and Python. We decided to use Python because, as a scripting language, it allows quick development. We used a Python port[6] specifically tailored for the S60[7] software environment of the N70. On the N70, Python can easily access the mobile's Bluetooth capabilities and create menus. We wrote a Python script that communi-

[4] The REXplorer, a mobile, pervasive game for tourists [Ballagas et al., 2007].

[5] http://www.symbian.com/

[6] http://opensource.nokia.com/projects/pythonfors60/index.html

[7] http://www.s60.com/

cated with the Arduino BT. and set up an ad-hoc communication protocol between the two devices which allowed to control the behavior of the Arduino board (Figure 6.2).

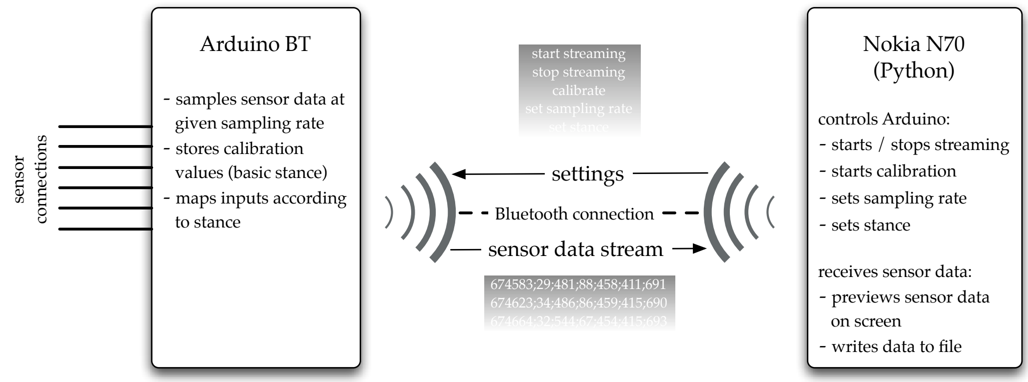


Figure 6.2: Communication protocol between Arduino BT and N70. The Arduino BT collects data samples from the sensors and streams them via Bluetooth to the N70 which stores the data. The N70 adjusts the settings of the Arduino.

In particular we implemented the following features:

- On the Arduino microprocessor:

Data sampling. Data is sampled from the six analog inputs with an adjustable constant sampling rate. The resolution of the onboard analog-to-digital converter is 10 bit, sensor values range between 0 and 1023.

Saving calibration values. We saved the sensor values while the snowboarder is in basic stance as calibration values, like we have done in the first prototype. These values can later be used to detect mistakes.

Streaming values. The 10 bit sensor values of the inputs are streamed over the serial line together with a timestamp. Streaming can be turned on and off.

Sending information. Information about the current settings is sent over the serial line when requested. This includes the sampling rate and the calibration values.

- Python script on the N70 (accessible via a menu):

  Connecting. This feature establishes a Bluetooth connection to the Arduino BT.

  Adjusting settings of the Arduino. This includes activating and deactivating streaming of data, setting the sampling rate, requesting information on the settings of the Arduino and displaying the sensor values on the N70.

  Logging. Streamed values from the Arduino are saved to a log file.

Data stream as separated strings

The data stream from the Arduino BT to the N70 consisted of CSV (character separated values) strings. The values of the six sensor inputs are separated with a semicolon and preceded by the current timestamp (ms) of the Arduino microprocessor.[8] One string contains one sample of every sensor.

Sensor mapping according to user's stance

Analogously to the first prototype (Chapter 5), we implemented a setting on the Arduino which stores the snowboarder's stance and maps the values as outlined in Figure 5.10 (p. 69) to a fixed position within the CSV string.

6.2.2 Off-line Sensor Data Analysis

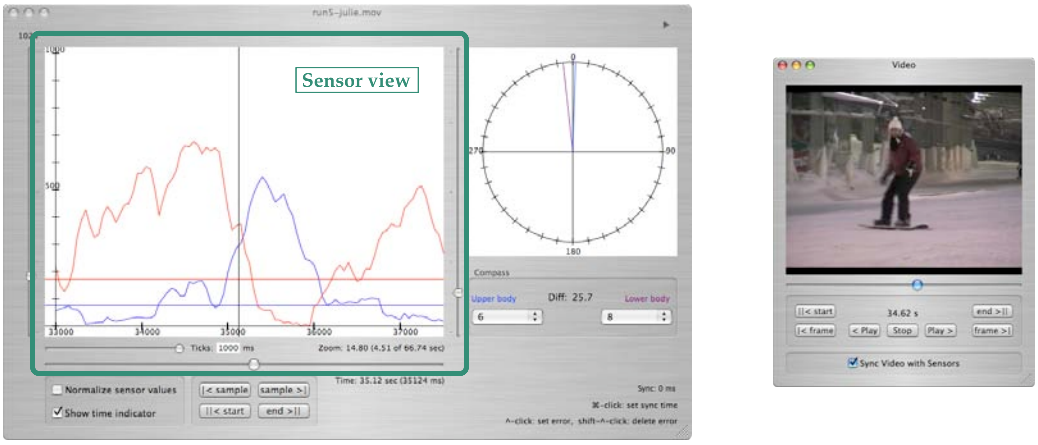
We tried to implement real-time processing on the Arduino as we had done before the lab prototype. On the slope, however, sensor data was too noisy to get satisfying results with the same approaches. Hence, we postponed real-time detection and focused on off-line analysis to explore possible algorithms. When appropriate algorithms have been found, they can be implemented directly on the Arduino or the N70.

Sensor values displayed as plotted graph

For the off-line analysis we developed a dedicated software tool which allows to import the sensor recording logs stored on the N70 and displays them as graphs on a timeline according to the timestamps of the sensors. Sensors are shown in different adjustable colors. The tool allows to set

[8] Time in milliseconds since powering the Arduino.

a calibration value for each sensor (which is also logged by the N70) which is shown in the graph as horizontal line. The software also allows to view video data and the sensor data synchronously (Figure 6.3) which enables us to identify sensor values associated with the users' mistakes.



(a) (b)

Figure 6.3: Sensor synchronization software. (a) Main window with sensor plots of the different sensors. (b) Video window displaying the current frame of the video recording.

Software still under development

The software was developed for Apple's Mac OS X[9] using the Xcode[10] development environment. For the data analysis of the user test (see Chapter 7), we made intensive use of the software. It is still under development and filters and algorithms to process the sensor data to test several approaches for mistake detection will be added.

6.3 Initial Self Tests on the Slope

Gathering sensor data

Initial self tests should verify that our hardware endures the conditions on a slope. Additionally, we gathered sensor data from this real-world setting. We did not yet implement any mistake detection.

[9] http://www.apple.com/macosx/
[10] http://developer.apple.com/tools/xcode/

Tests on indoor slope

We tested our prototype in SnowWorld[11] in Landgraaf, an indoor slope in the Netherlands. This location will also be the test environment for any further studies.

6.3.1 Problems with the Hardware

Bend sensors do not work well in real-world setting

Unfortunately, measuring the knee flexion with the bend sensors did not work as well as in the lab. Although they were attached with the knee pads, they did not stay in place on the slope. Sometimes the knee pads also folded the bend sensors which led to a sharp bent radius of the sensors. This created big gaps in the sensors' conductive layers (Figure 6.4) resulting in a sudden downfall of the sensor reading to almost zero even with slightest changes of the knee flexion.

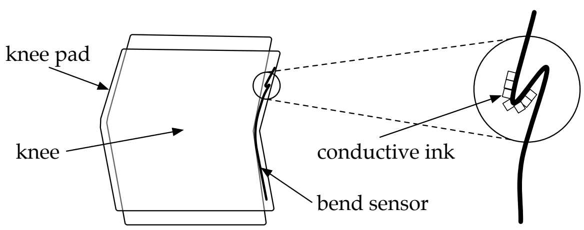


Figure 6.4: A bend sensor attached to the back of the knee with a knee pad. The bend sensor is bent sharply at one spot. This creates big gaps in the conductive ink.

FSR cabling is stressed

Moreover, we had some minor problems with the FSRs. The snowboarder's movements sometimes stressed the FSRs' cables inside the pants which led to a broken solder connection two times.

Considerable setup time

The setup time is considerably high. The insoles need to be put into the shoe and the cables need to be pulled through the pants. Additionally, attaching the knee pads and adjusting the bend sensors within them is a tedious task.

[11] http://www.snowworld.com

6.3.2 Successful Setup Features

The Bluetooth communication between the Arduino BT and the Nokia N70 worked well. Even on distances up to 50 m the Bluetooth connection transmitted the sensor values. Nevertheless, setting up a connection at this distance was not possible. Thus, the snowboarder who wore the sensors needed to carry the N70.

Bluetooth connection works properly

The FSRs inside the boots returned promising values by which different stages of a turn can be identified (Figure 6.5). As expected, when riding on the frontside edge the values of the FSRs on the toe-side are very high. In contrast, those on the backside edge are low. Especially the values of the front foot show the alternation between toes and heels. This is not surprising as the front foot is the leading foot and controls the snowboard. As we have already seen in the first prototype (see sec.5.4), the pivoting process cannot be divided into several stages. Nevertheless, the distinction between frontside edge, backside edge, and turning provides information about the turning direction:

FSRs distinguish frontside and backside edge

- change from frontside to backside edge: backside turn
- change from backside to frontside edge: frontside turn

The FSRs inside the boots were unnoticeable. The knee pads were not as unobtrusive as the FSRs, but still did not bother when riding. The cables running inside the pants were bothersome. However, they did not restrict one's freedom of movement at all. At this early stage, comfort is not the main goal, but in a sophisticated version cables could be woven into the pants.

Sensors are unobtrusive

Cables are disturbing

6.3.3 Summary

Although it was satisfactory to see that the communication between our self-made sensor box and the N70 worked, we

Mainly sensor problems

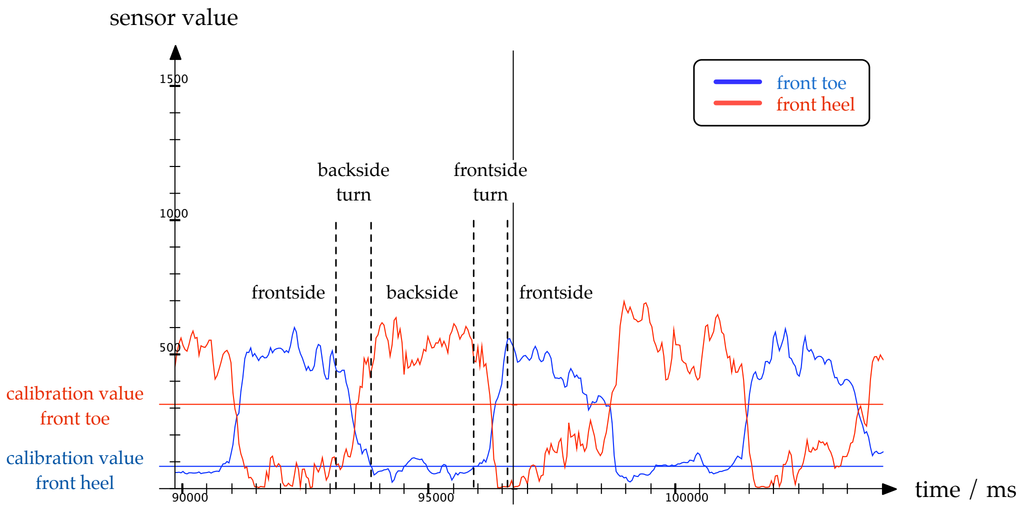


Figure 6.5: Data plots of the FSRs under the front foot of the author during alternating frontside and backside turns.

had severe problems regarding the bend sensors. They did not provide useful data. They even slipped out of the knee pads once after using the ski lift.

