

Bachelorarbeit,
Abteilung Informatik

Frühlingssemester
2012

Aufnahme- und Visualisierungssystem für Fahrtrainingszentren



Autoren:	Martin Boos und Jonas Hofer
Betreuer:	Prof. Dr. P. Heinzmann
Projektpartner:	Rolf Bader, A-Z Verkehrsschule,
Experte:	Dr. Th. Siegenthaler, CSI Consulting AG
Gegenleser:	Prof. Dr. O. Augenstein

Arbeitsperiode:	20.2.2012 – 15.6.2012
Arbeitsumfang:	360 Arbeitsstunden bzw. 12 ECTS pro Student

ABSTRACT

Ausgangslage	In Fahrtrainingszentren lernen Fahrschüler/innen, wie man auf kritische Situationen im Strassenverkehr richtig reagiert. Bei einer Vollbremsung gilt es, Reaktions- und Bremsweg möglichst gering zu halten. Die wenigstens Fahrtrainingszentren bieten jedoch Anschauungsmaterial und die Fahrschüler lernen aus Beobachtung anderer Fahrten.
Ziel	Ziel dieser Arbeit ist es, ein passives Aufnahme- und Visualisierungssystem für Vollbremsungen zu erstellen. Passiv bedeutet hier, dass das System ohne jegliche Änderungen an den Fahrzeugen funktionieren muss. Am Ende jeder Bremsung soll eine Auswertung dem Fahrer folgende Punkte grafisch aufzeigen: • Geschwindigkeitsverlauf • Reaktionszeit und Reaktionsweg • Bremszeitpunkt und Bremsweg • Geschwindigkeit bei Beginn der Bremsung • Anhaltestrecke
Vorgehen	Nachdem die grundsätzlichen Anforderungen definiert waren, wurde eine Evaluation verschiedener Technologien und Produkte durchgeführt. Die Evaluation zeigte, dass die erforderlichen Daten aus einer Vollbremsung nur durch Bildanalyse ermittelt werden können, wenn dabei die Anforderungen eines passiven Systems erfüllt werden sollen.
Ergebnis	Entstanden ist ein Programm, das Videomaterial analysiert und einzelne Videosequenzen der Vollbremsungen mit den ermittelten Daten speichern und wiedergeben kann. Es können auch zwei beliebige Fahrten verglichen werden, wobei diese dann untereinander abgespielt werden.

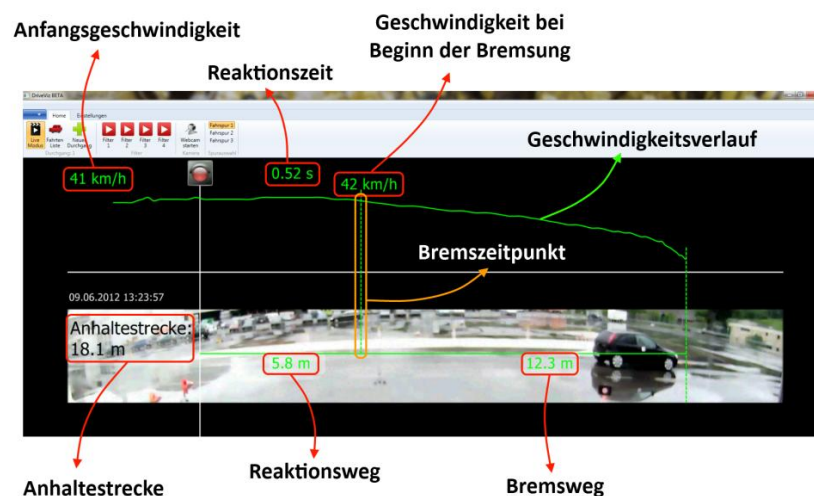


ABBILDUNG 1: RESULTAT

AUFGABENSTELLUNG „AUFNAHME- UND VISUALISIERUNGSSYSTEM FÜR FAHRTRAININGSZENTREN“

Studiengang: Informatik (I)

Semester: Frühjahrssemester 2012 (19.02.2012-25.05.2012)

Durchführung: Bachelorarbeit

Fachrichtung: Internet-Technologien und -Anwendungen

Institut: ITA, Internet-Technologien und Anwendungen

Gruppe: Boos, Martin
Hofer, Jonas

Betreuer: Heinzmann, Peter

Gegenleser: Prof. Dr. O. Augenstein

Experte: Dr. Th. Siegenthaler, CSI AG, Zürich

Industriepartner: Rolf Bader, <http://www.autofahren.ch>

AUSGANGSLAGE

In Fahrsicherheitszentren können Reaktionen auf kritische Verkehrssituationen bei kritischen Strassenverhältnissen trainiert werden. Die Fahrschüler lernen beispielsweise, wie lange der Bremsweg bei verschiedenen Fahrgeschwindigkeiten wird, wie man um Kurven fährt oder wie man auf plötzlich erscheinende Hindernisse reagieren soll. Die Fahrschüler fahren in ihren eigenen Fahrzeugen auf speziell präparierten Parcours. Teilweise werden mit Wasserfontänen plötzlich auftauchende Hindernisse simuliert oder es werden Bälle in die Fahrbahn geworfen. Diese Hindernisse sollen durch richtiges Brems- und Lenkverhalten, bzw. angepasste Geschwindigkeit, umfahren werden. Gegenwärtig werden in den meisten Fahrzentren keine Daten zu den verschiedenen Übungen angeboten. Man lernt in erster Linie durch die Beobachtung der anderen Kursteilnehmer.

ZIEL

Im Rahmen dieser Bachelorarbeit soll ein System realisiert werden, mit welchem die Bewegungsdaten der Fahrzeuge online erfasst und visualisiert werden können. Die Fahrten sollen unmittelbar nach jeder Übung ausgewertet und auf einem Bildschirm angezeigt werden.

TEILAUFGABEN

- Requirement Engineering
- Studium verschiedener Tracking-Systeme
- Anforderungen beim Fahrtraining (Bremsmanöver, Kurvenfahren)
- Auswertemöglichkeiten, Präsentation der Ergebnisse
- Flexibilität beim Einsatz bei verschiedenen Fahrtrainingszentren
- Übernahme von Auslösesignalen (z.B. Triggerpunkt für Bremsmanöver)
- Import der Daten zu den Hindernissen bzw. zur Übungsanlage
- Evtl. Kopplung mit Hindernis-Ansteuerungssystem
- Evtl. Kopplung mit Auto-Daten (Bremsvorgang, Bremszeitpunkt)
- Visualisierung der Fahrparameter (z.B. Geschwindigkeit, Bremszeitpunkt, Reaktionszeit)
- Visualisierung des Fahrweges und der Lage des Autos (Spuren, Geschwindigkeiten)
- Eventuell auch Erkennung von Bällen
- Analyse von Aufnahmesystemen (Objektverfolgungssystemen), Inbetriebnahme eines Positionsaufnahmesystems
 - Ubisense Ultrabreitband (UWB) Aufnahmesystem)
 - Optische Aufnahmesysteme
 - weitere
- Überprüfung der Genauigkeit und Messraten zu den Positionsangaben
- Vergleich, Beurteilung für die geplante Anwendung
- Design der Anwendung
- Systemarchitektur
- Realisierung Anwendung (möglichst .Net)
- Stand alone-Lösung
- Testing
- Installation und Inbetriebnahme auf dem Gelände in Gossau, SG

ERKLÄRUNG ZUR URHEBERSCHAFT

Die vorliegende Arbeit basiert auf Ideen, Arbeitsleistungen, Hilfestellungen und Beiträgen gemäss folgender Aufstellung:

Gegenstand, Leistung	Person	Funktion
Bachelor Arbeit "Aufnahme- und Visualisierungssystem für Fahrtrainingszentren Beide Personen waren in allen Bereichen involviert.	Jonas Hofer & Martin Boos	Autor der Arbeit
Korrekturlesen, Hinweise zur sprachlichen Überarbeitung	Nadine Walther & Juliane Seidel	Gegenleserin
Zurverfügungstellung des Testnotebooks	Adrian Stucky	
Idee, Aufgabenstellung, allgemeines Pflichtenheft, Betreuung während der Arbeit	Prof. Dr. P. Heinzmann	Verantwortlicher Professor
Zurverfügungstellung von Infrastruktur, Testgelände und Testfahrzeug	Rolf Bader	Industriepartner

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit gemäss obiger Zusammenstellung selber und ohne weitere fremde Hilfe durchgeführt habe,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.

Rapperswil, den.....

Student

Rapperswil, den.....

Student

VEREINBARUNG ZUR VERWENDUNG UND WEITERENTWICKLUNG DER ARBEIT¹

1. GEGENSTAND DER VEREINBARUNG

Mit dieser Vereinbarung werden die Rechte über die Verwendung und die Weiterentwicklung der Ergebnisse der *Bachelorarbeit „Aufnahme- und Visualisierungssystem für Fahrtrainingszentren“* von *Jonas Hofer* und *Martin Boos* unter der Betreuung von Prof. Dr. P. Heinzmann (für die Arbeit verantwortlicher Professor) geregelt.

2. URHEBERRECHT

Die Urheberrechte stehen der Studentin / dem Student zu.

3. VERWENDUNG

Die Ergebnisse der Arbeit dürfen sowohl von allen an der Arbeit beteiligten Parteien, d.h. von den Studenten, welche die Arbeit verfasst haben, vom verantwortlichen Professor sowie vom Industriepartner verwendet und weiter entwickelt werden. Die Namensnennung der beteiligten Parteien ist bei der Weiterverwendung erwünscht, aber nicht Pflicht.

Rapperswil, den.....

Student

Rapperswil, den.....

Student

Rapperswil, den.....

Verantwortlicher Professor

Rapperswil, den.....

Industriepartner

¹ Diese Vereinbarung basiert auf den Muster-Vereinbarungen in den *Richtlinien der HSR zur Durchführung von Projekt-, Studien-, Diplom- oder Bachelorarbeiten* vom 16. Februar 2009.

BROSCHÜRENTEXT

AUSGANGSLAGE

In Fahrtrainingszentren lernen Fahrzeugführer/innen, wie man auf kritische Situationen im Strassenverkehr richtig reagiert. Bei einer Vollbremsung gilt es, Reaktions- und Bremsweg möglichst gering zu halten. Bei den meisten Fahrtrainingszentren lernen die Teilnehmer/innen aus Beobachtung anderer Fahrten. Ziel der Arbeit war es, ein passives Aufnahme- und Visualisierungssystem für Vollbremsungen zu erstellen. Passiv bedeutet hier, dass das System ohne jegliche Modifikationen oder Erweiterung an den Fahrzeugen funktionieren muss. Am Ende jeder Bremsung soll eine Auswertung dem Fahrer folgende Punkte grafisch aufzeigen:

- Geschwindigkeitsverlauf
- Reaktionszeit und Reaktionsweg
- Bremszeitpunkt und Bremsweg
- Anhaltestrecke

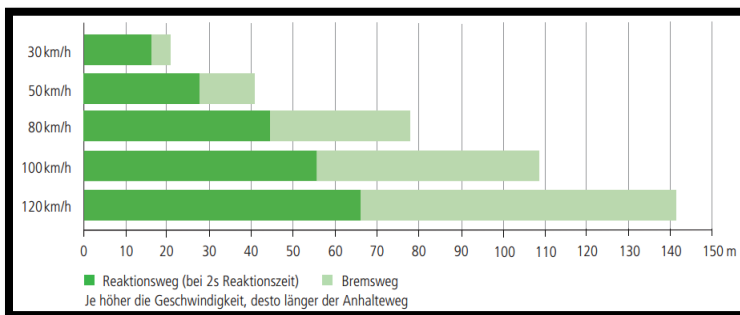


ABBILDUNG 2: STATISTIK DER BERATUNGSSTELLE FÜR UNFALLVERHÜTUNG (BFU)

VORGEHEN / TECHNOLOGIEN

Eine Evaluation verschiedener Technologien ergab, dass Videoanalyse die einzige Möglichkeit ist, die allen Anforderungen gerecht wird. Mit einer Weitwinkel Netzwerkkamera wird die Fahrstrecke gefilmt und das Bildmaterial in Echtzeit verarbeitet. Mit verschiedenen Videoanalysetechniken werden Fahrzeuge erkannt und verfolgt. Anhand der dadurch erhaltenen Daten lassen sich Positionen und Geschwindigkeiten berechnen. Der durch das Weitwinkelobjektiv entstandene Bildverzug beeinflusst die Berechnungen massgeblich und muss berücksichtigt werden.

