RWTH AACHEN UNIVERSITY

How's My Driving?
Providing Driver Feedback to Improve Driving

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

by
Christoph Vobis

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Bernhard Rumpe

Registration date: Dec 01th, 2011
Submission date: May 30th, 2012

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, May 2012
Christoph Vobis

Contents

List of Figures

List of Tables

Abstract

Traffic safety has always been a relevant topic since driving and mobility have become an important aspect in daily life. Over the years, vehicles have improved constantly regarding traffic safety aspects. However, in most traffic accidents not the vehicle nor the environment is at fault. Studies have found that in most cases accidents are caused due to human error. With dangerous and aggressive driving maneuvers some drivers expose themselves and others to a risk. Regarding teenage drivers, car accidents are the leading cause of death.

Several approaches were developed over the last years with the aim to identify those drivers to help them to improve their driving behavior. Large commercial fleets equip their cars with special developed devices to identify dangerous drivers. In addition, insurance companies and parents are interested in system that help them to observe their customers/children to identify and negate potential risks.

This thesis will explorer how modern smartphones could aid drivers in the traffic environment. With constantly rising smartphone sells and the possibilities modern ones offer, smartphones represent a great development platform. The thesis will present a mobile application developed for an iPhone 4 that analyzes and rates driving based on internal sensor data. The application only relies on its own capabilities and provides real-time feedback as well as it allows the user to review his driving later on and to compare it to other users.

An user test was performed to validate that the system is able to identify risky driving actions. The experiment showed that the application is able to simulate a passenger's view regarding certain driving actions. Thus, the application is able to act as an objective personal driving observer.

Überblick

Die Verbesserung der Verkehrssicherheit und das Verhindern von Unfällen war schon immer ein wichtiges Thema. Über die Jahre hinweg wurde die Sicherheit von Fahrzeugen stets verbessert. Die meisten Verkehrsunfälle werden jedoch durch menschliches Fehlverhalten verursacht. Mit gefährlichen und aggressiven Fahrmanöver bringen Fahrer sich und andere in Gefahr. Gerade unter Jugendlichen sind Verkehrsunfälle für einen großen Prozentsatz der Todesfälle veranwortlich.

In den letzten Jahren wurden verschiedene Ansätze entwickelt, um riskante Fahrweisen und Fahrer zu identifizieren und entsprechende Schritte zur Verbessrung einzuleiten. Vor allem große Logistikunternehmen nutzen solchen System und rüsten ihre Fahrzeuge mit speziell entwickelten Geräten aus. Daneben sind auch Versicherungen und Eltern an solchen System interessiert, die ihnen helfen ihre Kunden bzw. Kinder zu beobachten und potenzielle Risiken zu negieren.

In dieser Arbeit wird untersucht wie moderne Smartphones Fahrer im Verkehrsgeschehen unterstützen können. Die stetig steigenden Absatzzahlen im Smartphonebereich und die Möglichkeiten, die moderner Smartphones bieten, machen diese Geräte als Entwicklungsplattform interessant. Es wird eine mobile Anwendung für das iPhone 4 präsentiert, die das Analysieren und Bewerten des Fahrstils, beruhend auf den interne Sensor-Daten des Smartphones, möglich macht. Der hier vorgestellte Ansatz stützt sich nur auf die eigenen Fähigkeiten des Smartphones und bietet Echtzeit-Feedback. Zusätzlich erlaubt es dem Benutzer seine Fahrt später auszuwerten und sich mit Anderen zu vergleichen.

Mit Hilfe eines abschließenden Usertests wird überprüft, ob das System in der Lage ist riskante Fahrweisen zu identifizieren. Der Test zeigt, dass der Applikation in der Lage ist Passagiere zu simulieren. Dies macht die Applikation zu einem objektiven persönliche Fahrassitenten, den jeder Besitzer eines iPhone bzw. Smartphones nutzen kann.

Acknowledgements

I'm glad to finish my graduation with this thesis and I have to thank many people who made this possible.

At first, I want to thank Jonathan Diehl, my supervisor, who always supported my topic and gave me the possibility to work with smartphones. Throughout the thesis he supported me with feedback, suggestions, constructive criticism and took his time for several meetings.

I want to thank Prof. Dr. Jan Borchers for giving me the possibility to write this thesis. I also want to thank him for his great lectures and introducing me to interesting topics regarding HCI.

I also want to thank Prof. Bernhard Rumpe for having agreed to be my second examiner.

Special thanks to all the staff of the i10. Throughout my master and during my thesis, the chair was a great resource for feedback and interesting research projects. In addition, the chair provided me with all the hardware needed.

I also want to thank those who participated in my user tests and all my friends for their ongoing feedback and support. In addition, I want to thank those who took the time to read my thesis.

Last but not least I want to thank my family. You all guided and believed in me through good and bad times. Without your ongoing support and help throughout my whole study and this thesis, none of this would have been possible. I want to specially thank my mother and my father for helping me through difficult and stressful times.

Thank you!

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Definitions of technical terms or short excursus are set off in colored boxes.

EXCURSUS:
Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition: *Excursus*

Source code and implementation symbols are written in typewriter-style text.

`myClass`

The whole thesis is written in American English.

Download links are set off in colored boxes.

File: myFile[a]

[a]http://media.informatik.rwth-aachen.de/~ACCOUNT/thesis/folder/file_number.file

In boxplot diagrams, the whiskers extend to the maximal and minimal values. The lower box boundary represents the 25% percentile and the upper boundary the 75% percentile. The line inside the box indicates the median.

Chapter 1

Introduction

"Imagination is more important than knowledge..."

—Albert Einstein

Traffic safety has always been a relevant topic since driving and mobility have become an important aspect in daily life. Car manufacturers have invented car systems like the anti-lock braking system (ABS) and the electronic brake force distribution (EBD) to increase safety on roads. These improvements helped to reduce the fatality rate in car accidents . While car manufacturer are focused on improving the car itself, accidents caused due to human error are not affected. According to Rumar [1985] 57% of all traffic accidents in the United States and Great Britain are based solely on the driver. Because of that, in addition to the car centered approach, the driver himself has to be supported and guided to lower the overall crash rates.

Most traffic accidents are caused by human error

There were more than 5.5 million reported vehicle crashes in the U.S. in 2009 with 1.52 million that resulted in an injury (DoT [2011]). According to the Statistische Bundesamt, 2.4 million accidents including 288.297 injuries were recorded by the police in Germany [1] . Although the fatality rate decreases, the number of car accidents detected by the

1.2 million people die on roads each year

[1]https://www.destatis.de/DE/Publikationen/WirtschaftStatistik/Verkehr/Unfallentwicklung2010.pdf

police are higher than in previous years. Globally over 1.2 million people die in vehicle accidents and 20 to 50 million suffer non-fatal injuries (WHO [2004])(Peden and Organization [2004]). Especially for teens at the age from 16 to 19 car crashes are the leading cause of death with 30% in the United States.

Current approaches require additional hardware and setup

In 3—"Related work" several approaches regarding systems to lower these numbers and increase traffic safety are presented. Most of them include several devices that are installed in the car and provide feedback and suggestions to the driver. 3.1—"iDriveSafe" for example is a system which uses the On Board Diagnostics II (OBD-II) interface to gain access to the vehicle's internal system and use this data to process suggestions to the user on a mobile phone. Other systems like 3.8—"DriveCam" use custom designed devices that are installed in the car. These approaches have in common that they require extra hardware and a proper setup. This makes these approaches expensive and decreases the usability due to the complex installation for end users.

Identify dangerous driving behavior to improve it

Commercial fleet operators are highly interested in such systems. Several approaches during the last years showed that solutions aimed on the driver's skills can reduce the amount of crashes and the related costs within the fleet. Another party interested in that kind of systems are insurance companies because every crash less safes money. In addition rating the driver's skills in an automatic and objective helping these companies to set the insurance premium amount to a value tailored on the specific driver. Several studies also show the interest of parents in such systems. Knowing the above presented facts on traffic accidents, this seems understandable. Beside the examples mentioned one could imagine several other use cases. They all have in common that they aim to identify dangerous/unsafe driving behaviors and drivers to take further steps in educating the drivers and to lower the overall crash risk.

Smartphones becoming omnipresent

In the last years smart phones became very popular and accessible to everyone. Solely in 2011, up to 487.7 million smartphones were shipped world wide. That is an increase of 62.7% compared to 2010 with tendency to rise [2]. In addition, the processing power of these devices goes up constantly. Modern smartphones include a lot of sensors like GPS, accelerometer, gyroscope and magnetometer. This increases the range of possible mobile applications and through the large number of smartphone sales, making them available to a large number of people.

Building a mobile application to analyze and rate driving

The aim of this thesis was to build a mobile application that analyzes and rates driving. The analysis and rating should help drivers to get objective feedback on their driving skills, helping them to improve their driving. The rating represents a passenger's impression about the driving style as well as his judgment regarding certain driving actions. A passenger's impression is a good indicator for a safe driving style because he can use his experience to judge good from bad actions and is able to identify unnecessary actions. In addition, passengers perceive driving actions different than the driver because they are not in control. By not knowing the next driving actions, a passenger perceives events like acceleration, hard braking and turning, stronger. Thus, even when the driver determine his driving as safe, a passenger could have a different opinion.

This thesis investigates the research question if it is possible to build a mobile application that is able to simulate a passenger's impression and judgment about certain driving actions and driving as a whole. It contributes to the field of current research by presenting a mobile application that can reliably simulates a passenger's perception regarding three kinds of driving actions. The application is able to give real-time feedback about four different kinds of events. Compared to other approaches, this application makes use of the smartphone integrated capabilities only and thus requires no additional hardware. This allows everyone with access to a smartphone to use it. Through the existence of for example the App Store it is possible to distribute mobile applications globally and making them available to all smartphone owners. It represents a promising starting point in increasing driving safety by showing in later evaluations that the application can successfully judge driving

[2] http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011

behavior from a passenger's point of view.

The thesis will start to explain theoretical research about driving and accidents, including possible contributing factors in 2—"Theory". It includes a long-term study which was executed with the aim to learn more about driving, accidents and characteristics of the involved drivers (2.1—"The 100-Car Naturalistic Driving Study"). After this, the problem of anonymity in traffic and a successful solution to that is presented in 2.2—"How's my Driving placard".

In chapter 3—"Related work", several approaches regarding driving safety improvements that make use of smartphones are presented. It also includes general research about modern smartphones and their effect on driving. Most approaches use additional hardware and software and make use of the smartphone to provide feedback.

Chapter 4—"Design Approaches & Ideas" will describe initial ideas and thoughts regarding the design and purpose of the application. It includes a brainstorming mind map and states which of the ideas can be accomplished by using an iPhone 4. Later it describes three use cases illustrated with storyboards.

Chapter 5—"Implementation" gives a closer look at the implementation. This includes explanation about the motion filtering (5.2—"Filter Sensor Data") as well as the rating calculation (5.3—"Rate Driving").

In 6—"User Interface", the interface of the application and the way the feedback is presented is explained in detail. The chapter describes various functions of the system and includes detailed information how the application is used and how the user is able to evaluate his trips.

7—"Evaluation" presents two user-tests. The first user test consists of an experiment to estimate patterns that differentiate normal driving actions and possible unsafe ones from each other (7.1—"Determining Boundaries"). In 7.2—"System Validation" the system created, using the findings from the first experiment, is evaluated. This evaluation focuses on the correlation between the application and other passengers in the car, both rating the same driver.

Finally, a summary of the thesis is given in 8—"Summary and Future Work". The chapter recaps the contribution to current research and summarizes what has been achieved in this thesis. It finishes by talking about future research areas and work regarding the presented approach.

Chapter 2

Theory

Investigating contributing factors that lead to accidents

Many investigates focus on vehicle accidents and it's contributing factors that benefit accidents. The aim of these analysis is to identify characteristics in the driver's personality and physical factors that lead to car accidents. Knowing these could help to build systems that identify risky driving behaviors to establish supporting actions to lower the chance of accidents.

At first, the chapter will present a study that researched driver related contributing factors in accidents. It is one of the first studies that observed drivers over the time of one year. It includes several questionnaires and sensor data that was evaluated afterwards, making it possible to identify reasons leading to crashes. After that, an approach relying on the feedback of other drivers is explained. Although the approach itself is rather simple, it could successfully lower crash rates and crash costs in commercial fleets. This success gives hints about how potential solutions could be designed.

2.1 The 100-Car Naturalistic Driving Study

Observing 100 drivers for a year to better understand crashes

Starting in August 2001, the National Highway Traffic Safety Administration (NHTSA) developed a study named "The 100-Car Naturalistic Driving Study" (Neale et al. [2002]). The aim of this study was to observe drivers over the period of one year and collect data to investigate contributing factors to crashes, near-crashes and incidents.

Study buildup

In February 2009, the NHTSA published a report comparing real-world behaviors of drivers with high versus low rates of crashes and near-crashes (Klauer et al. [2009]). This study used the data collected from 109 participants in the above mentioned long term study.

NHTSA used a special developed instrument-package for data collection

To collect all the data, the NHTSA used an instrument package developed by the Virginia Tech Transportation Institute. It includes several independent sensors combined in a network as well as a Pentium-based computer. It was developed over a 15-year period. It includes:

- A connection to the vehicle network
- An accelerometer to detect forces in the car
- Multiple cameras as well as sensors for side obstacle detection
- A GPS subsystem to store vehicle location information
- An incident push-box to allow participants to flag events for researchers

Predefined triggers mark important data

Due to the length of the study, recording non-stop would require a lot of hard discs or regular downloading of the data. To reduce the amount of data being stored and evaluated, triggers were defined which flag recorded data for further research. The data 60 seconds prior to 30 seconds after the event were stored.

In the recruitment process of finding participants, drivers that drive very safe or very unsafe were avoided. The subjects could choose whether to use their own car or to receive leased vehicles. Since 100 cars have to be equipped with the instrument package, the possible vehicle types were limited to six. This also guaranteed that system works with the vehicle network. 78 Subjects used their own cars whereas 22 used leased ones.

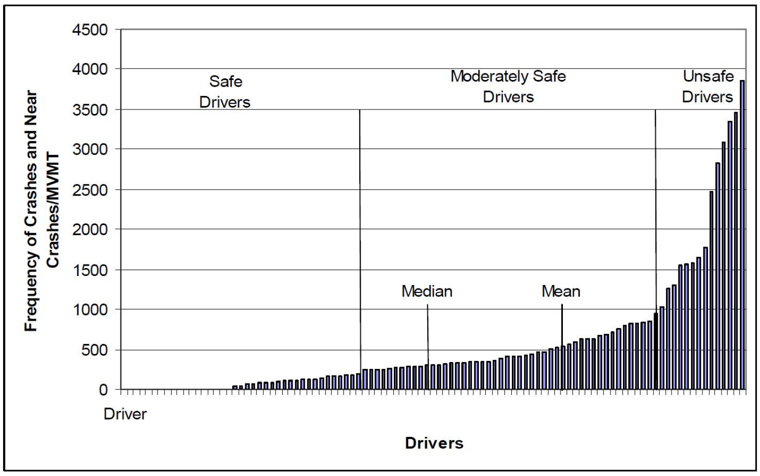


Figure 2.1: Driver classification used in the 100-Car Naturalistic Study

Classifying drivers into safe, moderate safe and unsafe

To research the different behaviors of safe versus unsafe drivers, the subjects had to be classified into categories. To do so, crashes/near-crashes per million vehicle miles traveled (MVMT) were compared for each participant. The mean of this crash rate per MVMT is used to separate safe from unsafe drivers. To get a higher separation between the two groups, some analysis used a three group model including a category named moderate safe drivers. Figure 2.1 shows the frequency of crashes and near crashes per MVMT and displays the limits used to classify each driver. In summary, this resulted in 72 safe drivers and 29 unsafe drivers, or using the second model in 39 safe, 47 moderate safe and 15 unsafe drivers.

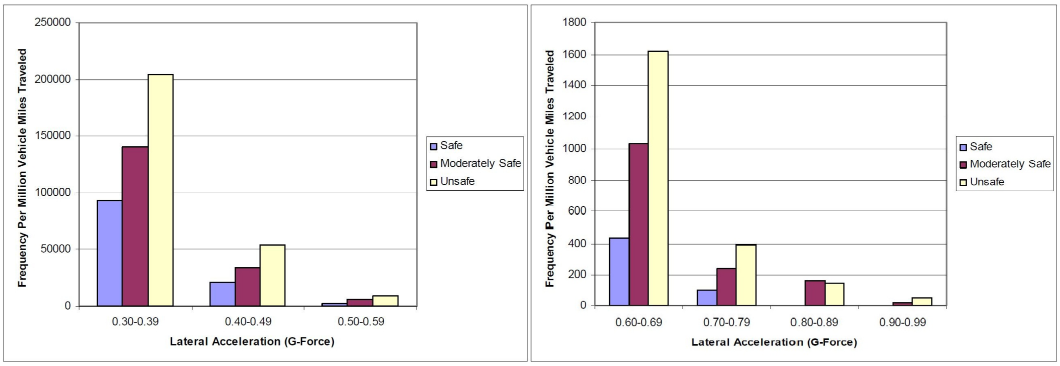


Figure 2.2: Lateral Acceleration frequency per MVMT for safe, moderate safe and unsafe drivers Klauer et al. [2009]

Study results

The research about significant differences between these groups ranges from demographics and psychological aspects, up to environmental conditions and potential patterns indicating unsafe driving. This also includes several questionnaires. Only results relevant to this thesis will be discussed. Other outcomes of this study can be seen in the original document.

The NEO Five-Factor Inventory's (Costa and McCrea [1992]) agreeableness scale measures altruistic and sympathetic tendencies compared to egocentric and competitive ones. The higher a driver scores on this scale, the more he is concerned about other drivers nearby. Low scores indicate that driving is seen more as a competition.

Unsafe drivers view driving as a competition

Comparing safe and unsafe drivers using the agreeableness scales shows that the safe driver group scores significantly higher than the unsafe driver group. This attitude can be used to build a system which raises competition in a way that participants challenge each other for the safest driving style. This attitude is also related to road rage (James and Nahl [2000]) and aggressive driving (Tasca [2000]).

Studying and searching for potential patterns between the groups, researches focused on lateral acceleration , longitudinal acceleration , longitudinal deceleration and swerving. Again, the frequencies of those events per MVMT were

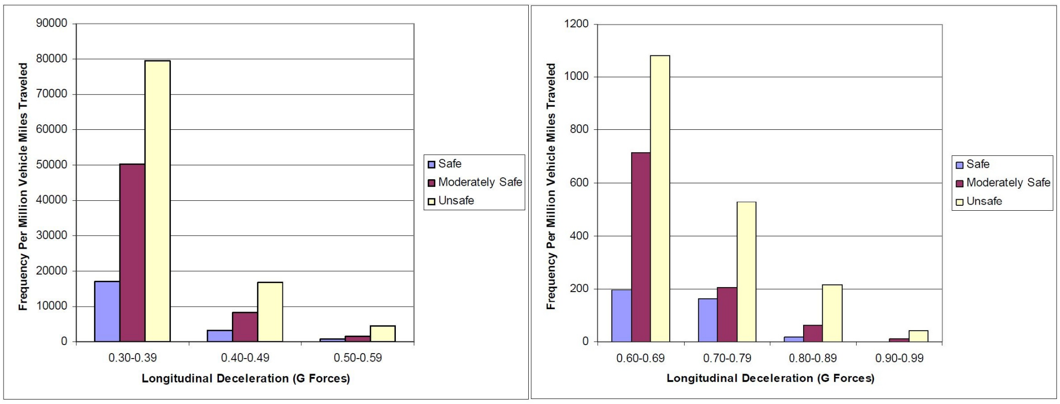


Figure 2.3: Longitudinal Deceleration frequency per MVMT for safe, moderate safe and unsafe drivers Klauer et al. [2009]

observed. Acceleration is measured in gravitational force (g-force). The g-force associated with an object is its acceleration relative to free-fall. On Earth, 1g corresponds to 9.81m/s^2. All measurements are grouped into seven levels ranging from 0.30g to 0.99g.

Unsafe drivers reached higher lateral acceleration levels more frequently

Lateral acceleration is measured in g-force and is a result of the driver turning the steering wheel. Figure 2.2 shows the frequency for each level per driver group. Post hoc Tukey tests show that safe drivers reached all the seven levels significantly less often than the moderate safe and unsafe drivers. Moderate safe drivers reached these levels significantly less often than the unsafe drivers, too. Engaging turns in shorter radius or at higher speed could be a contributing factor in the higher involvement in crashes and near-crashes of the unsafe drivers. An explanation for this could be that when engaging turns at a high speed, the time, the driver has to react to unexpected events, is reduced.

Longitudinal acceleration is the effect of the driver pressing the gas pedal or releasing the brake pedal. Compared to the level of acceleration ranging from 0.30-0.39g, stronger longitudinal accelerations were very rare and thus no significant differences could be found. Nonetheless, a post hoc Tukey test shows that unsafe drivers reached the first level of longitudinal acceleration significantly more often than the moderate safe and safe drivers.

Moderate safe and unsafe drivers reach high deceleration levels far more often than safe drivers

Longitudinal deceleration (figure 2.3) is the effect of the driver pressing the brakes. A post hoc Tukey test shows that the frequency of strong braking for safe drivers where lower in all seven levels compared to the moderate safe and unsafe drivers. The test also indicates a significant lower frequency for moderate safe drivers compared to unsafe drivers.

High acceleration as an indicator for unsafe driving

This study was one of the first which observed drivers over a long period of time and classified them according to their crashes. In summary, unsafe drivers tend to reach higher g-force levels more often than safe drivers. This observation can be used in the classification of drivers through the observation of lateral and longitudinal acceleration. Additionally viewing driving as a competition seems to be a contributing factor in the decisions leading to unsafe driving behavior.

2.2 How's my Driving placard

Figure 2.4: "How's My Driving" sticker on the bumper

HMD aims to lower the number of car accidents

The "How's My Driving" (HMD) program started in the 1980s and is based on feedback of other drivers with the aim to lower the number of accidents and improve safety of commercial fleets. It uses a reputation system like Ebay where other road users can provide positive and negative feedback.

Most drivers taking part in traffic show a safe driver behavior. The aim is to find those who don't possess such driving behavior. The police on the other hand tries the same but is only able to do sporadic analysis of easy detectable offenses such as speeding. While speeding is one of the leading factors in accidents, actions like improper braking or lane changes are rarely detected and punished by the police.

Finding the few bad drivers among the large number of good ones

These drivers raise insurance rates for all. Instead of classifying drivers in demographic groups, an insurance system that relies on the concrete driving behavior of the customer would be more appropriated. In addition, the chance of lowering the insurance rate could result in a higher motivation to improve the own driving in a safe way.

Bad drivers raise the insurance rate for all

If a fleet operator wants to participate in the HMD program, stickers (Figure 2.4) are attached to all vehicle bumpers in the fleet. These bumpers display a telephone number. If a nearby driver wants to inform about an incident or wants to give positive feedback, he is able to call this number. Call centers answer these calls and document the feedback. The given feedback is delivered to the fleet operator who then can take further suitable actions if necessary.

The program only requires stickers on the bumper displaying a telephone number

Investigating crash rates and crash costs of commercial fleets after one year of using the HMD program shows significant differences. The Hanover Insurance Co. measured a decrease of 22% in terms of crash rate and a decrease of 52% in crash costs regarding 11 different truck fleets (Knipling et al. [2003]).

HMD reduces crash rates and crash costs

Compared to the fact that the number of incident reports is relatively low, the effect of this program is huge. The pure presence of the stickers seems to remind the drivers of their accountability for their driving actions, resulting in an driving behavior that avoids annoying other drivers. In addition, this approach gives fleet managers the chance to identify bad drivers and initiate special training for them.

In the paper "How's My Driving? For Everyone (and Everything?)", Lior J. Strahilevitz suggests to bring this program to all automobiles (Strahilevitz [2006]). He argues that a huge issue in today's traffic are drivers who are feeling anonymous and unaccountable for their driving ac-

Expand the program to private vehicles and make it mandatory by law tions, causing thousands of accidents each year. Since the HMD program seems to decrease the crash count of commercial fleets, he suggests that it could have the same effect on private road users. Making HMD stickers mandatory by law could result in a less discriminating pricing politic of insurance companies and could make driving overall safer. Recently, the HMD program is also available for teenagers allowing parents to observe their children. On the other hand, this could lead to problems since there is no guarantee that the reporter is honest. For example conflicting parties could raise each others insurance rate by fake calls and making up a story about events that did not happen.

The presence of the sticker motivates to improve driving

The HMD program shows that even simple solutions, like a sticker on the bumper, can lower crash rates significantly. It also shows that just the presence of some kind of judgment and the reduction of anonymity motivates drivers to avoid annoying actions towards other drivers, resulting in a lower crash rate.

Chapter 3

Related work

"Do something. If it doesn't work, do something else. No idea is too crazy."

—Jim Hightower

By the increasing usage of smartphones, various sensors are introduced in everybody's daily life. Much research is going on in the field of mobile sensing and how to make use of those sensors. This chapter will concentrate on approaches that try to improve driving in terms of safety and sustainability by using smartphones. In addition, general research in this field is presented.

3.1 iDriveSafe

The iDriveSafe system is the outcome of an approach, to create a mobile knowledge-based system for on-board-diagnostics and car driving assistance (Ruta et al. [2010]).

Identifying dangers is iDriveSafe's main task

iDriveSafe assists the driver by identifying risks and providing solutions how to minimize or even eliminate those dangers. It is a prototypical application on the Apple iPhone. To identify risks in the current driving behavior and to give hints how to improve it, iDriveSafe needs several different types of data.

Figure 3.1: iDriveSafe UI Screenshot

OBD II, GPS and accelerometer data are used

Modern cars provide direct access to vehicle data like the actual RPM or additional components like ABS or ESP being active. This data can be accessed through the CAN-Bus using the *On Board Diagnostics II (OBD II) protocol*. To access this data via iPhone, a Kiwi Wifi PLX adapter is used. In addition, iDriveSafe uses GPS and accelerometer data as well as web-based data sources.

With the collected data the application is able to identify the road conditions by computing the standard derivation of the acceleration. Two driving styles are differentiated, imprudent and regular, by examining the standard derivation of the RPM through the OBD II interface. Speed, traffic conditions and wind speed are also considered. Through reverse Geo coding using the Google Maps API, the car's location is retrieved. Actual weather information can be requested by using The Weather Channels (TWC) XML data feed.

The system uses the Web Ontology Language (OWL-DL) to implement the modeling domain. An online matchmaking service (MaMaS) is used to identify dangers and to provide helpful hints to the driver.

Test scenarios showed that the system works as intended

The system was tested in three test cases with two settings each. Speed, driving style and safety equipment were varied in a way that the system identified no dangers in the first setting but displayed several warnings in the second setting. The system recognized all dangers as expected in all six situations.

iDriveSafe has shown that it is possible to give useful hints regarding the driving safety using a mobile device. The paper pointed out how web services for matchmaking and weather information can extend the installed sensors in the car and in the smartphone and should be considered when building the application. Although without the OBD II interface some important data will be missing, the matchmaking approach can be adopted in this work.