To sum up our experience gained on the slope:

- With the current setup, bend sensors do not work reliably in a real-world setting. We will have to find a new way of sensing knee flexion or build a more robust cover around the bend sensors.
- FSRs inside the boots are unobtrusive and work reasonably well. However, we must ensure that the solder connections are robust enough to bear stress.
- Bluetooth communication is sufficient for snowboarding lessons, where the instructor is at most 50 m away from his students.
- The overall comfort of the wearable system is satisfactory. In particular it does not restrict the wearer in his movements.

As a result of our test, we decided to focus on the fixation of the sensors, as this is essential for the quality of the sensor values.

6.4 Improvements

6.4.1 Bend Sensors

Foam to protect bend sensors

After experiencing difficulties fixing the bend sensors inside the knee pads, we tried to protect them with a robust but flexible cover. After trying out different rubbers and foams, we found a particular foam that worked well. The final version of the knee bend sensors is depicted in Figure 6.6. The cover protects the bend sensors and keeps them tightly fixed under the knee pads.

(a) (b)

Figure 6.6: (a) The right bend sensor is protected by a thick foam. (b) Bend sensors attached with knee pads.

6.4.2 Detecting Counter-Rotation

SHAKE compass unit to measure absolute orientation

As mentioned earlier, we dismissed gyroscopes for detecting counter-rotation. Instead, we incorporated two SHAKE SK6 sensor units. Among other parameters, they allow to measure absolute orientation by means of a digital compass. The values returned from the SHAKE compass range from 0 to 3600, i.e., 360° with 0.1° steps.

Bluetooth communication to SHAKEs

The SHAKE SK6 has a Bluetooth unit on its own and does not need to be connected to any sensor interface. We could communicate with the SHAKEs directly from the N70 by writing a new Python script that follows the protocol defined by these sensor units. The values of the SHAKEs were logged on the N70 by appending them to the CSV stream of the Arduino:

Unfortunatel

Sampling rate lowered to 20 Hz

y, the maximum sampling rate of the SHAKE when calculating absolute orientation is as low as 25 Hz. We streamed data from the SHAKE units and the Arduino at the same sampling rate. After experiencing synchronization problems between Arduino and the SHAKE units at 25 Hz, we lowered the sampling rate to 20 Hz. At this rate quick movements will not be captured correctly. However, for slow movements a sampling rate of even 10 Hz is sufficient [Knight et al., 2007]. We did not notice any disadvantages in the sensor recordings due to the lower sampling rate.

SHAKEs mounted on upper body and shin

To measure counter-rotation, we mounted one unit on the shin of the front foot and the other on the torso near the front shoulder (Figure 6.7(a)). We were interested in values indicating if the upper and lower body are aligned or twisted. Therefore, we chose a similar approach as when processing the FSR values. When the snowboarder is in basic stance, upper and lower body are aligned. The SHAKE values, however, will differ slightly (Figure 6.7(c)). During calibration, we store the difference in the readings as offset and declare this reading as the angle $\Delta\Phi = 0°$. Values above indicate a counter-clockwise twist of the upper body against the lower body, values below a twist in clockwise direction. We map counter-clockwise twists to an angle $0° < \Delta\Phi < 180°$ (Figure 6.7(b)) and clockwise twists to an angle $-180° < \Delta\Phi < 0°$ (Figure 6.7(d)).

Lab test of the SHAKEs

We have tested this setup in the lab with several twists of the upper body in both directions. Figure 6.8 shows the data plots of the values of $SHAKE_{upper}$, $SHAKE_{lower}$, and the angular difference $\Delta\Phi$ between these two values. The twists can directly be seen from the plot. First the person's upper and lower body are aligend, then he twists the upper body counter-clockwise and back. A twist in clockwise direction follows and thereafter a twist in counter-clockwise direction.

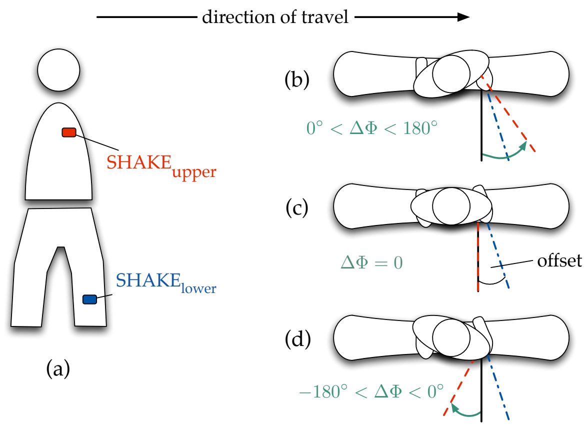


Figure 6.7: SHAKE: (a) attachment of the SHAKE units on a regular footed snowboarder. (b) Counter-clockwise body twist, (c) no body twist, (d) clockwise body twist of upper against lower body.

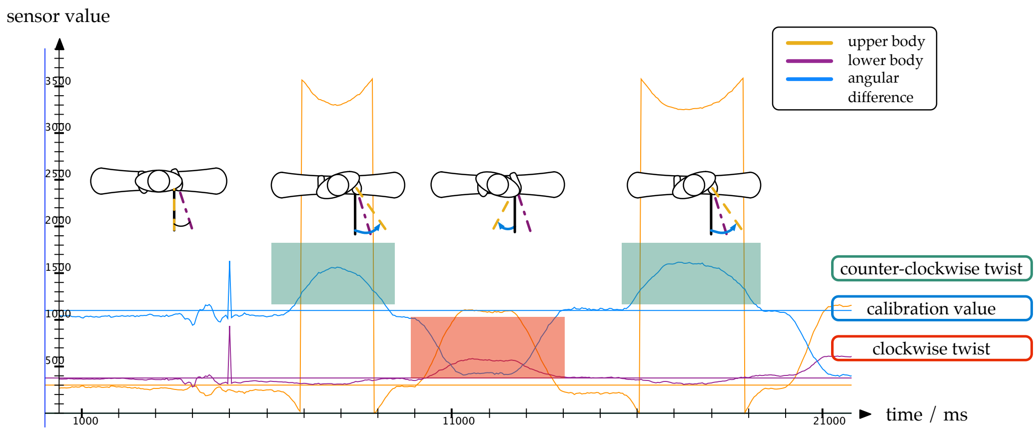


Figure 6.8: The plots show the values of the SHAKEs attached to the upper and lower body. The angular difference $\Delta\Phi$ compared to its calibration value reveals body twists ($\Delta\Phi$ is shifted by 1024 on the y-axis for visualization).

6.5 Final Setup

After incorporating the two SHAKE SK6 sensor units, the final setup of the prototype consisted of (Figure 6.9):

- four FSRs, two under each foot, fixed on an insole
- two bend sensors wrapped in thick foam, attached with knee pads at the back of the knees
- two SHAKE SK6 sensor units, one on the upper body, one on the front shin right above the boot

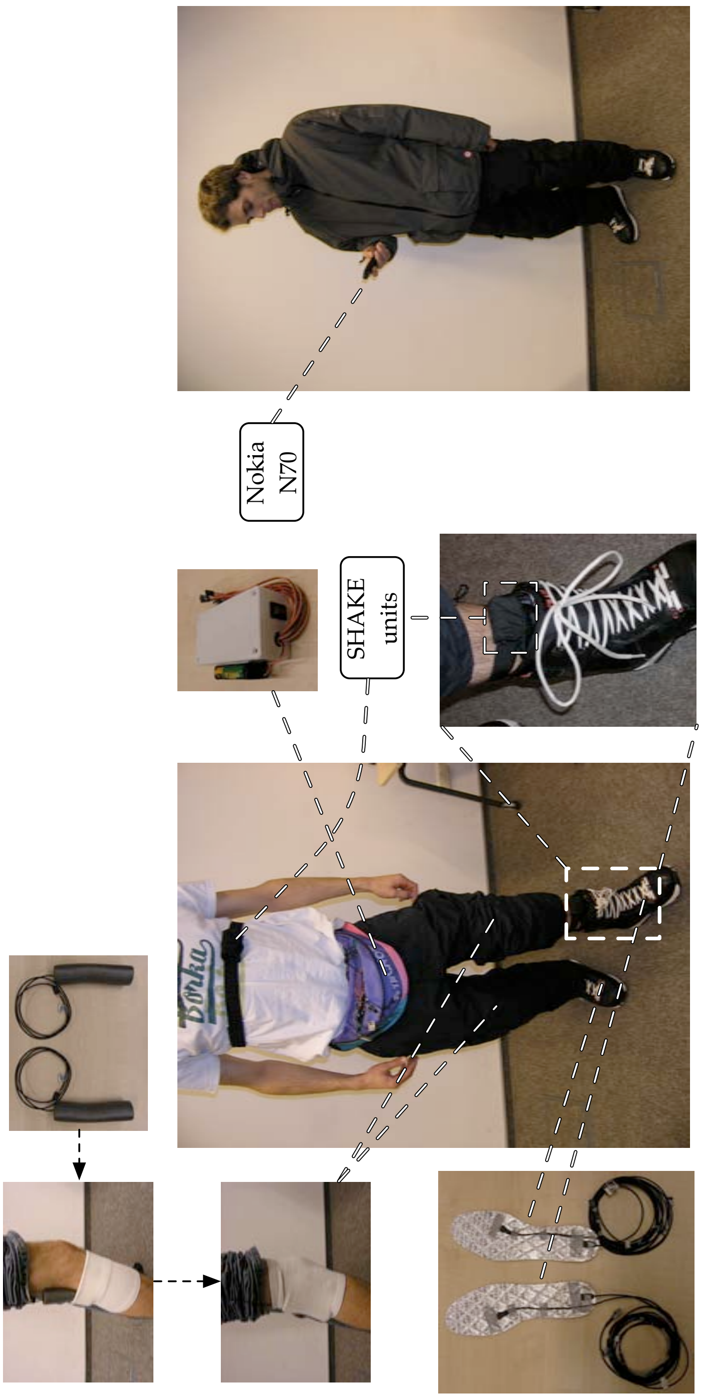


Figure 6.9: The final setup of our prototype. When wearing a jacket, the system cannot be seen.

Chapter 7

User Study and Data Analysis

"To acquire knowledge, one must study; but to acquire wisdom, one must observe."

— Marilyn vos Savant

In order to collect sensor samples under realistic conditions, we have conducted a user study with three snowboard beginners. We have analyzed the collected data with respect to common beginner mistakes. In the following chapter, we outline the test procedure as well as approaches to detect each mistakes made by the test subjects. Based on our findings, we judge the success of the outlined approaches and give recommendations for further improvements.

7.1 Tests with Snowboard Beginners

We recorded sensor and video data of three test subjects on the indoor slope of SnowWorld.