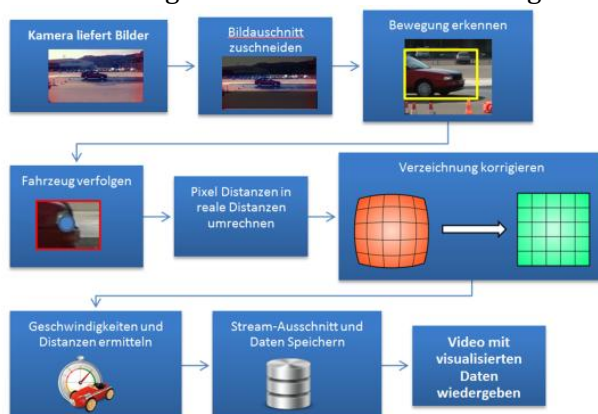


ABBILDUNG 3: ABLAUFDIAGRAMM

ERGEBNIS

Die realisierte Anwendung ist nach einer Kalibration des Benutzers in der Lage, Fahrzeuge zuverlässig zu erkennen und alle nötigen Werte genau zu errechnen. Die aus dem Bildmaterial ermittelte Geschwindigkeit hat eine Genauigkeit von ± 1 km/h. Aufgenommene Fahrten können einzeln zu einem beliebigen Zeitpunkt wieder einzeln oder zu zweit abgespielt werden, um ausgewählte Fahrten zu vergleichen. Am A-Z Verkehrszentrum wird die Anwendung ab sofort produktiv eingesetzt.

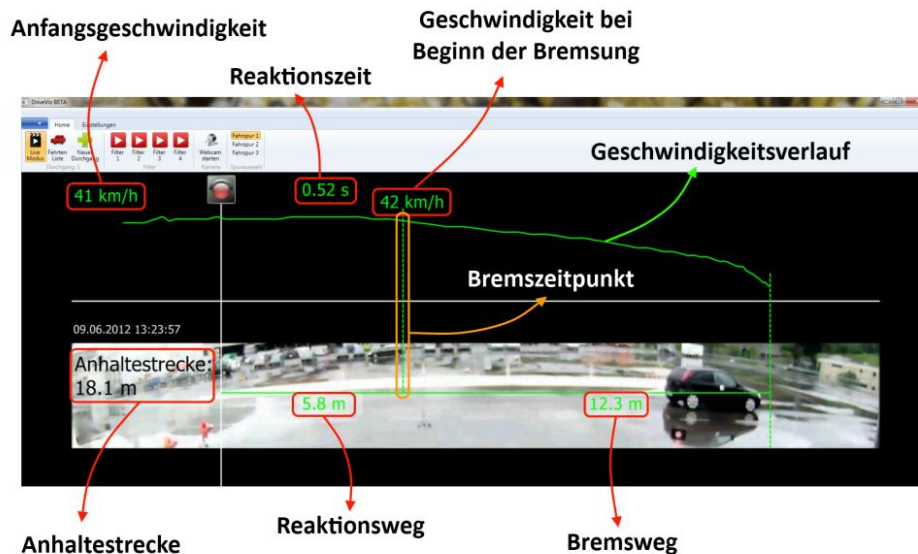


ABBILDUNG 4: AUSSCHNITT AUS DER ENDANWENDUNG MIT BESCHREIBUNG DER WERTE

MANAGEMENT SUMMARY

AUSGANGSLAGE

Neuliker von Autos müssen zwei Weiterausbildungen (WAB) Kurse absolvieren, worin unter anderem behandelt wird, wie man sich als Fahrzeugführer in verschiedenen Situationen richtig verhält.

Laut dem Bundesamt für Unfallverhütung (bfu) liegt die durchschnittliche Reaktionszeit des Fahrers in einer Notsituation bei zwei Sekunden. Folgende Grafik zeigt, zu welchen Teilen Reaktions- und Bremsweg sich die Anhaltstrecke zusammensetzt.

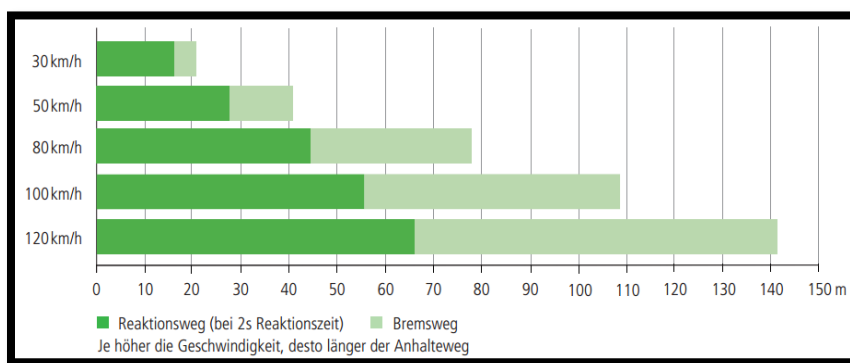


ABBILDUNG 5: STATISTIK DES BFU ZUM ANHALTEWEG

Die wenigsten Fahrtrainingszentren verfügen gegenwärtig über Daten, weshalb die Kursteilnehmer vor allem aus Beobachtungen lernen. Deshalb soll ein Programm erstellt werden, womit Vollbremsungen automatisch aufgenommen und analysiert werden. Nach der Analyse wird die aufgenommene Fahrt nochmals abgespielt und dazu zusätzlich folgende, aus dem Video errechnete Daten angezeigt:

- Geschwindigkeitsverlauf
- Reaktionszeit und Reaktionsweg
- Bremszeitpunkt und Bremsweg
- Geschwindigkeit bei Beginn der Bremsung
- Anhaltstrecke

VORGEHEN

Um von der Manöverdurchführung zu den errechneten Daten zu gelangen, werden die drei Schritte **Aufnahme**, **Analyse** und **Wiedergabe** durchlaufen, die im Folgenden beschrieben werden.

AUFNAHME

Jedes Bild, das eine installierte Netzwerkkamera liefert, wird an einen Bewegungserkennungsprozess weitergereicht. Sobald in einem definierten Bereich des Kamerabildes eine Veränderung auftritt, wird überprüft, ob die Höhe der Veränderung einen definierten Wert überschreitet und sich von links nach rechts bewegt.

Falls diese Bedingungen erfüllt sind, wird die Aufnahme gestartet und die Position des Fahrzeugs auf jedem Kamerabild erfasst. Nach einer fixen Länge von sechs Sekunden wird die Aufnahme wieder gestoppt. Damit ist die Aufnahme der Vollbremsung vollständig und sie wird an einen Analyseprozess weitergereicht.

ANALYSE

Nach der Aufnahme wird aus den erfassten Positionsdaten die zurückgelegte Strecke errechnet. Daraus können wiederum Geschwindigkeit und Beschleunigung errechnet werden. Die Beschleunigung wird benötigt um den Zeitpunkt zu bestimmen, an dem der Fahrer tatsächlich auf die Bremse tritt. Die komplett zurückgelegte Strecke lässt sich in zwei Teilstrecken klassifizieren. Zum einen in den **Reaktionsweg** und zum anderen in den **Bremsweg**.

Die errechneten Daten werden zusammen mit dem aufgenommenen Video abgespeichert, um sie zu einem beliebigen Zeitpunkt wieder abspielen zu können.

WIEDERGABE

Nachdem das Video einmalig analysiert wurde, wird es automatisch abgespielt. Zusätzlich dazu werden die errechneten Daten auf das Video gezeichnet.

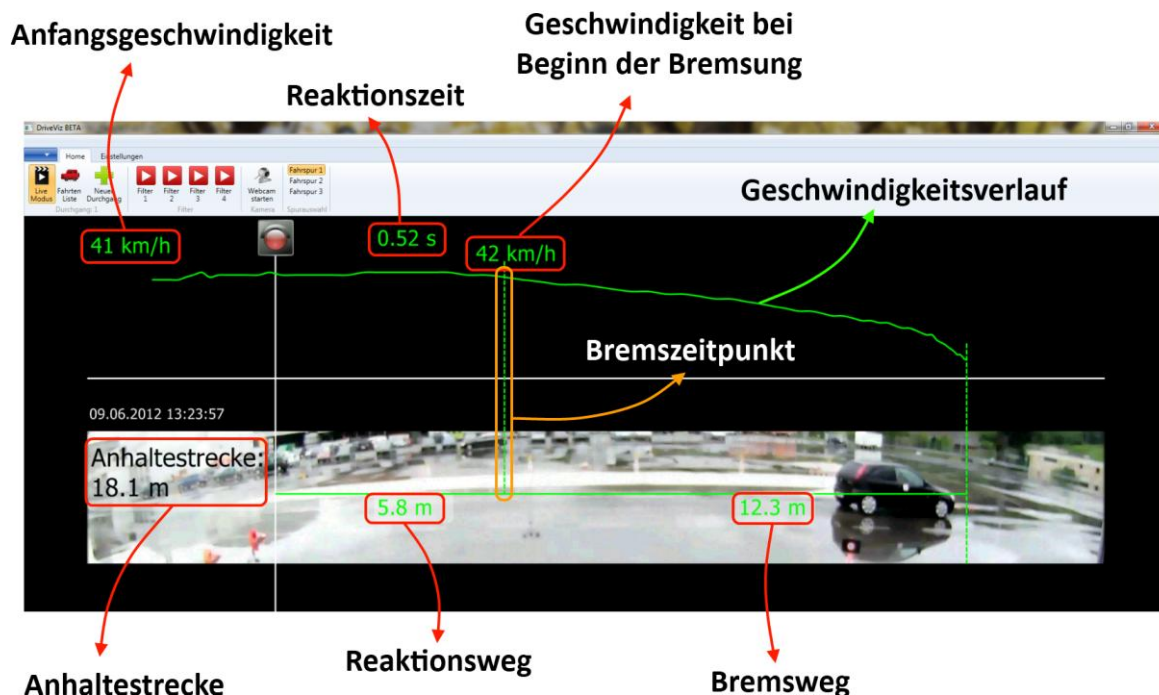


ABBILDUNG 6: BESCHRIFTETE ENDANWENDUNG

Einmalig aufgezeichnete Videos können später zu jedem beliebigen Zeitpunkt erneut abgespielt werden.

ERGEBNISSE

Das entwickelte Programm erkennt heranfahrende Autos und startet die Videoaufnahme automatisch. Die Analyse einer Videosequenz mit einer Dauer von sechs Sekunden dauert auf einem leistungsfähigen Gerät maximal 8 - 10 Sekunden.

Die errechnete Geschwindigkeit hat eine Abweichung von ± 1 Km/h. Die Anwendung wird ab sofort produktiv eingesetzt.

AUSBLICK

Der Industriepartner Rolf Bader ist an einer Weiterentwicklung interessiert und weitere Fahrtrainingszentren könnten potenzielle Kunden sein, da diese im Gespräch mit Rolf Bader auch grosses Interesse geäussert haben.

INHALTSVERZEICHNIS

Abstract.....	1
Aufgabenstellung „Aufnahme- und Visualisierungssystem für Fahrtrainingszentren“	2
Erklärung zur Urheberschaft	4
Vereinbarung zur Verwendung und Weiterentwicklung der Arbeit.....	5
Broschürentext	6
Ausgangslage.....	6
Vorgehen / Technologien	6
Ergebnis	7
Management Summary.....	8
Ausgangslage.....	8
Vorgehen.....	8
Ergebnisse.....	10
Ausblick.....	10
Inhaltsverzeichnis.....	11
Einleitung	13
Anforderungen.....	15
Passives System	15
Zeitliche Anforderung	15
Funktionsumfang	15
Evaluation	16
Gesamtüberblick.....	16
Fazit.....	17
Architektur.....	18
Design	19
Domainklassen	19
Bildverarbeitung.....	21
Persistenz	25
GUI.....	25
Tracking Algorithmus	31
Bewegung erkennen	31
Musterverfolgung.....	33
Patternmatching.....	35
Analyse	36
Von Pixel zu Meter	36
Weitwinkel und daraus entstehende Probleme.....	38

Kalkulationen.....	44
Qualitätsmanagement.....	47
Rohdatenanalyse.....	47
Testing.....	49
Exploratives Testing.....	49
Feldtests.....	50
Unittesting.....	51
Schlussfolgerung.....	52
Literaturverzeichnis.....	53
Abbildungsverzeichnis.....	53
Glossar.....	54
Personalien.....	56
Anhang.....	57
Persönliche Berichte.....	57
Entwicklungsumgebung.....	62
Datentransfer.....	62
Verwendete Bibliotheken.....	62
Klassendiagramme.....	63
Anforderungen.....	71
Poster.....	76
Projektmanagement.....	77

EINLEITUNG

Um Neulenker besser für den Strassenverkehr auszubilden, müssen zwei WAB Kurse absolviert werden. In einem der beiden werden hauptsächlich Manöver mit verschiedenen Strassenverhältnissen, bei dem anderen ökologisches Fahren trainiert.