3.2 Artemisa

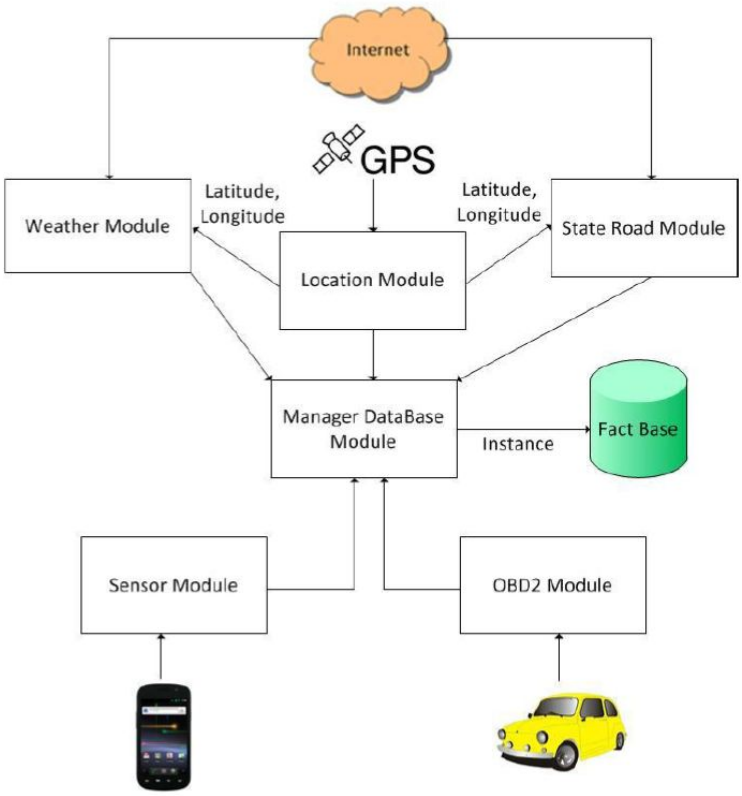


Figure 3.2: Artemisa Architecture Software

Artemisa is a mobile Eco-Driving assistant implemented on an android smartphone (V. Corcoba Magana [2011]). It

Artemisa provides Eco-Driving tips to the driver was developed by V. Corcoba Magaña and M. Muñoz Organero in May 2011. It tries to use data received through smartphone and car internal sensors to give hints how the driver can improve his driving regarding his ecological efficiency.

Save up to 25% fuel using Eco-Driving

The amount of gaseous pollution emitted by vehicles can be reduced by applying an ecological driving style by up to 25%. Due to the climate change and increasing energy cost these aspects have become more and more important in the last few years.

Artemisa makes use of many sensors as well as online services to collect data

Artemisa consists out of three main components: data acquisition system module, expert system module and user interface. The data acquisition system module collects the data needed for analyzing the driving behavior. Like iDriveSafe, Artemisa makes use of the OBD II interface to receive vehicle specific data. To access these a bluetooth adapter is installed. The features of the smartphone that are used by Artemisa are GPS, 3G, Bluetooth and the light, orientation and accelerometer sensors. The location is computed out of the GPS coordinates. With the exact location, two online services provide additional weather and traffic information.

Data is classified to compute Eco-Driving advices

Analyzing the collected data and giving ecological advice to the driver consists of four elements: facts base, preprocessing, knowledge base and a classifier. The fact base is a SQLite database that stores the data of the last ten minutes. The knowledge base contains information about ecological driving. It is used to identify meaningful driving advice and grows with each classification.

In contrast to iDriveSafe, Artemisa concentrated on changing the ecological driving in a positive way. Whereas it lacks testing with real users, it presents different algorithms to classify data gathered from different sensors and shows the effective use of different online services to collect weather information.

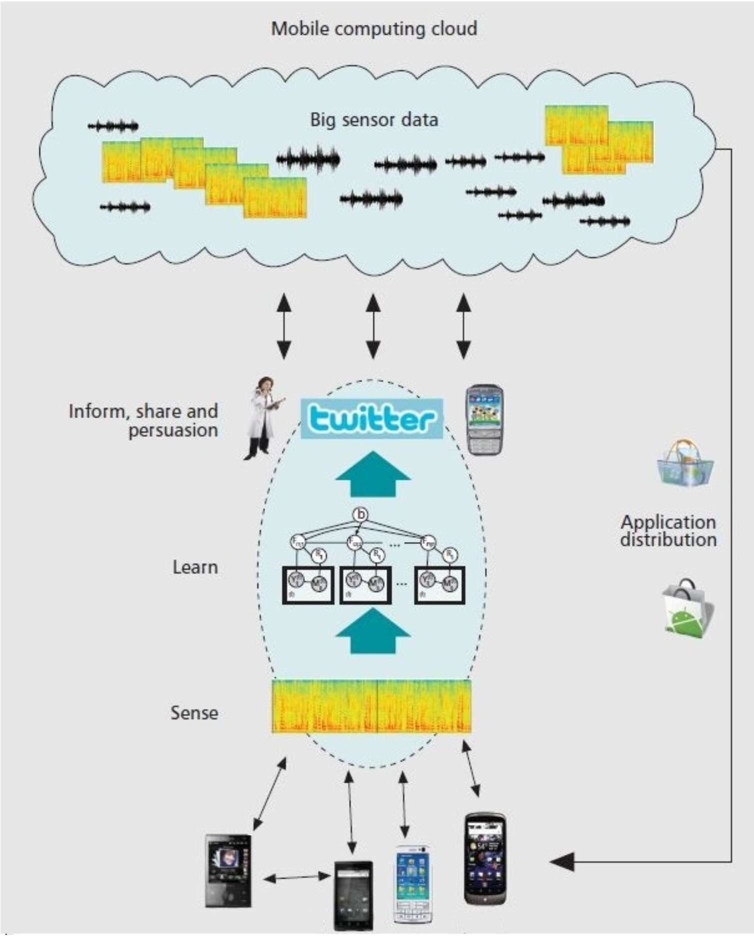


Figure 3.3: Mobile Phone Sensing Architecture

3.3 Mobile Phone Sensing

The paper "A Survey of Mobile Phone Sensing" (Lane et al. [2010]) describes the current state of the art regarding research in the field of mobile phone sensing. It talks about the technical possibilities with today's devices and problems which arise with today's widely accessible sensors.

Today's smartphones made mobile sensing ubiquitous

Distribution and availability of mobile sensing applications were an issue and made research in this field expensive and limited the user groups in scale. The paper states that these circumstances have changed since sensors become cheaper and were integrated in today's smartphones. Today's research also benefits from the devices' open APIs . Through application delivery services like the Apple App Store the distribution of applications became easy and made the collection of a huge amount of data possible.

Though the paper sees this as a great possibility for upcoming research, it also states that some important questions have to be answered to use these new possibilities to its fullest. This includes questions about privacy and how to validate experiments of large scales.

Two sensing paradigms: participatory and opportunistic sensing

Beside many examples of existing applications making use of mobile sensing, it differentiates between two kinds of sensing paradigms: Participatory sensing and opportunistic sensing. In participatory sensing, the user actively contributes to the sensor data gathering process whereas opportunistic sensing works automatically. Opportunistic sensing suffers from the fact, that the current context of the user is unknown. A disadvantage of participatory sensing is the higher burden and cost for the user.

The paper lists different sensing scales: personal sensing, group sensing and community sensing. Though some applications are only useful in a personal sensing scale, group sensing becomes very important due to the popularity of social networks.

Closing the sensing loop: User benefits from collected data

When talking about closing the sensing loop, the paper describes the importance of providing information computed out of the collected sensing data back to the user. The standard approach in terms of visualizing the data is the web. The gathered data can also be used to persuade the users in a way that it changes their behavior. Providing information from other users or even friends can result in a higher motivation. These effects can also be achieved by a game-like design of the system and a competition among groups.

User privacy as the most important aspect of mobile sensing

In the end, the paper states that respecting the privacy of the user is one of the most important aspects of a mobile sensing system. Raw data should not be provided in the web and highly personal data like the exact location with a time stamp should only be available to the user himself or within users in a trusted relationship because these information can have a major impact on the user's life.

This paper explains important aspects when using sensors. It describes the advantages and disadvantages of opportunistic and participatory sensing which should be considered when making use of sensors. The additional information about privacy and closing the sensing loop provides useful knowledge when defining the requirements of a new product in the sensing area.

3.4 Teen Driver Support System

The Teen Driver Support System (TDSS) (Janet Creaser [2011]) aims to increase safety of teenage drivers by providing feedback about insecure behaviors via mobile phones. In addition to the feedback directly presented to the driver, parents of the teenager are informed about incidents.

Combining car network and mobile phone to compute feedback

The TDSS uses the OBD II interface and multiple sensors in the car to evaluate the current situation and presents warnings if necessary. The system observes speeding in general as well as in curves. It includes lockout functions that prevent the car from being started if the driver has not fastened his seat belt and warnings if it is removed during driving. Tracking the amount of passengers in the car is also possible. Other features are preventing phone use while driving, detecting excessive maneuvers and watching if the driver stops at stop signs.

Incidents are reported to the parents

The system constantly observes if any of the above incidents occur. Incidents are reported to the parents, giving them the opportunity to control their teenager's driving behavior when driving alone. Through the near real-time feedback, parents can talk to their children about the incidents. In addition, parents have the possibility to view summaries of all trips on a web page.

The system was evaluated with 30 teenagers driving a 30 minutes predefined route after being introduces to the system. For each teenager at least one of his parents observed the driving using the given feedback of the system. The evaluation included several questionnaires about the usability of the system.

Figure 3.4: Visual feedback when exceeding the current speed limit in the Teen Driver Support System

Teenager and parents classify the system as useful and would recommend it

The majority of parents and teenagers rated the system as useful and marked it as reliable and accurate. Both mentioned that they believe that such a system would be a great help to improve driving in the first two years. Warnings about exceeding the speed limit were criticized because the warnings were presented in the moment the current speed limit is exceeded. This requires the drivers to constantly drive below the giving speed limit to avoid warnings. Teenagers also stated that they could see a privacy issue within the system as well as bringing conflicts into the family if every small incident is reported. Audible feedback were overall more distracting than visual. In summary, 95% parents would recommend such a system to their friends and were willing to pay a small fee of less than 20$ monthly.

The TDSS shows that a general interest in such a system from teenagers' and parents' perspective exists. It also shows that people would use such a system and think it could make them a safer driver. It also gives a bit insight into how feedback is perceived and points out concerns from both sides. Especially the fact that audible feedback is perceived as more distracting should be considered. In addition, warnings about exceeding the current speed limit should only be shown when the speed exceeds a certain threshold.

3.5 The Effect of an Eco-Driving System

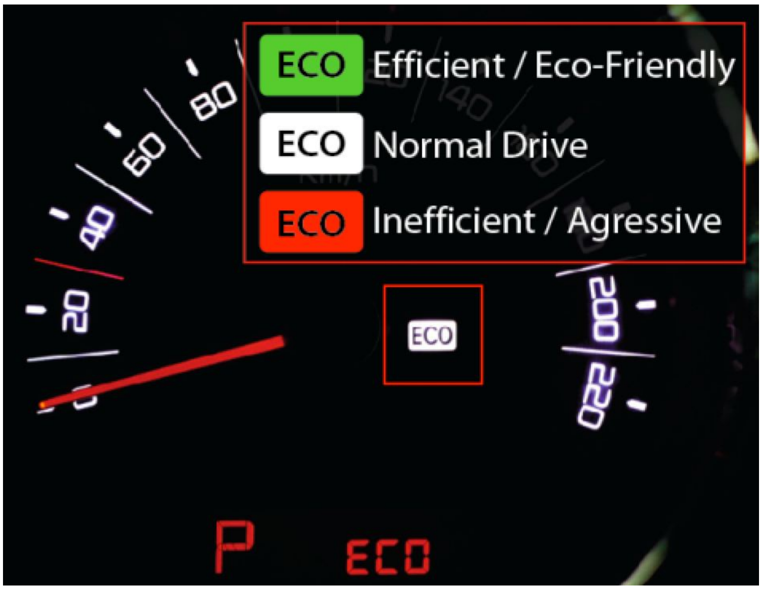


Figure 3.5: Eco-Driving System Indicator

This paper explorers the effects of an Eco-Driving System on the driver (Lee et al. [2010]) including an online survey with 60 participants and an user test with 14 participants. The Eco-Driving System was invented by the automobile manufacturer KIA and integrated in one of its car models named KIA Soul.

Eco-Indicator in the middle of the dashboard

The Eco-Driving System uses an indicator in the middle of the dashboard to inform the driver about his driving from an ecological point of view with the aim to save fuel. The indicator has three different colors where green is good, white is normal and red is bad regarding the fuel usage.

People thought they were saving fuel by using the system

60 people participated in the online survey that tried to find the benefits and the drawbacks of the Eco-Driving System using questions and a 5 point Likert scale. 78% of the participants stated that they thought they saved fuel. 64% also stated that saving gas and money is the main reason to use the system. 20% also wanted to improve their driving behavior to drive safer. The system was criticised for its poor feedback and missing concrete values about how much fuel was used/saved. The system's settings were also fixed so that for example road conditions and traffic did not influence the system's outcome.

Using the Eco-Driving System increases the workload of the driver

To see if using the Eco-Driving System helps the driver to save fuel, an user test with 14 participants was conducted. Participants drove a KIA Soul two times the same course, once with activating the Eco-Driving System and once without it. The fuel consumption was later on compared between those two driving sessions. In addition, the NASA Task Load Index (NASA-TLX) was conducted. The NASA-TLX is a subjective, multidimensional assessment tool that rates perceived workload. The test showed that there was no difference in fuel consumption using the system or not. Using the Eco-Driving System resulted in a significant higher mental demand, physical demand, effort and frustration. This leads to the conclusion that the users were aware of the system all the time and tried to maintain the green indicator. This ended up in an insecure and stressed driving behavior.

The user test and especially the online survey have shown that people are interested in improving their driving in terms of safety and sustainability. The importance of meaningful feedback has also been demonstrated by the user test. Displaying information to users while driving raises the awareness of such a system and affects the way of driving. These effects should be considered when designing the user interface and the way feedback is presented.

3.6 Predicting the Effects of In-Car Interfaces

Full-Manual	*Full-Voice*
Press **Power** Press **5, 5, 5, 4, 2, 8, 3** Press **Send**	Press **Power** Say **5, 5, 5, 4, 2, 8, 3** Listen for **5, 5, 5, 4, 2, 8, 3** Listen for "Connecting…"
Speed-Manual	*Speed-Voice*
Press **Power** Press **2** (speed number) Press **Send**	Press **Power** Say "**home"** Listen for "**home"** Listen for "Connecting…"

Figure 3.6: Dialing sequences for each scenario

Unlike desktop interfaces, in-car interface are not in focus all the time. The paper "Predicting the Effects of In-Car Interfaces on Driver Behavior using a Cognitive Architecture" (Salvucci [2001]) tries to estimate driver's distraction and effects on the driving performance when using a phone while driving.

For predictions, the Adaptive Control of Adaptive Control of Thought-Rational (ACT-R) cognitive architecture is used. The two tasks driving the car and using the phone are represented through the architecture models user model and driver model. The user model describes behavior whereas the driver model concentrates on the cognitive and motor processor. The ACT-R architecture allows to integrate these two models to simulate the situation and allows to make predictions about the driving performance. The investigated interface interaction is a dialing process. Four different ways of dialing are considered: Full-Manual, Full-Voice, Speed-Manual, Speed-Voice.

Executing the integrated model leads to predicted values for dialing time, derivation and velocity

In the integrated model, each of the previous mentioned models tries to fulfill its objective. The driver model drives the car and user model has the aim to finish one of the dialing tasks. In the simulation, the user model completes one small step of its execution sequence and hands over the control to the driver model which adjusts the car's position on the street. The driver model implements a simple driving task: driving on a single-lane straight highway at 60 miles/hour without traffic. Each dialing possibility is simulated 32 times in row with a 20 seconds brake of simple driving between each. The dependent values of this simulation are dialing time, lateral derivation and lateral velocity. In contrast to the other two values, the dialing is also considered when not driving.

Using a driving simulator in a user test to get empirical data

To compare the model predictions to empirical data, an experiment with 11 participants was conducted. A Nissan CBR fixed-base driving simulator was adjusted to collect the needed data and to provide the same driving conditions like the simulation. The participants then completed the driving and interface interaction task. Just like the simulation, the participants performed 32 dialing trials for each dialing method in 20 seconds intervals.

Comparing the integrated model and the empirical data shows a close fitting regarding the dialing time. Whereas the values of lateral derivation and velocity were overall to low, the ratio between the methods were predicted closely. Only the speed-manual predictions failed. In summary, the paper has shown that cognitive architecture models like the ACT-R are able to make useful predictions even for safety-critical tasks.

The paper compares four different interaction types when using a mobile phone and has shown that existing models can be used to calculate the effects on the driving behavior. The experiment provides real user data regarding driving safety and interaction time. When building a system that needs interaction while driving these results can be used to choose the safest type of interaction.

3.7 Green Multimedia

Figure 3.7: Mobile website screenshot showing CO_2 consumption

The paper "Green Multimedia: Informing People of their Carbon Footprint through Two Simple Sensors" (Doherty et al. [2010]) tries to provide CO_2 emissions of individuals to lower their energy consumption. Out of the three major sources of CO_2 named electricity, thermal and transport, they concentrated on two of them. In an user study with 22 participants, the paper investigated the electrical energy consumption and CO_2 emissions caused by driving. The paper studied whether displaying these information can cause a change in the participants behavior.

The major sources of CO_2 are electricity, thermal and transport

To collect data on the electricity usage, an EpiSensor ZEM-30 data logging unit was installed at each of the 22 participants homes. It is able to measure 11 different parameters whereas the watt hours are the relevant ones. The watt hours can be used to calculate the CO_2 emissions produced. The data is sent to a PC and stored for later research.

Gathering watt hours used from the participants' electric meter

To compute the produced CO_2 caused by driving, ac-

Using accelerometer data to compute fuel usage

celerometer data is used. Six of the 22 participants were handed a accelerometer installed on their key rings. The data was collected once a day by a researcher and stored. The first part when using the gathered accelerometer data was to find the data which can be related to driving using a driving classifier with optimized parameters was used. In the second part, a driving CO_2 estimator computed the CO_2 emissions.

Different visualization for different devices

Displaying the energy consumption to the users, played a key role in the process of behavioral change. The gathered data can be accessed in three different ways: a tablet application, a website and a mobile website. Whereas the website provides the richest user experience, the other data representations provide a reduced amount of information due to interaction possibilities, screen space and bandwidth. The data on the website and on the application were displayed using bar charts. Each bar was compared to the average consumption and its color changes accordingly. On the left side of the bar charts are numbers displayed that show the overall energy consumption. The 6 participants which were also using accelerometers, have a pie chart on the right that shows the proportion of driving and home CO_2 emissions. The mobile web page displays four bars showing the today's CO_2 consumption compared to yesterday's.

Collecting accelerometer data for one year to test and improve the driving CO_2 estimator

Before the experiment, one of the users was given an accelerometer for one year. The aim was to collect 20 weeks of data to improve the driving classifier and the driving CO_2 estimation to finally test it with the data collected in another 38 weeks. To validate the algorithm, the user documented his traveled kilometers, average speed and fuel consumption every week. In the end, the accuracy of the driving classifier could be improved and the driving precision was boosted up to 0.8203. In contrast investigating accuracy of the driving CO_2 estimator shows that the deviations of differences were very large and only 64.71% of the predictions had a degree of error within 1 standard deviation. This shows that calculating CO_2 emissions from accelerometer data is challenging.

Electricity consumption of 22 participants was tracked for one year. In addition, the accelerometer data was tracked for 6 weeks from 6 participants. When comparing the first two weeks and the last two weeks of the driving period, the 6 participants drove less which could be an effect of CO_2 visualization. Regarding the home electricity consumption an average saving of 3.88% could be measured whereas 6 users was also able to save up to 10%.

6 users were able to reduce their electricity consumption by up to 10%

The paper has shown that computing fuel usage only based on the accelerometer data is a complex and difficult task. Though the participants showed an overall lower consumption in the end of the experiment, the origin of this is unknown. A survey about the frequency of the website usage related to the CO_2 emission could be useful as well as statistics about the overall usage of the three different sources. To provide the gathered data via a website to the user additionally to a mobile website (or mobile application) is a good way to extend the user experience. Whereas the mobile website can be used to quickly view the consumption, the website allows more in depth view and interaction on the data.

3.8 DriveCam

DriveCam is a driver risk management company which is specialized on selling safety programs to commercial fleets, especially truck fleets. It uses a specially designed device which is mounted in the car. It includes two cameras to capture the inside of the car and to observe the area in front of it. The device constantly records audio and video data. Additional sensors in the device are able to identify certain events like hard braking, swerving or collisions. If one of these events occur, the data 10 seconds before and after is stored on the device. In addition, real-time feedback is provided in the cars rear mirror.

DriveCam offers safety programs for commercial fleets using specially design hardware

DriveCam uses a secure cellular connection to immediately transfer the data to the DriveCam's Data Center where the data is analyzed. This data can be used to identify risky drivers in the fleet which are then coached to improve their

Transmitted data is analyzed by experts

[1] http://www.drivecam.com/our-solutions/core-solution/safety-risk

Figure 3.8: DriveCam system feedback loop[1]

driving to limit the chance of an accident. Figure 3.8 shows each step in the improvement process of DriveCam.

Optional solutions provide a fuel management system which focuses on the driver. It includes feedback displayed in the rear mirror and a fuel score ranking which ranks drivers across the fleet. Another feature is fleet tracking which allows fleet managers to track all of their cars in real-time and getting their locations on a map using the GPS sensor included in the newest generation of the DriveCam device.

DriveCam achieved a collision related cost reduction of about 80%

DriveCam is a commercially successful solution that is currently installed in 170,000 vehicles in over 500 commercial fleets. According to DriveCam, the cost reduction for collisions were up to 80% and 12% fuel could be saved by using this system. The success and big interest of companies in combination with the success in numbers shows the possibilities of such systems. Especially computing a rating regarding fuel use makes it easy to compare the drivers among each other and is a great way to motivate people.

Chapter 4

Design Approaches & Ideas

"It's really hard to design products by focus groups. A lot of times, people don't know what they want until you show it to them."

—Steve Jobs

This chapter will describe the development process and the initial thoughts in detail. It starts with the initial design ideas and the intent behind the application including which aspects of the solutions presented in 3—"Related work" can be used.

4.1 Initial Design Ideas

Improve driving through feedback

As mentioned above, the aim of this thesis is to build a mobile application which detects driving mistakes from a passenger's point of view and to provide useful feedback to the driver. This feedback should help the user to improve his driving.

A mobile application should focus on one task but do this task very well (Gong and Tarasewich [2004]). This task

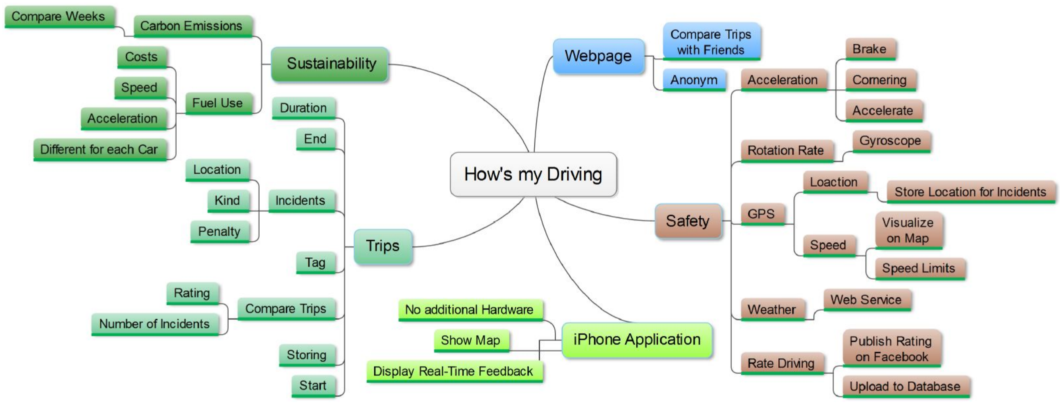


Figure 4.1: Initial Brain Storming for the application design

should solve a problem the user has, in this case to get objective feedback about his driving style. The most important aspects the application should fulfill are simplicity, intuitiveness and consistency, to obtain an application that is easy to use and its feedback easy to understand.

Application has no other dependencies than the device itself

The application should only use data that can be accessed through the features of the smartphone itself. It should be designed in a way that after installing the application on the device, no further requirements nor settings, in the user's car or the device itself have to be done.

Figure 4.1 shows the initial brain storming. Out of these ideas, a mind map was created. It consists of two major characteristics regarding driving. One aspect is safety and the other is sustainability. Judging sustainability in terms of fuel use and carbon emission is difficult when using only a smartphone. In 3.7—"Green Multimedia", researchers used an accelerometer to estimate the fuel usage. They stated that it is difficult to make accurate estimates about the CO_2 consumption. In addition, only one car was used in this approach. Trying to make general estimations for arbitrary cars would further raise the difficulty. To make good estimates about the CO_2 consumption would require access to the car network to access fuel level and other motor information. Especially the revolutions per minute would be of great interest since it is the best indicator for the current fuel usage. Because this approach excludes additional hardware, analyzing sustainability will not be part of the application designed in this thesis. Instead, it will focus on the safety aspect of driving.

Trips are the main component of the application

The application will record and store trips. A trip consists of several attributes. The most important attribute is a list of incidents that indicate points of interest regarding driving. These will be used to compute feedback such as an overall rating for the trip.

The feedback includes a unified trip rating

The feedback given to the user should make full use of the capabilities of the device. The application should be able to give real-time feedback as well as to allow the user to review his trip later on. The real-time feedback is the most important aspect. Informing the user directly after an unsuited driving action is important for the learning process. The driver knows exactly what he did and can remember his actions and circumstances that led to the driving action. Only reporting incidents after the trip would indeed inform the user about mistakes but due to the high temporal distance, it is unlikely the he is able to remember his actions to avoid it in the future. This feedback includes visual and audible feedback. In addition, a trip should result in a rating. The value of the rating is a number that is based on the measured incidents. This rating should allow the user to compare all his trips with each other and should make a comparison between different drivers possible. To achieve this, a website will be implemented that anonymously presents the user scores.

Minimize the time to start recording

Since the application has to be started before the user starts driving, it is important that the time the user needs to enable the recording is as low as possible. If it takes too much time for the user to setup the application before he can start driving, he could stop using the application. The user drives to reach a desired destination as fast as possible and does not want to spent additional time on setting up the application every time.