7.1.1 Test Subjects

Advanced beginners as test subjects

Table 7.1 shows an overview of the subjects and their snowboarding experience. Subject 2 has recently started snowboarding, whereas Subject 1 is snowboarding since 9 years. None of the subjects performed the sport on a regular basis, but all of them were able to perform turns on a slope. This justified the term 'advanced beginner', which was the target user group for our prototype.

Person	Subject 1	Subject 2	Subject 3
Gender	male	male	female
Age	25	27	24
Stance	regular	goofy	regular
Riding since	9 years	3 months	1 year
Riding weeks/year	1-2 weeks	3-4 weeks	2-3 weeks
Expertise	advanced beginner	advanced beginner	advanced beginner

Table 7.1: Test subjects in the user study.

7.1.2 Test Procedure

No major hardware flaws during the tests

Before entering the slope, we helped the subjects put on the sensors (FSRs, bend sensors, SHAKEs). Thereafter, we checked if the Bluetooth connection and the sensors worked correctly. Unfortunately, with the last test subject the bend sensor on the right knee was damaged and could not be recorded. Apart from that, no major hardware flaws occurred. On the slope, we set up a camera to record the subjects' runs. It was placed about 50 m away from the point where they started (Figure 7.1).

Adjusting settings and calibrating with the N70 before the run starts

Before each run, we adjusted the settings on the Arduino with the N70. We set the sampling rate to 20 Hz and the stance for the sensor mapping according to the subject's stance. Thereafter, we asked the subjects to stand in the basic stance on flat ground to save the sensor values as calibration data, which was logged by the N70.

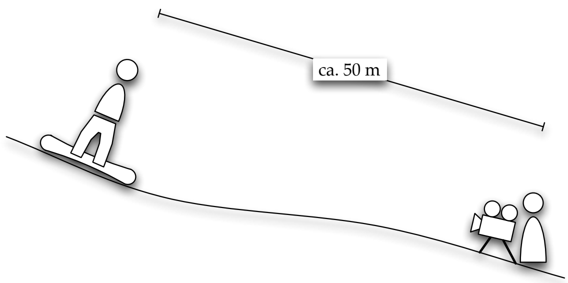


Figure 7.1: Outline of the user test. The snowboarder started his run about 50 m away from the camera.

At the beginning of each run, we asked the subjects to jump so that we could later synchronize sensor and video data. The jump can be clearly identified on both the video and the sensor recordings. During the run, the subjects were supposed to descend the slope as they would normally do. We recorded four to six runs of each subject for later analysis.

A jump at the beginning serves to synchronize video and sensor data

7.2 Approaches to Detect Mistakes

In the following sections, we will give a qualitative analysis of the sensor data. We provide approaches, showing how the sensor data can be analyzed with respect to common beginner mistakes. We analyzed the data with our dedicated software (see sec. 6.2.2—"Off-line Sensor Data Analysis"). The video of the run served as reference to verify, where the subjects made mistakes and to validate that mistakes were recognized correctly by our approaches. Additionally to the sensor data of the test subjects, we analyzed data collected during our initial self tests, including data from the author and a snowboard instructor, both 'advanced snowboarders'.

Qualitative analysis of the test data

In order to simulate a real-time analysis on the sensor data, so that it could also be processed on the Arduino or the N70 during a snowboard run, we adhered the following rules:

Analysis in ascending order. We analyzed the sensor values in ascending order according to their timestamp.

This was done in a for-loop starting from the first sample and ending at the last one.

No processing of future samples. In every step, we only processed data of the sensor samples with the current timestamp. Additionally, we used past samples. Succeeding samples were not considered as they were 'future' samples.

Use of calibration data. The calibration values for every sensor were known. These had been captured at the beginning of each run and stored on the Arduino.

Known time steps. The time step (in milliseconds) between each sample was known because of the timestamps logged by the microprocessor of the Arduino.

Detecting coarse levels of mistakes

To process the data, we proceeded like described in Section 5.2—"Design for Mistake Detection" by comparing the current values with the calibration values. To show the feasibility of real-time analysis of snowboard movements, we aimed at detecting coarse levels of mistakes, e.g., 'knees are straight' and 'knees are bent'(see sec. 5.2.1). For analyzing the slope data we made use of the following:

Removing noise. We used smoothing filters on the noisy slope data. We tested a simple moving average and an exponential moving average (see C—"Smoothing Filters").

Tolerance range. The tolerance range around the calibration value was set by hand. This would not be possible in real-time analysis. Later we will give suggestions on how it could be derived automatically.

Time thresholds. Even after applying smoothing filters, the recorded sensor data remained noisy. To prevent false detections when sensor data jumps up and down due to the uneven slope, we experimented with time thresholds: a value was only considered outside the tolerance range, if it stayed above or below for longer than the specified timeout.

To visualize the regions where the sensor values are outside and inside their tolerance range, we drew a new graph

$g_{map}(t)$ that mapped these regions to discrete values (an example is given in the following sections):

$$g_{map}(t) = \begin{cases} 800 & \text{if } value(t) \text{ is above the tolerance range} \\ 500 & \text{if } value(t) \text{ is within the tolerance range} \\ 200 & \text{if } value(t) \text{ is below the tolerance range} \end{cases}$$

Mapping is only for visualization

The values for the mapping were chosen to stay within the range of the sensor readings of the Arduino (0–1023), so that they could be viewed together with the original sensor plot. The values have no further meaning. The mapping is used solely as visualization. Three horizontal lines with different y-axis components represent the three different regions we distinguished. The graph $g_{map}(t)$ allows to directly interpret the results of our approaches.

7.2.1 Weight Distribution

We consider two aspects of the snowboarder's weight distribution: *toe–heel distribution* (see sec. 5.2.3) and *front–back distribution* (see sec. 5.2.3).

Toe–Heel Distribution

Figure 7.2(a) shows the values F_{ft} (front toe) and F_{fh} (front heel) of the FSRs under the front foot of the author. The previous chapter already outlined how to interpret these values (see. Figure 6.5). Figure 7.2(b) plots the difference between these two values $(\Delta F)_{ff}$ and the according calibration value $(\Delta F)_{ff}^{cal}$ (cp. sec. 5.2.3—"Toe–Heel Distribution"). Additionally, we set a tolerance range around the calibration value (tolerance value $= 70$).[1] In the graph, the calibration value is displayed as horizontal line. The tolerance range is visualized as the region between two horizontal lines above and below the calibration value.

Tolerance value was set by hand

For the analysis, the tolerance value was set by hand, based

[1] The value was set based on our experience with the sensor recordings.

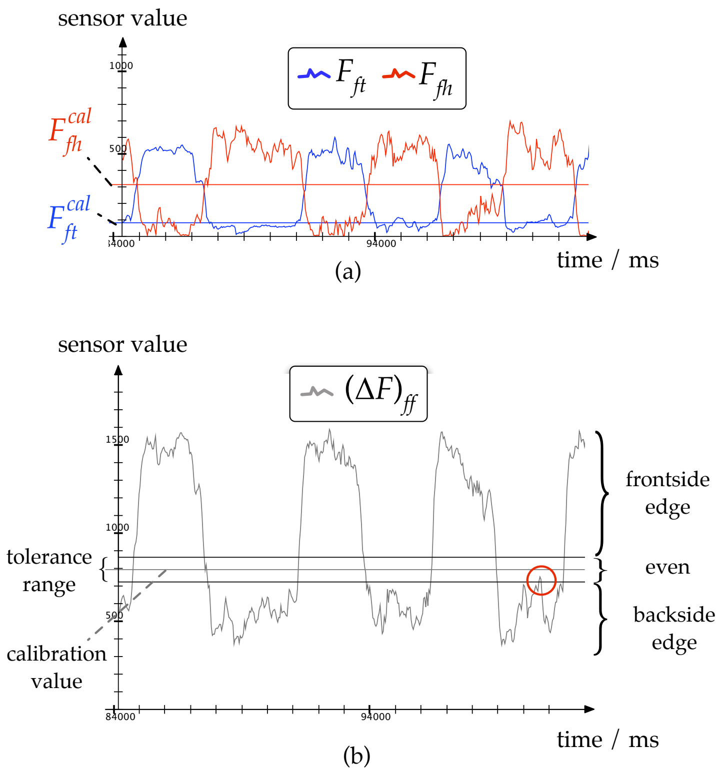


Figure 7.2: (a) The FSRs under the front foot of the author during turns. (b) The difference between the toe-side FSR and heel-side FSR (shifted by 1024 on the y-axis to prevent negative values). The red circle highlights a spike that reaches into the tolerance range.

on the sensor recordings. This violates the real-time behavior of this approach. As soon as enough analysis is conducted, we believe that we will be able to derive an appropriate tolerance value from the calibration values (similar to the *TR* in Section 3.2.2—"Biofeedback Wireless Wearable System").
Another approach would be to set the tolerance value on the slope via the N70. A smaller tolerance value could enforce more precise mistake detection. Moreover, in a real-world setting, it could be necessary to customize the mistake detection according to the student's skills.

Distinction between frontside and backside edge is possible

In accordance with the video, values above the tolerance range in Figure 7.2 mean that the snowboarder's weight is on his toes and thus he is riding on the frontside edge. Conversely, values below the tolerance range indicate that he is riding on the backside edge. Values within the tolerance range indicate the pivoting process from one edge to the other, e.g., during turns. Unfortunately, this pivoting takes place too fast to be recognized from the recordings. Nevertheless, a clear distinction between frontside and backside edge is possible.

Exponential moving average to filter noisy data

Figure 7.2(b) indicates a spike where the value of $(\Delta F)_{ff}$ briefly enters the tolerance range, although the snowboarder is still on his backside edge.[2] To smooth such spikes, we tested a simple moving average and an exponential moving average with different parameters. We chose the exponential moving average because compared to the simple moving average it stayed closer to the original sensor values and still filtered out spikes.[3] The behavior of the exponential moving average can be adjusted with the smoothing parameter $\alpha \in [0, 1]$. The higher the value the more the data is smoothed (Figure 7.3)

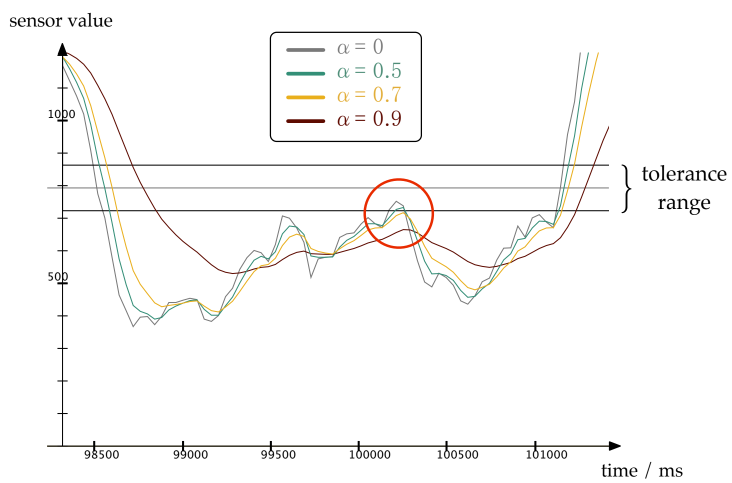


Figure 7.3: Detailed view on the spike highlighted in Figure 7.2(b). A higher smoothing factor introduces more latency relative to the original plot ($\alpha = 0$).