Bei einer dieser Übungen fahren die Schüler mit vorgegebener Geschwindigkeit auf einer Trainingsbahn, die über eine Lichtschranke verfügt. Dabei wird ein Rotlicht ausgelöst, was dem Fahrer signalisiert, sofort eine Vollbremsung einzuleiten.

Das Problem dabei ist, dass der Fahrlehrer nur sehr wenige Informationen für eine Diskussionsgrundlage hat. Es ist schwer die Fahrten miteinander zu vergleichen und Unterschiede aufzuzeigen. Des Weiteren ist es schwer, Dinge wie Reaktionszeit greifbar zu machen.

Gerade Neulenkern ist oft nicht klar, wie wichtig eine schnelle Reaktion in einer Notsituation ist. Laut der Beratungsstelle für Unfallverhütung (bfu) liegt die Durchschnittliche Reaktionszeit bei 1.5 – 2 Sekunden.²

Bei einer Innerortsgeschwindigkeit von 50 km/h entspricht die Reaktionszeit einer Distanz von 21 Meter (bei 1.5 Sekunden Reaktionszeit) bis 28 Meter (bei 2 Sekunden Reaktionszeit).

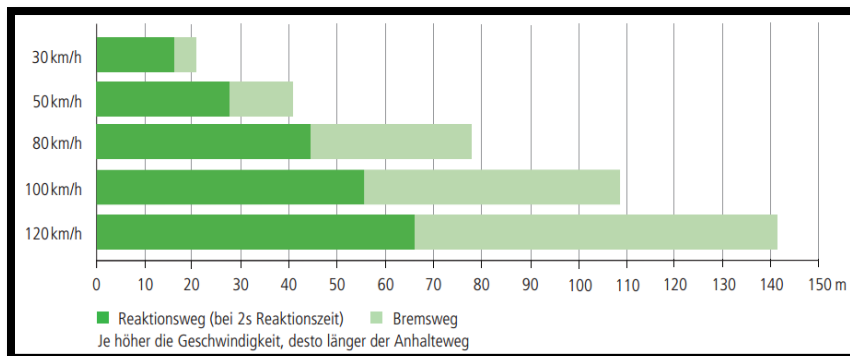


ABBILDUNG 7: REAKTIONS- UND BREMSWEG BEI VERSCHIEDENEN GESCHWINDIGKEITEN AUF TROCKENER FAHRBAHN³

Aus diesem Grund wurde ein System entwickelt, mit welchem es möglich ist, diese Diskussionsgrundlage zu schaffen. Dazu werden die Fahrten aufgezeichnet und mittels Videoanalyse werden Geschwindigkeitsverlauf, Reaktionszeit, Reaktions- und Bremsweg und Anhaltestrecke berechnet und auf dem Video dargestellt.

² <http://www.bfu.ch/German/lexikon/Documents/Kurzinfo%20Reaktionszeit.pdf>, 04.06.2012

³ http://www.bfu.ch/pdflib/1047_42.pdf, 04.06.2012



ABBILDUNG 8: SCREENSHOT

ZIELPUBLIKUM

Die nächsten beiden Kapitel erläutern, wieso man sich für Videoanalyse entschieden hat und nicht eine andere Technik verwendet.

Die nachfolgenden Kapitel richten sich an Leute, die sich für die technischen Details und die Funktionsweise interessieren.

Es wird einerseits konzeptionell gezeigt, wie das Tracking funktioniert und daraus Geschwindigkeit, Anhaltestrecke und Reaktionszeit abgeleitet werden, andererseits aber auch, wie die Software als solches konzipiert ist.

ANFORDERUNGEN

PASSIVES SYSTEM

Aus den Gesprächen mit dem Kunden ging hervor, dass es sich zwingend um ein passives System handeln muss. Das heisst, dass das System ohne jegliche Änderungen auskommen muss, wie das Anbringen eines GPS Gerätes am Auto, Anforderungen wie Mindestgrösse eines Autos oder eine spezielle Farbe. Das hat damit zu tun, dass sich das System im täglichen Gebrauch befinden wird und es einfach nicht praktikabel ist, jedem Teilnehmer einen GPS Sensor zu geben, wieder einzusammeln und zu warten.

Aus diesem Grund hat man sich dafür entschieden, die Problemstellung mit Hilfe von Video- und Bildanalyseverfahren zu lösen. Da dazu lediglich eine Kamera mit Weitwinkelobjektiv nötig ist, die die ganze aufzunehmenden Fahrstrecke abdeckt, hält sich der Installationsaufwand in Grenzen und die oben genannten Probleme mit den GPS Geräten fallen gänzlich weg.

Man konnte sich darauf einigen, dass es, sofern nötig, erlaubt ist, eine kleine Magnetscheibe (maximal 20x20 cm) mit vordefiniertem Muster am Fahrzeug anzubringen, um in der Videoanalyse das Fahrzeug gut zu erkennen.

ZEITLICHE ANFORDERUNG

Ein weiterer wichtiger Aspekt ist die zeitliche Abhängigkeit. Die Auswertung der Fahrt muss umgehend nach dem durchgeführten Bremsmanöver zur Verfügung stehen. Man konnte sich auf eine maximale Dauer von 30 Sekunden einigen.

Messungen haben ergeben, dass auf dem verwendeten System für einen Bremsvorgang mit einer Dauer von sechs Sekunden, ungefähr acht Sekunden Verarbeitungszeit benötigt werden, bis die Analyseergebnisse angezeigt werden.

Sechs Sekunden ist die Dauer, die maximal für einen Bremsvorgang auf dem Testgelände in Gosau benötigt wird.

FUNKTIONSUMFANG

Genau aufgelistete funktionale Anforderungen an das System befinden sich im Anhang.

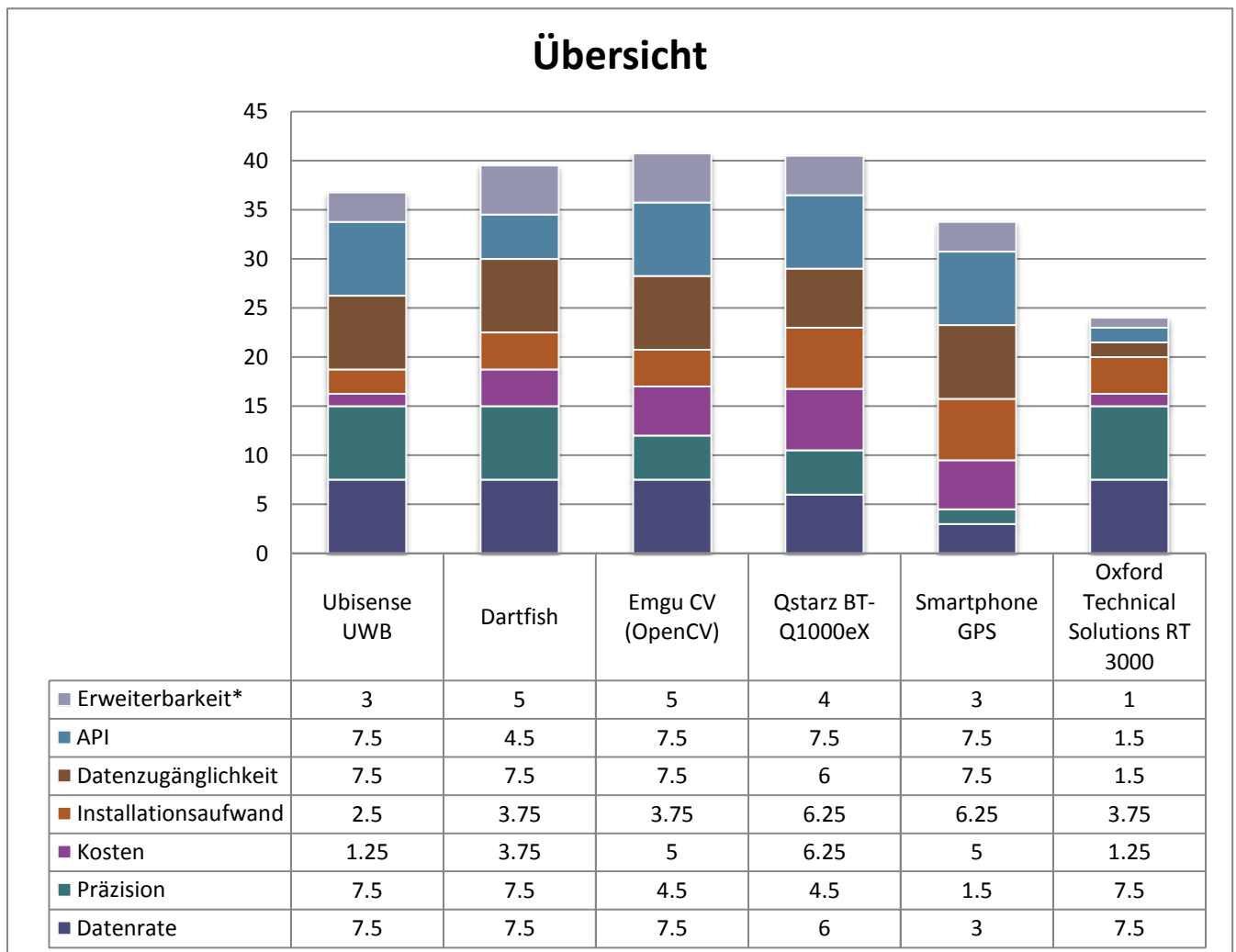
EVALUATION

In Rahmen dieser Arbeit war nicht klar, mit welcher Technologie die Problemstellung gelöst werden soll. Aus diesem Grund wurde in einem ersten Schritt recherchiert und verschiedene Möglichkeiten evaluiert. Zu diesem Zeitpunkt waren die Anforderungen jedoch noch kaum spezifiziert und es war nicht bekannt, dass es sich zwingendermassen um ein passives System handeln muss.

Wäre dies schon früher bekannt gewesen, hätte man sich auf Produkte aus der Videoanalyse beschränken können.

GESAMTÜBERBLICK

Nachfolgend werden die einzelnen Bewertungen im Überblick dargestellt.⁴



⁴ Genau Auswertung im Kapitel: Evaluations Details

FAZIT

Die zwei Favoriten sind also Emgu CV und Dartfish. Wobei beide auf der Basis der Video- und Bildanalyse beruhen und somit das Kriterium eines passiven Systems erfüllen.

Abklärungen haben ergeben, dass Dartfish dennoch ungeeignet ist, da kaum Entwicklungsschnittstellen für Programmierer vorhanden sind und Dartfish in sich geschlossen ist.

ARCHITEKTUR

Beim System handelt es sich um eine .NET 4.0 C# WPF Applikation, welche auf einem Windows 7 Rechner installiert wird. Dieser hat Zugang zur eingesetzten Netzwerkkamera.

Die Informationen, die aus der Videoanalyse gewonnen werden, werden in einer db4o-Datenbank⁵ direkt auf dem Rechner gespeichert. Die einzelnen Aufnahmen selbst werden aus dem Livestream extrahiert und auf der Festplatte abgelegt.