Distraction as a leading cause of crashes

Since the application aims to raise safety, the distracting effects of the application should be as low as possible. Operating wireless devices while driving is a major factor regarding distraction. Especially operating the device in form of dialing/typing and talking/listening leads to accidents. According to the NHTSA, 17% of all crashes in the

U.S. have at least one distracted driver involved (NHTSA [2010]). Texting while driving raises the crash risk by 23 times in contrast to non distracted driving (Government [2011]). Knowing this, the application should not require any input of the user while he is still driving.

In summary, the application should

- be easy to use and fast to start,
- not require additional hard- or software and only use the, features of the device itself,
- rate certain driving aspects from a passenger's point of view,
- give real-time feedback and a possibility to review trips later,
- assign a comparable rating to all trips,
- allow users to compare themselves to friends and other users of the application
- not require any user input while driving.

The iPhone 4 , operating on iOS 5.0, was chosen as development platform. The next section will describe the iPhone 4 in detail and which possibilities it provides including how these can be used to build the application.

4.2 The iPhone as Sensor & Feedback Platform

Modern smartphones contain several sensors

Modern smartphones are equipped with a lot of sensors and have high processing power. Since Apple presented the first iPhone in 2007, the device has constantly improved in design and equipment. The iPhone 4 provides developers with a lot of sensors and possibilities to design applications. Some of these capabilities are presented in Figure 4.2.

Acceleration as an indicator for unsafe driving

Most smartphones have an accelerometer installed. It is often used to detect orientation changes of the device. The iPhone is equipped with a 3-axis accelerometer that measures acceleration in relation to free-fall in gravitational force (g-force). Some of the approaches presented in the related work chapter use an accelerometer installed in the car

Figure 4.2: The iPhone 4 capabilities

to detect unsafe driving behaviors. In this application, the accelerometer will be used to identify possible incidents, too.

Detect turn taking via gyroscope

A new feature in the iPhone 4 is a 3-axis gyroscope. This allows to measure the position of the device in space. Combined with the accelerometer it is possible to calculate the rate of rotation about all 3 axis. This feature is interesting since it allows to detect turn-takings of the car. Figure 4.3 shows location and orientation of the axis regarding accelerometer and gyroscope. The rate of rotation is measured according to the right-hand rule.

Retrieve the current location through GPS

The integrated Global Positioning System (GPS) makes it possible to get the position of the device and thus of the car. This allows the use of maps and gives the possibility to inform the user about points of interests regarding his driving. The position can also be used to compute the distance traveled and to retrieve additional location based information like weather conditions and speed limits. In addition, the GPS sensor integrated in the iPhone is able to compute the current speed of the vehicle.

The iPhone allows a continuous Internet connection. This is important to retrieve map information and to access web

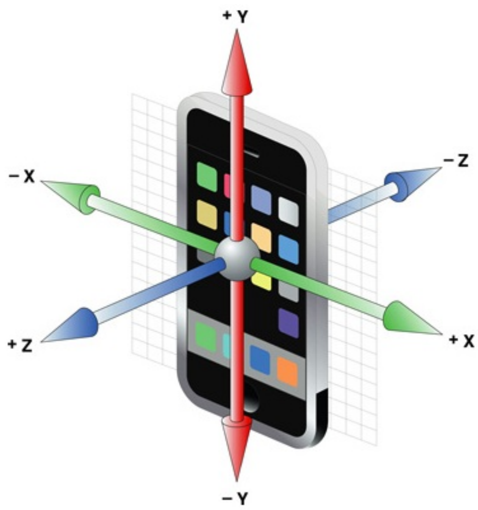


Figure 4.3: The 3 axis of the iPhone

based resources and web services. This is mandatory to get information about speed limits and weather. It also allows to upload the information about the trip immediately after it is finished.

Trip recording as a background process

Since iOS 4.0, it is possible to run applications in the background and to continue processing. This multitasking feature can be used when the user is not interested in real-time visual feedback which can be the result of an unfavorable device position. If the driver is not able to see the display, he can lock the iPhone and save battery power. It also allows the user to run other applications, such as navigation applications, while having his trip recorded.

4.3 Storyboards

Several storyboards were created during the development and planing process. This section will present three storyboards in detail. The storyboards represent the three main purposes of the application: competition, education and controlling. Whereas the storyboard regarding competition is included in this section, the other two can be found in C—“Storyboards”.

Figure 4.4: A storyboard describing a situation where the application is used to raise competition between drivers

Storyboard 1: Competition

Use the application's rating to enable competition

Moritz, fleet manager of a logistics company, has a problem. Recently his drivers caused a lot of accidents. He searches for an easy and cheap way to motivate his employees to improve their driving and to lower the accident rate. He knows that most of his employees are smartphone owners. Therefore, he came up with the idea to use the How's My Driving application to initiate a competition between all drivers. He informs all employees how they can participate and that the driver with the highest score each month will be rewarded.

Two of his employees named Sebastian and Max like the idea and are motivated to win the competition. After two weeks is Max slightly ahead of Sebastian regarding their average scores. Since there are still two weeks to go, Sebastian believes to be able to beat Max at the end of the month. In the end, Sebastian could constantly improve his driving and really beat Max. Max aims to win the next competition. In addition, to the fact that the competition motivated most employees to improve their driving, Moritz was able to identify very bad drivers within his company and initiates specially designed driving seminars for them.

Storyboard 2: Controlling

Recently, the 16 year old teenager Alex got his driving license (figure C.1). He is not yet owner of an own car and has to ask his father when he needs a car. Max father reads in the news that teenage drivers are the most prominent group in traffic accidents. In addition, the fact that traffic accidents are the major cause of death for teenage drivers makes him worry about his son's safety. He knows that it is impossible for him to always drive with his son. Therefor, he searches for a way to observe his son's driving style without the need to be in the car in person.

Use the application to observe novice drivers

He discovers the How's My Driving application, advises his son to install it on his smartphone and to record his trips with it. By viewing the application's reports on a website, the father is able to evaluate his son's driving. The son knows that his father is informed about any incidents during his trips and tries to avoid them. The father on the other hand can take further actions if he does not, giving him the feeling of still being in control when his son drives alone using his car.

Storyboard 3: Education

The storyboard (figure C.2) describes a situation where the young student Peter explores the App Store searching for interesting applications regarding driving. Recently, several passengers criticized Peter's driving style. He searches for a way to get his driving rated from an objective point of view with the aim to improve his way of driving.

While browsing, he discovers an application named How's My Driving that allows him to get objective feedback about his driving. Peter plans to test it the next day. He places his smartphone inside the car and starts driving to his desired location. The application reports several incidents during his trip. After the trip, Peter receives a score of 50.5 points. Knowing which driving actions were reported, he plans to improve his driving and aims for higher scores. After one week, he finally manages to reach the maximum score of 100 and proudly reports his achievement to his friends on facebook.

Chapter 5

Implementation

"Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots. So far, the Universe is winning."

—Rick Cook

In this chapter, the overall application structure will be described. It will present how the ideas from the previous chapter find their way into the mobile application. Figure 5.1 shows a rough overview of the processes and communication involved when recording a trip. The classes involved in the detection and rating process, and classes important for the overall structure of the application will be described in detail.

5.1 Architecture

Trip and TripAnnotations as the core components

One of the main components is the `Trip` class. It represents a trip recorded by the user. It consists of several attributes like start and end time, a rating, and the distance traveled. All attributes are shown in figure 5.2. In addition, a `Trip` has two lists containing objects defined by the classes `TripAnnotation` and `RoutePoints`.

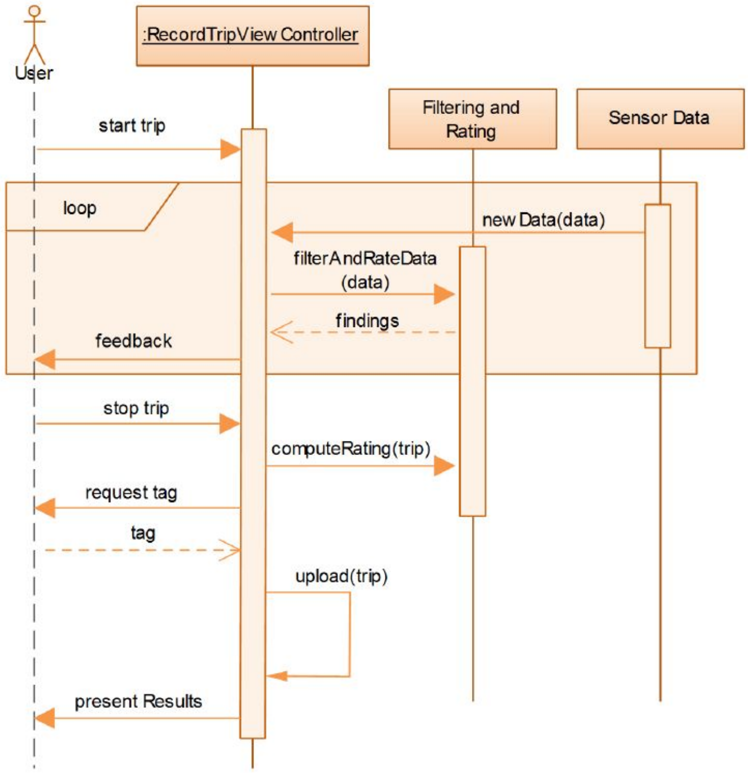


Figure 5.1: Sequence diagram visualizing the overall process of recording a trip

The `RoutePoints` class is used to store the route of the `Trip`. It stores longitude and latitude coordinates of geographical positions where the driver has been. With these points it is possible to retrace the complete trip. A `TripAnnotation` represents a point of interest in the trip. Like `RoutePoints` it contains latitude and longitude coordinates to define its location. In addition, it has a type and a time. A special form of `TripAnnotation` is an `Incident`. It is a subclass of `TripAnnotation` and adds a penalty and an acceleration attribute to the class. An `Incident` represents a possible driving error detected by the system.

Using CoreData for persistence

For persistence and storing `Trips`, the application makes use of the CoreData layer. To use it, each object that needs to be stored permanently needs to be a subclass of `NSManagedObject`. To prevent persistence errors and wrong initialization of objects, the factory pat-

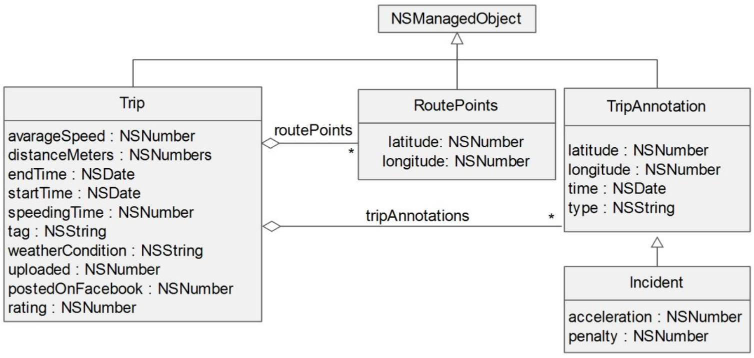


Figure 5.2: Class diagram showing the main model classes of the application that are used for storing the trips recorded by the user

tern (Gamma et al. [1994]) is used. All objects of type `NSManagedObject` are instantiated by this factory named `PersistenceController`. Through this singleton, a synchronized instantiation and deletion of objects is achieved.

Receiving information through CoreMotion and CoreLocation

Creating and recording a `Trip` is the main task of the `RecordTripViewController`. It make use of the CoreLocation and CoreMotion frameworks to get location and device motion updates. The `CLLocationManager` desired accuracy is set to `kCLLocationAccuracyBestForNavigation`. This gives the highest accuracy and includes course and speed information in the `CLLocation` object sent to the delegate.

Receiving motion updates at 25 Hz

For receiving motion updates, the `startDeviceMotionUpdates` possibility is used. Beside the pure device acceleration, the `CMDeviceMotion` object contains additional values such as rotation rate, user acceleration and gravity, which will be important for later computations. The update interval is set to 25 Hz. This interval represents a trade-off between high accuracy and high processor and battery usage. Whereas 100 Hz are theoretically possible, such a high frequency is not needed. Since several computations are done using this motion data, the applications performance could suffer. In addition, the amount of data would increase later evaluation's complexity. Receiving 25 updates per second seems to be a reasonable frequency to analyze driving. The motion updates are processed in an own NSOperationQueue. This queue allows other computations in the main queue to continue even under high processing overhead and guarantees that no motion update is lost.

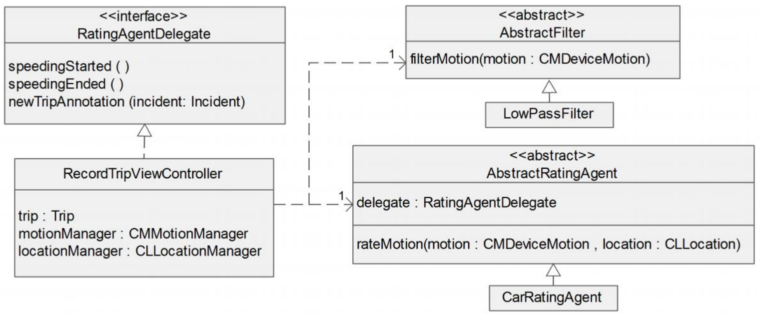


Figure 5.3: Class diagram showing the RecordTrip-ViewController and its two most important classes for trip recording and incident detection: AbstractFilter and AbstractRatingAgent

Filter and analyze data to find Incidents

The process of detecting an Incident is based on two tasks: filtering the incoming motion data and analyzing it. This is done by subclasses of the AbstractFilter and the AbsractRatingAgent displayed in figure 5.3. After setting up motionManager and locationManager, and creating a LowPassFilter and CarRatingAgent object, the received information can be used to detect driving mistakes. This architecture makes it possible to exchange the filter and the rating agent during runtime. In addition, this makes it easy to add new filters and rating agents to the system later on. This approach will focus on the use in cars. Other vehicle types like motorbikes or buses could possibly require other filtering and rating functions. Further research could then enhance the application by adding new filters and rating functions without much change.

The communication between those objects is illustrated in the sequence diagram 5.4. The RecordTrip-ViewController receives about 25 motion updates and one location update per second on average. Each time the motionManager sends a new motion object to the RecordTripViewController, it is forwarded to the filter. The motion filtering includes two steps. First,

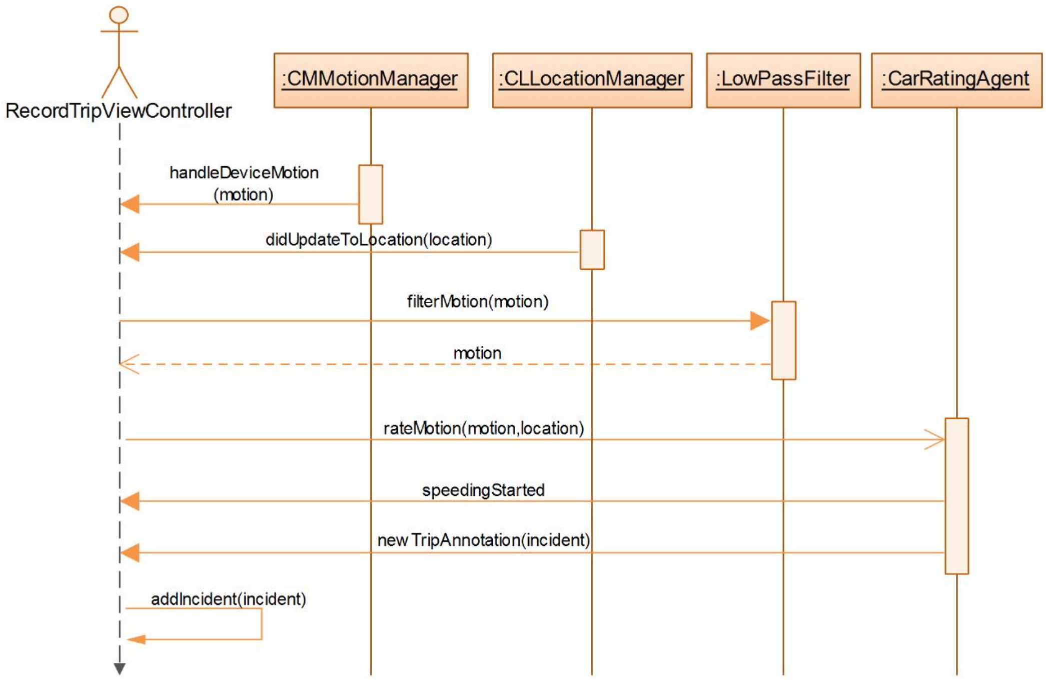


Figure 5.4: Sequence diagram showing the communication between the `RecordTripViewController`, the `LowPassFilter` and the `CarRatingAgent`

the motion measured by accelerometer and gyroscope is modified according to the device position. This is important for later operations on the motion data and provides a consistent format that makes it easier for other classes, like the `ratingAgent`, to interpret this data. Secondly, a low-pass filter is applied to the motion data. This process is described in detail in 5.2—"Filter Sensor Data".

RatingAgent analyzes filtered data to find Incidents

After the motion data has been filtered, it is sent to the `ratingAgent` . The `ratingAgent` is responsible for the generation of `TripAnnotations` that indicate a point of interest and to ensure compliance with the speed limit. The `ratingAgent` uses the filtered motion data and the current location object for this process. Since the location updates are not as frequently received as the motion updates, the `RecordTripViewController` always stores the last received location. This location is used by the `ratingAgent` to get the current speed of the car and to setup latitude and longitude of the generated `TripAnnotation`. The `RecordTripViewController` implements the `RatingAgentProtocol`. If an `Incident` is detected, a `TripAnnotation` that represents this `Incident` is generated and the `newTripAnnotation()` method is called transmitting the new `TripAnnotation` to the `RecordTripViewController`. To start or end speeding results in a call of `speedingStarted` or `speedingEnded`. The `RecordTripViewController` then adds the new `TripAnnotation` to the current `Trip` and computes feedback for the user. The process of detecting driving mistakes is described in 5.3—"Rate Driving" in detail. The `ratingAgent` also provides a function to compute the `Trip's` final rating.

Retrieving weather information via web services

In addition to the internal resources of the iPhone, the `RecordTripViewController` uses external resources in form of web services. One information already thought of in the initial brainstorming is the current temperature and the weather condition.

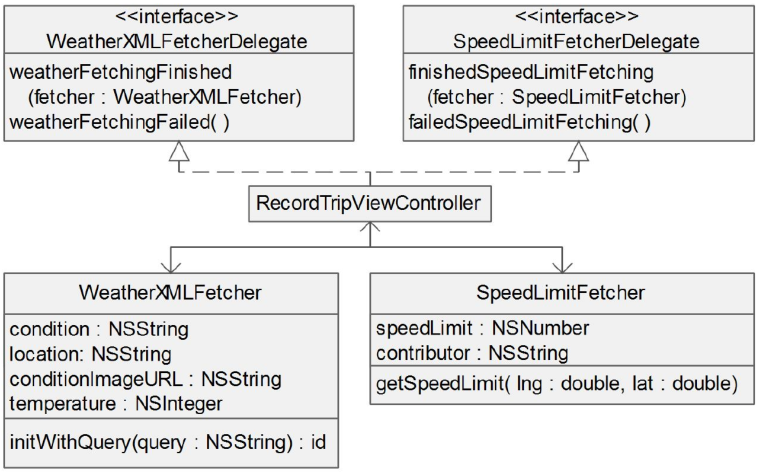


Figure 5.5: Class diagram showing two additional classes used by the RecordTripViewController to fetch additional resources through web services

Fetching the weather condition and the current temperature is the task of the `WeatherXMLFetcher` shown in class diagram 5.5. The web service used for this is provided by Google and can be accessed through an URL request. The `location` needs to be a string representing the current location. Since the current `location` object in the RecordTripViewController contains only latitude and longitude coordinates of the current position, the application needs to apply reverse geocoding to get the location in form of a string. This function is provided by the CoreLocation framework. After the location name has been determined, a new XMLWeatherFetcher object is initialized by using this name. The Google API is accessed through a NSURLRequest. After fetching weather information, the RecordTripViewController is informed about success or failure of the weather fetching process and can retrieve weather information from the XMLWeatherFetcher on success.

Fetching speed limit by using open source databases

Speeding is one of the most contributing factors in car accidents. To determine whether the current speed of the car exceeds the road's speed limit, the application needs to obtain the speed limit according to the current position. There exist a lot of commercial databases mostly run by navigation software producers like TomTom [1] or NAVTEQ [2]. The access to those databases is associated with a fee and therefor not interesting for this prototype. Beside those commercial databases, two free accessible and open source projects exist.

OpenStreetMap database storing map information in an open source database

The first is a project founded in 2004 and is called OpenStreetMap [3]. The objective of the project is to build an open source world map. Since it is an open source project, every one can contribute and add data. The project itself does not include a web service to access the database. One possibility is to setup an own server and to create an API to request the desired information out of the database or to use a web service that already makes use of it. One service that makes use of the OpenStreetMap database is MapQuest [4] which is focused on routing services. MapQuest offers a web service to request routing information to get from one location to another. The locations are both specified in form of geographical coordinates. Theses routing information include speed limits of the current street. A NSURLRequest requests the speed limit information of the current location inside the SpeedLimitFetcher shown in figure

[1] http://www.tomtom.com
[2] http://www.navteq.com
[3] http://www.openstreetmap.de/
[4] http://www.mapquest.com/

5.5. The MapQuest API returns a JSON object containing the speed limit if available. The `SpeedLimitFetcher` informs the `RecordTripViewController` through the methods defines in the `SpeedLimitFetcherDelegate` protocol about failure or success.

Wikispeedia gathers speed limit signs all over the world

Another speed limit database is Wikispeedia [5] . It aims to gather all speed limit signs around the world. Like OpenStreetMaps, everybody can contribute and add speed limits. A restriction when using this database is, that the application needs to display the contributor's name who added the used information. The speed limit database can be accessed through a web service. The web service requires to define the location in terms of longitude and latitude coordinates, and the direction the car is facing to estimate the correct speed limit sign. The direction can be read out of the `CLLocation` object's attribute `course`.

Both services are available in the system. Since these two databases are incomplete and speed limits can be missing or wrong, the user will be able to choose which of those two he would like to use and whether he wants to use this feature at all. Choosing between those two databases allows the user to find the service which provides the best results for his area. This selection can be done through the settings application.

At the end of a trip, the user is asked to enter a name describing the trip. After the user selected a name, the recording process is finished and the final trip rating is computed. An overall summary of the trip is presented to the user, allowing him to view his results and `Incidents` detected by the system.

Uploading anonymous trip information

After a trip, the results are uploaded to make them available through the web page. To achieve this, the most relevant information of the trip are mapped into a JSON object and uploaded through a PHP interface into a mySQL database. These information include the trip rating, duration, start time, weather condition and the `Incidents` detected including their acceleration and penalty. To prevent unauthorized access and uploading of fake trips, a MD5

[5] http://www.wikispeedia.org/

hash of the trip data is included in the JSON object. No location or device related information is uploaded to the database to ensure anonymity. If a trip upload was successful, the `uploaded` attribute is set to `YES`. Otherwise it is set to `NO`. Each time the application starts, if there exist trips that have not been uploaded yet to repeat the uploading process.

This thesis will continue to explain the previously mentioned filtering and detection processes in detail. Section 5.2—"Filter Sensor Data" explains how the motion data is filtered. After this, section 5.3—"Rate Driving" describes in detail what causes a driving action to be classified as `Incident` and how the final rating is calculated.

5.2 Filter Sensor Data

Aligning iPhone and car axis

One challenge in building a system running on a mobile phone compared to other approaches is the device positioning. Whereas other approaches uses fixed installed sensors in the car, this approach will only use the sensors of the iPhone. The position of the iPhone and so the orientation of the sensor's axis are very important for the detection process. The objective is to build the application in a way that it works independent of the device position and without extra user settings. As mentioned, the iPhone uses a 3-axis accelerometer and gyroscope displayed in figure 4.3. Before the sensor data can be used, the iPhone's accelerometer axis needs to be aligned with those of the car. In the end, the positive y-axis should be directed in driving direction and the x-axis should point to the side of the car. The axes locations regarding the car are displayed in figure 5.6. This position will be called the reference frame.

The difference in orientation regarding the reference frame in 3D space can be described by roll, pitch and yaw. Roll is the rotation around the y-axis, pitch the rotation around the x-axis and yaw the rotation around the z-axis.

Imagine two different device positions. In the first position the iPhone is placed in the car so that the positive y-axis

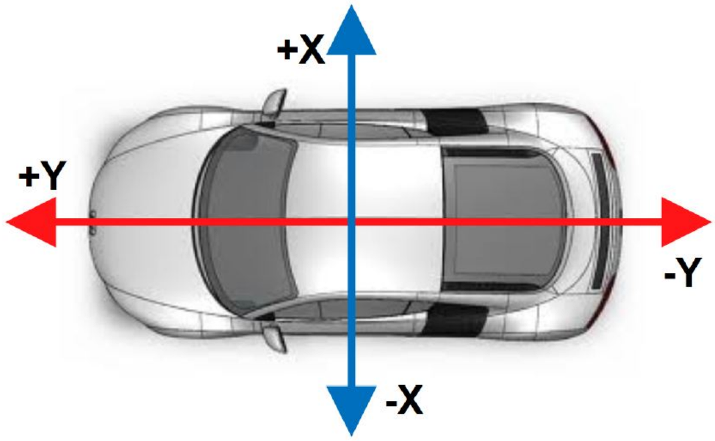


Figure 5.6: The desired reference frame the iPhone axis should be aligned to. It defines they-axis to be parallel to the driving direction

is directed in driving direction and the display is facing in the positive z-axis direction (reference frame). Pushing the brakes while driving straight would result in a positive g-force on the y-axis whereas the value on the other two axis would not change. Another device position with a pitch of $90°$ resulting in the negative z-axis is directed in driving direction and the positive y-axis pointing up. The same action would now result in a measured g-force on the negative z-axis.

Filter device position to result in a unified data format

Later classes working on the motion data expect it to be in a unified format. For example the `ratingAgent` always expects that accelerating or decelerating the car will result in a g-force measured on the y-axis. Whereas excessive braking would be recognized in the first device position, it would be missed in the second one because there would be no g-force on the y-axis at all.

Two steps have to be taken in the filtering process. First, the current device orientation has to be determined. Secondly, the measured accelerations have to be redistributed on the 3 axis to match the acceleration vector for a device resting in reference frame position.

5.2.1 Pitch & Roll

Using gravity to determine roll and pitch

Determining pitch and roll of the device, one can use the earth's gravitational force that always affects the accelerometer. The gravitational force of the earth is a force pointing towards its center with an amount of 1g. Imagine the device resting in the reference frame. The gravitational force would cause the value on the accelerometer's z-axis to be -1g, resulting in an overall acceleration vector of $(x, y, z) = (0, 0, -1)$. In fact, a resting device's acceleration vector results in such a unit vector for all possible positions. For example a pitch of $90°$ would result in an acceleration vector of $(0, -1, 0)$ and a roll of $45°$ results in an acceleration vector of $(0.707, 0, -0.707)$.