[2] This can be seen in the video recording.

[3] This observation is based on experiments with the sensor data. For future developments we will have to test more filters to decide which one is the most appropriate.

Filtering introduces latency

Figure 7.3 shows a close up of the spike and various smoothed graphs with different settings for α. A higher value for α smoothes the graph more but introduces latency. A value of $\alpha = 0.7$ seems a good compromise between smoothing behavior and latency. Values below 0.5 have almost no smoothing effect and values above 0.7 introduce a latency of almost 500 ms (Figure 7.3). After smoothing the graph, the spike no longer reaches into the tolerance range and it is possible to clearly separate the different regions.

As stated earlier we want to map the different regions to a graph $g_{map}(t)$ with discrete values. For the *toe–heel distribution* we interpret the mapping as follows:

$$g_{map}(t) = \begin{cases} 800 & \text{on frontside edge at time } t \\ 500 & \text{evenly distributed at time } t \\ 200 & \text{on backside edge at time } t \end{cases}$$

Figure 7.4 shows the original graph and the discrete mapping. We see that the approach indeed separates *toe–heel distribution* clearly between frontside and backside.

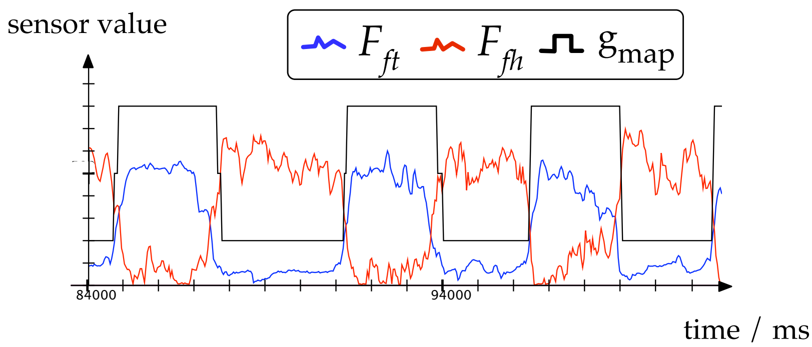


Figure 7.4: The result of the discrete mapping.

Problems determining *toe–heel distribution* with the beginners' *front* feet

When using the same approach to evaluate the sensor recordings of the beginners, we realized that using the *front* foot to identify *toe–heel distribution* only worked correctly for advanced riders. Nevertheless, for the beginners we were able to derive information on the *toe–heel distribution* from the *back* foot.

Figure 7.5(a) shows the plots of the FSRs under the front foot of Subject 1. In this case the discrete mapping for the *front* foot does not clearly indicate when he is riding on the backside edge. Figure 7.5 shows the results of the same approach using the subject's *back* foot. The mapping is much clearer.

Evaluating the *back* foot leads to a clear indication of *toe–heel distribution*

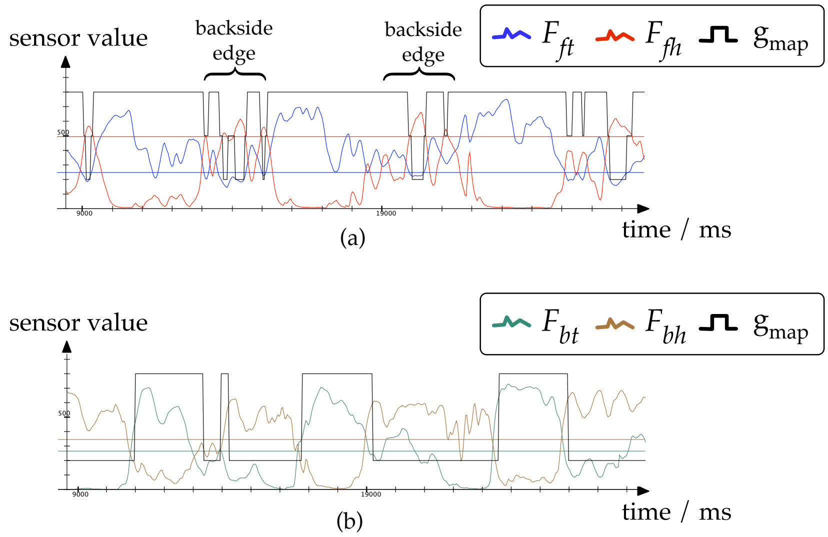


Figure 7.5: (a) The graph shows the FSRs under the front foot of Subject 1. The discrete mapping does not clearly indicate when the subject is on the backside edge. (b) This graph shows the same time segment with the values of the back foot. The discrete mapping yields clearer results.

We stated earlier that the front foot shows the alternation between frontside and backside edge because it is the leading foot. However, beginners tend to lean too much towards their back foot (cp. 4.2.1—"Wrong Weight Distribution"). More pressure on the back foot keeps it more stable than the front foot. This explains the results. Thus, to infer *toe–heel distribution* reliably, we must observe the foot with the highest pressure. This information can be derived from the *front–back distribution*.

Foot with highest pressure indicates *toe–heel distribution*

Front–Back Distribution

Combining information on *toe–heel distribution* and *front–back distribution*

To analyze *front–back distribution* we calculate values as described in 5.2.3—"Front–Back Distribution". The difference $(\Delta F)_{ff-bf}$ between the front foot and the back foot of Subject 2 is shown in Figure 7.6(a) along with its calibration value $(\Delta F)^{cal}_{ff-bf}$ and a tolerance range (tolerance value = 100).[4] Additionally, we have plotted the discrete mapping of the *toe–heel distribution* to see where the subject performs turns. In this special case, we shifted the mapping along the y-axis so that all graphs can be viewed separately without interference. As stated earlier, the actual values of the mapping do not have any further meaning. The relation between the values is important.

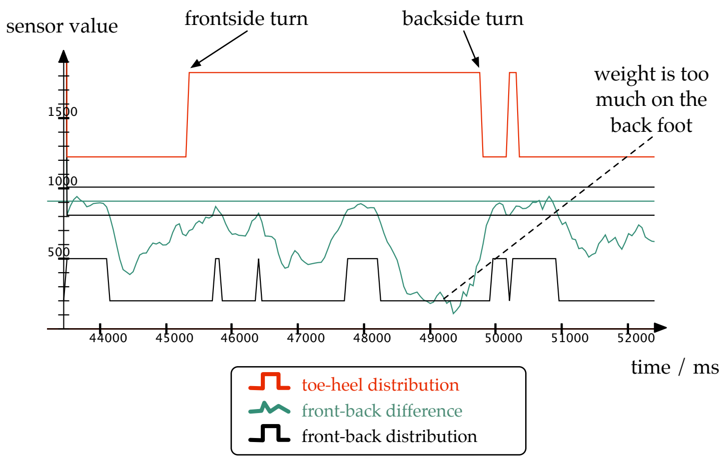


Figure 7.6: The discrete plot in red indicates the three levels of *toe–heel distribution*. The discrete plot in black visualizes the three levels of *front–back distribution*. The difference between the values of the front foot and the back foot is plotted together with its calibration value and the surrounding tolerance range. Values within the tolerance range indicate a centered weight distribution between front and back foot. Values below indicate a shift towards the back foot and values above a shift towards the front foot.

When taking a closer look, we see that the subject's weight balance is not centered. Especially before turns, he shifts

[4] This value was set through experimentation.

his weight more towards the back foot, which results in a value below the tolerance range. This is a common beginner mistake as identified in 4.2.1—"Wrong Weight Distribution". The discrete mapping $g_{map}(t)$ shows the three levels of *front–back distribution*.

$$g_{map}(t) = \begin{cases} 800 & \text{weight on front foot} \\ 500 & \text{weight equally distributed} \\ 200 & \text{weight on back foot} \end{cases}$$

For comparison Figure 7.7 shows the *front-back distribution* of the author. The weight is never on the back foot— except for one spike.

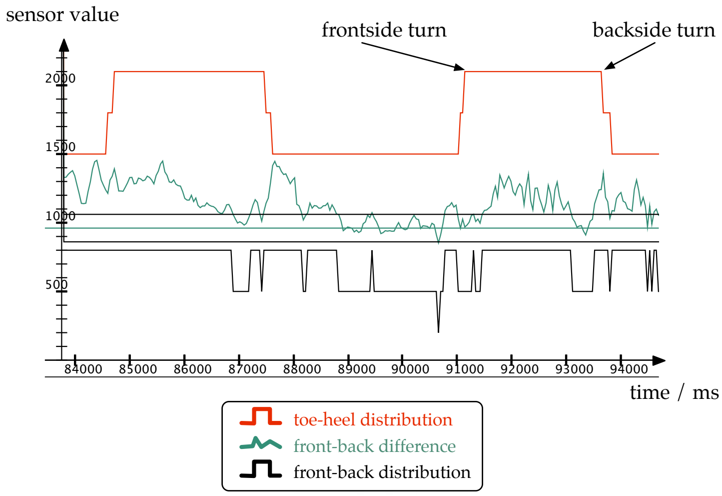


Figure 7.7: Plot of the author's *front–back distribution*. Weight is mostly on the front foot.

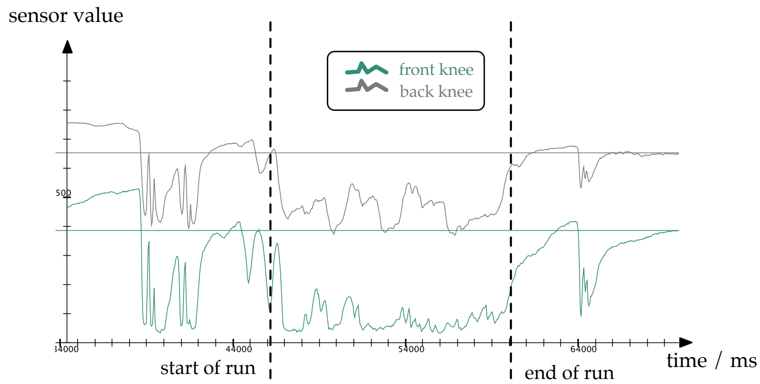
Evaluating the other sensor recordings led to similar results. However, in some runs the *front–back distribution* could not clearly be separated into the three levels. This was due to poor calibration values. We will discuss problems during the calibration process in Section 7.3—"Summary". Nevertheless, when proper calibration values were available, we could identify wrong weight distribution with the FSRs.

Poor calibration values prevent proper evaluation in some cases

7.2.2 Knee Bending

When recording Subject 3 the bend sensor on her back knee ceased to work and could not be evaluated. However, for the other test subjects the bend sensors worked properly. Furthermore, we did not encounter problems with the proper fixation of the sensors on the knees like we did in our initial test runs (cp. sec. 6.3.1). Figure 7.8 shows two data plots, each of them showing values of the two bend sensors on the knees.