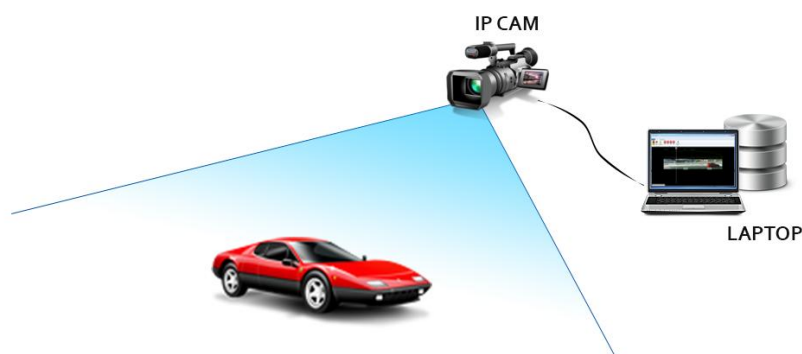
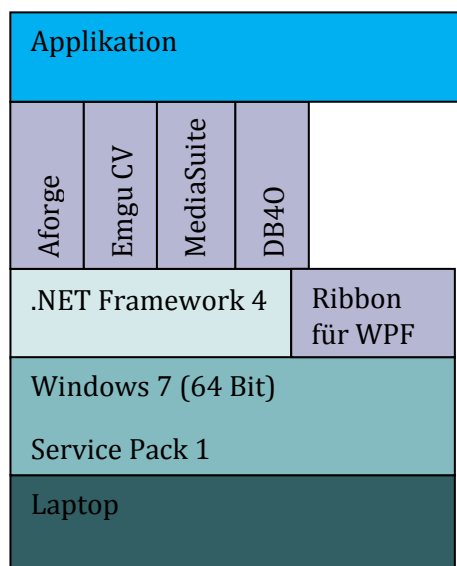


ABBILDUNG 9: ARCHITEKTUR

SYSTEMDIAGRAMM



Produkt

Im Umfang des Produkts ist die Software und die verwendeten Bibliotheken.

MINIMALE SYSTEMANFORDERUNGEN⁶:

Betriebssystem: Windows 7 Service Pack 1
Systemtyp: 64 Bit-Betriebssystem
Prozessor: Intel® Core™ i7-2620M CPU @ 2.70 GHz
Arbeitsspeicher: 8.00 GB Ram

Voraussetzung

NÖTIGE VORINSTALLIERTE SOFTWARE

Microsoft .Net Framework Version 4
 Microsoft Ribbon für WPF Version 4⁷

⁵ <http://www.db4o.com/>, 07.06.2012

⁶ Mit diesem System wurde die Applikation erfolgreich getestet, es kann durchaus sein, dass es auch auf einem schwächeren System läuft.

⁷ <http://www.microsoft.com/en-us/download/details.aspx?id=11877>, 07.06.2012

DESIGN

Die folgenden Kapitel erläutern das Softwaredesign und die wichtigsten Klassen, sowie ihren Zweck.

Die kompletten Klassendiagramme befinden sich im Anhang.

DOMAINKLASSEN

Die Informationen, die man mittels der Klasse **TripAnalyzer** gewinnt, werden in den Domainklassen abgelegt.

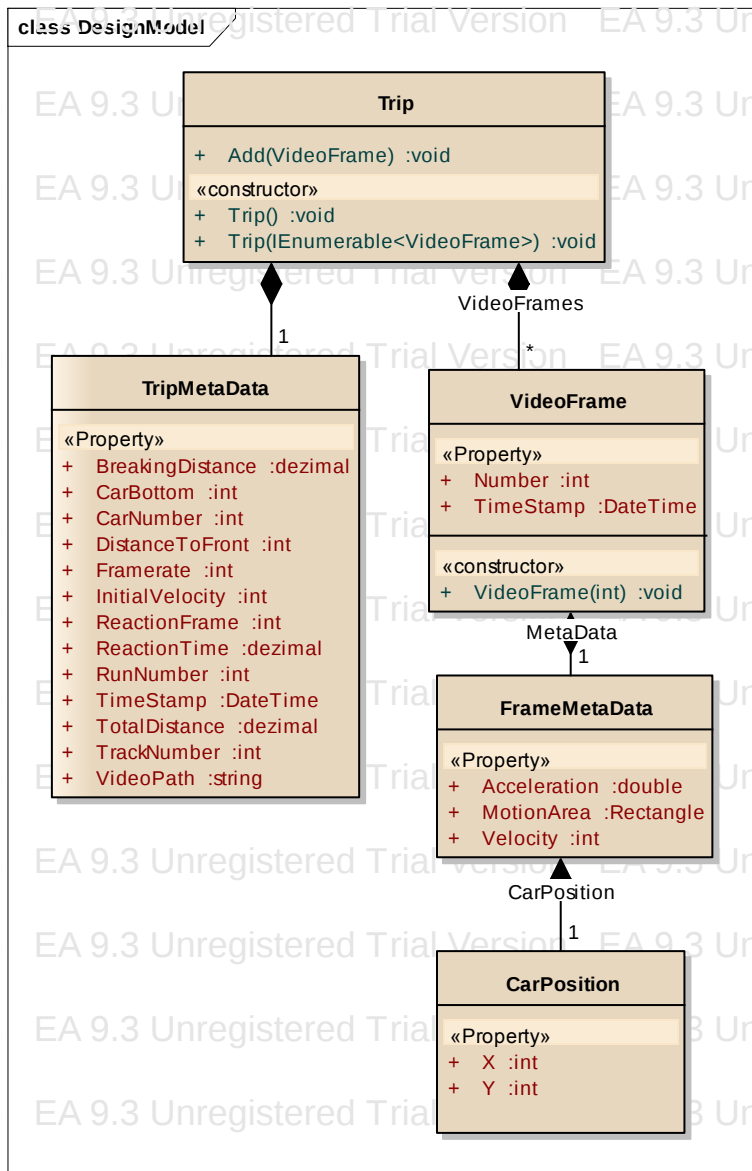


ABBILDUNG 10: DOMAINKLASSEN

Trip	Zu jeder aufgenommen Fahrt besteht ein Trip-Objekt. Ein Trip besteht aus einer Menge von Videoframes und Metainformationen über den Trip.
VideoFrame	Zu jedem Bild einer aufgenommen Fahrt besteht ein VideoFrame-Objekt. Es enthält sowohl Verarbeitungszeitpunkt, Nummer und weitere Metainformation
TripMetaData & FrameMetaData	Sowohl der Trip, als auch das VideoFrame haben ein MetaData-Objekt, auf welchem Werte für die weitere Berechnung und berechnete Werte gespeichert werden können.

BILDVERARBEITUNG

Das folgende Bild (Abbildung 11: Videohandlergrobkonzept) zeigt ganz grob, wo sich die Komponenten befinden und wie sie zusammen arbeiten.

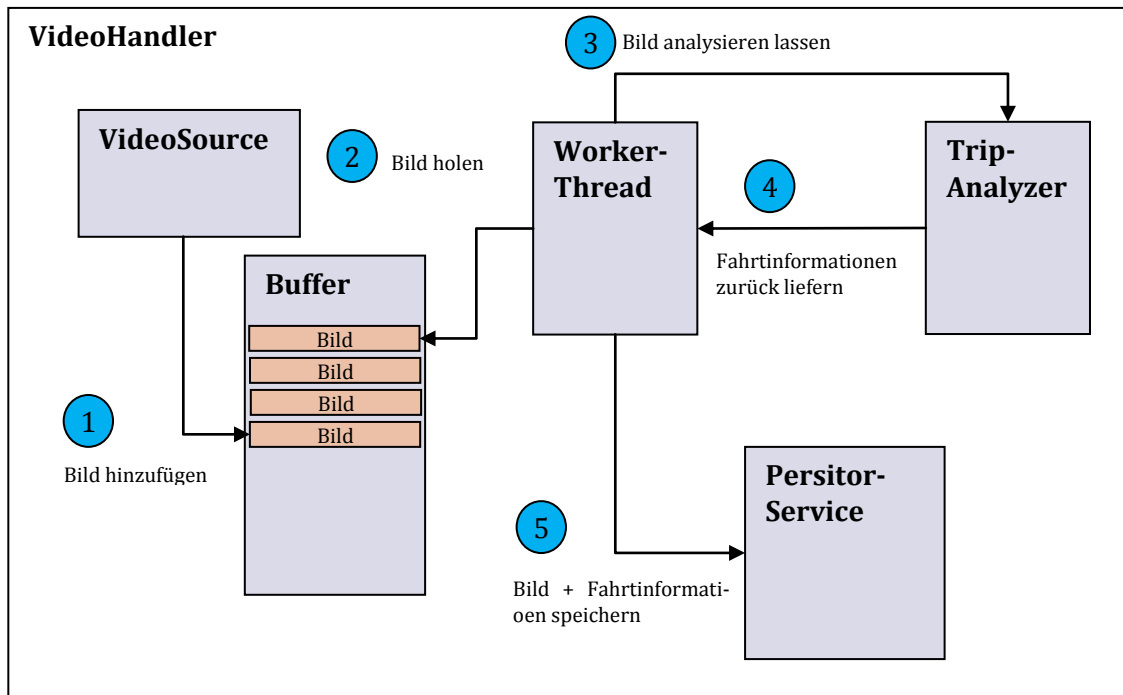


ABBILDUNG 11: VIDEOHANDLERGROBKONZEPT

1. Die Videosource fügt neue Bilder in den Buffer ein.
2. Der WorkerThread holt Bild für Bild wieder aus dem Buffer.
3. Lässt die Bilder vom TripAnalyzer analysieren
4. Erhält entsprechend Informationen zurück
5. Persistiert sowohl Bild, als auch die gewonnen Informationen mittels des PersistorServices ab.

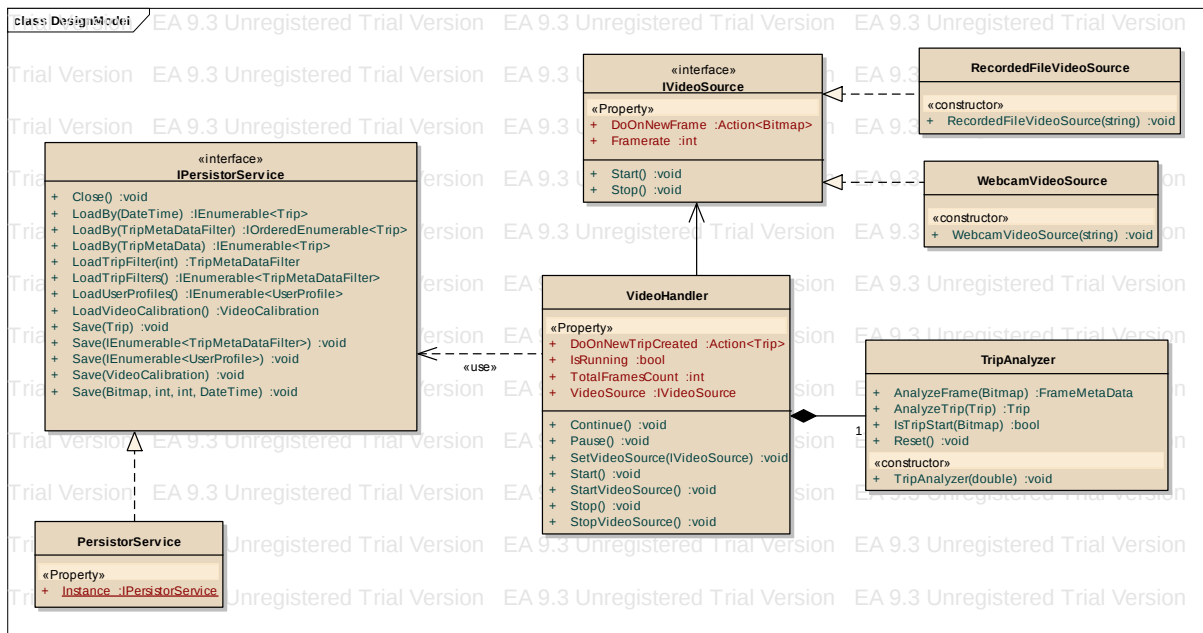


ABBILDUNG 12: VIDEOHANDLING

VideoHandler Die Klasse *Videohandler* verwaltet die zu verarbeitenden Bilder.

Auf dem *Videohandler* kann jeder Zeit eine andere *VideoSource* gesetzt werden. Beim Setzen einer *VideoSource* registriert der *Videohandler* seine Action, die dann aus der *VideoSource* mit dem jeweils neuesten Bild aufgerufen wird. Bilder, die der *VideoHandler* erhält, werden in einem Buffer gespeichert und von einem Workerthread innerhalb des *VideoHandlers* verarbeitet.

Da die Analyse der Bilder sehr zeitintensiv ist, wird nur jedes zweite Bild aus dem Buffer zur Detektion einer neuen Fahrt verwendet und das andere Bild verworfen. Sobald mittels des *TripAnalyzer* festgestellt wird, dass eine Fahrt beginnt, wird wieder die volle Bildrate verarbeitet.

Dies ist nötig, da es auf gängigen Rechnern nicht möglich ist, die Bilder gleich schnell oder schneller zu verarbeiten, als die *VideoSource* die Bilder liefert. Andernfalls wäre eine Überfüllung des Buffers das Resultat.