The goal is to compute a rotation matrix R with:

$$\begin{aligned} r &= (0, 0, -1) \\ g &= (x, y, z) \\ R \cdot r = g &\Leftrightarrow R^{(-1)} \cdot g = r \end{aligned}$$

where r is the desired reference frame, g the current acceleration vector caused by gravity and R being a 3 x 3 rotation matrix.

Calculate a rotation matrix to redistribute the measured g-force

The iOS framework already separates the user acceleration from gravity. Otherwise this separation could be done by applying a low-pass and a high-pass filter to the measured motion data. With obtaining the rotation matrix R^{-1}, the application is able to redistribute the measured acceleration caused by the user.

The first step in calculating R is to calculate the unit vector u orthogonal to r and g and using it as rotation axis. After this, rotate about this axis with an angle α:

$$\begin{aligned} u &= \tfrac{r \times g}{\|r \times g\|} \\ \alpha &= \arccos(r \cdot g) \end{aligned}$$

Using the above calculations results in R being:

$$\begin{pmatrix} u_x^2 + (1 - u_x^2 c) & u_x u_y (1-c) - u_z s & u_x u_z + u_y s \\ u_x u_y (1-c) + u_z s & u_y^2 + (1 - u_y^2) c & u_x u_z (1-c) - u_x s \\ u_x u_z (1-c) - u_y s & u_y u_z (1-c) + u_x s & u_z^2 + (1 - u_x^2 c) \end{pmatrix}$$

where $c = \cos\alpha$, $s = \sin\alpha$ and u being the axis of rotation.

If $|r \cdot g| \geq 0.99$ the computation becomes unstable due to the vectors being nearly parallel. In this case u is defined as $u = (0, -1, 0)$.

After calculating the inverse of the rotation matrix R^{-1}, it can be used for distributing the measured accelerations to the reference frame:

$$R^{-1} \cdot a = a'$$

where a is the measured user acceleration. The new acceleration vector a' is the acceleration vector that would have been measured if the device would rest in the defined reference frame without any pitch or roll. The same calculations are done for the measured rotation rates.

5.2.2 Yaw

Changes in yaw ratio do not effect the gravity vector

Using the gravity as in 5.2.1—"Pitch & Roll" is not possible regarding the yaw ratio. This is due to the fact that rotating the device around the z-axis does not effect the gravity vector. Imagine the device resting inside the car with a gravity vector of (0,0,-1) and the positive y-axis directed in driving direction. Rotating the device around the z-axis would not change that vector, even if after rotating the negative y-axis would be pointing in driving direction. Whereas roll and pitch can be determined very accurately, it is not possible to do that for yaw as well.

The most important aspect for this system is which of the accelerometer axis is directed in driving direction. This allows a distinction between acceleration and deceleration. Three approaches were tested during the development process to determine this axis. The first approach requires a calibration at the start whereas the second uses a combination of the car's speed and the measured user acceleration after extrapolating pitch and roll. The third approach makes use of the course and heading information stored in the `CLLocation` object that has been received by the application.

First Approach

Requesting certain driving actions from the user to calibrate the device

To determine the yaw rotation, the first approach needs to know the force generated by acceleration and deceleration events on the accelerometer. One possibility would be to start the application in a calibration state where the user is advised to brake while driving straight. The measured acceleration vector would allow the application to compute the yaw rotation by observing the ratio between g-force measured on the x and y-axis. Whereas this would allow the computation of the yaw rotation, requiring a braking event at each start is not feasible for end users. In addition, certain situations like dense traffic would not allow a calibration in this way.

Second Approach

Instead of advising the user to brake at the start of the trip, an automatic calibration over time is considered in which the user is not required to take certain driving actions. Deceleration (braking) as well as acceleration events can be identified by observing the `CLLocation` object's `speed` attribute representing the current car speed. In addition, the rotation rate on the z-axis should be close to zero to make sure the car is driving straight. An increase in speed suggests acceleration whereas a decrease suggests deceleration. When detecting such a change in speed, the next values measured by the accelerometer are analyzed to find the axis directed in driving direction. Whereas this approach is able to determine this axis, tests also showed that this process induced a high error rate. This is the result of the GPS data being inaccurate due to low signal strength and a delay in receiving this data. This causes the measured user acceleration and the current speed being out of sync, making it hard to determine the axis. Using this approach could cause the application to be in an uncalibrated state, leading to no or false detections.

Third Approach

Yaw determination through heading and course information

The third approach does not require force on the accelerometer. Instead it uses the `heading` of the device that is calculated by the integrated magnetometer and the `course` information from the GPS data. Both values contain a value from 0 to 360 degree with the `course` representing the direction the car is driving and `heading` the direction the device is pointing to. Using `heading` requires additional setup of the `CLLocationManger` so that the `RecordTripViewController` gets heading updates. Comparing these two values allows to determine which axis is directed in driving direction. For example a `course` of $300°$ and a heading of $210°$ would suggest that the device's yaw rate is about $90°$ resulting in the positive x-axis pointing in driving direction. While this approach provides a good estimation of the device's yaw rotation, it also includes several drawbacks. After turning the car, the `course` information needs some time to adjust. Also some device positions could result in very inaccurate values for `course` and `heading` making the calculation, which axis is directed in driving direction, wrong.

Device position defined by the user

As mentioned, a yaw ratio detection in an automatic way is very difficult and suffers from a high error rate. The application could allow other device positions by allowing the user to choose the reference frame himself. Some car environments could make it more comfortable for the user to place the device in a position where one of the x-axes is directed in driving direction. A solution to allow this would be an option where the user can select the position he would like to place the device in by defining the axis directed in driving direction. Knowing this axis would allow the application to perform the appropriate rotation to map the current position to the reference frame.

5.2.3 Device Position Discussion

The device position is one of the major problems when using the iPhone in the car environment because it is not possible to to detect the precise device orientation in the car. The gravity measured by the accelerometer is the only hint about the orientation regarding the device's pitch and roll.

Approaches to automatically determine the yaw ratio showed non satisfying results and using one of these could result in false detection and feedback. Since the yaw ratio is still very important for the detection, the positions, the user can place the phone, are limited and the device has to rest in a yaw ration close to zero.

Defined target vector does not include yaw ratio information

Using the above computation of the rotation matrix induces some problems. First, adding roll and pitch to the device at the same time can result in a rotation matrix that rotates the device in a position where the positive y-axis is not directed in driving direction. This is caused by the fact that the target vector $(0, 0, -1)$ does not include any information about the yaw ratio of the device. The calculation computes the rotation matrix that rotates the measured gravity vector to the target vector the fastest way.

Inaccuracy of the gravity vector

Gravity vector can change during driving

Another problem is the gravity vector itself. The gravity vector is computed by combining accelerometer and gyroscope measurements. Although the framework works well in separating user acceleration and gravity, motions still effect the calculated gravity vector. The decision that have to made is which gravity vector can be used to calculate the rotation matrix. Three possibilities are considered: Continuous calibration, initial calibration and periodic calibration.

Continuous calibration would recalculate the rotation matrix with every new motion update received. The advantage of this would be that during driving, changing from one valid device position to another would be possible. The disadvantage of the continuous calibration is that the gravity vector can change through motion although the device position, in relation to the car, does not change. This causes the computed rotation matrix to be inaccurate and therefore resulting in an imprecise calculated motion. Especially on roads in a bad condition this effect could become problematic.

In the initial calibration, the gravity vector is measured in the moment the recording starts. A fixed device position during the trip implied, this rotation matrix would be valid for the complete duration. The advantage of this would be that during all driving events and conditions, the iPhone axes would be aligned with those of the car. Obviously, changes in the device position relatively to the car could lead to false positives in the detection process since the rotation matrix is not adapted to the new device position.

Calculating rotation matrix while the car is not moving

A periodic calibration would be a combination of the previous mentioned methods. It aims to combine the advantages of the continuous and the initial calibration. Whereas the continuous approach always updates the rotation matrix, here only gravity vectors measured while the car is not moving are used to calculate the rotation matrix. Since this approach excludes "false" gravity vectors through motion but allows changes in the device positioning, it was determined to be the best solution and will be used in the application's calibration process. To determine whether the car is currently moving or not, the application makes use of the speed attribute delivered within the location object.

Although the above calculations do not allow an arbitrary positioning, it significantly increases the number of positions the device can be placed in. This includes all position where at least the x or the y-axis is aligned with those of the car. For example, the user is allowed to place the device on the dashboard (on/off-switch pointing in directed direction) or place it in portrait mode. Of course all positions between these are also possible, allowing for example the positioning of the device in the cup holder. Instead of rotating the device around the x-axis the user can also roll the device around it's y-axis when in reference frame position. When starting the application for the first time, the user is informed about the possible positions the phone can be placed in.

5.2.4 Smoothing Data

After the filter handled the position of the device, one can focus on the analysis of driving by interpreting the motion data. The relevant values are the g-force on the x and y-axis as well as the rotation rate over the z-axis. G-force caused by acceleration or deceleration is now visible on the y-axis. Cornering and turning the car causes a change in the rotation rate on the z-axis and a change in the acceleration measured on the x and y-axis.

Since these sensors are highly sensible, the interpretation of the raw data is difficult. Tests done using the raw data showed that an additional filtering is needed to make an analysis possible. Vibrations of the car are constantly influencing these sensors by adding a significant amount of noise to the data. At higher speed levels this noise increases even more. In addition to that, bad road conditions and certain road characteristics can cause high peaks measured by the accelerometer. For example curbs or potholes cause the iPhone's accelerometer to measure strong g-forces that could be interpreted as actions caused by the driver. To avoid that, this noise has to be filtered out of the data.

Car engine and road conditions add noise to the motion data

To achieve this, a low-pass filter is used for smoothing the motion data to filter out most of the noise. The filter is applied after the position related transformations have been done. Test drives have proven the effectiveness of the low-pass filter, making it possible to remove most of the noise while keeping the motion information caused by the driver. It is defined by the equation:

Low-pass filter data to remove noise

$$a_i = (a_i \cdot \alpha) + (\ell_i \cdot *(1.0 - \alpha))$$
$$\ell_i = a_i$$

where a_i is the measured user acceleration on one axis and ℓ_i being the last calculated value for this axis. The variable α defines how much of the measured value will be added to the last calculated value for the specific axis. Setting $\alpha = 0.05$ pointed out to be the best, although it lowers the overall detected maximum g-force peaks significantly but it is more important that most of the noise is filtered out than to keep the height of the g-force peaks. This has to be kept in mind in later evaluation of g-forces measured by the application.

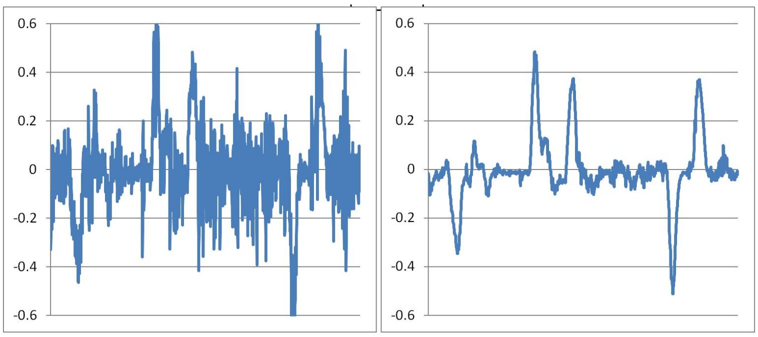


Figure 5.7: Showing the values of the rotation rate on the z-axis before applying the low-pass filter (left) and after applying it (right)

After applying this filtering, most of the noise in the motion data could be removed. Figure 5.7 illustrates the effect of the filter on the z-axis rotation rate. Most of the noise has been filtered out and only the relevant events, in this case when the car is driving a curve, become visible. The same effect occurs on the other axes as well. Another benefit of this is that a constant force in one direction is required to generate g-force peaks. This makes the application resistant to false detection caused by the user moving the phone.

Removing sudden changes in the motion data

In addition to the removal of the noise, the motions caused by the driver became more smooth and sudden changes became rare. This effect will be important in the detection and rating of the `Incidents` detected and is explained in the next section. After this filter has been applied, the motion is returned to the `RecordTripViewController` for further processing.

5.3 Rate Driving

After finishing the filtering process, the computed motion is used to analyze the driving. This is done by a subclass of the `AbstractRatingAgent`. It distinguishes between four types of `Incidents`: excessive acceleration, deceleration (strong braking) , cornering (fast turn- takings) and speeding. For this computation it uses the filtered motion as well as the last received location and the current speed limit. This section will describe the detection process in detail, as well as explain calculations regarding the trip rating.

Detecting acceleration, deceleration, cornering and speeding

Strong acceleration, strong braking and fast turn-takings are detected using the accelerometer data on the x and y-axis combined with the rotation rate on the z-axis. For each `Incident` exists a lower boundary defining the minimal g-force that classifies a driving event as a possible `Incident`. In addition to that, certain patterns were defined to lower the rate of false positives.

Exceeding defined lower boundaries indicating the start of an `Incident`

The lower boundaries are used to trigger the `Incident` detection process and the duration, the g-force stays above the boundary, is measured. The duration is used to additionally separate events caused by the driver from those that are not driver related. Even after filtering, some combinations of road condition and driving speed caused short peaks in the measured g-force and lead to an false detection. Setting the minimal duration for `Incidents` to at least half a second verifies that a constant force is effecting the sensors. Thus, the force lasts a certain time frame it is unlikely that it is caused by outer influences and increases the possibility that the g-forces measured are caused by an actual driving action.

The next sections will describe in detail what causes a driving action to be classified as `Incident`. This includes the algorithm to detect those `Incidents` as well as their effect on the rating.

5.3.1 Acceleration, Deceleration and Cornering

At first, the `RatingAgent` looks at the rotation rate on the z-axis to decide if the car is currently driving a curve or is driving straight. Tests showed that an absolute rotation rate higher than 0.15 rad/s indicates that the car is driving a curve. This results in the rating agent searching for an `Incident` regarding cornering. Otherwise, based on the

Using the rotation rate to identify turn takes g-force measured on the y-axis, the rating agent searches for an acceleration or deceleration `Incident`.

```
if |rot.z| >= 0.15
 possible cornering Incident
else
 if acc.y >= 0
  possible deceleration Incident
 else
  possible acceleration Incident
```

To investigate the rotation rate in addition to the g-force measurements was necessary since observing only acceleration on the x and y-axis lead to false detection. Some situations and device positions lead to an increase in the g-force measured on the x-axis, which then leads to a detection of a cornering `Incident` although the car was not driving around a corner at all. Using the rotation rate in the first place for separating straight driving from turn-taking solved that problem.

Acceleration is the cause of pressing the gas pedal or releasing the brake resulting in a negative g-force on the y-axis. Exceeding the lower boundary for half a second results in an acceleration `Incident`.

To detect braking related `Incidents`, the g-force on the y-axis is considered. A positive value indicates the deceleration of the car, potentially caused by the driver braking. Test drives have shown that driving elongated curves can result in a positive g-force on the y-axis. This effect becomes troublesome if the rotation rate does not indicate that the car is driving a curve. This effect is related to Newton's first law. When driving a curve, the car is kept from driving straight through the occurring centripetal force that is generated between the tires and the road. Since the device is somehow connected with the car, it is also kept from moving straight, resulting in a similar effect like braking the car and therefore a positive g-force measurement on the y-axis. This can cause an exceeding of the lower boundary resulting in a detected deceleration `Incident` by the `ratingAgent`. This mostly happens when during driving at high speed levels.

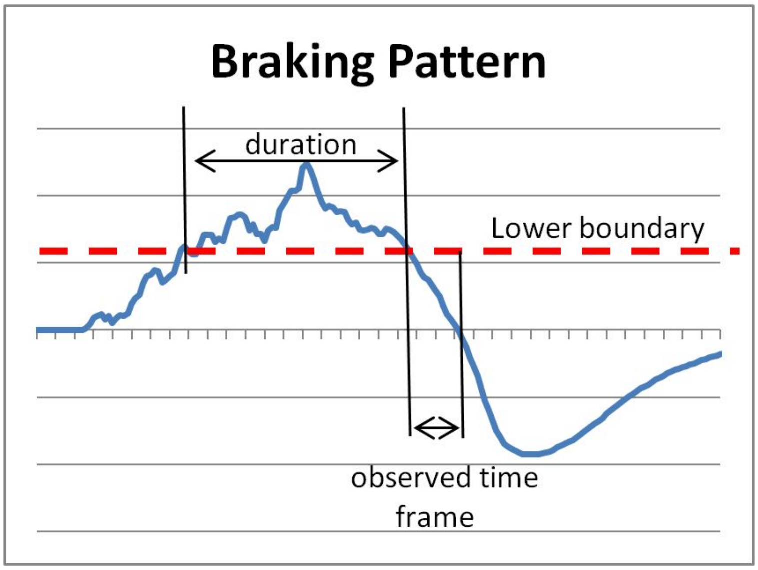


Figure 5.8: Measured g-force on the accelerometers y-axis during braking involving the pressing of the brake pedal and releasing it.

NEWTON'S FIRST LAW:
Every body persists in its state of being at rest or of moving uniformly straight forward, except insofar as it is compelled to change its state by force impressed

Definition:
Newton's First Law

Separating true and false braking events

To distinguish between true and false braking events, the fact that releasing the brakes results in negative g-force on the y-axis is used. Whereas in false events the g-force on the y-axis stays positive after returning to a value lower than the boundary, a true event results in a negative value within a short time frame afterwards. A true braking event is shown in figure 5.8. This means for the detection process that if the g-force falls under the lower boundary and in the observed time frame is smaller than one second, an braking `Incident` is detected. Otherwise no `Incident` is created.

Combining rotation rate and g-force to detect excessive cornering

To detect excessive cornering (turning the steering wheel), the rating agent observes the g-force on the x-axis as well as the rotation rate on the z-axis. If the rotation rate indicates that the car is driving a curve and in addition the g-force on the x-axis exceeds a predefined boundary for a duration of at least half a second, the driving action is classified as an `Incident`. The system distinguishes left and right cornering. Although the criteria for left and right cornering `Incidents` are the same, to know the direction of the curve allows the system to choose an appropriate visualization of the event. This classification does not matter in the later rating process.

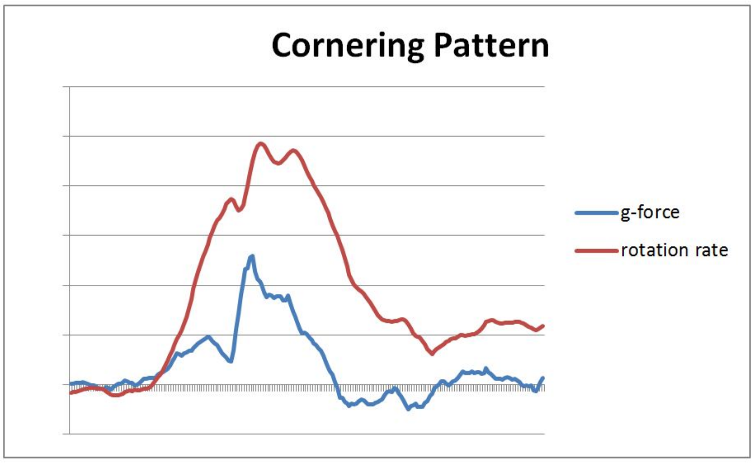


Figure 5.9: Rotation rate on the z-axis and g-force on the x-axis during cornering

The rating agent creates these `Incidents` by using the `PersistenceController`. The `penalty` value as well as the overall trip rating is explained in detail in 5.3.3—"Rating Incidents".

5.3.2 Speeding

Driving 5 km/h above the speed limit triggers speeding notifications

Detecting speeding is done by using the `CLLocation` object's `speed` attribute that is compared to the current speed limit. Creating speeding warnings in the moment the speed limit is exceeded is not a good design. Minor transgressions of the current speed limit are considered normal in the process of adapting the car's speed to current speed limit. Thus, only driving with a speed 5 km/h faster than the current speed limit will be classified as speeding. This decision is based on the outcome of the usability study of the 3.4—"Teen Driver Support System". Parents as well as teen drivers criticized speeding warnings in the moment the speed limit is exceeded. Most of the participants suggested a threshold of up to 5 km/h before a notification is produced. When speeding is detected, the rating agent informs the `RecordTripViewController` by calling its `RatingAgentDelegate` method `speedingStarted` and calls `speedingEnded` when the speeding ended.

5.3.3 Rating Incidents

The rating calculated for each trip is based on the `Incidents` that were detected. Instead of directly using the measured g-force, a penalty is calculated. This penalty describes the severity of an `Incident`. To use the measured g-force directly would be inappropriate because the same g-force for two different `Incidents` does not mean that both events are perceived with the same severity. In addition, `Incidents` added in the future may not involve g-forces at all. Therefore, the penalty allows to compare the severity for different kinds of `Incidents` to each other. This section will describe this value and its computation in detail. These computations are done by the `RatingAgent`.

Penalties for incidents vary from 1.0 to 3.5

As mentioned, three types of `Incidents` are differentiated: acceleration, deceleration and cornering. After the system detects one of these `Incidents`, it calculates a penalty ranging from 1.0 to 3.5 where 1.0 indicates a minor `Incident` and 3.5 a major one. The penalty itself consists of three parts: Basic penalty, g-force peak and duration. Adding up these three values in the final penalty assigned to the `Incidents`:

The penalty consists of 3 components: Basic penalty, g-force and duration

$$penalty = bP + gP + dP$$

where bP is the basic penalty, gP is the g-force penalty and dP is the penalty based on the duration.

To calculate the severity for each `Incident`, the system uses three boundaries that are individually defined for each of the three regarded `Incident` types. Each set of boundaries includes a lower, an upper and a critical boundary.

Whereas the lower and upper boundaries define the range of the regarded interval, the critical boundary is used to additionally identify actions with a g-force in the upper part of the interval. The upper boundary is important to judge the strength of `Incidents` in relation to each other. As mentioned above, the same g-force for two different `Incidents` does not necessarily mean that both event are perceived with the same severity.

The basic penalty can be 1.0 or 1.5 based on the g-force of the incident

The basic penalty for each `Incident` is predefined in the `Defines.h`. Based on the g-force peak, the `RatingAgent` decides whether the `Incident` is considered normal or critical. The g-force peak is defined as the maximum g-force measured in the interval the g-force stayed above the lower boundary. This decision is based on the critical boundary mentioned above. If the g-force exceeds this boundary, the basic penalty is set to 1.5, otherwise it is set to 1.0. To distinguish between normal and critical g-forces allows a higher discrimination in the later overall rating between drivers frequently reaching high levels of g-force and those who only reached the lower levels of the predefined intervals.

```
if gForcePeak >= INCIDENT_CRITICAL_BOUNDARY
  basicPenalty = 1.5;
else
  basicPenalty = 1.0;
```

Another part of the penalty computation is the g-force peak measured. In addition to the lower g-force boundary, an upper g-force boundary for each `Incident` is defined. Based on the g-force peak and the lower and upper boundary, a value between 0.0 and 1.5 is calculated.

$$gForcePenalty = 1.5 \cdot (\frac{(gForcePeak - \ell B)}{mB - \ell B})$$

where ℓB is the lower boundary and mB is the upper boundary for the specific `Incident`.

The last part of the final penalty is computed using the duration the penalty lasted. This duration is defined by the time the measured g-force is higher than the lower boundary. This value lies between 0.0 and 0.5. The duration is included since the longer an event is, the more it can be considered to be on purpose. In addition, the longer an event takes, the higher its risk and the more the driving action effects the passengers.

5.3.4 Calculating Trip Rating

Calculating and presenting a rating for each trip is the most important feedback given by the application. It allows the user to get an idea about his driving performance by just viewing this rating. It allows an easy comparison between different trips and drivers. This section will describe the calculation in detail.

The trip rating makes it easy to compare different trips

The aim is to use the penalties assigned to the `Incidents` described in 5.3.3—"Rating Incidents" and to calculate a reasonable trip rating. This rating can go from maximum score of 100 to 0 where scoring 100 indicates a trip without any `Incidents`. The challenge is to make trips of different length, `Incident` occurrence and `Incident` strength comparable. In the following, several rating functions including their advantages and disadvantages are described.

Ratings range from 100 to 0

At first, it was the intention to use the weather condition in the rating process. Especially weather conditions like fog, freeze, rain or snow were meant to influence detection criteria and also influence the `Trip's` rating. Many studies investigated relationships between different weather conditions and traffic accidents. These investigations found conflicting results about the influence of weather on the crash rate (Brijs et al. [2007]) and its effect on the road safety. In addition, the weather information can be wrong. These concerns lead to the decision that weather conditions should not influence the detection nor the rating.

Weather information does not influence the overall rating

In the current system, speeding does not influence the trip rating nor results in any penalty. The user is just notified that he is currently speeding and receives visual and audible feedback. Due to the possibility of missing speed limit signs in the used open source databases and the possibility

Speeding does not influence the overall rating for the user to turn the speeding detection off, could negatively influence the rating, making the overall comparability of ratings less accurate. In addition to that, the speed limits in these databases can be wrong resulting in a penalty while not speeding at all. In later versions of the application it can be considered to use commercial databases of speed limits which would increase the reliability of the speeding detection. With high reliability, speeding should influence the rating. A possibility is to add "speeding-incidents" to the system including a penalty that is based on the duration the driver was speeding. In addition, the overall speeding time already measured and displayed by the system, should effect the rating as well. The penalty assigned to this `Incident` should be based on the amount by which the speed limit was exceeded.

Excluding weather conditions and speeding in the rating calculation, the function calculating the rating should fulfill the following characteristics:

- The rating should allow to compare all trips with each other, regardless of distance and time,
- the maximum rating is 100.0,
- the rating should never go lower than 0,
- the rating should allow drivers to compare all their trips to each other as well as to compare them to other drivers,
- the rating should be a result of the trip duration/distance, `Incident` count and `Incident` penalty,
- the rating function should be continuous,
- the rating should be reasonable to the user by reviewing his trip.

Distance vs. Duration

Normalizing Trips based on their duration and/or distance

All `Trips` have to be normalized to allow users to compare them to each other. To normalize the `Trips`, the application can use the distance and/or the duration of the

`Trip`. Using the distance would have the advantage that the same trip, for example driving to work, would also be normalized with the same value since the distance between the two start points does not change. The duration on the other hand can vary very much. Having a higher duration for the same trip could indicate a higher traffic volume making `Incidents` more likely.

In the end it comes down to the point whether it is harder to drive longer distances, or whether it is harder to drive for a longer duration regarding the avoidance of detected `Incidents`. This highly depends on the context the trip is recorded in. Driving on the highway, the driver could travel long distances in less time compared to driving through a city. Imagine a person driving for 30 minutes on the highway and one driving 30 minutes through dense city traffic, both causing 5 equal `Incidents`. Whereas the person driving on the highway is able to drive a distance of 50 km (with an average speed of 100km/h) the other person driving through the city only drove 15 km (with an average speed of 30km/h). The rating for the highway trip would result in a higher value than the rating for the city trip does. This seems wrong because it can be expected that driving through dense city traffic results in more driving actions regarding acceleration, deceleration and cornering than driving on a highway.