(a)



(b)

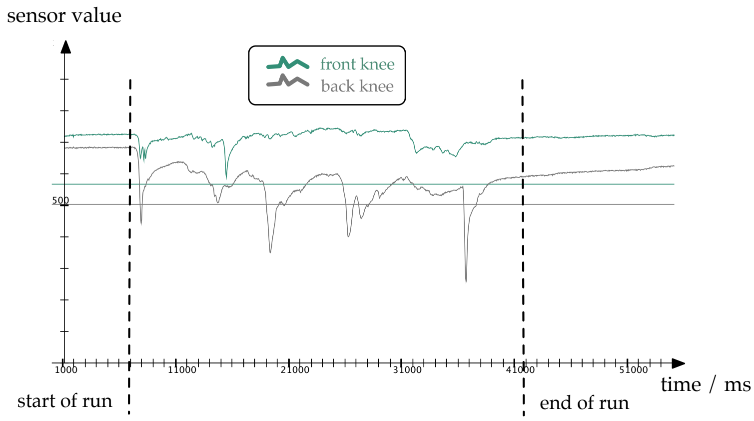


Figure 7.8: Plots of the bends sensors on the front knee and the back knee of two snowboarders: (a) the snowboard instructor, (b) Subject 1. During the run the values of (a) were below the calibration value. Those of (b) were above with few exceptions, indicating too straight knees.

Bend sensors make the distinction between advanced snowboarders and beginners possible

Plot (a) shows the snowboard instructor's values, plot (b) the recordings of Subject 1. During the run, the values in (a) are always below the calibration data, whereas in (b) they are above most of the time. As we stated earlier, the higher the value the straighter the knees. Therefore, Subject 1 did not have both of his knees bent enough. We set the tolerance value based on the video recording to indicate too straight knees (tolerance value = 50). In the case of the snowboard instructor's plot and the front foot of Subject 1 we omit the discrete mapping. The raw data plots can directly be interpreted. Figure 7.9 shows the plot of the back knee of Subject 1 with the discrete mapping:

$$g_{map}(t) = \begin{cases} 800 & \text{knee is straight} \\ 500 & \text{knee is bent} \\ 200 & \text{knee is bent even more} \end{cases}$$

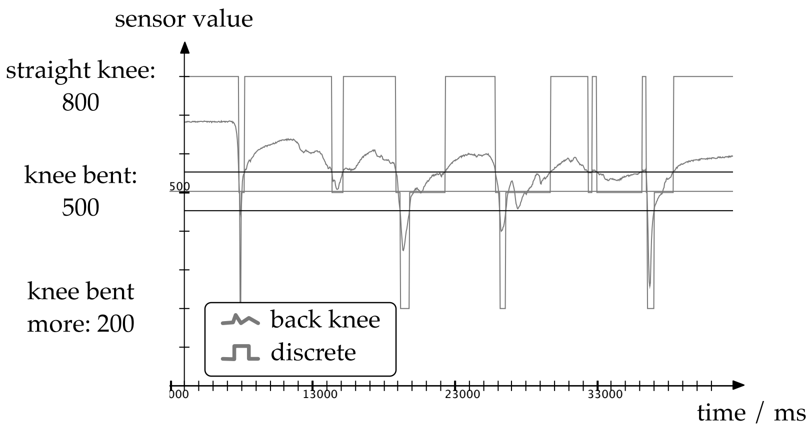


Figure 7.9: Sensor values of the back knee of Subject 1 and corresponding discrete mapping (tolerance value = 50).

We see from the mapping that his back knee was only temporarily too straight (Figure 7.9), but on a regular basis. Analysis of the plots of Subject 2 and the author led so similar results. For Subject 3 we could only analyze her front knee, which was not bent sufficiently.

Resistive bend sensors are sufficient for snowboard beginners

The resistive bend sensors we used enabled us to detect coarse levels of knee bending. This was sufficient for snowboard beginners. For advanced snowboarders we will have to measure knee bending more accurately. The used bend sensors will most likely not provide this information. The measurement of the bend sensors was highly dependent on the placement on the knees, which can be seen on the data plots in Figure 7.8(a). The readings of the bend sensors differ greatly in their y-axis component, even when they are flexed to the same extent.[5] We will have to take other sensors into consideration whose outputs depend only on the flexion angle. A promising choice could be optical sensors like described in [Kuang et al., 2002]. These react linearly on bending.

7.2.3 Upper Body Posture

Accelerometer tested with one subject

Although we did not further investigate the use of an accelerometer to measure the tilt of the upper body, we tested it with Subject 1. As we stated in the previous section, this subject did not bent his knees sufficiently. Additionally, he did not keep his upper body upright during runs. Figure 7.10 shows the values of the accelerometer on his upper body, which was attached as in the first prototype (Figure 5.4).

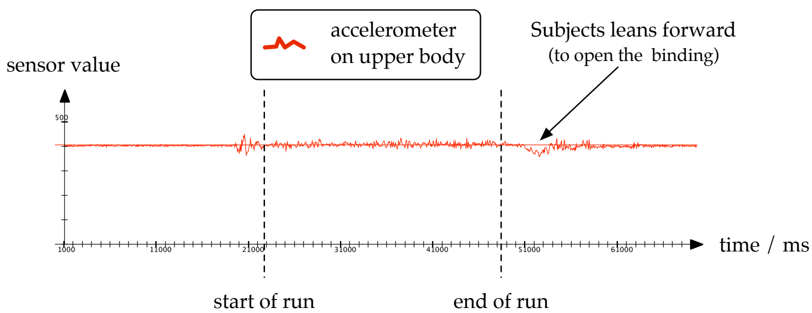


Figure 7.10: Values of the accelerometer attached on the upper body of Subject 1. The reading do not show body leanings during the run.

Dynamic acceleration is too high to derive information about tilt

The values do not provide information about the current body tilt of the subject. The dynamic acceleration during

[5] This was observed on the video recordings.

the run was too high to clearly indicate acceleration due to gravity. Only when the subject was at rest, his upper body tilt could be measured. At the end of the run Subject 1 leaned forward to open his bindings. This can be seen in the sensor plot.

As we stated in Section 4.2.1—"Wrong Upper Body Posture", problems with the upper body posture occur often in combination with insufficient knee bending. Since we are able to detect the knee flexion, the inability to measure the upper body posture does not impact the mistake detection to a great extent. Moreover, in the next section we will provide alternatives to detect this mistake.

Poor upper body posture occurs often with insufficient knee bending

7.2.4 Counter-Rotation

In the lab the values of the SHAKE units were smooth and could give a clear indication on the current orientation (cp. sec. 6.4.2—"Detecting Counter-Rotation"). On the slope these values were distorted through the movements of the snowboarders. Figure 7.11 shows the angular difference $\Delta\Phi$ between SHAKE$_{upper}$ and SHAKE$_{lower}$ during a run of Subject 1. He did *not* twist his upper body against his lower body. However, from the readings we would conclude that this happened often.

Readings of the SHAKE units are distorted through movements

Unfortunately, the SHAKE units could not measure absolute orientation reliably when they were subject to quick movements. We contacted their manufacturer to get more information. Although the SHAKE units have a gyroscope unit, this is not incorporated in the calculation of the absolute orientation. It depends solely on the magnetometer and the accelerometer. Rapid movements affect the accelerometer readings, compromising the calculation of the absolute orientation. We will discuss alternatives for the SHAKE units at the end of this section.

SHAKE units do not reliably provide orientation on the slope

Despite the inaccuracies of the SHAKE units, we could evaluate the recordings of Subject 2 with respect to counter-rotation. As the least experienced subject, he descended the slope slowly and made no quick movements on the snowboard. The SHAKE values were only slightly distorted and

Evaluation of Subject 2 was possible due to his slow movements

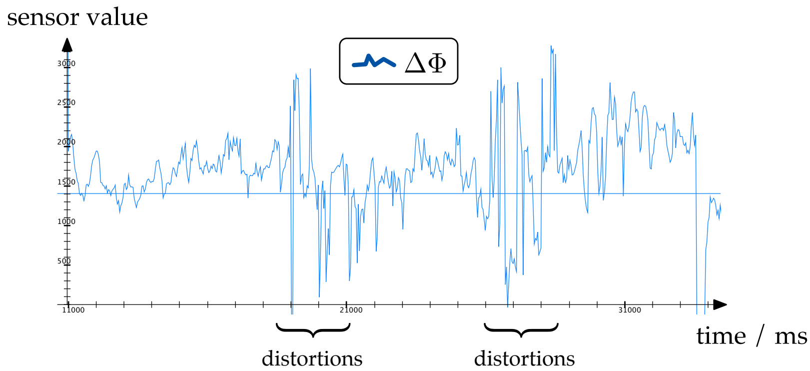


Figure 7.11: Plot of the angular difference $\Delta\Phi$ between the SHAKEs on the upper body and the lower body of Subject 1. The distortions evoked through the movements of the snowboarder are too great to derive meaningful information.

could be analyzed. Figure 7.12 shows the angular difference $\Delta\Phi$ between the two SHAKE units attached to Subject 2. He often used counter-rotation to perform turns. Three video snapshots from the run illustrate how the subject performed backside turns: To turn around the snowboard, he twisted his upper body against his lower body in counter-clockwise direction. This can be seen from the raw data plot, where the value of $\Delta\Phi$ rises high above the calibration value for one second.

To account for the unstable readings of the SHAKE units, we employed a timeout. A twist was only indicated, if the reading exceeded the tolerance value for more than 250 ms.[6] The resulting mapping of this approach which maps $\Delta\Phi$ on a discrete graph is also shown:

$$g_{map}(t) = \begin{cases} 800 & \text{counter-clockwise twist} \\ 500 & \text{upper body and lower body aligned} \\ 200 & \text{clockwise twist} \end{cases}$$

Mapping indicates body twists

We see that the timeout introduces latency and lacks about 250 ms behind the actual twist. Nevertheless, the mapping indicates the twist successfully. To show the relation-

[6] Like the tolerance value, this was set by hand.

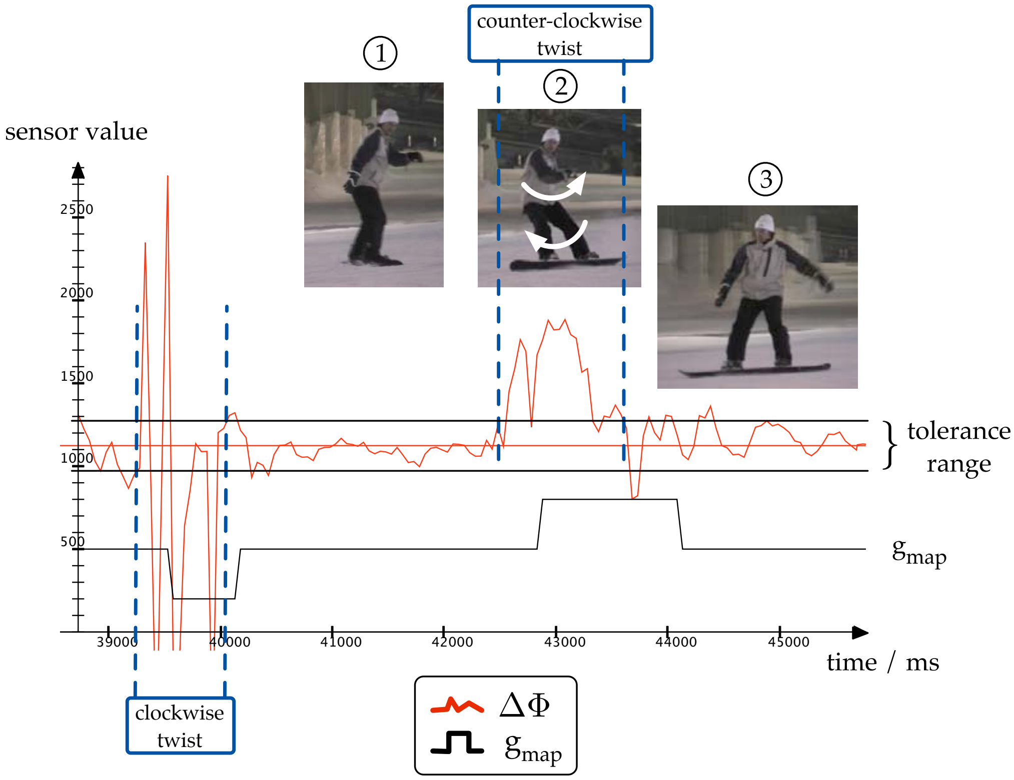


Figure 7.12: Counter-rotation of Subject 1. Snapshot ② from the video shows that the upper body of the subject is twisted counter-clockwise against his lower body. This is indicated by an increase in the reading of the angular difference $\Delta\Phi$ high above the calibration value.

ship between body twists and the stage within a turn, Figure 7.13 shows the discrete mapping of the subject's *toe–heel distribution* and the discrete mapping of the angular difference. The graph shows the whole run of the subject. Backside turns are indicated by a shift from the frontside edge to the backside edge, frontside turns vice versa.