Nachdem detektiert wurde, dass eine neue Fahrt gestartet hat, werden die Bilder der nächsten drei Sekunden dem *TripAnalyzer* zur Verarbeitung übergeben.

Die mittels dem *TripAnalyzer* gewonnen Metainformationen, wie beispielsweise Geschwindigkeit und Bremsstrecke, werden mit Hilfe der Klasse *PersistorService* gespeichert.

IVideoSource *IVideoSource* ist ein Interface für eine Bildquelle. Dabei ist es möglich, die Quelle zu starten, wieder zu beenden und eine Delegate-Methode (*DoOnNewFrame*) zu registrieren, die jedes Mal aufgerufen wird, wenn die Video-Quelle ein neues Bild für den *VideoHandler* hat.

Es gibt zwei verwendete Implementationen:

- **WebcamVideoSource**, welche einen RTSP-Stream öffnet. Diese Implementation basiert auf der MediaSuite-Bibliothek von Streamcoders.⁸
- **RecordedFileVideoSource**, welche es ermöglicht eine Videodatei (z.B. im Format WMV) zu verarbeiten.

Zusätzlich gibt es eine alternative *VideoSource*, für die in diesem Projekt eingesetzte Netzwerkkamera der Firma Vivotek.

VivotekVideoSource, diese *VideoSource* basiert auf der ActiveX Komponente des SDKs von Vivotek. Wenn möglich sollte die Source NICHT verwendet werden, sondern die *WebcamVideoSource*. Diese ist performanter und funktioniert nicht nur mit Kameras der Marke Vivotek, sondern Dank Unterstützung von RTSP mit jeder Kamera, die RTSP ebenfalls unterstützt.

TripAnalyzer

Der *TripAnalyzer* bietet zwei Hauptfunktionen:

AnalyzeFrame(...) und *AnalyzeTrip(...)*

- **AnalyzeFrame(...)** gibt zu einem Bild ein *FrameMetaData* Objekt zurück, welches Grundinformationen wie beispielsweise die Position des Fahrzeuges oder den Bewegungsbereich enthält. Diese Informationen werden danach im Kontext eines *TripMetaData*-Objekts wieder verwendet, um Geschwindigkeit, Bremsweg und weiteres zu berechnen. Um diese Informationen zu ermitteln werden die Klassen *TripDetector* und *CarTracker* verwendet.
 - **AnalyzeTrip(...)** wird ein Trip-Objekt übergeben. Diese Methode reicht durch die bestehenden Informationen, die über *AnalyzeFrame* gewonnen wurden, den Trip, wie auch die einzelnen Frames, mit weiteren Meta-Informationen an.
-

Das folgende Sequenzdiagramm (Abbildung 13: Sequenzdiagramm Videohandler) zeigt konzeptionell den Ablauf zwischen den verschiedenen Teilkomponenten. Dabei gibt es drei Teile. Als erstes werden die Komponenten instanziiert und initialisiert. Danach wird der Buffer des Videohandlers kontinuierlich mit neuen Bildern eingespeist, welche vom Worker wieder ausgelesen und verarbeitet werden. Die Methoden sind nicht immer eins zu eins so im Code zu finden, sondern dienen mehr dem Verständnis. Die Nummern eins bis fünf, die in *Abbildung 11: Videohandlergrobkonzept* zu sehen sind, befinden sich auch im Sequenzdiagramm.

⁸ <http://www.streamcoders.com/products/msnet.html>, 04.06.2012

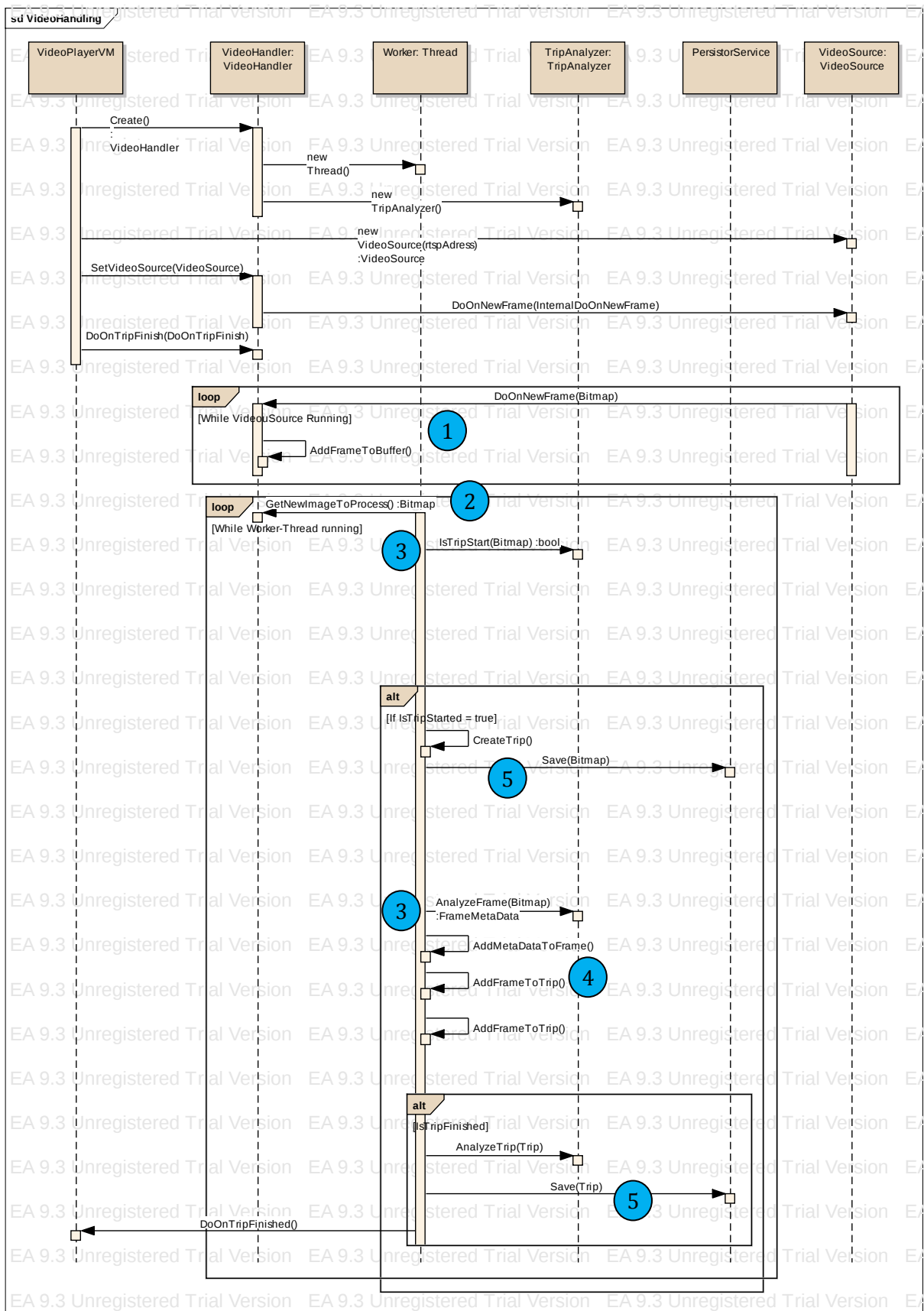


ABBILDUNG 13: SEQUENZDIAGRAM VIDEOHANDLER

PERSISTENZ

PersistorService	Um die Daten zu speichern, steht der <i>PersitorService</i> zur Verfügung. Mit Ausnahme von Videoaufnahmen von Fahrten werden alle Daten in einer db4o Datenbank ⁹ gespeichert und wieder davon ausgelesen. Videos hingegen werden direkt auf der Festplatte des Systems abgespeichert.
-------------------------	---

GUI

Für die Benutzeroberfläche gibt es die drei folgenden Klassen, die im Programm an verschiedenen Stellen verwendet werden.

ViewModelBase	<i>ViewModelBase</i> ist die Basisklasse für alle <i>ViewModels</i> . Sie implementiert das Interface <i>INotifyPropertyChanged</i> , womit die ableitenden Klassen über die Methode <i>FirePropertyChanged(propertyName)</i> das GUI aktualisieren können. <i>ViewModelBase</i> hat ein Property vom Typ <i>SynchronizationContext</i> , dem beim Instanzieren der Klasse <i>ViewModelBase</i> die Instanz <i>System.Threading.SynchronizationContext.Current</i> zugewiesen wird. Bei jedem Aufruf des <i>PropertyChangedEventHandler</i> wird überprüft, ob der aufrufende <i>SynchronizationContext</i> derselbe wie im instanziierten <i>ViewModel</i> ist. Falls das nicht so ist, wird der Aufruf über den <i>SynchronizationContext</i> des <i>ViewModels</i> geleitet. Würde das nicht gemacht werden, hätte ein Aufruf des EventHandlers aus einem anderen Thread eine Exception zur Folge, da das GUI nur aus dem Main Thread der Applikation manipuliert werden darf.
SelectionModel	Das <i>SelectionModel</i> ist auch ein <i>ViewModel</i> und enthält Programmweite GUI Selektionen, die an mehreren verschiedenen Stellen gebraucht werden, wie beispielsweise das aktuell ausgewählte Benutzerprofil.
InvariantModel	Das <i>InvariantModel</i> enthält Objekte, die einmal beim Programmstart geladen werden und dann an verschiedenen Stellen verfügbar sein müssen. Durch die Kapselung im <i>InvariantModel</i> kann auf einfache Art und Weise sichergestellt werden, dass keine unnötigen Datenbankzugriffe gemacht werden.
RootModel	Das <i>RootModel</i> enthält je eine Instanz vom <i>Invariant-</i> und <i>SelectionModel</i> . Die Klasse <i>RootModel</i> ist als Singleton instanziiert, was es erlaubt, aus dem XAML direkt darauf zu binden.

⁹ <http://www.db4o.com/>, 07.06.2012

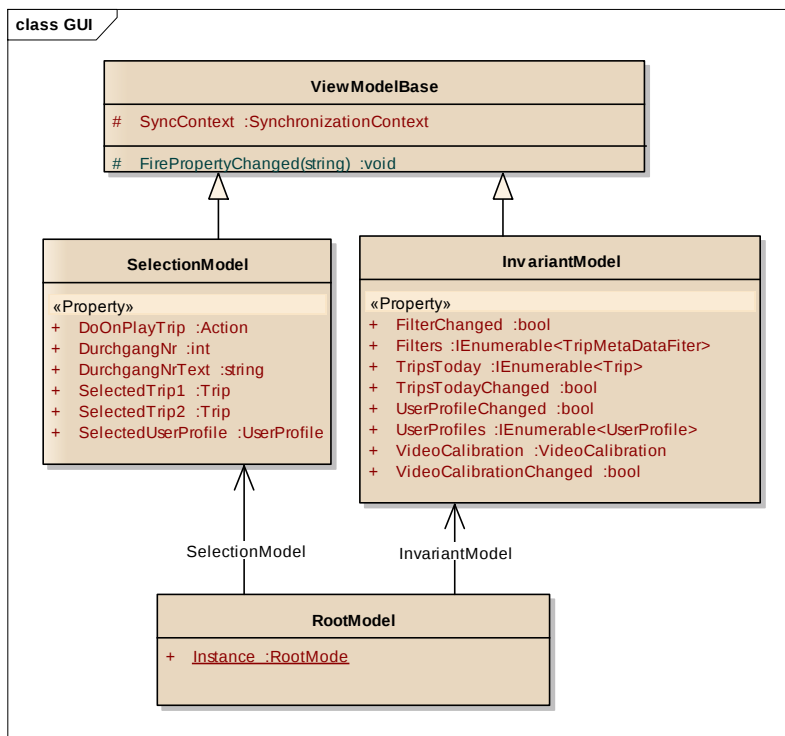


ABBILDUNG 14: VERSCHIEDENE MODELS

USER CONTROLS

Nachfolgend werden die verschiedenen *UserControl*-Klassen für die Benutzeroberfläche aufgeführt und deren Zweck und Funktionsweise erklärt.

MainWindow

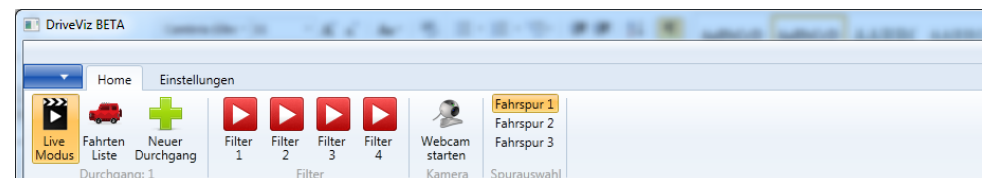


ABBILDUNG 15: NAVIGATIONSLEISTE

Das Hauptfenster der Applikation. Durch die Navigationsleiste gelangt man zum Live Modus, zur Fahrtenliste und zu den Einstellungen.