Trip distance does not include the frequency of measured events

Now imagine another person is driving on the highway for 9 minutes, traveling the same distance as the person driving in the city. Again, both receive 5 `Incidents` with the same strength. Using the distance, both trips would receive the same rating. A rating based on the duration would benefit the city trip. From an objective point of view this rating would be more appropriate since the highway driver caused his `Incidents` far more frequently than the city driver did. In addition, it can be expected that passengers feel more uncomfortable with more `Incidents` in a shorter time frame. Therefore, the duration is used to normalize the trips including one minor change. The duration used will not be the duration between start and finish time of the trip. Instead, the duration the car is actually moving is considered. This excludes a major disadvantage of the duration, that standing still increases the duration and therefore increases the overall rating. Causing no

`Incidents` while not driving at all should not benefit the rating.

Dividing the trip into several parts

Splitting up the trip into several parts and rating each part individually

The first idea was to split up the trips into several parts and to rate each part independently. A 40 minute `Trip` would be divided into four parts with a length of 10 minutes each. Each part contains the `Incidents` occurred in its time frame. For example the first part would include all `Incidents` detected in the first 10 minutes. The second would include all `Incidents` from trip minute 10 to 20, the third from 20 to 30 and the fourth from 30 to 40. Each part is rated with a value reaching from 100 to 0. The overall rating is calculated by building the average of all part:

$$rating = \frac{\sum f(part_i)}{n}$$

where f is a linear rating function and n is the number of parts the trip has been divided into. The function assigns a value based on the `Incident` count and the average of the `Incidents` penalties in the part. Using a fixed time frame of a maximum of 10 minutes would make it easier to setup the rating function regarding its parameters and the points subtracted per `Incident`.

A problem occurs if a part contains `Incidents` in number and strength, that the part's rating would result in a negative value. Imagine the rating function is defined in a way that it allows 5 `Incidents` with an average penalty of 2 resulting in a rating of 0. This means that more than 5 `Incidents` in this part do not influence its rating any further . The positive effect of this is that the driver is able to achieve a high rating, although he had a short period of time where he caused many `Incidents`. This effect could be useful for long trips.

Table 5.1 shows an example of a `Trip` with a length of 40 minutes resulting in 4 parts. R_1 displays the computed rating by the function for part 1 containing the numbers of

Trip	I_1	I_2	I_3	I_4	R_1	R_2	R_3	R_4	Rating
1	1	1	1	1	80	80	80	80	80
2	8	0	0	2	0	100	100	60	65
3	3	3	3	1	40	40	40	80	50

Table 5.1: Comparing different trips with a length of 40 minutes split into 4 parts. Each part is shown with the number of incidents(I_i) and a rating (R_i)

`Incidents` that are displayed in I_1. The last row shows the overall `Trip` rating. For simplicity, the function used in this example does not include the average penalty and simply subtracts 20 points per `Incident`. The problem that occurs now is the fact that the second `Trip` receives a rating of 65 whereas the third `Trip` only receives a rating of 50, while both `Trips` contain 10 `Incidents`. This shows that the rating is strongly effected by the occurrence time of the `Incidents`. A user reviewing and comparing these two `Trips` and their rating would not be able to understand the reason why the two ratings differ that much. To understand that, the application would need to display the rating for each part. This would increase the overall complexity of reviewing and understanding the `Trip's` rating. In addition, the definition of the part's length adds some kind of randomness to the rating where the occurrence time of an `Incident` plays an important role.

Putting incidents in context

Average time between incidents as part of the computation

Another idea was based on the time between `Incidents`. The motivation behind this approach was that several `Incidents` detected in a short time frame could be an indicator that the driver currently is in an aggressive driving state and should receive additional penalties. The value of interest could be the average time between the detected `Incidents`. This value could be used as part of the formula. Again, including the time the `Incidents` occur in the rating could make it difficult for the user to understand the rating. Depending on the situation, two small `Incidents` close to each other could result in a lower rating than two strong ones that are far apart in time.

A penalty factor increases the penalty of incidents temporally close to each other

Instead of using the average between all `Incidents` as part of the final trip rating computation, `Incidents` temporally close to each other could effect the penalty calculation. Additional `Incidents` detected in a certain time frame after the last detection increase a penalty factor, resulting in an overall increased penalty for the detected `Incident`. This would indirectly effect the final rating as well. An advantage of this would be that the user could relate the trip rating to the number of the `Incidents` and their penalties. A disadvantage of this is the fact that the penalty of the `Incident` itself is not only related to its specific characteristics used for the previous penalty calculation (5.3.3—"Rating Incidents"). The penalty, as well as the feedback, would suggest a strong `Incident` to the user, while the `Incident` alone would have been classified as minor. Although this penalty factor is implemented in the system, it is currently disabled because the penalty should represent the `Incidents` severity.

Rating the trip as a whole: linear vs degressive rating calculation

The linear function subtracts a fixed amount of points for each incident

Another solution is to rate the `Trip` as a whole using a linear and a degressive function. In the linear approach a fixed point subtraction is calculated and the maximum `Trip` rating is lowered by this number for each `Incident`. First, the average penalty of all `Incidents` is calculated as $averagePenalty = \frac{\sum_{i=1}^{k} p_i}{k}$, where k is the number of `Incidents` the trip contains and p_i is the penalty of `Incident` i.

As mentioned, the `Trips` will be normalized over their duration. For example a 10 minute trip with one `Incident` with a penalty of 1.0 and a 20 minute trip with two `Incidents` and an average penalty of 1.0 should result in the same rating. The duration is measured in seconds. A value called $allowedPenaltiesPerSecond$ is defined which indicates how many possible `Incidents` per seconds are allowed before the rating reaches 0. In the following $allowedPenaltiesPerSecond$ is 0.025 which indicates that more than one `Incident` per 40 seconds with an average penalty of 1.0 results in a rating equal to 0.

Defining the strictness of the rating function

This value controls the strictness of the rating calculation. It is difficult to say what number of `Incidents` should be allowed in a trip of a certain length. The value has changed a lot during the development and testing process. It was chosen by calculating the rating of several trips with different parameters and comparing these ratings. This value represents an approximation to calculate a reasonable rating for trips of a certain duration, `Incident` count and average strength. As long as this value is not chosen too high or too low, it does not affect the purpose of the rating because the ratings are still comparable although they might seem to high or low.

Duration	Penalty Count	Average Penalty	Rating
300s	1	1.0	86.67
600s	1	1.0	93.33
600s	2	1.0	86.67
900s	2	1.5	86.67
1200s	2	1.0	93.33
1200s	4	1.0	86.67
1200s	2	2.0	86.67
1200s	15	1.0	50
2400s	30	1.0	50

Table 5.2: Comparing trip ratings for trips with different duration, average penalty strength and number of incidents calculated by a linear function

After defining the $allowedPenaltiesPerSecond$, the $maxAllowedPenalties$ for the trip are calculated. Finally, the points subtracted for each `Incident` and the final rating can be calculated with:

$$mP = duration \cdot allowedPenaltiesPerSecond$$
$$pLp = (100 \cdot (aP/mP))$$
$$rating = 100 - (pLp \cdot k)$$

where aP is the average penalty and mP represents the $maximumAllowePenalties$ mentioned above.

Table 5.2 shows some examples of trip ratings calculated by this function. It displays the linear dependency between the components and their influence on the rating. For example a trip of 600 seconds with two `Incidents` results in the same rating as a trip with a duration of 1200 seconds and four `Incidents`.

The degressive function subtracts points based on the remaining trip rating for each incident

Instead of using a linear function, an approach using a degressive function will is discussed. Most of the calculations done in the linear approach will stay the same but instead of subtracting a fixed value for each `Incident`, the amount of points that will be subtracted is based on the remaining amount of points left. This results in

$$rating = \frac{100}{(1+aP/mP)^k}$$

where aP is the average penalty, mA the maximum allowed penalties and k represents the numbers of `Incidents`.

Duration	Penalty Count	Average Penalty	Rating
300s	1	1.0	88.23
600s	1	1.0	93.75
600s	2	1.0	87.89
900s	2	1.5	87.89
1200s	2	1.0	93.65
1200s	4	1.0	87.71
1200s	2	2.0	87.89
1200s	15	1.0	61.15
2400s	30	1.0	60.90

Table 5.3: Comparing trip ratings for trips with different duration, average penalty strength and number of incidents calculated by a degressive function

Ratings calculated by the degressive function are higher

Table 5.3 shows the ratings calculated by the degressive function. Compared to table 5.2, the ratings are higher. In addition, to double the trip duration and number of `Incidents` does not result in the very same rating. Most of the time the difference is smaller than one point making it acceptable regarding the previous defined goals for the rating function.

Comparing the linear to the degressive function, it has a

Figure 5.10: Comparing a linear and degressive approach for calculating the trip rating. The trip has a duration of 1200 seconds and an average penalty of 2.0

huge disadvantage. Once the `Incident` count exceeds the number of $maximumAllowedPenalties$, the rating becomes negative. Since negative ratings don't make much sense these ratings would result in a rating of 0.

Figure 5.10 shows the rating computed by each of the described functions in relation to the number of `Incidents`. An additional `Incident` would lead to a linear rating smaller than 0 whereas the degressive rating would still be above 0. In fact it is impossible for the degressive rating to result in a value smaller than 0. This allows a comparison between drivers with a high number of `Incidents` with the degressive function (in this case more than 30) whereas this is not possible with the linear one.

The degressive rating keeps drivers from receiving a 0 rating

Another advantage is the degressive course of the rating. The first `Incidents` result in a higher point subtraction making high ratings harder to achieve which raising the ambition of the user to avoid even a single `Incident`. On the other hand, users with a lot of `Incidents` do not finish a trip with rating of 0 because with each new `Incicent` the amount of points that gets subtracted gets smaller.

Although both functions were implemented, later research will use the degressive function because of its advantage to assign a higher rating to `Trips` with a lot of `Incidents` while still making it hard to achieve a very good rating.

In addition, the amount of points that are going to be subtracted for each `Incident` is highly controlled by the value set in the $maximumAllowedPenaltiesPerSecond$ parameter. Since this approach has not been tested in a large scale, choosing a disproportionate value could result in very high or low ratings when using the linear approach.

Chapter 6

User Interface

"Beauty and brains, pleasure and usability - they should go hand in hand"

—Donald Norman

Detecting and rating possible `Incidents` is only one aspect of the application. Another aspect is the visualization and notification about detected `Incidents` to help the driver to avoid these mistakes in the future. The user interface should provide an instant usability and the feedback given to user should be understandable and intuitive. The application's user interface as well as the development process are described in this chapter in detail.

6.1 First Iteration

At first, navigation through the application was possible via a `UINavigationController` and a `UITabBarController`. The `UITabBarController` included three tabs that made it possible to access the recording function, the trips overview and the settings of the application.

UITabBar and UINavigationBar to navigate through the application

To record a new `Trip`, the user selects the "Record" tab of the `UITabBarController`. The view that appears can be seen in figure 6.1 on the left and represents the

Figure 6.1: The `RecordTripViewController` (left) and the driver's logbook (right) in their first design

`RecordTripViewController`. The view included several information like the current time and date as well as general information about the trips already recorded. In addition, weather information including temperature and the current condition are displayed. Next to that the distance driven and the current speed of the car are shown. In the middle of the screen a huge four digit number displays the current `Trip` points. Below that a graph displays the current acceleration measured on the three axis. At the views bottom is a huge button that allows the user to start and stop the recording of a `Trip`. Pushing the button in the upper right corner allows the user to name the `Trip`.

The tab Logbook opened a `UITableView` with two groups. First, a summary displaying the number of all trips recorded and the average points regarding theses trips are displayed. Below that all `Trips` are listed in descending order. Selecting a `Trip` would cause a new `UITableView` to open that displays detail information about that `Trip`.

6.2 Second Iteration

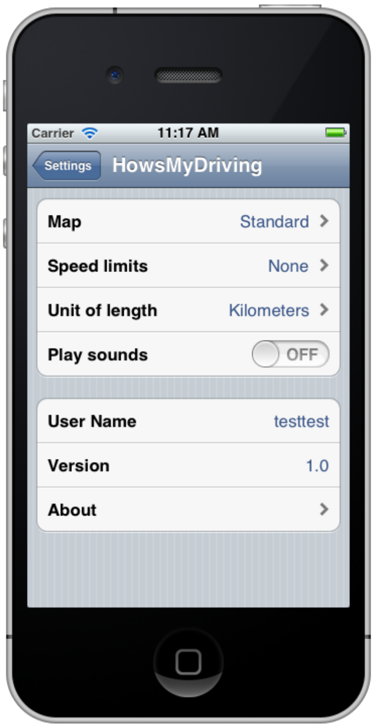


Figure 6.2: Possible settings the user can choose from. These settings can be accessed through the settings application of the iPhone

Moving settings to the iPhone's setting application

Analyzing this first approach of designing the user interface led to several changes. At first, the idea of using an extra tab for displaying the settings of the application was discarded. Instead, the settings were moved to the settings application of the iPhone as recommended by development guide lines. Several new options were added to the settings (figure 6.2). In addition to the options already included in the first prototype, the user can choose whether or not he would like to include speed limits. When choosing to use speed limits, the user has the options to choose which of the previously mentioned speed limit databases he would like to use.

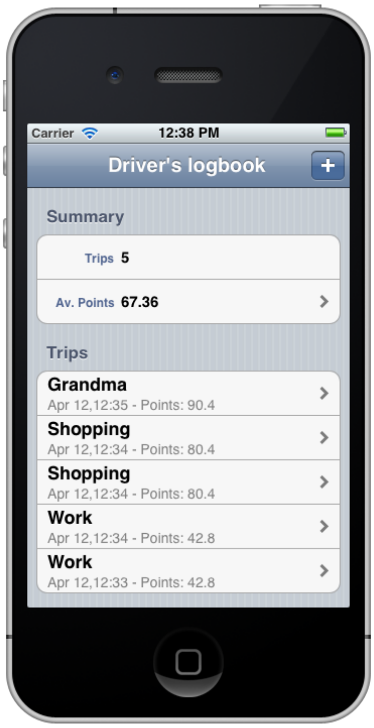


Figure 6.3: Logbook of the application represents the initial view controller for the application

Removing the UITabBar as navigation element

After removing the settings from the application, the `UITabBarController` remained with only two tabs. This led to the decision to remove the `UITabBarController` entirely. The starting point of the application is now the "Logbook" displaying all trips in an `UITableView`. To record a new `Trip`, the user adds a `Trip` to this table view by selecting the "+"-button in upper right corner shown in figure 6.3. This opens the `RecordTripViewConroller` which is responsible for recording a new `Trip` and giving feedback to the user while driving.

The next section will describe the `RecordTripViewController` in detail and will explain the feedback given to the user while driving.

6.2.1 Trip Recording

When starting the recording, the `RecordTripView-Controller` checks whether the location service and a network connection is available. If one of those services is missing, an alert view notifies the user to enable these services. After dismissing the alert view, the interface returns to the "Logbook". Since the recording itself does not require any user interaction, the application is kept from idling and switching off the screen by setting `idleTimerDisabled` to `YES`. When leaving the `RecordTripViewController` this value is set back to `NO`.

Check for location and network connection

Refine the First Iteration

The first interface design of the `RecordTrip-ViewController` (figure 6.1) was completely revamped. The first version included much information that is not relevant for the user and the overall aim of the system to provide feedback regarding his driving. This led to the decision to remove information about about time, date and trips. In addition to that, the rating label was removed. Although it would be possible to compute the rating based on the current `Incidents` and the duration, the rating would constantly increase or decrease during driving. Only the rating calculated at the end of the trip is relevant, and displaying a constantly changing rating the whole time has no benefit for the user.

Unnecessary information is removed

The same is true for the graph view. Displaying the current acceleration measured, only suggests the user that the application is recording and running. It includes no information about the driving itself nor indicates whether `Incidents` were detected in an understandable way to the user. The graph view as well as the button to start the recording were removed.

After removing all unimportant information, only information about weather, distance and vehicle speed remained. The complete screen is covered by a

Using a MKMapView to present feedback

Figure 6.4: The new interface of the `RecordTripViewController` containing a map as the main resource for feedback (left). The right image shows the view controller that allows the user to name his trip

`MKMapView`. Figure 6.4 shows the new design of the `RecordTripViewController` which contains the map with the mentioned information displayed on the top. When the user chooses to retrieve information about speed limits, the current speed limit is displayed in the upper right corner.

Presenting Real-Time Feedback

Report incidents via map annotations

Using this map in combination with location updates allows the application to display the current position of the car. Instead of just showing the current location, the map can be enhanced to display additional information via annotations and overlays. An annotation is a small image on the map defined by location coordinates. Classes that would like to function as an annotation must implement the methods provided by the `MKAnnotation` protocol. The system distinguishes between five annotations (figure 6.5). This map and its annotations will be the main resource of real-time feedback given to the user while driving.

Figure 6.5: Five annotations displayed on the `RecordTripViewController's` map. They indicate trip start, trip end and `Incidents` regarding acceleration, deceleration, cornering (from left to right)

The map is centered around the car's current position. The position is marked with a small image of a car named `CarAnnotation`. The `CarAnnotation` is following the car's current position and changes in location cause the `CarAnnotation` to move to that location. The map's view point is always centered around the `CarAnnotation`.

Automatically starting the trip recording by reaching 15 km/h

The recording does not start immediately after the `RecordTripViewController` appears. This could cause the system to detect `Incidents` while the user is placing the device in the car. Since the button to start the actual recording has been removed, a new way to start the recording is required. The actual recording starts when a speed of at least 15 km/h is detected. This allows the user to start the application even on the way to his car without having to fear to receive penalties while not driving. The user can start the application and then place it in the car and start driving.

Annotations mark the trip's start and end points

As long as the speed measured does not exceed 15 km/h, only the `CarAnnotation` is updated on the map. After the speed reaches at least 15 km/h, the recording starts by setting the `Trip's` start annotation at the current position. From this time on, each location update updates the user interface regarding speed and distance. In addition, each location update results in the instantiation of a new `RoutePoint`. As mentioned, the framework allows to mark certain regions by placing overlays. Overlays make it possible to display complex shapes on the map. With the list of `RoutePoints` it is possible to add a line to the map showing the route of the `Trip`.

Overlay displaying the trip's route

To create this overlay a `MKPolyline` is instantiated. The according view to this is an instance of `MKPolylineView`. The `MKPolyline` cannot be edited so it is necessary to create a new `MKPolyline` with each location update. This means that with each location update the old `MKPolyline` overlay is removed from the map and a new one containing all `RoutePoint` coordinates is created and added to the `MKMapView`.

The framework constantly tries to estimate the location as accurate as possible. This can cause the application to receive several coordinates describing the current position. Adding all these received coordinates to the list of `RoutePoints` would effect the `MKPolyline` negatively. Thus, only location updates that have at least a 10 meter distance to the last stored `RoutePoint`.

Speeding results in changing the overlay's color

If the user has enabled speed limits before the recording started, the current speed limit is displayed in the upper right corner. If speeding is detected by the `ratingAgent`, the text color of the speed limit label turns red. In addition, the route section the driver is or was speeding in is colored orange. This allows the user to adjust his speed according to the limit and allows him to review the sections where on his route he was speeding. The `MKPolylineView` visualizing the overlay that describes the route can only be single-colored. In addition, an `MKPolyline` cannot contain any gaps between the points. This makes it necessary to add a new `MKPolyline` for each speeding section in the trip. Another list containing the coordinates where the driver was speeding is added to the `Trip`. Using this list, it is possible to create new overlays for each speeding section and adding them above the overlay for the route. This results in the route being classified into driving below and driving above the speed limit.

Detecting an `Incident` results in the according annotation being added to the map. In addition, an image version of the annotation fades in. The image is larger than the actual annotation to make it easier for the user to see what kind of `Incident` was detected. The image scales down and becomes invisible shortly after the detection. The annotation remains on the map. This allows the user to see which `Incident` was detected by looking at the annotation. In addition to that, a sound notifying the user is played if sound has been enabled in the settings. The same holds for speeding. This sound is important to notify users who placed their device in a position where they are not able to see the display.

Trip recording as background process

If that is the case, keeping the display on would waste battery power. Since Apple introduced multitasking in iOS 4.0, applications can continue processing even if they are not in the foreground. This allows the user to use for example a navigation software while still being able to record his trip with the application. In addition, background processing makes it possible to keep recording even if the iPhone is locked. To keep the detection running, the application needs to receive location and motion updates even when it is currently in the background. To achieve this, the `UIBackgroundModes location` and `external-accessory` are defined inside the `info.plist`, allowing the application to continue receiving those updates.

Ending the Recording Process

By reaching the desired destination, the user finishes the recording by pressing “Done” in the upper right corner. If the `Trip` is valid, meaning the `Trip` recording started and the distance of the `Trip` is at least 1 km, the user is able to choose a name for the trip. Otherwise the `RecordTripViewController` is popped and the application returns to the driver’s logbook. The user can choose one of his previously entered names from an `UIPickerView` or enter a new one which is afterwards added to the list. If the user does not enter a trip tag, the application automatically sets the name to “Trip i”, where i is the trip number. After pressing “Done”, the recording process is finished. All attributes of the `Trip` are set and the trip is uploaded to the database.

6.2.2 Trip Review

Figure 6.6: Table view showing the recorded information providing the user the possibility to compare and review his trips

Table view presenting the trip's information

After a `Trip` is successfully recorded, a detail view of the `Trip` is presented to the user. It shows all information in a table view displayed in figure 6.6. At the top the calculated trip rating is shown since it represents the most important information regarding the driving evaluation. The background color of this cell ranges from green to red where green indicates the best (100 points) and red the worst (0 points) possible rating. Using this natural mapping of the colors allows a faster interpretation of the results. Besides the rating, the table view includes all other information recorded.

By pressing the button in the upper right corner a `UIActionSheet` is presented allowing the user to choose between two tasks. The first one allows the user to compare the current `Trip` with other `Trips` recorded. After choos-

Figure 6.7: Comparing two trips by choosing the "Compare trip" option from the `UIActionSheet`

ing to compare the trip, the user picks the trip he wants to compare from a list containing all recorded trips. This opens the `CompareViewController` that lists the most important facts of each trip next to each other making it easy to compare certain aspects of the trips (figure 6.7).

Figure 6.8: Rating posted on the user's facebook wall

Posting trip rating on facebook

The other option allows the user to post his score on his facebook wall. Since facebook is the leading social network with over 500 million users, it represents a great way to allow the user to share his rating with his friends even if they do not have the application. It is also a great way for making the application more popular. This feature is integrated by using the official iOS facebook SDK[1] . After logging into facebook, a post similar to the one shown in figure 6.8 appears on the user's wall. The `Trip's` rating can only be posted on facebook once. If a `Trip` was already posted, the option "Post on Facebook" is not displayed in the `UIActionSheet` making it easy for the user to see if whether already posted the results.

Figure 6.9: Paper-like transitions deliver the feeling of browsing through a real driver's log

Since the application wants to suggest the feeling of a driver's log, the user is able to page through the applications logbook like in a real driver's log. This is achieved by using an `UIPageViewController` that allows to page through different `Trip` summaries by using paper-like transitions. The visual effect of this can be seen in figure 6.9

[1]http://developers.facebook.com/docs/reference/iossdk/

Annotation Map

In addition to review the trip on the basis of numbers, the complete trip can be reviewed by using a `MKMapView`. This view can be accessed by selecting the cell at the bottom of the `Trip's` detail view as well as by pressing the button in the compare view. This results in a view composed of a map that shows the complete trip. The map includes all annotations as well as the overlays describing the route of the trip. The zoom level of the map is selected in such a way that the complete trip is visible. At the top of the view the frequency of each `Incident` as well as the time the user was speeding is displayed.

Review trip using a map view

Figure 6.10: The user can review his trip by using this map view. The right image show how rotating the annotations solves the problem of overlapping

Low zoom levels combined with a lot of annotations lead to an overlapping of the annotations and due to the their same coloring, it became hard to identify the type of the annotation. Therefore, the visualization of `Incidents`

Improving annotation images for a better visualization was enhanced by surrounding the symbol indicating the `Incident` type with a red triangle. This triangle represents a warning sign on German roads. The sign is rotated so that one edge of the triangle points on the location where the incident happened. This keeps the annotation borders from merging.

Rotating annotations to dissolve overlapping

Although this improved the visualization significantly, `Incidents` very close to each other overlap in a way that even on very high zoom levels, one `Incident` is hidden below the other (figure 6.10 left). To solve this the problem one of the `Incidents` is rotated (figure 6.10 right). The application checks whether two `Incidents` are close in distance and they are, it chooses the appropriated rotation based on the location of the two `Incidents`. To achieve this, a new subclass of `MKAnnotationView` is created. Overriding the `draw` method allows to compute a rotated representation of the `Incident`. This is also done during the recording process in the `RecordTripViewController`.

Annotation size indicating incidents strength

Since all `Incidents` are rated based on their severity determined by the `ratingAgent`, this information should be included in the representation of the annotation. Annotations are scaled according to the penalty assigned to it. This results in `Incidents` with a high penalty being larger than `Incidents` with a small penalty. Therefore, the user can determine the strength of the `Incident` by viewing at the annotation's size. Doing this on low zoom levels results in incidents with low strength being overlapped by those with high strength. The reason for this is that the annontations are closer together and their size is not related to the current zoom level. Therefore, scaling the annotations according to their strength is done when the zooming level reaches a certain value.

Annotation call-outs provide additional information

Selecting an annotation causes a `call-out` to appear. This `call-out` consists of a title indicating the type. `Incidents` also have a label containing time and penalty below that. On the right is an image additionally suggesting the strength of the `Incident`. The higher the penalty, the higher the bar in the image is (figure 6.10 right).

6.2.3 Overview & Statistics

In addition to the possibility of reviewing a single trip, an overview listing some facts regarding all recorded trips can be accessed through clicking on the "Av. points"-cell in the "Logbook". This opens a table view that adds up the individual numbers of the recorded trips regarding points, average points, time driven, average trip time, time speeding and the number of `Incidents` for each type. Below that the user can access the leaderboard and achievement view controller and the online statistics web page.

Leaderboards & Achievements

Using Game Center for motivation and comparing results to others

The Game Center introduced in iOS 4.1 is a technology included in the Game Kit framework that helps developers to build up social games. It allows the developer to set up leaderboards and achievements by using one account through all Game Center supporting applications. The auto-match function allows a simple way to build up network games. Since this approach includes game-like characteristics, some of the possibilities offered by the Game Kit are used.

To raise the user's motivation to use the application as well as to encourage him to avoid `Incidents`, specific leaderboards and achievements are integrated. The application includes leaderboards for high scores in rating, the overall score, the number of trips recorded and the distance traveled. In addition, several achievements were added regarding those categories.