We note the following:

- Before and during a backside turn, i.e., a turn in clockwise direction for the subject, he twisted his upper body in counter-clockwise direction.
- After having performed a frontside turn, i.e., a turn in counter-clockwise direction for goofy snowboarders,

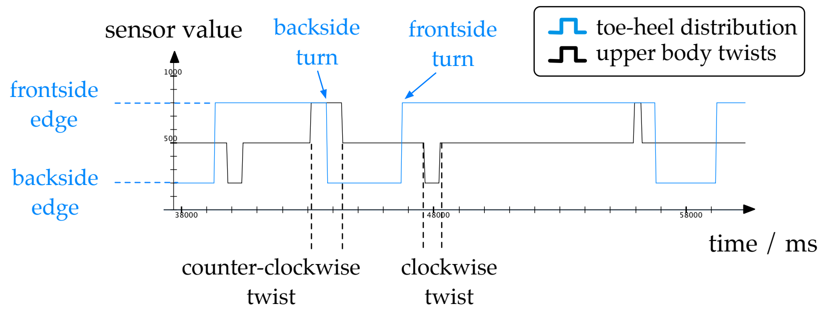


Figure 7.13: Discrete plot of the *toe–heel distribution* and the body twists of Subject 1

Subject 2 twisted his upper body in clockwise direction.

Combining values of FSRs and SHAKEs makes detection of counter-rotation possible

In both cases the upper body was twisted contrary to the turning direction, which we already saw on the video recording. The mappings, however, show that this can also be derived by combining the information of *toe–heel distribution* from the FSRs and the angular difference between upper and lower body from the SHAKE units. By this counter-rotation, a common mistake among snowboard beginners, can be detected.

Alternatives for the SHAKE units

As the SHAKE units only worked reliably, when the subjects were moving slowly, we will have to look for alternatives. As mentioned in Section 2.5—"Inertial Measurement Unit (IMU)", another IMU is the *MTx* from XSens. According to its data sheet, it measures absolute orientation angles in three dimensions even when moved quickly. This would also solve the problem of measuring the upper body tilt. The device has already been used to monitor speed skaters.[7] . We assume that it would also work for snowboarders. Unfortunately, these devices are expensive. That is why we dismissed them in the first place and used the SHAKE units. The challenge for future developments will be to find a reliable way to measure absolute orientation during snowboarding runs. The approach to detect counter-rotation has been outlined successfully in this sec-

[7]http://www.xsens.com/

tion.

7.3 Summary

Table 7.2 shows an overview of the analysis of the sensor values. We were able to address most of the target common mistakes. Although not every mistakes could be detected for every subject, we outlined approaches to detect the mistakes in real-time.

Parameter	Author	Instructor	Subject 1	Subject 2	Subject 3
Toe–heel dis.					
W/ front foot	$\surd$	$\surd$	–	($\surd$)	–
W/ back foot	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Front–back	$\surd$	$\surd$	($\surd$)	$\surd$	($\surd$)
Counter-rotation	–	–	–	$\surd$	–
Knee bending	$\surd$	$\surd$	$\surd$	$\surd$	–

Table 7.2: Success of our approaches to derive several parameters from the test subjects' sensor readings ($\surd$ = Success, – = No success, ($\surd$) = Partly successful).

We will give a brief discussion on the success of our approaches:

Toe–heel distribution. Riding on the frontside edge and riding on the backside edge could be distinguished for all test persons. To infer the *toe–heel distribution* for advanced snowboarders and beginners alike we needed to observe the foot with the highest pressure on it. This can be determined by the *front–back distribution*.

Front–back distribution. The *front–back distribution* could be derived from most of the sensor data. In some cases the approach did not lead to meaningful results due to poor calibration values, e.g., in some runs of Subjects 2 and 3. The calibration process remains an open problem. We will discuss it at the end of the section.

Upper body posture. The upper body posture could not be inferred by the 3-D accelerometer on the slope. This problems needs to be addressed in further iterations. However, as this mistake often occurs together with straight knees, measuring knee bending partly solves this posture related mistake.

Counter-rotation This mistake was successfully detected with the least experienced subject. He descended the slope slowly enough, so that the values of the SHAKE units were only slightly distorted by the accelerations of his movements. Although we could only show the detection for Subject 2, the approach is still promising. If other sensor are incorporated that return the absolute orientation of the upper body and the lower body reliably then the outlined approach can be used for all subjects.

Knee bending. We have outlined an approach to detect too straight knees, a common beginner mistake.

Calibration process is difficult with snowboard beginners

One problem remains the calibration process. Snowboard beginners do not know how to stand properly in the basic stance. Even when standing still on flat ground they had problems to distribute their weight equally on their feet. In particular, they often stood on either their toes or their heels. This could be seen afterwards from the sensor readings, when the calibration values of the FSRs on the heel and on the toe side differed noticeably.

Nonetheless, the FSRs have proven to be useful to analyze the weight distribution under the feet. Two FSRs for each foot enabled us to detect coarse levels of weight distributions. For a more detailed information on the weight distribution we should incorporate more FSRs. Musselman et al. [2007], for example, have developed an insole incorporating fourteen FSRs to enhance the *GaitShoe* (sec. 3.2.1).

Tolerance range needs to be derived from the calibration values

We have outlined how the common beginner mistakes can be identified with the sensor readings. We still need to implement algorithms that act autonomously. Right now we have set the tolerance range by hand based on the experience with the data recordings. These values should be derived from the calibration data automatically (similar to the Target Range in 3.2.2—"Biofeedback Wireless Wearable System"). We still need to evaluate sensor data of more snowboarders. Nevertheless, the approaches discussed in this section can be extended to work in real-time, as they do not consider 'future' samples.

Chapter 8

Summary and Future Work

> *"Always remember that the future comes one day at a time."*
>
> *—Dean Acheson*

This chapter sums up the results from the previous chapters and outlines future work that needs to be done in the development of the *Snowboarding Assistant*.

8.1 Summary and Contributions

In this thesis we have presented initial steps towards a wearable *Snowboarding Assistant*. Based on the results of interviews with snowboard instructors, we selected appropriate sensors to detect common beginner mistakes in snowboarding. In a first lab prototype we explored possibilities to process the sensor data. In the next step we developed a mobile sensor system that was robust enough to endure the conditions on the slope. After initial tests on the slope, we improved the hardware resulting in the final prototype which we have tested with three snowboard beginners. Data analysis of sensor values collected during this test have shown the feasibility of real-time analysis of snowboard movements with the current setup.

With respect to Section 1.1.4—"Requirements" we have achieved the following:

Exploring the Application Domain. We have given an in-depth overview of important terms and techniques in the snowboarding domain in Chapter 4—"The Snowboarding Domain".

Opportunities for Change. Through literature review and interviews with four snowboard instructors we have identified common beginner mistakes. The instructors have confirmed that incorporating a device as the *Snowboarding Assistant* into snowboarding lessons would be useful. They have also contributed new ideas that we will consider for our future developments.

Robust Hardware. In order to detect common mistakes, we have selected appropriate sensors. After exploring the possibilities to process the sensor data in the lab, we have built a robust, mobile sensor platform which we tested in several trials on the slope. The hardware is unobtrusive to wear and does not hamper the snowboarder. After improving the prototype based on the experience gained from the initial tests on the slope, we have conducted a user study with three snowboarder beginners on the slope.

Algorithms to Detect Mistakes. Based on our knowledge about the snowboarding domain, we have developed several approaches to detect common mistakes. After having collected sensor data from the field test, we have presented different real-time approaches to identify each of the common mistakes. We have tested these approaches with the recorded data sets and discussed their success. Although we did not implement the approaches on the mobile device, the results suggest that real-time detection of snowboard mistakes is feasible with a wearable mobile device as presented in this thesis.

8.2 Future Work

Our final prototype is a good starting point for any further developments towards a wearable *Snowboarding Assistant*. As this was only the first step of the project a lot of work remains to be done

Final prototype as starting point

The hardware was sufficient for an initial mobile prototype. We were able to identify common beginner mistakes with the selected sensors. However, for advanced snowboarders the hardware needs to be improved to achieve more accurate sensor values. The quality of the mistake detection depends on the quality of the sensor data. In particular, the following improvement should be considered for detecting the more subtle mistakes of advanced snowboarders:

Hardware must be improved

Weight distribution. Incorporating more FSRs under the feet could reveal more details of the snowboarders weight distribution. For example Musselman et al. [2007] have extended the insole of the *GaitShoe* to incorporate 14 FSRs to measure the weight distribution under the feet more precisely.

Knee bending Similar to the FSRs, a more precise way of measuring knee flexion should be investigated, e.g., optical sensors like presented in [Kuang et al., 2002].

Counter-Rotation We have outlined an approach to detect counter-rotation with the SHAKE units. Unfortunately, they only provided good result for slow descending snowboarders. For monitoring advanced snowboarders, the sensor units from XSens, e.g., the *MTx*, might be able to measure absolute orientation even with fast movements. Other approaches to capture human motions with body-worn sensors are still under development and yield promising results. The *Snowboarding Assistant* could benefit from such technology, e.g., the motion capture approach in [Vlasic et al., 2007]. This would also solve the problem of tracking the snowboarder's upper body posture.

We have limited our prototype to one Arduino board with

Incorporating more sensors

six sensor inputs. To get more reliable sensor data we will have to incorporate more sensors in future prototypes. We plan to use several Arduino boards that communicate with the N70 or another mobile device. The mobile device combines the information from the sensors connected to the different Arduino boards and responds with appropriate feedback.

Type of feedback needs to be investigated

One of the most important aspects that needs to be investigated in the future is the type of feedback that could be given to snowboarders. Visual feedback might distract the beginners to much. Auditory or tactile feedback seems more appropriate.

Does the *Snowboarding Assistant* improve the learning process?