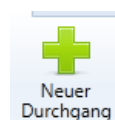


ABBILDUNG 16: DURCHGANGSNUMMER ERHÖHEN

Mit dem Knopf „Neuer Durchgang“ hat der Benutzer die Möglichkeit, Fahrten in logische Gruppen zu gliedern. Die aktuelle Durchgangsnummer wird jeder neu aufgenommenen Fahrt zugewiesen. Dadurch hat der Benutzer bei der Filterung der Fahrten die Möglichkeit, nach seiner logischen Gruppierung zu filtern.

Die Durchgangsnummer ändert sich nur, wenn sie vom Benutzer manuell

erhöht wird. Wird das Programm beendet und am selben Tag wieder gestartet, ist die aktuelle Durchgangsnummer die höchste, auf die am selben Tag bereits erhöht wurde. Wird das Programm am nächsten Tag wieder gestartet, ist die Durchgangsnummer wieder bei eins. Die jeweils aktuelle Durchgangsnummer wird in der Mitte unter den drei Knöpfen „Live Modus“, „Fahrten Liste“ und „Neuer Durchgang“ angezeigt.

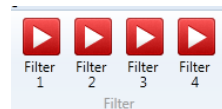


ABBILDUNG 17: KONFIGURIERTE FILTER

Durch die vier konfigurierbaren Filterknöpfe besteht die Möglichkeit, bereits aufgenommene Fahrten sehr einfach abzuspielen. Wie Filter konfiguriert werden, ist beim *UserControl FilterCtrl* beschrieben.

Die Filterung, welche durch die vier Filterknöpfe gemacht wird, kann auf zweierlei Arten genutzt werden:

1. **Der *Live Modus* ist gewählt:** Wird einer der Filterknöpfe angeklickt, geschieht die Filterung und es werden automatisch die zwei obersten Fahrten des Resultats (eine Liste mit Fahrten) der Filterung abgespielt. Liefert die Filterung nur ein Resultat, wird nur dieses abgespielt. Hier hat der Benutzer also keinen weiteren Einfluss auf die Auswahl der Fahrten.
2. **Die *Fahrten Liste* ist geöffnet:** Wird einer der Filterknöpfe betätigt, wird das Resultat der Filterung in der Fahrtenliste angezeigt. Der Benutzer kann nun eine oder zwei Fahrten davon manuell auswählen und diese abspielen lassen. Hier kann also eine weitere Unterauswahl neben der bereits gemachten Filterung getroffen werden.

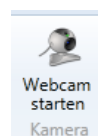


ABBILDUNG 18: WEBCAM EIN- UND AUSSCHALTEN

Durch diesen Knopf kann die in den Einstellungen konfigurierte Webcam gestartet, oder falls sie bereits läuft, beendet werden.

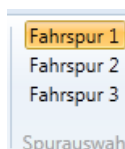


ABBILDUNG 19: SPURAUSSWAHL

Durch diese Knöpfe wird definiert, auf welcher Spur gefahren wird.

VideoPlayerCtrl

Das *VideoPlayerCtrl* ist das *Control* für das Abspielen neuer oder bereits vorhandener Fahrten.

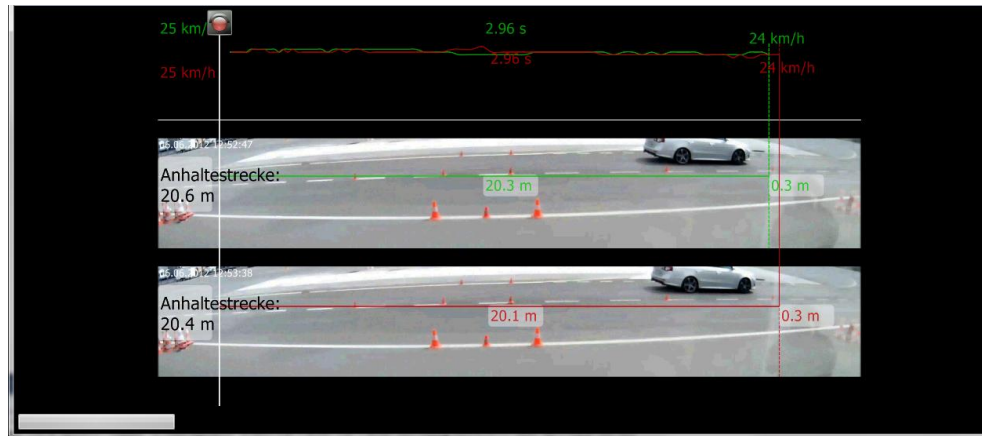


ABBILDUNG 20: ABSPIELEN ZWEIER FAHRTEN

TripListCtrl

Hier werden die Fahrten des aktuellen Tages aufgelistet, wovon der Benutzer eine oder zwei Fahrten per Mausklick auswählen und dann abspielen kann. Durch das Abspielen wechselt das Programm auf das *VideoPlayerCtrl*.

Auf diesem *Control* hat der Benutzer auch die Möglichkeit, an verschiedenen Werten Änderungen vorzunehmen, die dann automatisch gespeichert werden. Folgende Werte können manuell geändert werden:

- Autonummer
- Zustand der Fahrspur (nass oder trocken)
- Nummer der Fahrspur

Zeitpunkt	Durchgang	Autonummer	Geschwindigkeit	Anhaltstrecke	Bremsweg	Reaktionszeit	Zustand	Spurnummer	Video
06.06.2012 12:53 1	0	25 km/h	20.4 m	0.3 m	3.0 s			0	E:\Cartracking\Videos\201266_125327.avi
06.06.2012 12:52 1	0	25 km/h	20.6 m	0.3 m	3.0 s			0	E:\Cartracking\Videos\201266_125239.avi
06.06.2012 12:42 1	0	24 km/h	20.1 m	0.3 m	3.0 s			0	E:\Cartracking\Videos\201266_124221.avi
06.06.2012 12:40 1	0	24 km/h	20.1 m	0.3 m	3.0 s			0	E:\Cartracking\Videos\201266_124024.avi
06.06.2012 12:22 1	0	25 km/h	20.6 m	0.3 m	3.0 s			0	E:\Cartracking\Videos\201266_122238.avi
06.06.2012 11:04 1	0	26 km/h	12.6 m	3.7 m	1.2 s			0	E:\Cartracking\Videos\201266_110458.avi
06.06.2012 11:04 1	0	25 km/h	20.4 m	0.3 m	3.0 s			0	E:\Cartracking\Videos\201266_110435.avi
06.06.2012 10:55 1	0	26 km/h	6.8 m	-1.0 m	1.1 s			0	E:\Cartracking\Videos\201266_105552.avi
06.06.2012 10:51 1	0	21 km/h	4.7 m	0.0 m	0.8 s			0	E:\Cartracking\Videos\201266_105109.avi
06.06.2012 10:49 1	0	24 km/h	20.1 m	0.3 m	2.9 s			0	E:\Cartracking\Videos\201266_104910.avi
06.06.2012 10:48 1	0	78 km/h	63.9 m	0.8 m	1.9 s			0	E:\Cartracking\Videos\201266_104807.avi

ABBILDUNG 21: AUFLISTUNG DER FAHRTEN DES AKTUELLEN TAGES

FilterCtrl

Auf dem *FilterCtrl* kann der Benutzer vier verschiedene Filter definieren. Er kann dazu verschiedene fakultative Werte als Ober- und Untergrenze festlegen. Beim Filtern werden alle Fahrten angezeigt, welche die festgelegten Werte alle erfüllen. Zusätzlich kann eine Sortierreihenfolge angegeben werden, nach der das Resultat sortiert werden soll.

Durch einen Klick auf den Knopf mit den zwei Pfeilen, wird der Filter ausgewertet und das Resultat wird sofort angezeigt. Nun kann der Filter weiter konfiguriert oder mit einem Klick auf die Diskette abgespeichert werden.

Der Filterknopf auf der Navigationsleiste hat diese Konfiguration automatisch übernommen und filtert nach dieser.

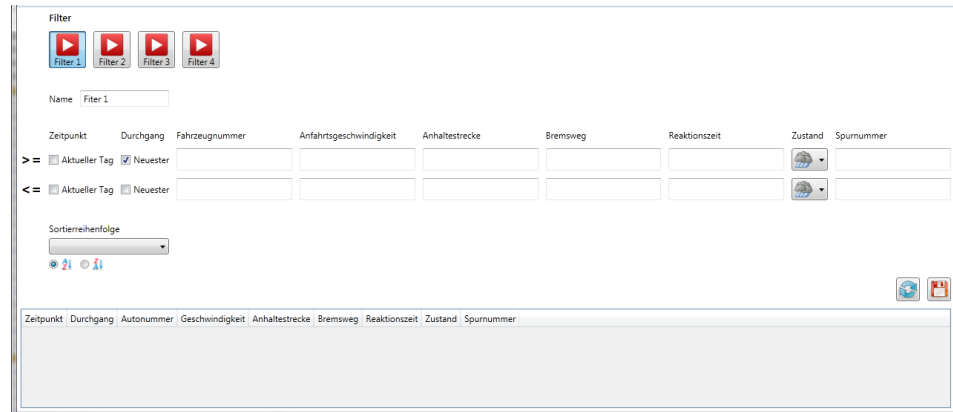


ABBILDUNG 22: FILTER KONFIGURIEREN

Configurati- onCtrl

Durch die Navigationsleiste gelangt man über den Reiter "Einstellungen" und dann den Knopf "Globale Einstellungen" zum *Control*, womit Benutzerprofile konfiguriert werden. Standardmässig existiert nur ein Profil.

Für die Fahrten die abgespielt werden, können verschiedene Farben definiert werden, um je nach Situation (Lichteinfall, Tageszeit) den besten Kontrast auf dem Anzeigegerät zu erreichen.

Durch einen Haken bei „Algorithmus anzeigen“, werden beim Abspielen einer Fahrt Videoanalytische Daten, wie Bewegungsbereich, Trackingpunkt und Positionsverlauf des Trackingpunktes angezeigt.

Als Kamera URL kann eine RTSP Adresse einer Webcam angegeben werden. Durch den Knopf mit dem grünen Haken wird die Adresse überprüft. Kann eine Verbindung aufgebaut werden, wird für eine Dauer von fünf Sekunden das Kamerabild live angezeigt, danach trennt sich die Verbindung automatisch. Kann keine Verbindung aufgebaut werden, erscheint eine Fehlermeldung.

Eine Bildrate kann manuell eingetragen oder automatisch ermittelt werden. Bei der automatischen Ermittlung können folgende Bildraten erkannt werden: 20, 25, 30, 50 und 60 Bilder pro Sekunde.

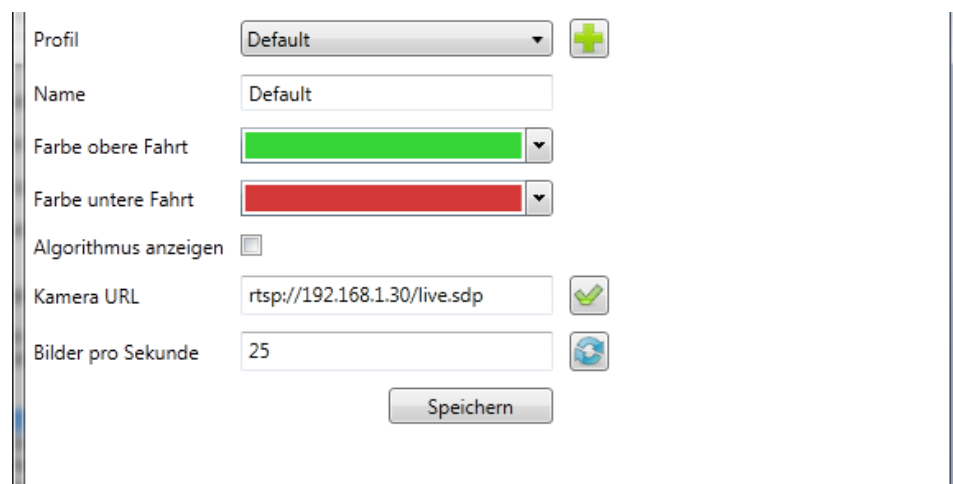


ABBILDUNG 23: GLOBALE EINSTELLUNGEN

Kalibration

Damit Geschwindigkeiten und Distanzen errechnet werden können, muss das aktuelle Kamerabild kalibriert werden. Dazu gehören folgende Schritte:

- Bereich auswählen, in dem Autos detektiert werden sollen.
- Linie festlegen, bei der ein Signal für die Vollbremsung ausgelöst wird.
- Zwei Distanzen auf unterschiedlichen Höhen vermessen.
- Höhe von 3 Spuren festlegen, auf welchen die Fahrzeuge fahren.