Leaderboards and achievements are updated after a trip has been successfully recorded. When the user is forwarded to the trip's detail view, newly earned achievements show up at the top if achieved. The leaderboards and achievements can be accessed from inside the application by using the `GKLeaderboardViewController` and the `GKAchievementViewController` shown in figure 6.11.

The idea of using the integrated match making system was

Figure 6.11: `GKLeaderboardViewController` and `GKAchievementViewController` presenting the user performance regarding all trips

Match making could lead to unintended results

discarded. Although it would be possible that different users challenge each other in an asynchronous way, this could cause unintentional results. Challenging other users could lead to some kind of racing application in which the purpose of the application, to make driving safer, is lost.

Since the application only uses the Game Center to raise the user's motivation and allowing the user to compete with other users and friends, it is not mandatory. Therefore, the application can be used even without a Game Center account.

Figure 6.12: Initial start page (left image) and page containing the top user classified in several categories (right image)

Online Statistics

A web page displaying the uploaded data can be accessed by selecting the "Online Statistics" cell. This opens an `UIWebView` which shows the web page. Using a web page in addition to the leaderboards offers the users another way to compare themselves to other users. In contrast to the leaderboards, the possibilities how information is presented are not limited to any restrictions. The `UIWebView` controls look the same as the ones used in Safari and allow basic navigation as well as to dismiss the view.

Web page displaying uploaded data

The web page should be easy to use on a mobile device making it comfortable for the user to browse it. To achieve this the jQueryMobile framework [2] is used. It allows the developer to build up touch optimized web pages. It con-

Web page specially designed for mobile devices

[2]http://jquerymobile.com/

Figure 6.13: Web page showing the list of all users (left) and a user summary containing the rating and incidents as charts (right)

tains several UI elements that are optimized for the use on mobile devices.

The web page contains an overall user list (figure 6.13). Users are identified by a previously selected user name. The user himself can decide who gets to know his user name to guarantee anonymity. To allow a faster navigation, the user list allows the user to search for a specific user name.

Charts displaying the users progress

By selecting a user, two charts are rendered. The first one shows the user's achieved ratings for the last 20 trips in a line chart. Below that, a bar chart indicates the incidents detected during the trip. At the bottom, all trips recorded by the user are listed. Selecting a trip reveals further details about the trip. Remember that all information stored and presented on this web page cannot be related to a specific device and does not contain any location information to ensure anonymity.

Chapter 7

Evaluation

"True genius resides in the capacity for evaluation of uncertain, hazardous, and conflicting information."

—Winston Churchill

This chapter includes two user tests. The first section describes an experiment which aims to determine the lower boundary for each acceleration, deceleration and cornering events. Participants in this experiment were advised to report driving actions they classify as unsafe.

The second user test aims to validate the application's main task, to analyze driving and to calculate an appropriate rating for each trip.

7.1 Determining Boundaries

Reasonable lower boundaries are needed for a reliable detection and rating

The rating of a `Trip` is based on the detected `Incidents`. As described in 5.3—"Rate Driving", predefined lower boundaries are used to classify driving actions as `Incidents`. Exceeding the lower boundaries starts further investigation that can result in an `Incident` detection. To compute a reasonable rating for a `Trip`, those lower boundaries have to be set to a proper value. Setting up these values too low could result in false incident detection. The application should only report driving actions that are related to an unsafe and inappropriate driving behavior. If the application reports every driving action, the overall feedback and learning effect would suffer. In the worst case, with too many false detection, the application would not be used anymore. On the other hand, too high boundaries would only allow the detection of near crashes and extreme situations.

Since there exist no data on g-forces that clearly separate safe from unsafe driving actions which could be used for this classification, the values have to be determined in an experiment. Even if such data would exists, these values had to be adopted because the detected motion data is filtered which influences the intensity of the g-forces used in the detection process.

Passengers report situations where they feel uncomfortable

An experiment was designed to determine the lower boundaries for the three incident types. In this experiment, passengers reported driving actions regarding acceleration, deceleration and cornering, they judge to be unreasonable. This helps to get an overview which g-forces cause passengers to feel unsafe and unpleasant to calibrate the application accordingly. Using these values, the application becomes some sort of back-seat driver who informs the driver about unreasonable driving actions, like a passenger would possibly do.

Passengers flag inappropriate actions by pressing a button

The experiment includes several trips. Each trip includes one driver and one passenger. The application is enhanced to store all measured motion data as well as the timestamp they were measured. The recorded data is stored on the device in CSV-format. The user interface for this test consists of three large buttons named "Acceleration", "Deceleration" and "Cornering" covering the complete application screen. Pressing one of these buttons sets a hint in the data that allows to identify when and what button was pressed for later research.

Placing the iPhone in a car mount to ensure a fixed position

The iPhone is placed inside a car mount that is attached to the front window of the car. This is necessary to guarantee a fixed device position even if the passenger is operating the device. Other positions, like the passenger holding the iPhone, could modify the accelerations measured by the device. This could cause the later determined boundaries to be inaccurate. Before the trip starts, the application is started showing the three buttons. While driving, the passenger is advised to press a button if he thinks that the specific driving action was inappropriate. For example the driver is driving around a corner. If the passenger thinks that the cornering was too sharp or the car is driving to fast around the corner, he is advised to press the “Cornering”-button.

Investigating 15 trips with 141 reports

The experiment included 15 trips, all with different driver-passenger combinations and a total of 10 different users. In addition, several cars were used. The trips were conducted under clear weather conditions. The trip time reached from 10 to 25 minutes. The route was not predefined and was chosen by the driver. After all trips were finished, the areas of the stored data that were marked through the passengers pressing one of the buttons, were reviewed. Each report is associated with the measured g-force on the specific axis. Since it cannot be expected that the passenger presses the button in the moment of the highest g-force peak, the g-force values in the interval 5 seconds around the actual report are considered. During these trips, 141 driving actions were reported including 25 acceleration, 52 deceleration and 64 cornering related events. Whereas the number of deceleration and cornering incidents is almost the same, only a few acceleration incidents were reported. In addition, half of them occurred in one trip.

Acceleration events are rarely reported

As mentioned above, acceleration events were very rare compared to deceleration and cornering events. A reason for this could be that the drivers avoid to strongly accelerate the car at all. Another explanation for this is that the possibility to reach high g-force peaks regarding acceleration is limited by the car.

Although the number of reports were far higher than expected, to determine exact and general applicably lower boundaries from such a small set of test drives is difficult. This is due to the fact that the participant’s classification, whether a driving action is marked as an incident, is subjective. In addition, this experiment is only able to evaluate the maximum g-force that occurred. These g-forces do not directly indicate at which exact g-force level a passenger starts to view a certain driving action to be inappropriate. To evaluate such boundaries, as well as to better reflect a passenger's perception generally, a huge amount of people had to participate in a specially designed experiment. Such an experiment should somehow directly apply increasing levels of g-force to participants in a regulated test environment. This would allow to clearly research at which g-force level passenger's begin to feel uncomfortable and relate it to unsafe driving.

Determine lower and upper boundaries

Nonetheless, for the system to operate the lower and upper boundaries for each event have to be defined. The lower boundaries are the most important since they are involved in the detection process. The lower boundary should have a very high agreement with a passenger's perception, resulting in a very high reliability in the application's reports. They should be defined in a way that $application\ judgment \leftrightarrow passenger\ judgment$.

In this equation it is more important that passengers would agree on the the application reports ($a \rightarrow p$) than to detect all incidents reported by a passenger ($a \leftarrow p$). Of course this detection rate should also be as high as possible. This is due to the fact that a complete agreement between application and the passenger's perception is hard to achieve. A passenger's judgment varies from person to person. What some consider as strong braking, other might consider as normal. In addition, the passenger has much more information about the current situation and context, whereas the application's context is limited to the sensor data. This allows the passengers to detect additional incidents that are not detected by the smartphone's sensors.

Reliability and detection rate

It is important that the application's reports correspond with a general and reasonable judgment and do not correspond to the judgment of an extreme sensitive passenger. Of course the lower boundaries could be set to a very high value, making false detection highly unlikely, but this would also cause the system to miss many events. The difficulty is to find a trade-off between a high reliability of the application's reports and an overall high detection rate. As mentioned above, the application should be calibrated in a way that if there is an incident reported by the applica-

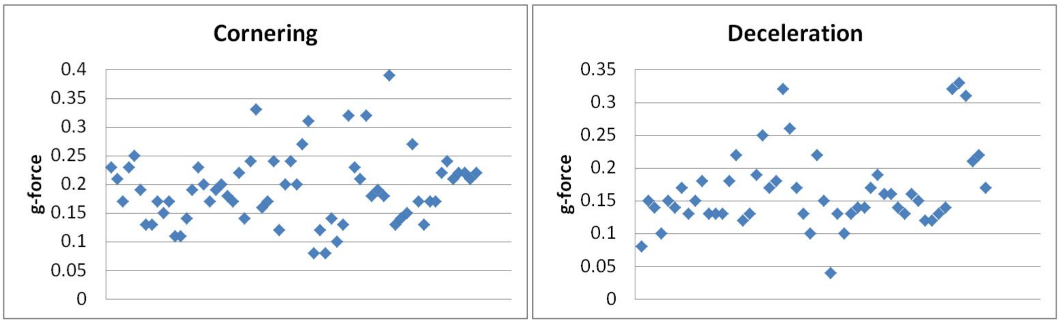


Figure 7.1: Point diagram containing the g-force peaks associated with the passenger reports for cornering (left) and deceleration (right)

tion, there is a high probability that the driving action can be classified to be inappropriate.

7.1.1 Lower Boundaries

Figure 7.1 shows the g-forces peak that occurred when the passenger pressed the button to mark an incident regarding deceleration and cornering. Due to the mentioned problems that this method of determining the lower boundaries includes, it is not possible to use the lowest reported g-force as lower boundary. To start the evaluation, the lower boundaries are defined in a way that the application would have reported about 80% of the events as well. Later analysis of the measured sensor data will investigate if it is reasonable to reduce these boundaries further or if it is even necessary to increase them.

Improving the boundaries

Regarding deceleration, most of the reports had an associated g-force peak higher or equal than 0.13g (84.6%). Regarding cornering, the g-force peaks reached higher values than 0.15g more frequently than it is the case for deceleration and acceleration reports. Whereas several passenger reports start at around 0.11g, most included a g-force greater or equal 0.14g (81.3%). Figure 7.2 shows the g-force peaks associated with reports regarding acceleration events. Compared to deceleration and cornering reports, the reports for acceleration are overall lower in number and strength. To keep 80% of the reports above the in lower boundary, it is set to 0.12g.

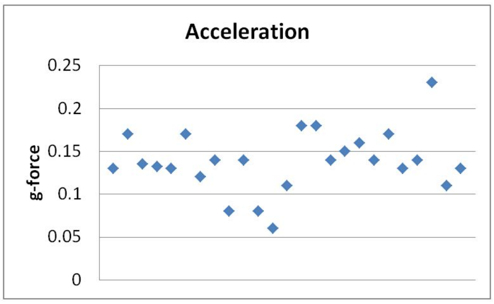


Figure 7.2: Point diagram containing the g-force peaks associated with the passenger reports for acceleration

Since several reports are excluded by theses boundaries, driving actions that were not reported and their associated g-force peaks are investigated. Especially the frequency these g-forces reach the interval around the previously defined boundaries is important. If the analysis shows that many not reported driving actions had a g-force as high as the lower boundary, the boundary needs to be increased to prevent false-positive detection. If that is not the case, it is investigated if it is reasonable to decrease the lower boundaries and therefore to increase the applications detection rate.

Investigating not reported driving actions

Investigating the g-forces measured on the x-axis showed that turn-takings that were not reported reached the interval of 0.12 - 0.14 frequently whereas g-forces higher than 0.14 were very rare. Regarding braking related driving actions, the positive g-forces measured on the y-axis showed the same. Whereas the interval of 0.11-0.13 is reached frequently, values higher than 0.13 are rare. This led to the decision to leave the two lower boundaries unchanged.

Investigating acceleration events revealed that there were several not reported acceleration events with a g-force of close or slightly above the previous defined boundary. Keeping the boundary at 0.12 could result in many false-positives regarding acceleration. Therefore, the boundary is increased by 0.01 to 0.13 which excluded the mentioned events.

7.1.2 Upper Boundaries

Define upper boundaries based on the maximum reported g-force

As mentioned in 5.3—"Rate Driving" the upper boundaries are needed to calculate the penalty value each incident receives. Compared to the lower boundary, this value is not that important to the overall functionality of system. It is needed to define the interval where g-forces are separated. Furthermore it is required for the penalty calculation and to scale the incidents on the map. The boundary should include g-forces that are likely to occur during driving. For example setting the upper boundary to 1 would cause roughly 99% of all incidents detected are classified as minor. To make a decision using the gathered data is difficult. The decision is based on the maximum measured values.

Regarding these values the maximum boundary for deceleration is 0.35 and 0.4 for cornering. Observing 7.2, an upper boundary of 0.25 could be used for acceleration. This would cause the acceleration interval to be relative small. The low g-forces and the low number of events can be related to the low horse power of the cars that were involved. It is likely that cars with more horse power can exceed a boundary of 0.25. This leads to the decision to increase this boundary to 0.35.

Each boundary is slightly higher than the highest measured values because the boundary should at least be higher than all measured accelerations. It is possible that even higher boundaries for each event are reasonable but this decision cannot be made using the gathered data.

This experiment provides an insight which g-forces cause a passenger to report certain driving actions under the current filter settings. Obviously more research and tests with a larger number of cars and passengers is needed to adjust and verify these numbers.

Testing the Boundaries

First test drives using the above determined lower boundaries were satisfying for cornering and deceleration. Regarding acceleration, the system tends to be very sensitive. When the car starts moving, sometimes high acceleration peaks were measured if not operating accelerator and clutch perfectly, causing the lower boundary to be exceeded. Instead of just raising the lower boundary for acceleration, the system was enhanced to distinguish between accelerating the car during driving and accelerating the car while standing. While standing, the lower boundary for acceleration incidents is increased, whereas it stays the same while driving. This solution allows to filter acceleration incidents detected by the system when starting to move the car and still be able to detect strong accelerations while driving. To determine an appropriate value for this additional lower boundary is difficult since this effect was rarely observed.

Add a second lower boundary for acceleration events

Using the small data set available, the boundary was set to 0.17. This value represents the lowest possible value that would have prevented an detection in the first place. Of course more research is needed to validate this decision. In addition, it could be considered to scale the boundary in accordance to the actual driving speed. Since there is no data to define a reasonable scaling factor, the fixed boundary is used.

7.2 System Validation

To validate whether the system is able to predict a passenger's sensations about certain driving actions, a user test is performed. To judge the application's results, the test uses experienced drivers as experts. Both, application and participants, rate trips and will report certain driving actions as incidents.

7.2.1 Experiment Setup

Trips rated by passengers

The experiment requires participants to drive a trip. The trips take place on a public road. Although this field test includes many uncontrolled variables like traffic, weather, road conditions and trip duration, it permits to evaluate the system under real conditions.

To minimize the effect of these variables, each participant drives the same route. This guarantees same trip length and similar duration. The route includes several types of roads with different speed limits. To limit the traffic, rush hours are avoided. The iPhone is mounted on the front shield in portrait mode.

Exchanging drivers and passengers throughout the experiment

Throughout all trips, the independent variables are the driver, the passengers including their position in the car and the car that is used. Exchanging the driver allows to test the system against various driving styles. As mentioned, the system's trip rating and analysis will be compared with that of the passengers. To evaluate if the application reports and ratings represent a general judgment of passengers, multiple participants are included in the experiment. Allowing several passengers to rate and analyze the trip at once, makes it possible to compare the passengers perception to each other. The more passengers will mark a certain driving event, the more likely the event can be considered as a faulty driving action. This increases the importance for the application to report this event, as well. In addition, multiple participants make it possible to identify outlier.

Disable all feedback

The application's interface is modified that no visual or audible feedback about reports is perceived by the driver or the passengers. The application logs all measured sensor data as well as information about the detected incidents for later evaluation.

Using iPod Touches as input devices for passengers

An iPod Touch is handed to each passenger. Participants use a specially developed application to report strong acceleration, hard braking or sharp turn turn takings. The application (figure 7.3) consists of two screens. The first screen displays three buttons for each of the mentioned driving

Figure 7.3: The iPod Touch application used by the passengers to report driving actions and to rate the trip

actions at the screen's bottom. On top of this is a text field summarizing the reports, allowing the participant to validate his button presses. After the trip is finished, the passenger gets to a second screen by pressing "Done" in the top right corner. It includes a summary of the reports and two segment controls. The first one is used to specify the participants position in the car, the second one is used to assign a rating to the trip. The participant's position is important to match the logs to a certain participant later on. Pressing "Done" again generates a log stored on the device that is used in the later evaluation process.

7.2.2 Procedure

Each test run starts at the same location. The application is fixed on the front shield using a car mount. Each passenger receives an iPod touch with the application running.

The driver is instructed to drive as usual and gets informed about the predefined route.

Reporting incidents based on type and strength

At first, all participants fill out a questionnaire. After that, the participants get a short explanation about their task and the interface of the iPod application. During driving, each passenger reports driving actions regarding acceleration, deceleration and cornering which he considers to be inappropriate. The participant can choose from three buttons for each event. The buttons are labeled with an image related to the event. The images on the buttons differ in size and are the same the application uses to provide feedback in the map views. The three buttons allow the participants to report a certain driving action with a strength of 1 (small image), 2 and 3 (large image). Pushing the button with the smallest image reports the last driving action to be possibly inappropriate and if, it can considered to be a minor transgression. Pressing the button in the middle indicates that the driving action is clearly considered to be unsuited. The last button displaying the large image is used to report excessive and aggressive driving actions.

The devices are synchronized by pressing a start button at the start of each trip. This is needed to match the logs from the passengers and the application in the later evaluation. By reaching the destination, the driver as well as the passengers finish the logging process by pressing the "Done"-button.

Each set of passengers completed four trips, each with a different driver.

7.2.3 Participants

Two groups, four participants each

Eight participants were recruited for the final evaluation. It was mandatory that each of the participants is owner of a valid driving license. The participants were split into two groups. Each group exchanged the driver until every participant drove once. This results in a total number of eight trips, four for each group.

Two female and six male people took part in the test. Six

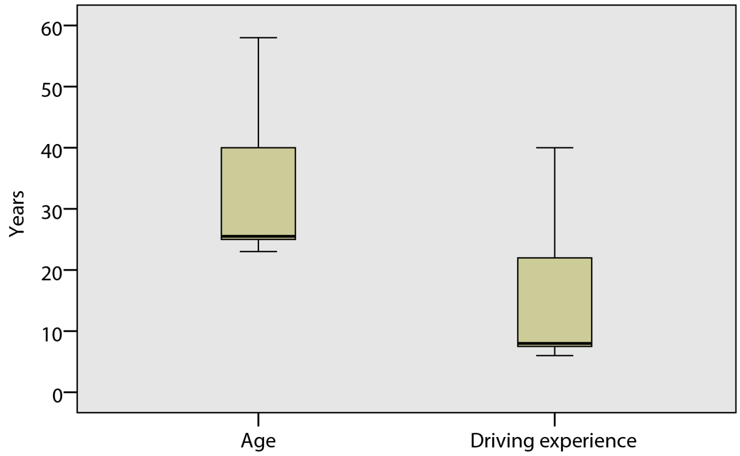


Figure 7.4: Distribution of the participant's age and driving experience

of the participants were between 23-27 years old and two were older than 50 years. Every participant had a driving experience of at least six years with a median of eight years. 75% drive daily and seven of them own a car. Only one participant stated that he is driving sporadically and does not own a car. Most participants estimated their driving behavior as safe and speedy. Most of the participants stated good driving experience on country roads and urban driving, the kind of streets where the test took place. All results of the questionnaire can be seen in B.1.

A total of four different cars were used in the test. The cars were varied to test the system in different cars and was in some cases necessary to guarantee insurance cover.

7.2.4 Data Processing

Merging the logs of all devices

Before the gathered data can be analyzed, the logs for each trip has to be merged. Each of the four devices (1x iPhone, 3x iPod touch) created a log for each trip. The application logs contain sensor data, reports and the final trip rating. The iPodTouch logs contain the reports of one passenger. Both devices add a timestamp to each entry which indicates the time that has passed since trip start and button press. As mentioned above, the start of a trip is set by pressing the start button on each device.

The evaluation aims to compare the agreement between application and passenger. To achieve this, the reports that describe the same driving action have to be merged to events. This process is based on the timestamp and the type of each report.

Grouping together events of the same type and temporally close

Several rules were applied to merge the reports from the different devices and thus classifying them as describing the same driving action. Obviously only reports of the same type are merged. In addition, there are three differentiated possibilities:

- A passenger reported a driving action that was also reported by the application
- A passenger reported a driving action that was also reported by another passenger, but was not reported by the application
- A passenger reported a driving action that was not reported by the application or any other passenger

Defining time intervals to compensate different reporting behaviors

At first, a time interval is chosen that indicates which reports are merged. Regarding reports of the application, all reports in an interval within 6 seconds around the report's timestamp are merged. This interval might seem large but it is needed to compensate possible differences in the reporting behavior of each passenger. Some might tend to report a driving action in the moment it begins. Other might wait till the driving action ended. Certain driving actions, especially driving a curve and braking, can have a duration of up to 4 seconds. This requires the interval to be at least 4 seconds to avoid that reports that describe the same action get split.

In addition, the interval needs to compensate inaccuracy in the calibration process at the start. It is unlikely that all four start buttons are pressed in the exact same moment. Furthermore, the passenger needs some time to decide whether or not to report the driving action and which strength to select. As mentioned in 5.3.3—"Rating Incidents", the application triggers an incident in the moment the g-force falls back below the lower boundary (and of course lasted long enough) for acceleration and cornering events. This means that the temporal distance between report and the actual driving action can vary and probably result in a report at the end of the driving action. This can increase the time difference given by passengers that tend to report early. To compensate all mentioned concerns, the interval is extended by 2 seconds. If several possibilities exist to merge a passenger report with an application report, the report is merged with the closest one.

Merging passenger reports to events

Regarding passenger reports that do not lie in an interval of an application report, another way to match possibly related report is needed. Passenger reports are defined as belonging together if they define a group where each event is at most 5 seconds away from the others. If more than one event could be placed in a group, the one which is the closest to the others is used. The interval is 1 second smaller than previous interval because in this case the difference between the actual driving action and the moment it is reported by the application does not need to be compensated.

7.2.5 Results

Evaluating 8 trips including 86 unique events

The experiment includes 8 trips with 8 different drivers. During all trips the system detected 39 incidents and overall 161 passenger reports were logged, resulting in 86 unique events. As in the previous experiment, the number of events is higher than expected.

Both, application and passengers, reported deceleration incidents at most, followed by cornering. Events regarding acceleration were rare compared to the other two. Especially the application reported only four acceleration events. This matches the result of the 100-Car Naturalistic Driving Study 2.1—"The 100-Car Naturalistic Driving Study" and the first experiment 7.1—"Determining Boundaries". Again, a explanation of this could be that drivers just avoided strong acceleration. Another could lie in the possibility to reach high acceleration values which highly

	Acceleration	Deceleration	Cornering	Total
Events	18	39	29	86
1 Passenger	10	16	10	36
2 Passengers	6	8	8	22
3 Passengers	1	15	11	27
App	4	19	16	39

Table 7.1: Reports by the application and the passengers regarding the three differentiated driving actions and their total number of events. Passengers are split into three groups representing the number of passengers reporting the same event

depends on the power of the car. In addition, this is reduced further by the fact that 4 people increase the weight of the car significantly. Cornering and breaking actions are more independent of the car.

The test was conducted under clear weather conditions, dry roads and a temperature of about 10 degree. Both groups drove on a Sunday afternoon to avoid dense traffic. Each run took the participants about 1.5 hours including all explanations and the questionnaire. The driving time was about 12 minutes for each trip resulting in a total of about 50 minutes driving time for each group.

The experiment took each group about 1.5 hours

The route included all speed limit zones ranging from 30-100 km/h. The route's streets included newly asphalted roads as well as roads in bad condition. The experiment and the application are evaluated based on previous defined hypotheses regarding report and rating agreement.

Investigating Reporting Agreement

At first, the evaluation will focus on comparing reports by passengers and application to investigate if the system is able to detect driving actions that were classified to be inappropriate and unsafe by the passengers. The study wants to prove the following hypotheses:

H1: The application represents a passenger's perception and judgment about certain driving actions and driving style

Proving the applications capabilities to detect certain driving actions

This hypotheses aims to show that it is possible to detect improper driving actions and that it is possible to simulate a passenger's impression regarding driving style based on accelerometer and gyroscope sensor data. It also aims to show that the defined lower boundaries and the classification algorithm represent a robust basis for decision making whether or not an driving action is unsuited. In summary it aims to prove

$$application\ judgment \leftrightarrow passenger\ judgment$$

H1 will be evaluated based on the events that were generated by merging the individual reports. The study is based on 86 events (table 7.1). In summary 86 events will be investigated.

Dividing the hypotheses

It was already mentioned in 7.1—"Determining Boundaries" that comparing reliability and rate of conformity with the passengers is important to decide if the systems is working and to prove H1. Thus, the hypotheses is split and the mentioned aspects $a \rightarrow p$ and $a \leftarrow p$ are investigated separately.

H1.1: Driving actions reported by the application are also reported by the passengers ($a \rightarrow p$)

As previously mentioned, this is the most important aspect of H1. The system should work robust and should not generate incident reports out of nowhere. In addition, the application should not report driving actions that passengers have not reported.

The application reported 39 incidents

The hypotheses will be research based on the set of 39 events that were reported by the application. This includes 4 acceleration, 19 deceleration and 16 cornering events. Again, compared to the other driving actions, acceleration events were very rare. This matches the results from the first experiment.

Regarding the 39 events reported by the application, 38 of them were reported by at least one passenger. This high

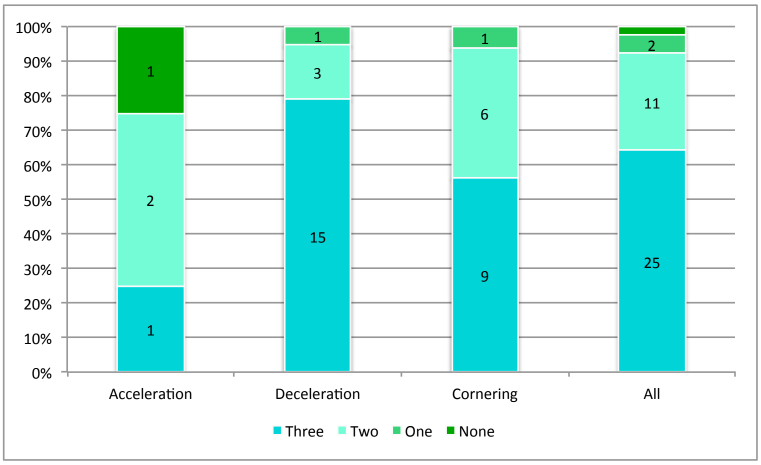


Figure 7.5: Comparing the number of passengers that responded to a certain event reported by the application for each of the three differentiated driving actions

correspondence of 97% shows that the system works robust and does not generate reports for driving actions that no passenger have reported. In addition, it proves that the current calibration and filtering successfully removes noise from the motion data and prevents reports that are unrelated to the current driving.