As soon as the *Snowboarding Assistant* incorporates both components — mistake detection and feedback — the main question whether or not it improves the learning process needs to be evaluated under real conditions. Results found here could be applied to other sport areas as well.

Appendix A

Interview Guideline

When we conducted interviews with snowboard instructors we followed an interview guideline. Section A.1 shows the original german guideline, section A.2 the english translation.

A.1 Interview Guideline (German)

Ablauf eines Anfängerkurses:

- Wie lange dauert typischerweise ein Anfängerkurs (Stunden pro Tag, Dauer insgesamt in Tagen)?
- Mit welchen Übungen beginnt der Snowboardunterricht? Wann werden Kurven gefahren?
- Was sind die Ziele eines Anfängerkurses? Was sollten die Schüler nach dem Kurs beherrschen?
- Wie gehst du vor, wenn du den Schülern die Fahrtechnik beibringst (Vorfahren, Erklären, Übungen, . . .)?

Typische Anfängerfehler

Typische Anfängerfehler:

- Gibt es Fehler die typisch sind für Anfänger (bitte aufzählen und beschreiben)?
- Wie kannst du Fehler bei deinen Schülern erkennen?
- Hat du manchmal Probleme zu erkennen wo der Fehler liegt?
- Warum treten diese Fehler deiner Meinung nach auf?

Verbessern der Fehler

- Bitte beschreibe jeweils, was du im Fehlerfall machst, um dem Anfänger zu helfen.
- Wie gehst du vor wenn du **während des Fahrens** einen Fehler beim Schüler beobachtest und ihn verbessern möchtest (z.B. zurufen, warten bis Schüler runtergefahren ist und danach Verbessern)?
- Ist räumliche Trennung von Anfänger und Lehrer eine Situation die häufig auftritt? Welche Probleme gibt es dadurch? Wie versuchst du diese Probleme zu umgehen?
- Können alle Schüler deine Verbesserungen sofort umsetzen? Wenn nicht, woran könnte das liegen?
- Hast du das Gefühl, das einige Schüler während des Fahrens vergessen was du Ihnen zuvor gesagt hast?
- Glaubst du dass du den Schüler besser unterstützen könntest, wenn es möglich wäre, dass du während der Fahrt neben ihm stehst und ihm direktes Feedback gibst?

Vorstellen der Idee *Snowboarding Assistant*

Nach dem Präsentieren der Idee folgende Fragen:

- Glaubst du, dass direktes Feedback für die Schüler sinnvoll wäre? Für Anfänger / Fortgeschrittene / Experten?
- Welche Körperteile sollten gemessen werden?
- Wie könnte das Feedback aussehen (Audio, Video, taktil, ...)? Was ist deiner Meinung nach wichtig?
- Wie könnte der Assistent den **Snowboardlehrer** unterstützen?
  - Fehler besser erkennen (z.B. bei Gewichtsverteilung)
  - Fehler den Schülern am Display zeigen etc.

A.2 Interview Guideline (English)

Content and Steps of Snowboarding Lessons

- Duration of the course: What is the average amount of hours per day and the total of days?
- First lessons exercises: What exercises are done at the beginning? When do you teach your students how to perform turns?
- Goals of the course: Which are the students' achievements at the end of the course?
- Approaches to convey the technique: How do you explain turning techniques to your students (e.g., by demonstration, by explanation, by exercises)?

Typical Mistakes of Beginners

- Do you consider several mistakes common for snowboard beginners? Please specify and describe these characteristic mistakes in detail.
- Describe how you recognize each of the mistakes.
- Do you sometimes have problems recognizing the mistakes?
- Can you think of reasons why these mistakes occur?

Handling Problems of Beginners

- Please describe what you usually do in order to help your students resolve their mistakes.
- What opportunities do you have to communicate suggestions for improvement to your students **while they are riding** and you are not close-by (e.g., call out to them, tell them when approaching)?

- Is spatial distance between students and instructor a common situation during snowboarding lessons? What problems are raised due to spatial distance? How do you usually work around these problems?
- Are your students able to put your instructions into practice immediately? If not, can you think of reasons why?
- Do you feel that some students forget the just given instructions when they start riding?
- Do you think that direct feedback would improve the students performance (imagine you stand on the same board with them)?

Presenting the Idea of The *Snowboarding Assistant*

After presenting the idea of the *Snowboarding Assistant* ask the following questions:

- Can you imagine direct feedback being useful for all levels of expertise? For beginners / advanced riders / experts?
- Which parts of the body should be measured?
- How could the Feedback be realized (audio, video, tactile, ...)? What do you consider important?
- The *Snowboarding Assistant* could not only support the student, but also the instructor. Can you imagine how the *Snowboarding Assistant* could support you?
  - help to recognize mistakes (e.g., better analysis of weight distribution)
  - show students their mistakes on a screen etc.

Appendix B

MAX/MSP Patches

B.1 MyCubePatch.pat

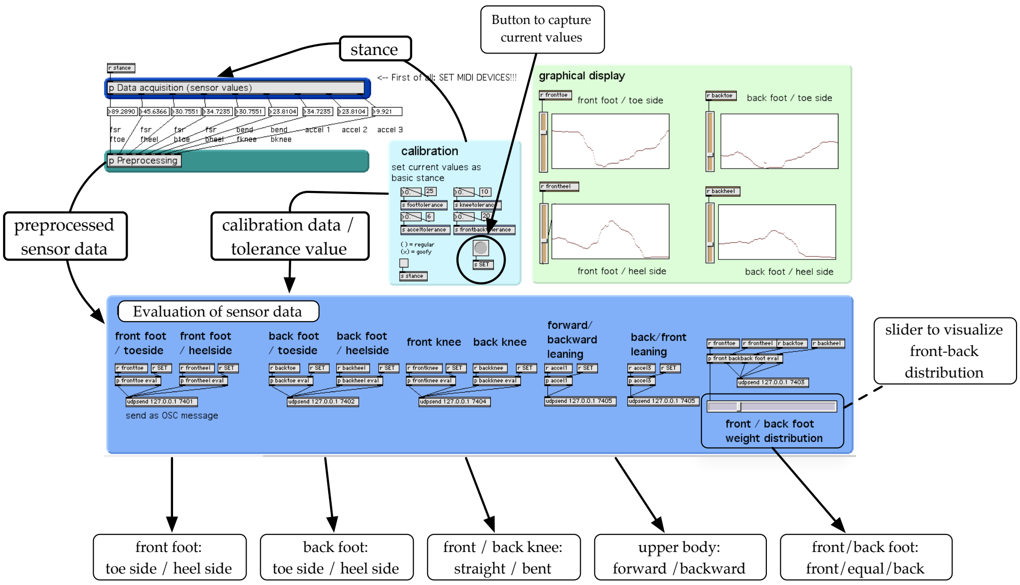


Figure B.1: Screenshot of `MyCubePatch.pat`

B.2 MyEventReceiver.pat

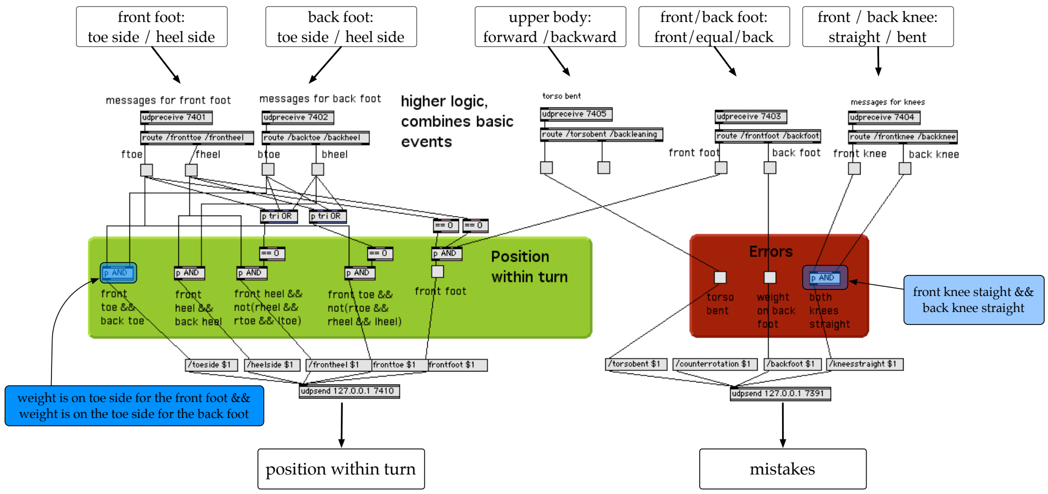


Figure B.2: Screenshot of `MyEventReceiver.pat`

B.3 MyFeedbackGenerator.pat

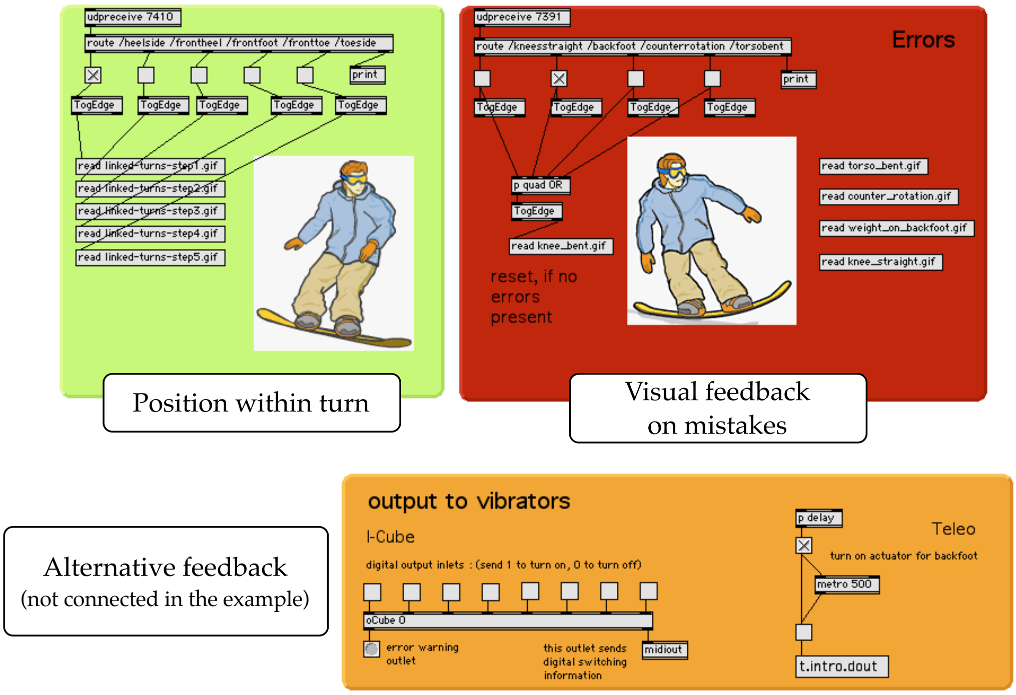


Figure B.3: Screenshot of `MyFeedbackGenerator.pat`

Appendix C

Smoothing Filters

C.1 Simple Moving Average (SMA)

In a series of data samples $x(t)$, the simple moving average $SMA(t)$ calculates the mean over the last N data samples:

$$SMA(t) = \frac{\sum_{i=0}^{N-1} x(t-i)}{N} = \frac{x(t) + x(t-1) + \cdots + x(t-N+1)}{N}$$

In a successive series of samples the calculation of the whole sum is not necessary in every step. To update the SMA, only the oldest value needs to be subtracted from the sum and the newest needs to be added:

$$SMA(t+1) = SMA(t) - \underbrace{\frac{x(t-N+1)}{N}}_{\text{oldest value}} + \underbrace{\frac{x(t+1)}{N}}_{\text{newest value}}$$

The length N of the sliding window is the smoothing factor for the SMA. A higher N results in better smoothing but in decreased responsiveness with respect to the original samples.