Mit dem Knopf „Neues Bild“ wird ein neues Bild von der Kamera geholt, worauf man einen Bereich von 1920x300 Pixeln auswählen kann. Die vertikale Position dieses Bereichs wird gespeichert.

Vom Videostream der Kamera wird dann jedes Bild auf dieser Höhe auf diese Grösse zugeschnitten, um einerseits die zu verarbeitende Datenmenge zu reduzieren und andererseits, damit beim Abspielen einer analysierten Fahrt auch zwei Videos untereinander Platz haben.

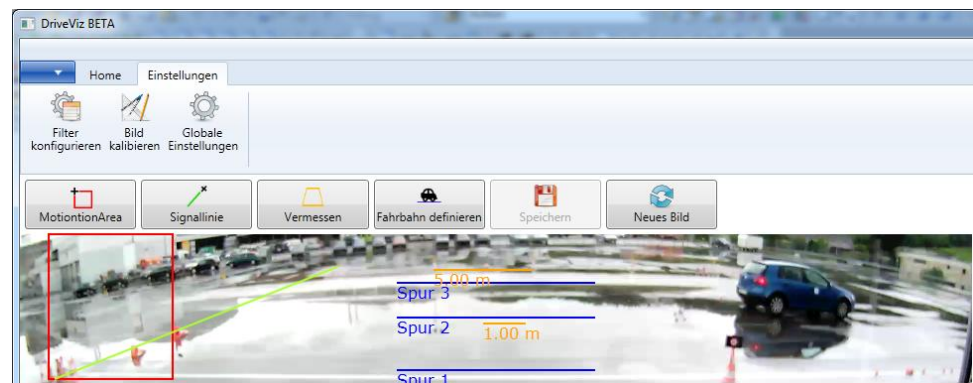


ABBILDUNG 24: BILDKALIBRATION

TRACKING ALGORITHMUS

In diesem Abschnitt wird erläutert, wie das System ein Auto erkennt und weiter verfolgt.

Der Algorithmus besteht grob aus zwei Teilen. Als erstes muss in einem vordefinierten Bereich eine Bewegung erkannt werden, um in einem zweiten Schritt dieses Objekt, welches als bewegtes Objekt identifiziert wurde, weiter zu verfolgen.

BEWEGUNG ERKENNEN

Die Bewegungserkennung hat zwei Ziele. Das erste ist festzustellen, wo sich ein Objekt bewegt, damit dann ein Muster davon erstellt werden kann, welches genau mit verfolgt werden kann (mehr dazu weiter unten).

Das zweite Ziel ist es, den untersten Punkt des Fahrzeugs (Punkt auf dem Bild wo die Räder die Strasse berühren) und die vorderste Stelle des Autos festzustellen. Dies geht am besten, wenn das Objekt fast still steht. Also kurz vor dem Ende des Bremsvorgangs.



ABBILDUNG 25: BEWEGUNGERKENNUNG

Als erstes wird ein grober Bereich definiert, innerhalb dessen die Bewegungserkennung das erste Mal eine Bewegung bzw. Bildveränderung erwartet (**roter Rahmen**). Diese Information und die einzelnen Videoframes können der Aforge-Bibliothek übergeben werden, welche die Frames auswertet und die Koordinaten des bewegten Teils des Bildes zurück liefert (**gelber Rahmen**). Dieser Rahmen kann nun mit dem der letzten Bewegung verglichen werden und wenn er sich von links nach rechts bewegt sowie eine bestimmte Minimalgrösse aufweist, kann davon ausgegangen werden, dass es sich bei der Bewegung um ein Fahrzeug handelt, welches vor die Kamera fährt.

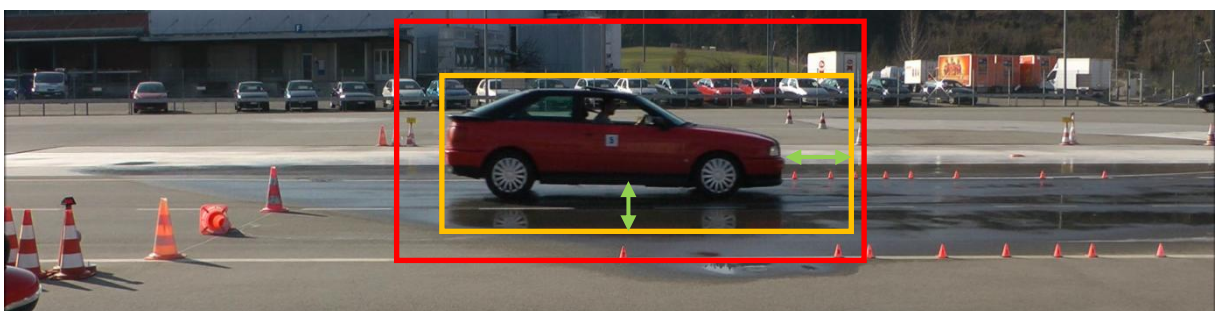


ABBILDUNG 26: SCHNELLE BEWEGUNG UND RESULTIERENDE ABWEICHUNG DER BEWEGUNGERKENNUNG



ABBILDUNG 27: LANGSAME BEWEGUNG

Nun, da die Position und ungefähre Grösse des Fahrzeugs bekannt sind, muss man den Rahmen, in welchem die Bewegungserkennung sucht, mit dem bewegten Objekt mitverschieben, damit keine falschen Objekte verfolgt werde, wie beispielsweise Fahrzeuge, die nach dem Bremsvorgang zurückfahren oder Personen, die auf dem Platz herum gehen.

Dieser gelbe Rahmen ist bis kurz vor dem Ende des Manövers ungenau in der Grösse (siehe Abbildung 26: Schnelle Bewegung) und nimmt erst gegen Ende, wenn sich das Fahrzeug nur noch sehr langsam bewegt, genau dessen Grösse an.

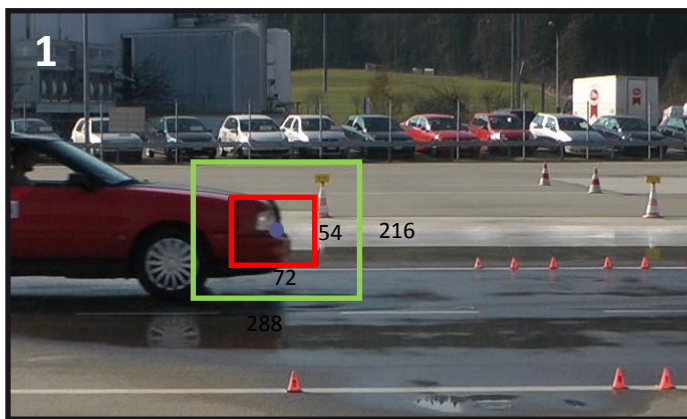
Zu diesem Zeitpunkt ist es dann möglich, die Spitze des Fahrzeugs, respektive die Distanz vom Trackingpunkt zur Spitze des Fahrzeuges, zu ermitteln.

Wie weiter oben erwähnt, war ursprünglich geplant, dass mittels dieses Verfahrens der Boden des Fahrzeugs gefunden werden kann. Tests haben aber gezeigt, dass Einflüsse wie Schatten, Spiegelungen im Wasser, sowie die Verzeichnung durch die eingesetzte Weitwinkelkamere zu viele Ungenauigkeiten mit sich bringen. Aus diesem Grund hat man sich entschieden, diese Information, manuell über das GUI zu kalibrieren.

MUSTERVERFOLGUNG

Wie oben bereits erwähnt, reicht die Bewegungserkennung alleine nicht aus, um ein Objekt zu verfolgen, da sie zu unpräzise ist. Für das genaue Tracking wurde ein Patternmatching-Verfahren verwendet. Die Idee und Teile der Implementation des Tracking werden auch in der OpenSource Software Kinovea¹⁰ verwendet. Die Autoren fanden dieses Projekt in der Recherche Phase der Bachelorarbeit und übernahmen einen Teil des Tracking nach einer Evaluation in einer modifizierten und an die eigenen Bedürfnisse angepassten Version.

Die zwei betroffenen Methoden finden sich in der Klasse *CarTracker* und heissen **CreateTrackPoint(...)** und **Track(...)**.

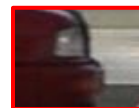


Blau: Trackpunkt

Am Anfang, wenn der Algorithmus die Bewegung erkennt, definiert er einen Punkt innerhalb dieses Bereiches, welcher verfolgt wird. Dieser muss nicht zwingend ganz vorne am Fahrzeug sein. Es kann durchaus vorkommen, dass dieser sich etwas weiter hinten befindet.

Rot: Muster (Pattern)

ABBILDUNG 28: MUSTER ERSTELLEN



Ein 72*54 Pixel grosser Ausschnitt um den Trackpunkt legt der Algorithmus als Muster fest, das dann in allen folgenden Bildern wieder gesucht wird.

¹⁰ <http://www.kinovea.org/en/>, 13.06.2012

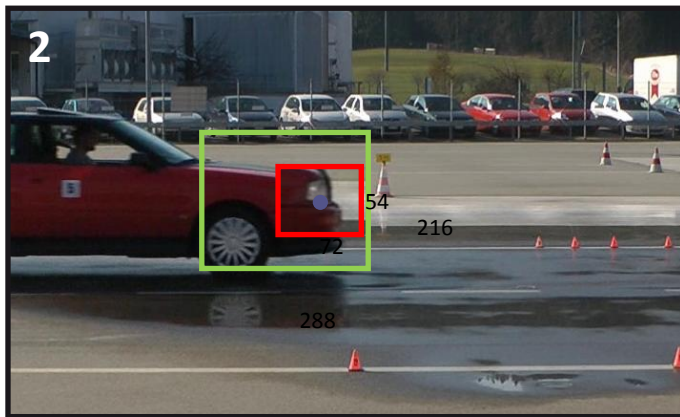


ABBILDUNG 29: MUSTER SUCHEN

Grün: Suchfenster



Das Suchfenster ist ein Bereich mit einer Grösse von 288×216 Pixeln, in welchem das Muster gesucht wird. In diesem Fall hat sich das Fahrzeug nun nach rechts bewegt und das Muster wurde entsprechend etwas weiter rechts im Suchfenster gefunden.

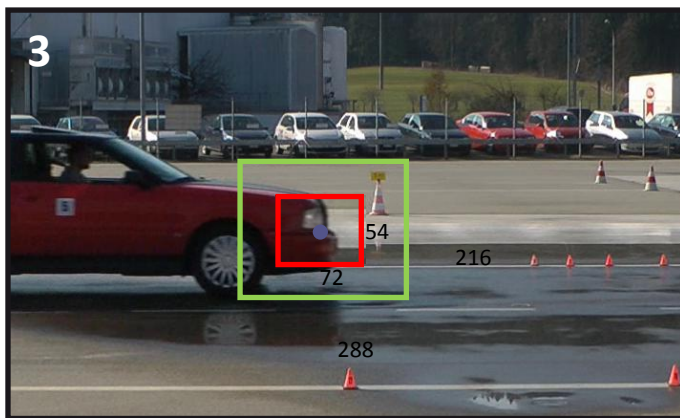


ABBILDUNG 30: SUCHFENSTER NEU AUSRICHTEN

Nachdem das Muster gefunden wurde, richtet der Algorithmus das Suchfenster wieder neu um die neue Position des Trackpoints aus.

PATTERNMATCHING

In diesem Abschnitt geht es darum, wie der Algorithmus das Muster genau sucht. Wie oben erwähnt, nimmt er einen Ausschnitt aus dem Bild und verwendet diesen als Muster, welches er dann mit den folgenden Bildern vergleicht. Der Algorithmus beginnt dabei oben links im Suchfenster und sucht nach unten rechts. Danach verwendet er den Punkt mit der grössten Übereinstimmung als neuen Trackingpunkt.