Most incidents detected are also reported by all passengers

The events are further analyzed by looking at the number of passengers that also reported the event. Figure 7.5 displays these numbers. Obviously, the more passengers report an event, the better. Overall 64.1% of all events were marked by three passengers. This value indicates that the application has a successful classification rate of at least 64.1%. Especially regarding deceleration events, all passengers agreed on 78.9% of the application reports.

That is a satisfying result since it can not be expected that all passengers share the same opinion among all events. Even if only two passengers report an event, it can be assumed that the application's report is justified . Considering events with two passengers, the overall agreement increases to 92.3%. In addition, 76.3% of the application reports were reported by passengers with a strength of 2. Regarding each driving action alone, acceleration events come up with the lowest agreement rate. Since only 4 events are contained in the set of events, it is hard to make any further conclusions.

Confirming H1.1 based on the high overlapping

In summary, in 64.1% of the cases, all 3 participants also reported the incident. In 92.3%, at least 2 participants, in 97.4%, at least 1 participant. From this high overlap we can conclude that H1.1 is confirmed in the majority of cases.

Discussion

One incident included no other reports

One event was only reported by the application and no other passenger. Looking at figure 7.5, the event missed is an incident regarding acceleration. Why was this acceleration event missed by all passengers? One reason could be that the lower boundary for acceleration is set too low. The incident's penalty of 2.74 indicates that the boundary was not just slightly exceeded. In addition, two other acceleration events received a lower penalty but were detected by two passengers. Thus, a too low lower boundary is not responsible.

Reviewing the sensor data around the time the incident was reported revealed that this acceleration incident was the second in a sequence of two acceleration events in the same trip. The sensor data as well as the location of the incidents suggests that the driver accelerated, changed the gear and accelerated again. Although two passengers reported the first, they did not report the second. Knowing this leads to the conclusion that the passengers perceived the process of accelerating the car through several gears as one event. After reporting the first acceleration, they probably thought that they already reported it when the second happened. Regarding the application's reports, acceleration reports close in time could be merged.

H1.2: Driving actions reported by passengers are also reported by the application ($a \leftarrow p$)

The detection rate is 44.7%

This hypotheses aims to investigate the number of events that were reported by at least one passenger and how many of them are reported by the application. Overall 85 events were reported by passengers. 38 of them were also reported

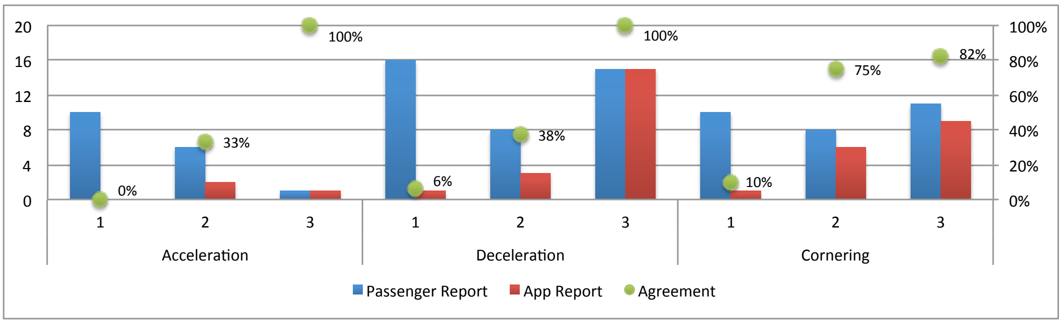


Figure 7.6: Events reported by passengers compared to the number of events reported by the application. For each number of passenger and each event two bars display the number of reports by passengers (blue) and the number of reports by the application (red). The green dot indicates the how many of these events could be detected.

by the application resulting in an detection rate of 44.7%.

To further research the detection rate, the number of passengers that reported an event are compared for each event. The more passengers report an event and the higher the strength of it, the more important it is that the system also detects it. In addition, it is more likely that the driving action can be considered to be inappropriate.

Comparing detection rate for events that received 1,2 or 3 reports

Figure 7.6 displays a bar chart comparing the number of events reported by a certain amount of passengers and the application's respond. Regarding events reported by all passengers, the detection rate is 100% for acceleration and deceleration events, and 82% regarding cornering events. In summary this results in an detection rate of 92.6% of events all passengers reported. The detection rate for events reported by one (6%) or two passengers (50%) are significant lower. Regarding the strength, 100% of all events at least reported by one passenger with a strength of 3 were detected. Thus, no driving action that a passenger reported as excessive and aggressive was missed. In addition, the system was able to detect 85% of the events reported by two or more passengers with a strength of 2.

Filtering events based on their reported severity

Excluding events reported with a strength of 1

The detection rate of 44.7% is calculated based on all events reported by passengers and does not consider the strength

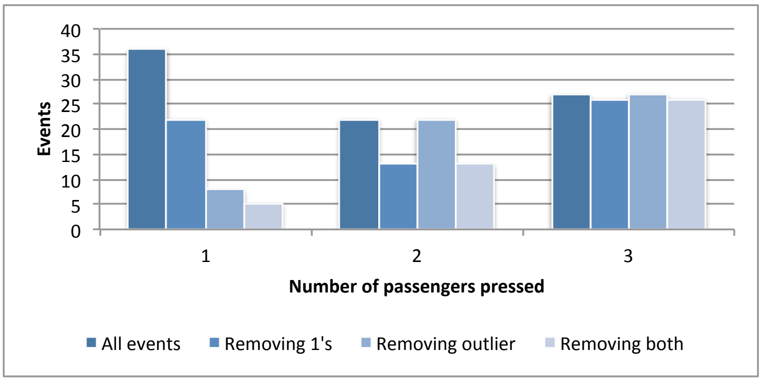


Figure 7.7: Effect on the number of events by excluding certain events. Removing 1's removes all events from the data set which are only reported with a strength of 1. Removing outlier removes events from the data set only participant 3 or 6 reported

of the passenger reports. Because the overall detection rate regarding events reported by one or two passengers seems low, the strength of the reports is reviewed. As mentioned in 7.2.2—"Procedure", participants could choose from three different strength when reporting an event. If the participant was not sure whether or not he considers an driving action to be inappropriate and unsafe, he was advised to report it by pressing the button displaying the smallest image and therefore generate a report of strength 1.

How do these events that are only reported with a strength of 1 effect the detection rate? Reviewing all events revealed that 24 of the 85 were only reported with a strength of 1. Especially in the set of acceleration events, 47% fit this condition. After removing these events, the overall detection increases to 62%. The effect of this filtering process on the number of events and the detection rates is visualized in figure 7.8 and 7.7 and is labeled with "Removing 1's". Especially the detection rate regarding events that were marked by two passengers, increased by 35%.

Identifying and filtering outlier

Identify outlier that negatively effect the detection rate

In addition to the high number of events only reported with a strength of 1, 42% of all 85 events were only reported by

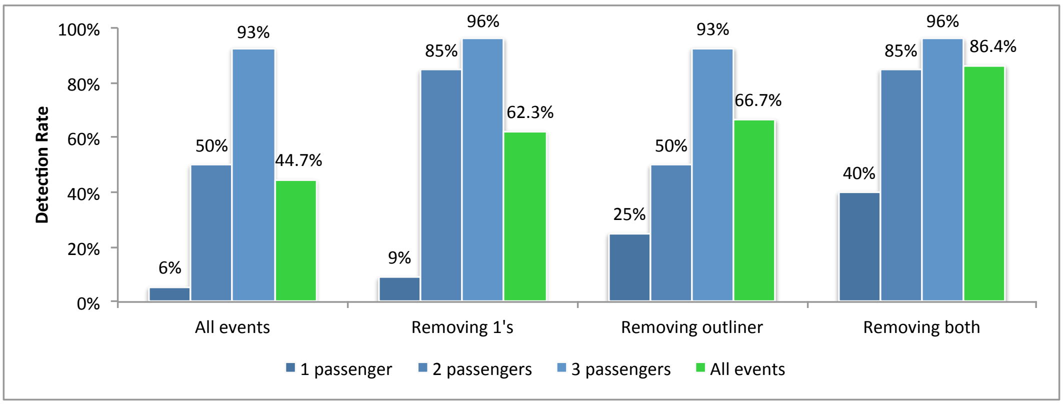


Figure 7.8: Effect on the detection rate by excluding certain events

one passenger alone. Researching how often a certain passenger reported an event alone could identifier outlier in each group. A high number of single presses by one participant could indicate that he does not represent the general opinion and is very sensitive or even worst, that he did not understood his task causing him to press buttons unrelated to the current driving actions.

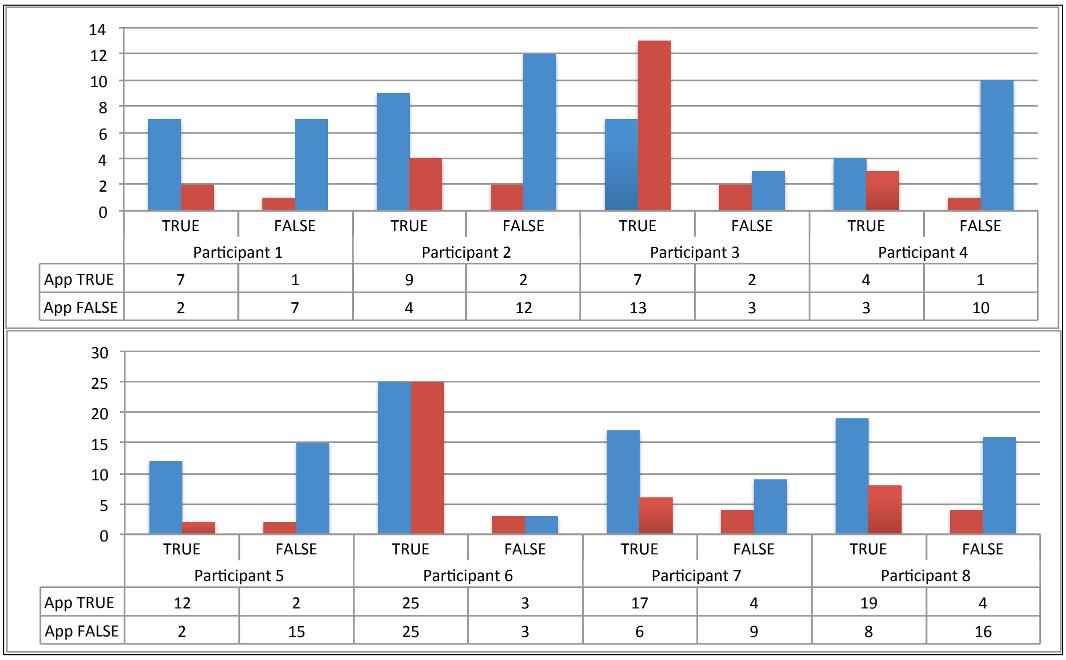
At first, using all events, confusion matrices are created to evaluate the agreement of each participant with the application. This helps to get an overview if there is a significant disagreement between any participant compared to the other participants and the application.

Visualizing the agreement using confusion matrices and bar charts

Figure 7.9 shows the confusion matrices as well as a bar chart making it easier to interpret this data. Blue bars indicate agreement whereas red bars indicate disagreement on events. Whereas six participants show an overall high agreement with the application, participant 3 and 6 disagreed with application on many events. For both, most events include a report by the participant and no report by the application. Because no other participant in their group shows such a strong disagreement with the application, this is an indicator that participant 3 and 6 are outlier and are responsible for most of the single reports.

Analyze confusion matrices using Fisher's Exact Test

The confusion matrices can be further research by using Fisher's Exact Test(Field [2005]) to prove that there is an association between participants and application. Table 7.2



	Participant 1 TRUE	Participant 1 FALSE	Participant 2 TRUE	Participant 2 FALSE	Participant 3 TRUE	Participant 3 FALSE	Participant 4 TRUE	Participant 4 FALSE
App TRUE	7	1	9	2	7	2	4	1
App FALSE	2	7	4	12	13	3	3	10

	Participant 5 TRUE	Participant 5 FALSE	Participant 6 TRUE	Participant 6 FALSE	Participant 7 TRUE	Participant 7 FALSE	Participant 8 TRUE	Participant 8 FALSE
App TRUE	12	2	25	3	17	4	19	4
App FALSE	2	15	25	3	6	9	8	16

Figure 7.9: Confusion Matrices to research the agreement between participant and application on all events. The first chart visualize the results for the first group and the second one for the second group of participants

lists the resulting p-values. As expected, there exists a significant agreement between participant 1, 2, 4, 5, 7, 8 and the application (p-value < 0.05). Only for participant 3 and 6, the null hypotheses, that there is no agreement, could not be rejected. Therefore, participant 3 and 6 are assumed to be the outlier in each group and their reports have to be viewed with caution when investigating the detection rate.

Fisher's Exact Test	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
p-value $<$	0.015	0.006	1.0	0.047	0.001	0.671	0.023	0.001

Table 7.2: P-values of the Fisher's Exact Test using a two a two-sided hypothesis to test whether an association between each passenger and application reports exists

Two participants generated 61% of all single presses

How do these two participants effect the overall detection rate? Reviewing all events revealed that in fact those two were responsible for 61% of the generated events that were only reported by one passenger. Regarding the differentiated driving actions, those are responsible for all single reports regarding acceleration and 62.5% regarding deceleration events.

As shown in figure 7.8, removing the two outlier from the set of all events increases the detection rate to 67%. Removing the events reported with a strength of 1 and in addition removing events only participant 3 or 6 reported, the detection increased up to 86%, with a detection of 96% of events reported by all passengers and 85% of events reported by two passengers. From this high overlap it can be conclude that H1.2 is confirmed in the majority of cases.

Discussion

Talking to participant 3 and 6 about their reporting behavior regarding acceleration events revealed interesting information. Participant 3 stated that he also reported driving actions like speeding or any other dangerous situation by pressing one of three acceleration buttons. Participant 3 as well as 6 stated that they also based their decision, whether to report or not, on the engine sound. Although the sound can be related to unnecessary acceleration and badly operating gas and clutch, as long as no strong acceleration in terms of g-force happen, the driving action cannot be detected by the application nor it can be related to unsafe driving. Nonetheless, acceleration events should be investigated closer in future work, since they were rarely reported and caused problems throughout all experiments and test drives.

Investigating missed events to improve the detection rate

To improve the detection rate, events reported with a strength of 3 or events reported by more than one passenger are further investigated. This includes all events in the set where outlier and events with a strength of 1 are removed.

One cornering event reported by all passengers, whereas two reported it with a strength of 2, was not detected by the application. Reviewing the sensor data for this event shows that the g-force measured had a peak of 0.11g and in consequence is not detected by the application. A reason that all passengers reported it could be the relative high rotation rate indicating that it was a tight curve. Although high rotation rates does not imply a strong driving action, in future work, the combination of the height in rotation rate and measured g-force can be used in the classification process.

Missed events' g-forces were far lower than the lower boundaries

The missed events do not include any report with a strength of 3. Regarding all events where two passengers reported it with a strength of 2, the g-force levels were overall far lower than the defined lower boundaries. Lowering these to that level represents no option because it would cause the system to report almost every driving action. It is concluded that the decision to report the driving action was based on external event the application can not detect.

Conclusion regarding H1

Because H1.1 and H2.2 are confirmed, H1 is confirmed as well. The evaluation could successful prove the applications reliability and, after filtering out events that are not necessarily related to the excessive driving actions, a high detection rate and thus a high overall overlapping between the passengers perception and the application.

Investigating Rating Agreement

This evaluation will focus on the rating each trip has received. As mentioned, each participant received four ratings, one system rating ranging from 0-100 and three passenger ratings ranging from 1-10. Figure B.1 shows a boxplot including the minimum and maximum rating as well as the mean rating each participant received from his passengers.

Passengers chose similar ratings

The participants shared a common mind regarding the rating. In each of the 8 trips, the ratings do not differ by more than two points and in 7 of 8 trips two passengers even chose the same rating. The fact that the ratings are all very close suggests that the rating each passenger chose is appropriated.

H2: Application and passengers rank the drivers in the same order

The system should represent a passenger's perception and judgment about a certain driver and his driving style. As mentioned, each participant was a passenger in 3 trips. Therefore, it is researched if there is a correlation between

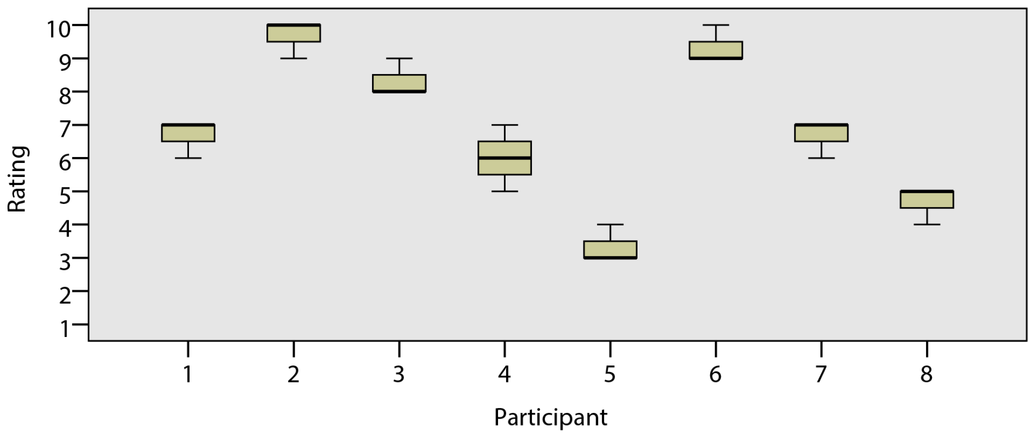


Figure 7.10: Visualizing the minimum, maximum and mean rating each user received for his driving by the passengers

the application's and the passenger's ratings. It is not important that the passenger's rating matches the application's rating in numbers. It cannot be expected that the passengers match the application rating in numbers because he does not know about its calculation. In addition, the passengers can only choose from 10 different ratings making it highly unlikely that the ratings match each other. It is more important that they ranked the drivers in the same order.

Ranking the drivers based on their rating

This is achieved by calculating the Spearman's rank correlation coefficient r_s (Myers and Well [1995]). It is a non-parametric statistical method that investigates the correlation between two variables. The coefficient suggests how strong the relationship between two variables can be described by a monotonic function. The result ranges from -1 to 1. Reaching 1 or -1 means that the variables build a perfect monotone function. Instead of using the variables directly, the variables are ranked in relation to each other. Then the Pearson product-moment correlation coefficient is calculated.

7 of 8 participants show strong correlation

Table 7.3 shows the ratings given by the passengers as well as the related ratings given by the application for the same trip for user group 1. A Spearman's rank correlation coefficient of 1 suggests that there is a significant correlation between participant 1, 2, 4 and the application. This means that the passengers ranked the drivers in the same order.

Group 1	Participant 1			Partipant 2			Participant 3			Participant 4		
P_{rating}	5	8	10	6	7	9	7	6	10	7	8	9
P_{rank}	1	2	3	1	2	3	2	1	3	1	2	3
A_{rating}	54.6	86.5	100	54.6	72.8	86.5	54.6	72.8	100	72.8	86.5	100
A_{rank}	1	2	3	1	2	3	1	2	3	1	2	3
r_s			1			1			0.5			1

Table 7.3: Calculating the Spearman's rank correlation coefficient (r_s) between participant and application. It compares the rating given by each passenger (P_{rating}), to the related application rating (A_{rating}).

Compared to the application, Participant 3 three ranked the drivers differently. Whereas participant 3 and the application ranked equal regarding the best trip, the ranks for the other two trips are swapped.

Group 2	Participant 5			Participant 6			Participant 7			Participant 8		
P_{rating}	4	9	7	5	6	3	5	9	4	10	7	3
P_{rank}	1	3	2	2	3	1	2	3	1	3	2	1
A_{rating}	43.2	87.4	44.1	43.2	44.1	19.0	43.2	87.4	19.0	87.4	44.1	19.0
A_{rank}	1	3	2	2	3	1	2	3	1	3	2	1
r_s			1			1			1			1

Table 7.4: Calculating the Spearman's rank correlation coefficient between participant and application. It compares the rating given by each passenger (P_{rating}), to the related application rating (A_{rating}).

The Spearman's rank correlation coefficient is calculated for group 2 as well. The values are presented in table 7.4. Looking at the correlation coefficient it shows that there is a significant correlation between all participants and the application regarding the rating.

Confirming H2 based on a significant correlation in 7 of 8 cases

In summary, a significant correlation between 7 passengers and the application. Only the ratings of one passenger do not correlate with the application. Nonetheless, the hypotheses, that there is an association between passengers and application regarding the rating, could be confirmed.

Discussion

Since the system was in line with all other participants, it is investigated if a reason can be found that caused participant 3 to rank the mentioned two trips differently compared to the application.

Participant 3 was the only one that rated trip 1 higher (7) than trip 2 (6) (table 7.3). This ranking could be the result of participant 3 really perceived trip 1 to be better than trip 2 and thus has less reports for trip 1. Looking at figure 7.11 it is clear that this is not the case. Although trip 1 includes more reports in total and as well more or the same number of reports for each strength as trip 2, it was higher rated. Talking to participant 3 about this inconsistency he stated that he can not remember what causes his decision. Thus, it cannot be determined if the difference is caused by an inconsistent rating behavior of participant 3 or that other external factors, that could not be reported, happen in trip 2, resulting in an overall lower rating.

Investigating the rating behavior of participant 3

H3: Passenger base their rating on the number of reports

This hypotheses aims to show that passengers base their decision about the rating on the number of reported incidents, like the application does. Figure 7.11 displays a bar chart comparing the number of reports to the assigned rating. The number of reports is further distinguished by their strength. At first sight, a higher number of reports seems to result in a lower rating of the trip for most of the passengers. This suggest a high correlation between the number of reports and the rating.

	Pearson Correlation		Spearman's Rank Correlaton	
	r-value	p-value <	r-value	p-value <
$\sum$ Reports	-0.840	0.001	-0.845	0.001
$\sum$ Weighted reports	-0.858	0.001	-0.890	0.001

Table 7.5: Using Pearson and Spearman's Rank Correlation Coefficient to investigate if the is a relation between the number of reports and the assigned rating

To prove that there is a correlation in general, the Pearson product-moment correlation and the Spearman's rank correlation coefficient, using the 24 data pairs of numbers of incidents and ratings gathered from the participants, are calculated. As shown in table 7.5, there exists a significant correlation between the number of reports and the rating (p-value < 0.01). Therefore, the null hypotheses that there is no association, can be rejected and the hypotheses is confirmed. In addition, the negative coefficient shows that a higher number of reports cause the rating to decrease.

There is a significant correlation between the number of reports and the rating

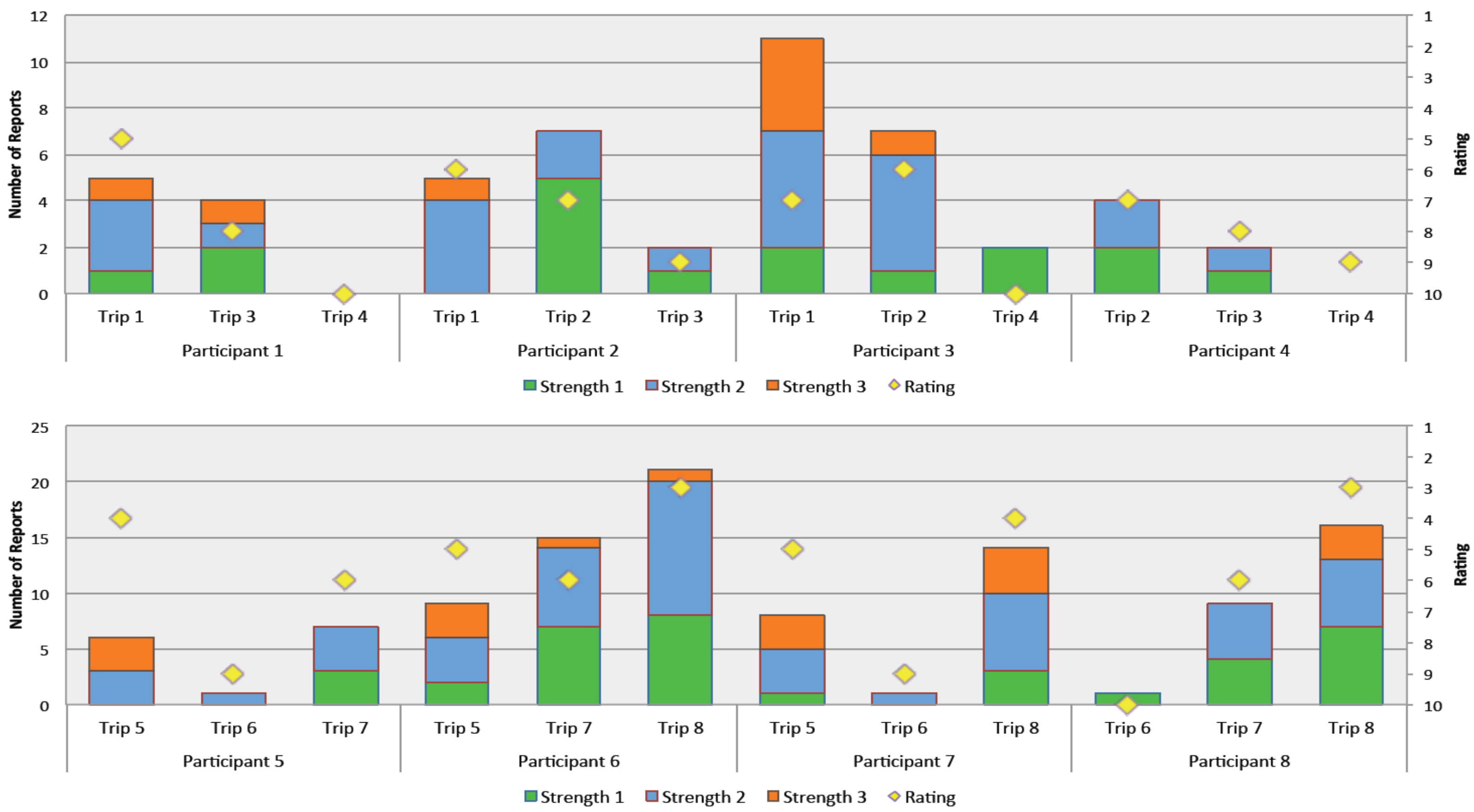


Figure 7.11: The chart displays the overall number of reports for each passenger for each trip. In addition, the bars show how often a specific strength was used during the reporting. The yellow diamond shows the rating the passenger assigned to the trip. It corresponds to the y-axis on the right which is scales from 10 to 0 to better illustrate the correlation between the number of incidents and the rating

The sum of reports does not account the strength of the reports. To do this the sum of reports is calculated by weighting each report with its reported strength. This causes the correlation coefficient to approach -1 further, indicating an even stronger correlation. This lead to the conclusion that the participants additionally account the strength of their reports when deciding the rating.