C.2 Exponential Moving Average (EMA)

In a series of data samples $x(t)$, the exponential moving average $EMA(t)$ is calculated as follows:

$$EMA(t) = \alpha \cdot EMA(t-1) + (1-\alpha) \cdot x(t)$$

The smoothing factor $\alpha \in [0, 1]$ determines the relationship between $EMA(t-1)$, i.e., the result from the previous calculation, and the current sample. Choosing a high value for α emphasizes the past data, whereas for a low value for α the current data sample is predominant for the calculation. At $\alpha = 0$, for example, there is no smoothing at all because no past samples are used for the calculation.
Compared to the SMA, the EMA is more responsive with respect to the original data values. The EMA places more importance to recent values, whereas the SMA treats all past values within the time window of length N the same in the mean calculation.[1]

[1] Details can be found, e.g., at:
http://lorien.ncl.ac.uk/ming/filter/filewma.htm.

Bibliography

Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 304–307, London, UK, 1999. Springer-Verlag. ISBN 3-540-66550-1.

U. Anliker, J.A. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E.B. Schenker, F. Catarsi, L. Coluccini, A. Belardinelli, D. Shklarski, M. Alon, E. Hirt, R. Schmid, and M. Vuskovic. Amon: a wearable multiparameter medical monitoring and alert system. *Information Technology in Biomedicine, IEEE Transactions on*, 8(4):415–427, Dec. 2004. ISSN 1089-7771. doi: 10.1109/TITB.2004.837888.

C.W. Bachmann, E.R.; Xiaoping Yun; Peterson. An investigation of the effects of magnetic variations on inertial/magnetic orientation sensors. *IEEE International Conference on Robotics and Automation*, 2:1115–1122, 2004. ISSN 1050-4729. doi: 10.1109/ROBOT.2004.1307974.

Rafael A. Ballagas, Sven G. Kratz, Jan Borchers, Eugen Yu, Steffen P. Walz, Claudia O. Fuhr, Ludger Hovestadt, and Martin Tann. Rexplorer: a mobile, pervasive spell-casting game for tourists. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 1929–1934, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-642-4. doi: http://doi.acm.org/10.1145/1240866.1240927.

S.J.M. Bamberg, A. Y. Benbasat, D. M. Scarborough, D. E. Krebs, and J. A. Paradiso. Gait analysis using a shoe-integrated wireless sensor system. *IEEE Transactions on Information Technology in Biomedicine :*

Accepted for future publication, Volume PP(Issue 99):1 – 1, 2007. URL `http://ieeexplore.ieee.org/iel5/4233/26793/101109TITB2007899493.pdf?tp=&isnumber=26793&arnumber=101109TITB2007899493&punumber=4233`.

Massimo Banzi, David Cuartielles, and Tom Igoe. Arduino: An open electronics prototyping platform, February 2007. URL `http://www.viktoria.se/altchi/submissions/submission_mellis_0.pdf`.

R. Barbieri, E. Farella, L. Benini, B. Ricco, and A. Acquaviva. A low-power motion capture system with integrated accelerometers. In *Consumer Communications and Networking Conference*, pages 418–423, Bologna, Italy, January 2004.

Michael Beigl, Tobias Zimmer, Albert Krohn, Christian Decker, and Philip Robinson. Smart-its — communication and sensing technology for ubicomp environments. Technical Report 2, TeCo, University Karlsruhe, 2003.

Davide Brunelli, Elisabetta Farella, Laura Rocchi, Marco Dozza, Lorenzo Chiari, and Luca Benini. Bio-feedback system for rehabilitation based on a wireless body area network. In *PERCOMW '06: Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops*, page 527, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2520-2. doi: http://dx.doi.org/10.1109/PERCOMW.2006.27.

Ed H. Chi, Jin Song, and Greg Corbin. "killer app" of wearable computing: wireless force sensing body protectors for martial arts. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 277–285, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. doi: http://doi.acm.org/10.1145/1029632.1029680.

Thomas Degen, Heinz Jaeckel, Michael Rufer, and Stefen Wyss. Speedy: A fall detector in a wrist watch. In *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*, page 184, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-2034-0.

Elisabetta Farella, Augusto Pieracci, Luca Benini, and Andrea Acquaviva. A wireless body area sensor network for posture detection. In *ISCC '06: Proceedings of the 11th IEEE Symposium on Computers and Communications*, pages 454–459, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2588-1. doi: http://dx.doi.org/10.1109/ISCC.2006.21.

Jacob Fraden. *Handbook of Modern Sensors: Physics, Designs, and Applications (Handbook of Modern Sensors)*. SpringerVerlag, 2003. ISBN 0387007504.

John Geen and David Krakauer. New imems angular-rate-sensing gyroscope. *Analog Dialogue*, 37(1):12–15, March 2003. URL `http://www.analog.com/library/analogDialogue/archives/37-03/gyro.html`.

Nicky Kern and Bernt Schiele. Multi-sensor activity context detection for wearable computing. In *European Symposium on Ambient Intelligence*, Eindhoven, The Netherlands, November 2003.

James F. Knight, Huw W. Bristow, Stamatina Anastopoulou, Chris Baber, Anthony Schwirtz, and Theodoros N. Arvanitis. Uses of accelerometer data collected from a wearable system. *Personal Ubiquitous Comput.*, 11(2):117–132, 2007. ISSN 1617-4909. doi: http://dx.doi.org/10.1007/s00779-006-0070-y.

K.S.C. Kuang, W.J. Cantwell, and P.J. Scully. An evaluation of a novel plastic optical fibre sensor for axial strain and bend measurements. *Measurement Science and Technology*, 13(10):1523–1534, 2002. URL `http://stacks.iop.org/0957-0233/13/1523`.

Kai Kunze, Michael Barry, Ernst A. Heinz, Paul Lukowicz, Dennis Majoe, and Jeurg Gutknecht. Towards recognizing tai chi - an initial experiment. In *Proceedings of the Third International Forum on Applied Wearable Computing*, Berlin, Germany, April 2006. URL `http://www.xsens.com/download/papers/Tai-Chi-IFAWC06.pdf`.

Doo Young Kwon and Markus Gross. Combining body sensors and visual sensors for motion training. In *ACE '05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 94–101, New York, NY, USA, 2005. ACM Press.

ISBN 1-59593-110-4. doi: http://doi.acm.org/10.1145/1178477.1178490.

Kristof Van Laerhoven and Ozan Cakmakci. What shall we teach our pants? In *ISWC '00: Proceedings of the 4th IEEE International Symposium on Wearable Computers*, page 77, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0795-6.

Jonathan Lester, Tanzeem Choudhury, Gaetano Borriello, Sunny Consolvo, James Landay, Kate Everitt, and Ian Smith. Sensing and modeling activities to support physical fitness. In *UbiComp '05 Workshop: W10 – Monitoring, Measuring, & Motivating*, September 2005.

Robert W. Lindeman, Yasuyuki Yanagida, Kenichi Hosaka, and Shinji Abe. The tactapack: A wireless sensor/actuator package for physical therapy applications. In *VR '06: Proceedings of the IEEE Virtual Reality Conference (VR 2006)*, page 106, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 1-4244-0224-7. doi: http://dx.doi.org/10.1109/VR.2006.130.

Florian Michahelles. *Innovative Application Development for Ubiquitous and Wearable Computing*. Phd (dr. sc. techn.) dissertation, ETH Zurich, Perceptual Computing & Computer Vision Group, Institute for Scientific Computing, ETH Zurich, Switzerland, December 2004.

Florian Michahelles, Erich Crameri, and Bernt Schiele. Design, implementation and testing of a wearable sensing system for professional downhill skiing in cooperation with trainers. In *2nd International Forum on Applied Wearable Computing (IFAWC 2005)*, March 2005.

Stacy J. Morris. *A Shoe-Integrated Sensor System for Wireless Gait Analysis and real-time Therapeutic Feedback*. PhD thesis, Massachusetts Institute of Technology, June 2004.

Joshua Musselman, Minu Gandhi, and Stacy Morris Bamberg. Development of an in-shoe sensor insole for motion analysis, physical therapy, and rehabilitation. Technical report, BioInstrumentation Lab, Dept. of Mechanical Engineering, University of Utah, 2007.

B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C.J. Bula, and P. Robert. Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. *IEEE Transactions on Biomedical Engineering*, 50(6):711– 723, June 2003.

Nuria Oliver and Fernando Flores-Mangas. Healthgear: A real-time wearable system for monitoring and analyzing physiological signals. *bsn*, 0:61–64, 2006. doi: http://doi.ieeecomputersociety.org/10.1109/BSN.2006.27.

Joseph A. Paradiso, Stacy J. Morris, Ari Y. Benbasat, and Erik Asmussen. Interactive therapy with instrumented footwear. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1341–1343, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-703-6. doi: http://doi.acm.org/10.1145/985921.986059.

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296, 1990.

Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *Proceedings of the Seventeenth Innovative Applications of Artificial Intelligence Conference*, Pittsburgh, Pennsylvania, July 2005.

Nagy Reil, Joe Sauter, Martin Schamböck, and Timm Stade. *Snowboard Lehrplan*. Blv Buchverlag, München, 2003.

Bill Riedel. A surface-micromachined, monolithic accelerometer. *Analog Dialogue*, 27(2):3–7, 1993.

Albrecht Schmidt and Kristof Van Laerhoven. How to build smart appliances. *IEEE Personal Communications*, pages 66 – 71, 2001. URL `citeseer.nj.nec.com/schmidt01how.html`.

Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 89–101, London, UK, 1999a. Springer-Verlag. ISBN 3-540-66550-1.

Albrecht Schmidt, Michael Beigl, and Hans-W. Gellersen. There is more to context than location. *Computers and Graphics*, 23(6):893–901, 1999b. URL `citeseer.ist.psu.edu/schmidt98there.html`.

Masami Takahata, Kensuke Shiraki, Yutaka Sakane, and Yoichi Takebayashi. Sound feedback for powerful karate training. In *NIME '04: Proceedings of the 2004 conference on New interfaces for musical expression*, pages 13–18, Singapore, Singapore, 2004. National University of Singapore.

Kristof van Laerhoven, Nicky Kern, Hans-Werner Gellersen, and Bernt Schiele. Towards a wearable intertial sensor network. In *IEE EuroWearable*, Birmingham, UK, September 2003.

Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović. Practical motion capture in everyday surroundings. *ACM Trans. Graph.*, 26(3):35, 2007. ISSN 0730-0301. doi: http://doi.acm.org/10.1145/1276377.1276421.

Index

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, December 20^{th}, 2007

Typeset December 20, 2007