It is important to note that the high correlation could be caused by actively require the passenger's to report incidents. Nonetheless, the high correlation does not indicate that the approach, to base the rating on the number of reports, is wrong.

Discussion

In the system designing process and especially in the decision how the rating should be calculated, a degressive function was assumed to be the best (5.3—"Rate Driving"). Because there is a significant correlation between the rating and the number of reports, it is possible to use the gathered data to research if the degressive function really performs better compared to the linear one. Ratings are calculated based on the reports using the degressive and the linear function to reveal which fits the user's expectation the best. Because application and passengers rated on different scales, the ratings calculated are transformed to match the passenger's scale.

The degressive ratings match the actual ratings better

Table 7.6 compares the participant's rating to the calculated rating using both functions. The ratings of the degressive function are very close to the actual rating each participant assigned. Scaling up the rating the participants assigned with a factor of 10 to match the application's rating interval, the average in difference is 0.8432 (standard deviation $\sigma = 0.83839$) points for the degressive and 1.9277 ($\sigma = 1.67646$) for the linear approach.

This shows that under the current settings, the degressive function fits the participant's mental model the best. Even for trips with many incidents (trip 8 figure 7.11), passengers seem to avoid assigning very low ratings. This confirms that a degressive rating computation fits a passenger's rating model. It is important to note that a rating of 1 actu-

	Participant 1			Participant 2			Participant 3			Participant 4		
Rating	5	8	10	6	7	9	7	6	10	7	8	9
Linear*	4.5	6.4	10	4	5.1	8.5	1	2.3	8.9	6.7	8.5	10
Degressive*	6.1	7.2	10	5.8	6.3	8.6	3.3	5	9	7.3	8.6	10

	Participant 5			Participant 6			Participant 7			Participant 8		
Rating	4	9	6	5	6	3	5	9	4	10	6	3
Linear*	2.1	9	4.5	1	1	1	1	9	1	9.5	2.7	1
Degressive*	5.0	9	6	4.2	3.4	2	4.4	9	2.5	9.5	5.1	2.6

Table 7.6: Participant's ratings compared to the rating that would have been calculated by using the degressive and the linear function presented in 5.3—"Rate Driving". The * indicates that the calculated ratings are transformed to match the passenger's scale from 1-10

ally indicates a rating where the linear function calculated a value below 0.

Of course these ratings are highly controlled by the value defining the $maxAllowedPenaltiesPerSecond$ and adjusting this value could improve the average distance of the linear rating. The fact that the current ratings using the degressive function are so close to the actual rating suggest that there is no reason for changing the value. In addition, in the underlying case such an adjustment would be difficult since the number of reports for two trips were so high that an adjustment of $maxAllowedPenaltiesPerSecond$, in way that the linear rating matches the passenger ones, would cause all other trips to receive a rating close to 100.

Chapter 8

Summary and Future Work

"The best way to predict the future is to invent it"

—Alan Kay

This chapter will give a brief summary over the thesis in two sections. The first section will repeat the goals of the thesis and will summarize which of these goals could be accomplished. Afterwards it explains who could benefit from such an application and where it could be used. The second section points out interesting futere topics and work.

8.1 Summary and Contribution

Car accidents are responsible for many injuries and death all over the world. Several approaches and studies were conducted with the aim to identify contributing factors in car crashes. The 100-Car Naturalistic Driving Study was the first study that could find a relation between the frequency a driver reaches certain g-force levels and his involvement in crashes, near-crashes and incidents. The How's my Driving placard could successfully reduce crash rates and costs in commercial fleets by reducing the anonymity of the driver in traffic. In the last years, smartphones have become common and due to increasing processing power and possibilities, they developed to an interesting platform. In addition, mobile applications represent a great way to reach a lot of people.

Analyze driving from a passenger's point of view

This thesis aimed to research how modern smartphones can be used in the driving environment to analyze driving behavior. Whereas other approaches use smartphone in combination with additional hardware, this thesis studied which problems and challenges have to be faced when using a smartphone only. The thesis investigated the possibilities modern smartphones offer and how these can be used to analyze and generate feedback concerning the driving performance from a passenger's point of view.

A mobile application providing real-time feedback

A mobile application was developed to analyze and rate driving behavior. The application was developed for an iPhone 4 and only uses its internal sensors. The system differentiates between four kinds of incidents: acceleration, deceleration, cornering and speeding. It is able to provide real-time feedback as well as it allows the driver to evaluate his trips later. Incidents detected by the system are visualized on a map allowing the user to see what happened when and where. Based on the detected incidents, each trip receives a unified rating ranging from 0-100. This makes a quick estimation of the driving performance possible as well as it provides an easy way to compete with others.

Involving experienced drivers in the classification process of good and bad driving actions

The thesis includes two user experiments. The first aimed to determine lower g-force boundaries and to identify patterns that indicate a faulty driving actions regarding acceleration, deceleration and cornering. Involving experienced drivers in this process allowed a calibration of the system in a way that the application developed to an objective passenger, observing and rating the driving behavior.

The second experiment was conducted to validate the system. The system was tested under real driving conditions and its reports are compared to those of other passengers in the car. The experiment could successfully show that the system is able to compute reliable and accurate results under real driving conditions. In addition, it validates the decisions and previously estimated values used by the system.

In summary, this thesis could successfully build a mobile application that can detect and visualize inappropriate and unsafe driving actions regarding acceleration, deceleration and cornering. Since the application does not require additional hardware or any other kind of installation, it can be used by everyone owning an iPhone 4 or higher. Actually, the system could be implemented for every mobile platform and would operate on every device that includes an accelerometer, a gyroscope and a GPS sensor.

Use Cases

Educating drivers

The application represents a low cost system to rate driving from a passenger's perspective and as such, several areas of application are imaginable. The application could be used to educate professional drivers. For example a package delivery company could hand out the application to new drivers or install it on the driver's smartphone. The application could be modified in a way that it is possible to change the defined lower boundaries for each driving action. The company's fleet manager could define g-forces that should not be exceeded. By the application's real-time feedback it would be possible for the new driver to adapt his driving according to fit the fleet managers expectations. This allows an easy and cheap way to introduce the new driver to the companies rules without another person being involved.

Another use case could be the long term observation of drivers. The calculated rating could be used to build leaderboards and rankings between drivers within a company or a group and motivates them to avoid driving actions resulting in an incident detected by the application. A similar approach to this is 3.8—"DriveCam". The advantages of the application in contrast to DriveCam are the low costs. Another disadvantage of the DriveCam device is that it belongs to certain car. The smartphone on the other hand is a personal device which belongs to the driver. When changing the car, the driver is able to take his smartphone and thus the application with him. In addition, the application could be enhanced by reporting incidents in the moment they appear. Additionally uploading location based information would make it possible to build a real-time tracking system allowing a third party to observe drivers in real time.

Allow parents to observe their children's driving and enabling fair insurance rates

Especially teenage drivers represent in matters of car accidents. Parents show a huge interest in finding a way to observe their children while they are driving alone. By using this application, parents can evaluate their children's driving by installing the application on the teenager's mobile phone and provide them with some kind of driving teacher. The system could be further enhanced by sending reports to the parents summarizing each trip after the teen's destination has been reached. Insurance companies on the other hand could make use of such an application to lower the insurance rates of drivers with high scores. This would create fair insurance rates based on the customer's driving behavior of the customer instead of using demographic aspects. This could be a motivation for drivers to drive properly and safe, and could result in an overall more appropriate calculation of insurance rates. Although this is theoretically possible, the reliability of the application have to increase a lot before insurance companies can use them. In addition, the application is very susceptible to manipulations because of the none fixed device position.

At last, the application can be used by individuals. It can be used to track all trips and represent a motivation to drive safe and attentive. This driving style gets rewarded with high scores and achievements. The user can additionally use the time he spends driving to improve his own driving behavior. Therefore, the education process includes no additional time expense.

8.2 Future Work

This thesis represents a first step in making use of the growing smartphone distribution in the traffic environment. Using a mobile phone in the car offers many challenges and includes many interesting topics that could be worth further investigation.

Detect additional incident types

Currently, the system differentiates between three kinds of incidents named acceleration, deceleration and cornering. Future work should investigate which driving actions can be added to this list and if the current incident types can be further classified. Cornering incidents for example could be divided into subtypes of 90 or 180 degree turns. Additional events could be swiping or any other driving action that could possibly be detected by using the smartphone's internal sensors.

Another interesting topic would be to review sensor data in a larger context and investigate if patterns can be found that can be related to unsafe driving. These patterns could be used to overall classify the drivers attitude in addition to the rating.

Bring context to the device

Enabling context by the using the smartphones cameras

When judging certain driving actions only by evaluating g-force and rotation rate data, the missing context is a problem. The application does not know the situation of the driver which could lead to false reports. Whereas sharp turn-takings and strong acceleration can always be considered as unnecessary, this does not apply for deceleration actions. There might be situations were strong braking is needed and the driver had no chance to avoid it. An example of this kind of situation could be a child suddenly running on the street. Although these situations are very rare, the system would create an incident and deduct points for it. Whereas recognizing and visualizing the incident is not wrong, reducing the driver's rating is. Instead of acting unsafe, he possibly prevented an accident.

The system should be enhanced to differentiate between incidents where the driver is at fault and those where he is not. The rating should then represent the evaluation of incidents caused by the driver. The device used in 3.8—"DriveCam" solves this context problem by making use of several cameras installed inside. The device constantly records video material that can be reviewed to learn more about the context. Most smartphones, as well as the iPhone 4, are equipped with at least one camera. This camera could be used to gain information about the current context and conditions the driver is in. Placed in a car mount, the camera at the back of the iPhone would be oriented in driving direction. The camera images could identify the distance to objects in front of the car and could additionally identify suddenly appearing objects. Knowing the distance would allow the application to identify further incidents like tailgating and driving too close to the car in front. In addition, this feature could solve the problem of identifying braking incidents where the driver is not at fault.

Sustainability

Enhance the application with a second rating about sustainability

In times of rising energy costs, sustainability is becoming more and more important. To integrate an analysis of driving in terms of sustainability was already mentioned in 4.1—"Initial Design Ideas". Future work could focus on finding a way to use accelerometer and gps-sensor data to determine unnecessary driving actions that cause higher fuel consumption.

In addition to the rating regarding safety, the application could be enhanced by computing a rating regarding sustainability. New ecological incidents would be added which visualize trip sections and events that can be related to fuel wastage.

Research should focus on whether and what kinds of events and patterns can be determined by observing the smartphones internal sensors. The challenging part will be how accurate a generalized prediction regarding sustainability can be estimated without additional information about the car and the current context. Nonetheless, safety should always be more important than sustainability. This means that reaching a high ecological rating should not affect safety in a bad way.

Bring the application to other vehicles

The application was build and tested for the use in cars. Adding support for additional vehicle types would increase the number of possible users. Actually, the application could already be used in any kind of vehicle, but it is important to note that using the application in vehicles that are very different compared to a car, could require other filtering and rating algorithms.

Determining lower boundaries and patterns for other vehicles

Future work should concentrate on evaluating the use of the application in other vehicle types. The research should focus on determining the lower boundaries and eventually finding new incidents based on the vehicle type. One interesting research could be the evaluation of such an application in buses. The fact that buses are responsible for transporting people fits the applications idea and purpose. When searching for incidents and determining lower boundaries, the research should differentiate between people standing and sitting. Testing this application in buses could additionally reveal interesting information about how professional drivers react and use such a system. The final system could be handed out to several bus drivers to validate the system and research its competitive aspects.

Another interesting area are motorcycles. Riding on a motorcycle is more dangerous than driving in a car. Comparing the risk of a fatal crash for car drivers and motorcyclists revealed a 16 times higher rate of serious injuries per 100 million vehicle kilometers[1] . These numbers increase the importance to prevent accidents in the first place. Future work should focus on finding driving behaviors that lead to those accidents and research the possibilities to identify those using a smartphone.

[1] http://www.milemuncher.com/dft_rdsafety_035422.pdf

Long term study and application effect

Investigating whether or not the application can cause a change in driving behavior

In the end, the application should help to identify dangerous and unsafe driving behaviors and actions. This should help the user to improve his driving behavior. The thesis presents a first step in that direction. Future work should try to determine the effect of this application in a larger scale. Especially if and how the driving behavior is affected by using this application.

Since this evaluation requires a long term study, it cannot be examined in a controlled experiment. It rather requires to analyze the participants driving prior to handing out the application to determine any changes. The high amount of independent variables makes this evaluation even more difficult.

Future work should focus on how such an experiment can be designed and especially how the independent variables can be limited. This is important to allow later analysis to make conclusions about whether or not the application was able to change the driving behavior.

Appendix A

User Questionnaire

Fragebogen zum Erfassen Persönlicher Daten

Teilnehmernummer :

1.1 Wie alt sind Sie?

Jahre

1.2 Welches Geschlecht haben Sie?

☐ Männlich ☐ Weiblich

1.3 Wie lange sind Sie im Besitz einer gültige Fahrerlaubnis?

Jahre

1.4 Was für ein Fahrzeug besitzen Sie?

☐ PKW ☐ Motorrad ☐Anderes ☐Kein Fahrzeug

2.1 Wie oft fahren Sie?

☐ Täglich
☐ Mehrmals die Woche
☐ 1-2 mal die Woche
☐ Unregelmäßig
☐ Garnicht

2.2 Ich fahre überwiegend alleine:

Trifft nicht zu				Trifft zu
☐	☐	☐	☐	☐

Wo fahren Sie am häufigsten?

3.1 Autobahn:

Sehr Selten				Sehr Oft
☐	☐	☐	☐	☐

3.2 Landstraße:

Sehr Selten				Sehr Oft
☐	☐	☐	☐	☐

3.3 Innenstadt:

Sehr Selten				Sehr Oft
☐	☐	☐	☐	☐

3.4 Innerorts:

Sehr Selten				Sehr Oft
☐	☐	☐	☐	☐

4.1-4.3 Wie schätzen Sie ihren Fahrstil ein?

Unsicher				Sicher
☐	☐	☐	☐	☐

Unruhig				Ruhig
☐	☐	☐	☐	☐

Langsam				Zügig
☐	☐	☐	☐	☐

Appendix B

User Test Results

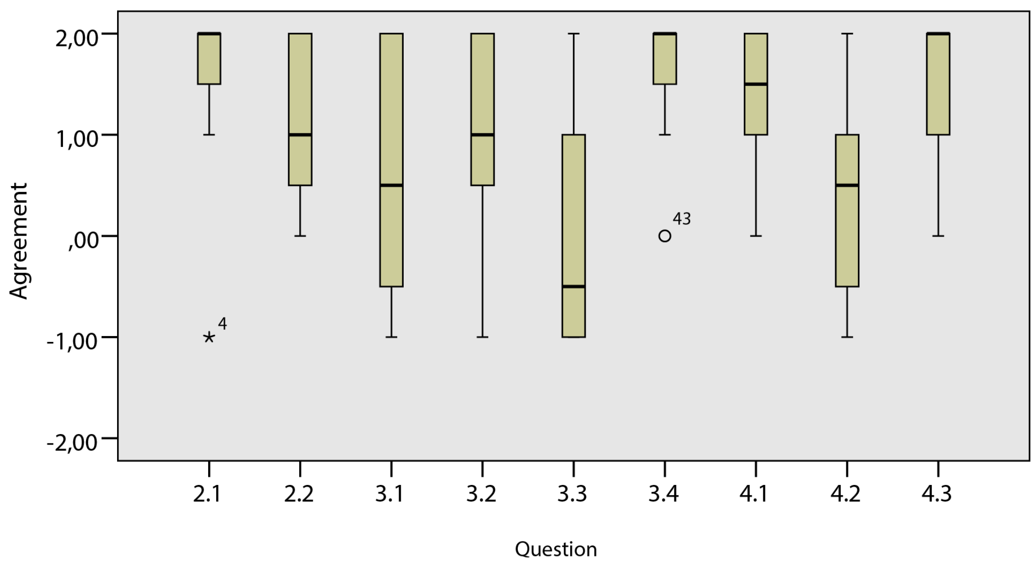


Figure B.1: Results of the questionnaire. The chart displays min and max value as well as the median of the given answers. The box shows the quantil between 25% and 75%

Appendix C

Storyboards

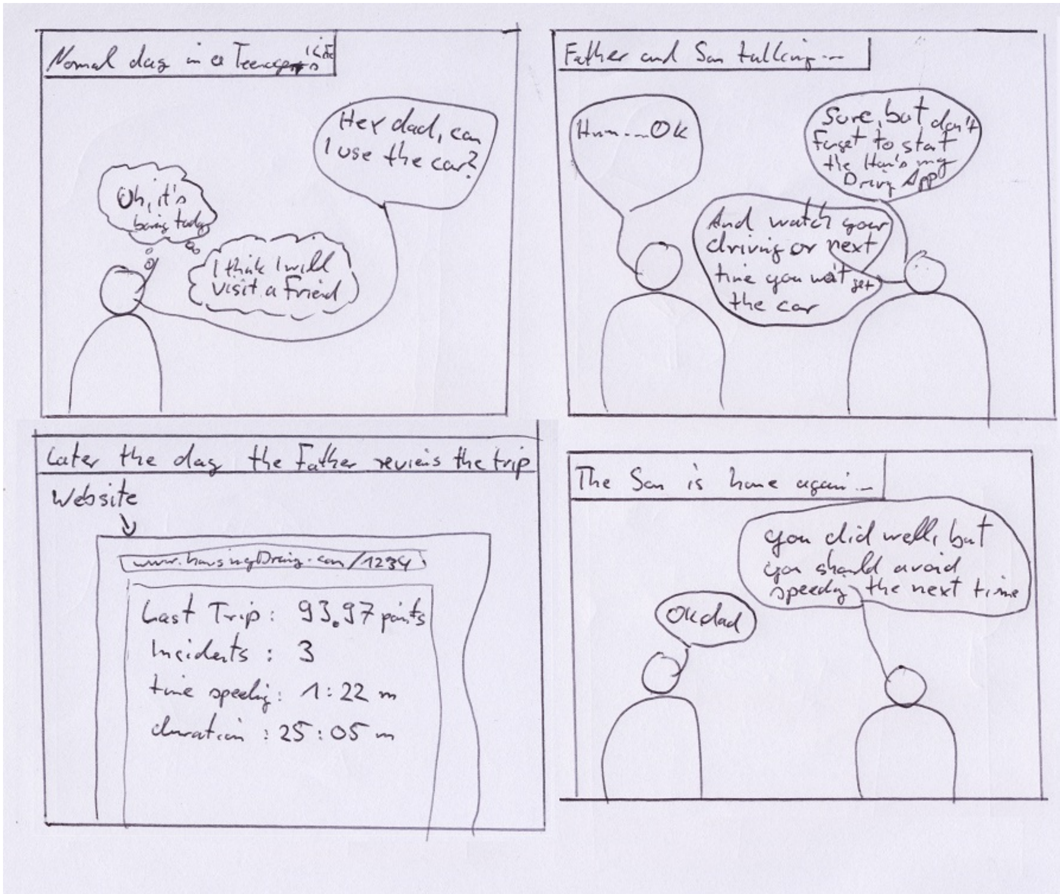


Figure C.1: A storyboard describing a situation where the application is used to control a certain driver

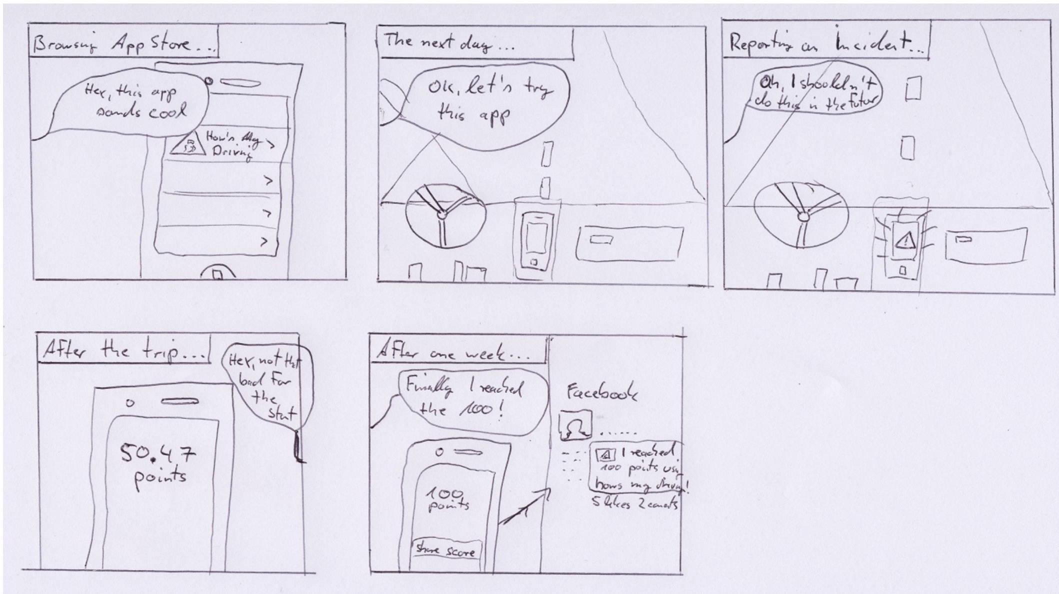


Figure C.2: A storyboard describing a situation where the application is used to control a certain driver

Bibliography

World Report on Road Traffic Injury Prevention: Summary. Technical report, World Health Organization, Geneva, 2004.

Tom Brijs, Dimitris Karlis, and Geert Wets. Studying the effect of weather conditions on daily crash counts using a discrete time series model, 2007.

Paul T Costa and Robert R McCrea. *Revised NEO Personality Inventory (NEO PI-R) and NEO Five-Factor Inventory*. Psychological Assessment Resources, 1992.

Aiden R. Doherty, Zhengwei Qiu, Colum Foley, Hyowon Lee, Cathal Gurrin, and Alan F. Smeaton. Green multimedia: informing people of their carbon footprint through two simple sensors. In *Proceedings of the international conference on Multimedia*, MM '10, pages 441–450, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-933-6. doi: http://doi.acm.org/10.1145/1873951.1874015. URL `http://doi.acm.org/10.1145/1873951.1874015`.

U.S. DoT. Traffic safety facts 2009: A compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system. In *DOT HS 809 337, U.S. Department of Transportation, National Highway Traffic Safety Administration, National Center for Statistics and Analysis*, 2011.

Andy Field. *Discovering Statistics Using SPSS*. SAGE Publications, 2005. ISBN 0761944524.

Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. *Design Patterns: Elements of Reusable Object-*

Oriented Software. Addison-Wesley Professional, 1 edition, 1994. ISBN 0201633612.

Jun Gong and Peter Tarasewich. Guidelines for handheld mobile device interface design. In *In Proceedings of the 2004 DSI Annual Meeting*, 2004. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.87.5230`.

U.S. Government. *Driver Distraction in Commercial Vehicle Operations*. General Books LLC, 2011. ISBN 9781234526085. URL `http://books.google.de/books?id=5RTjygAACAAJ`.

L. James and D. Nahl. *Road Rage and Aggressive Driving: Steering Clear of Highway Warfare*. Prometheus Books, 2000. ISBN 9781573928465. URL `http://books.google.de/books?id=UaOGQgAACAAJ`.

Michael Manser Max Donath Janet Creaser, Alec Gorjestani. Usability evaluation of a smart phone-based novice teen driver support system (tdss). 2011.

Sheila G Klauer, Thomas A Dingus, V L Neale, Jeremy Sudweeks, and D J Ramsey. Dot hs 811 091 comparing real-world behaviors of drivers with high versus low rates of crashes and near-crashes. *Event London*, (February), 2009.

R.R. Knipling, J.S. Hickman, G. Bergoffen, Commercial Truck, Bus Safety Synthesis Program (U.S.), and United States. Federal Motor Carrier Safety Administration. *Effective commercial truck and bus safety management techniques*. Synthesis (Commercial Truck and Bus Safety Synthesis Program (U.S.))). Transportation Research Board, 2003. ISBN 9780309087544. URL `http://books.google.de/books?id=mTNPAAAAMAAJ`.

Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. A survey of mobile phone sensing. *Comm. Mag.*, 48:140–150, September 2010. ISSN 0163-6804. URL `http://dl.acm.org/citation.cfm?id=1866991.1867010`.

Heewon Lee, Woohun Lee, and Youn-Kyung Lim. The effect of eco-driving system towards sustainable driving behavior. In *Proceedings of the 28th of the international* *conference extended abstracts on Human factors in computing systems*, CHI EA '10, pages 4255–4260, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-930-5. doi: http://doi.acm.org/10.1145/1753846.1754135. URL `http://doi.acm.org/10.1145/1753846.1754135`.

Jerome L. Myers and Arnold D. Well. *Research Design & Statistical Analysis*. Routledge, 1 edition, June 1995. ISBN 0805820671. URL `http://www.worldcat.org/isbn/0805820671`.

Vl Neale, Sg Klauer, Rr Knipling, Ta Dingus, Gt Holbrook, and A Petersen. The 100 car naturalistic driving study phase i experimental design. *DOT Report HS*, 809 (December):1–136, 2002. URL `http://www.nhtsa.gov/DOT/NHTSA/NRD/Multimedia/PDFs/Crash%20Avoidance/2002/100CarPhase1Report.pdf`.

NHTSA. Traffic safety facts 2009. Technical report, 2010. URL `http://www.distraction.gov/research/PDF-Files/Distracted-Driving-2009.pdf`.

M.M. Peden and World Health Organization. *World report on road traffic injury prevention*. Nonserial Publication. World Health Organization, 2004. ISBN 9789241562607. URL `http://books.google.de/books?id=iY5RrUKTkeQC`.

K. Rumar. *The Role of Perceptual and Cognitive Filters in Observed Behavior, Human Behavior in Traffic Safety*. 1985.

Michele Ruta, Floriano Scioscia, Eugenio Di Sciascio, and Politecnico Di Bari. A mobile knowledge-based system for on-board diagnostics and car driving assistance, 2010.

Dario D. Salvucci. Predicting the effects of in-car interfaces on driver behavior using a cognitive architecture. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '01, pages 120–127, New York, NY, USA, 2001. ACM. ISBN 1-58113-327-8. doi: http://doi.acm.org/10.1145/365024.365064. URL `http://doi.acm.org/10.1145/365024.365064`.

Lior Strahilevitz. 'how's my driving?' for everyone (and everything?). *NYU Law Review, Vol. 81, p. 1699, November 2006*, 2006.

Leo Tasca. *A Review of the Literature on Aggressive Driving Research*, page 25. 2000. URL http://www.stopandgo.org/research/aggressive/tasca.pdf.

M. Munoz Organero V. Corcoba Magana. Artemisa: Using an android device as an eco-driving assistant, 2011.

Typeset May 30, 2012