Verglichen wird mittels dem Fast Normalized Cross-Correlation (NCC) Verfahren, welches die EmguCV Bibliothek anbietet. Dabei liefert das Verfahren einen Rückgabewert, welcher von -1 bis 1 reicht, wobei 1 einer 100%igen Übereinstimmung entspricht.

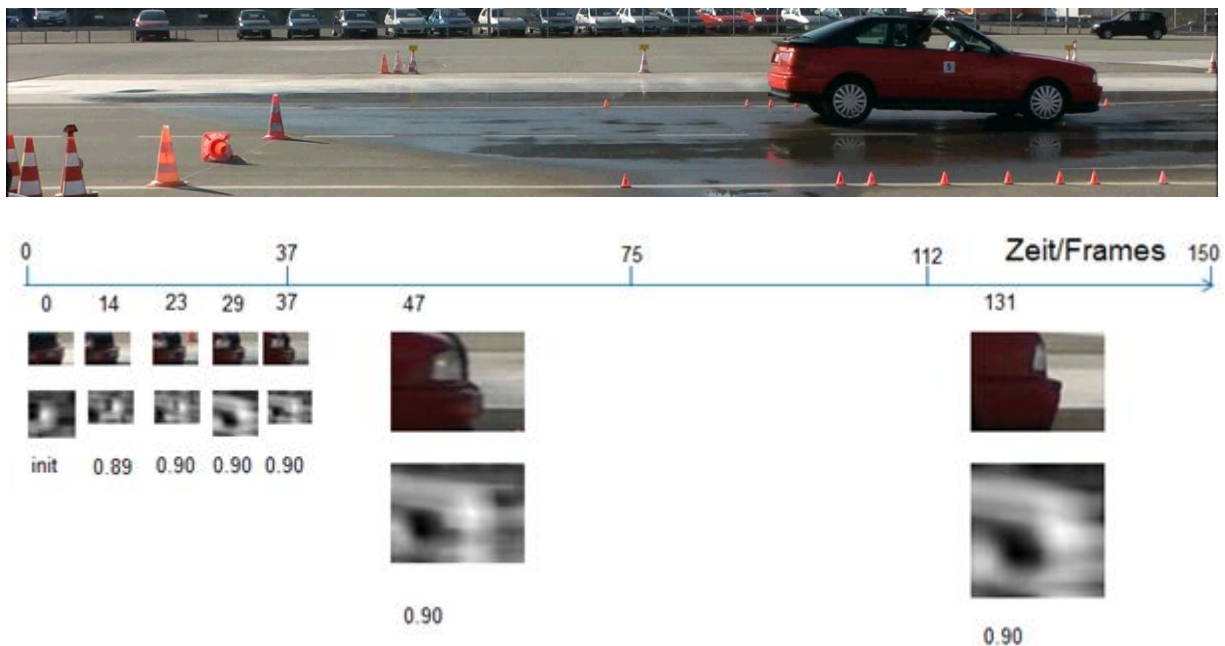


ABBILDUNG 31: MUSTERNEUERSTELLUNG

Das NCC Verfahren erstellt eine Ähnlichkeitskarte: Ein Schwarzweiss-Bild, welches die Übereinstimmungen aufzeigt. Dabei stellen die weissen Flächen Übereinstimmungen und die schwarzen Differenzen dar. Daraus lässt sich ein Wert berechnen der bestimmt wie stark die Bilder übereinstimmen.

Durch die Perspektive kann es vorkommen, das sich das Muster verändert (siehe Abbildung 31: Musterneuerstellung). Anfangs sieht man das Fahrzeug leicht von vorne und später nur noch von der Seite. Das Resultat ist, dass der berechnete Übereinstimmungswert unseren Schwellwert von 0.9 oder kleiner erreicht. Falls dies der Fall ist, wird ein neues Muster erstellt, welches dann weiter verfolgt wird.

Dies ist in der Abbildung 31: Musterneuerstellung zu sehen:

Bei Frame 0 wird Initial das erste Muster erstellt und verfolgt, bis es beim 14. Bild das erste Mal unseren Schwellwert mit 0.89 unterschreitet. Darauf wird ein neues Muster erstellt, welches dann weiter verfolgt wird. Das Selbe ereignet sich dann wieder bei den Frames 23, 29, 37, 47 und 131.

ANALYSE

Durch das Tracking des Fahrzeugs ist es nun zu jedem Zeitpunkt (auf jedem Frame) möglich zu bestimmen, wo sich das Fahrzeug befindet. Aus diesen Informationen lassen sich dann Kennwerte wie Geschwindigkeit, Beschleunigung und Distanzen berechnen.

VON PIXEL ZU METER

Um auf jeder Höhe des Bildes zu bestimmen, wie viele Pixel einem Meter entsprechen, muss der Benutzer über ein Konfigurationsmenu das Bild kalibrieren. Dazu werden zwei Referenzdistanzen auf zwei Höhen angegeben (siehe Abbildung 32: Bildkalibration). Wie der Benutzer diese Kalibration genau vornehmen muss, ist im Abschnitt *Kalibration* des Kapitels *GUI* beschrieben.

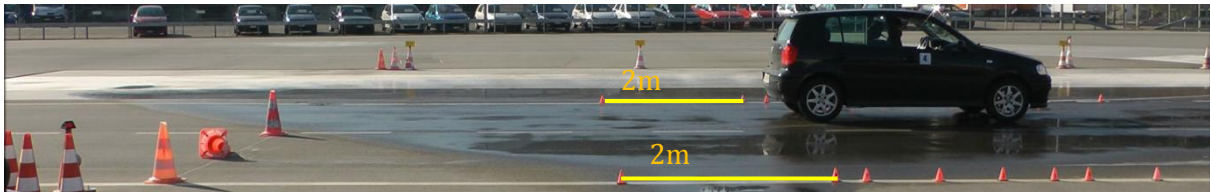
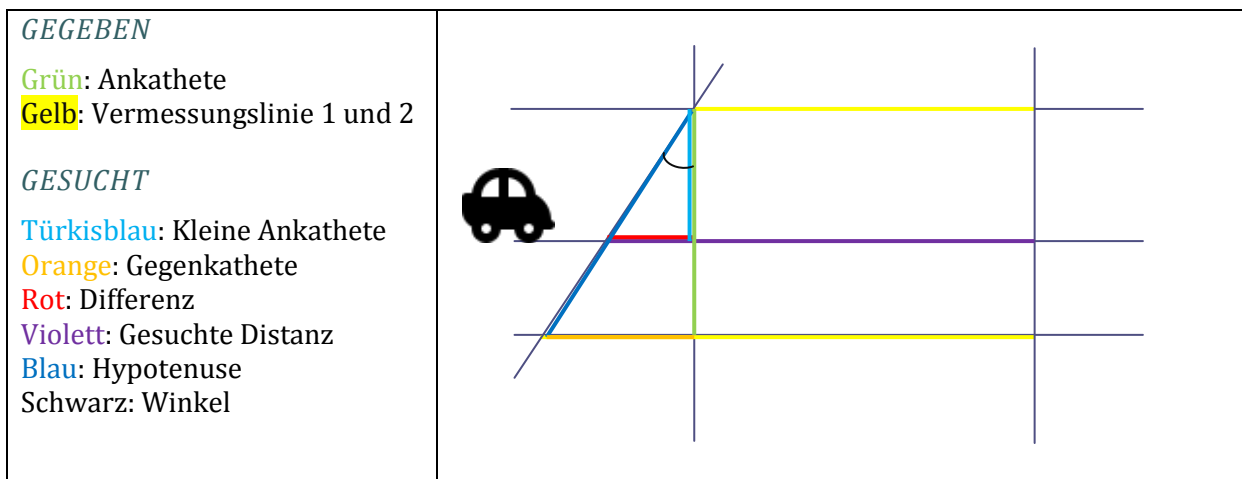


ABBILDUNG 32: BILDKALIBRATION

In diesem Beispiel wurde der Abstand von zwei Pylonen auf zwei verschiedenen Höhen vermessen. Die angegebene Distanz wird dann auf einen Meter erweitert oder gekürzt, um bei beiden Werten dasselbe Verhältnis zu haben. Aus dem vertikalen Abstand beider Vermessungen kann dann die Distanz auf jeder anderen Höhe berechnet werden:



Aus den gelben Linien kann die Höhe (grün) und die Gegenkathete (orange) berechnet werden:

$$\text{Grün} = \text{Gelb.Y} - \text{Gelb2.Y}$$

$$\text{Orange} = \text{Gelb1.X} - \text{Gelb2.X}$$

Aus diesen Werten kann mittels Pythagoras die Hypotenusen(blau) berechnet werden:

$$\text{Blau} = \sqrt{\text{Grün}^2 + \text{Orange}^2}$$

Mittels Hypotenuse (blau) und Ankathete (grün) kann nun der Winkel (schwarz) berechnet werden.

$$Schwarz = \cos^{-1}\left(\frac{Grün}{Blau}\right)$$

Nun hat man alle Größen, um auf jeder Höhe die Pixel per Meter zu bestimmen. Beispielsweise fährt das Fahrzeug auf der Höhe der roten Linie. Nun berechnet man den Höhenunterschied zur oberen gelben Linie was der türkisblauen Linie entspricht. Diese Distanz als Ankathete verwendet und dem Winkel (schwarz) lässt sich mittels des Tangens die Differenz (rot) zur oberen gelben Linie berechnen:

$$Rot = \tan(Schwarz) * Türkisblau$$

Die Distanz, welche auf dieser Höhe einem Meter entspricht, ist dann Violett. Dies ergibt sich aus Gelb und Rot:

$$Violett = Gelb + Rot$$

WEITWINKEL UND DARAUS ENTSTEHENDE PROBLEME

Gegen Ende der Arbeit wurde die bis anhin verwendete Kamera durch eine, mit einem Weitwinkel Objektiv mit 110 Grad Blickfeld in der Horizontalen, ersetzt. Da man durch enge Platzverhältnisse nicht die Möglichkeit hatte, die bestehende Kamera weiter zurückzusetzen, um die ganze Fahrstrecke abbilden zu können.

Durch den Einsatz eines Weitwinkelobjektives hat man das Problem, dass das Bild verzerrt abgebildet wird, was man **tonnenförmigen Verzeichnung** nennt.

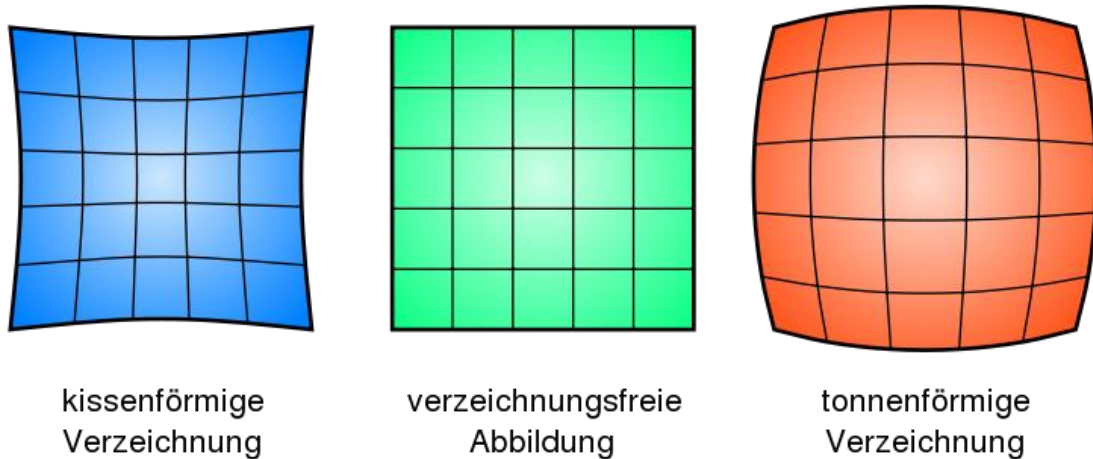


ABBILDUNG 33: VERZEICHNUNG

Dadurch wird eine Strecke in der Mitte des Bildes grösser und mit mehr Pixeln dargestellt, als links oder rechts aussen am Bildrand. Zusätzlich sind gerade Linien auf dem Bild nicht mehr gerade.



ABBILDUNG 34: GERADEN SIND DURCH DAS WEITWINKELOBJEKTIV GEBOGEN



ABBILDUNG 35: VERMESSUNG MIT PYLONEN

Um diesen Effekt abzuschätzen, wurden Pylonen in einem Abstand von 1 Meter über die Breite des Bildes verteilt und dann auf dem Bild in Pixelwerten vermessen. Daraus gab sich folgendes Bild:

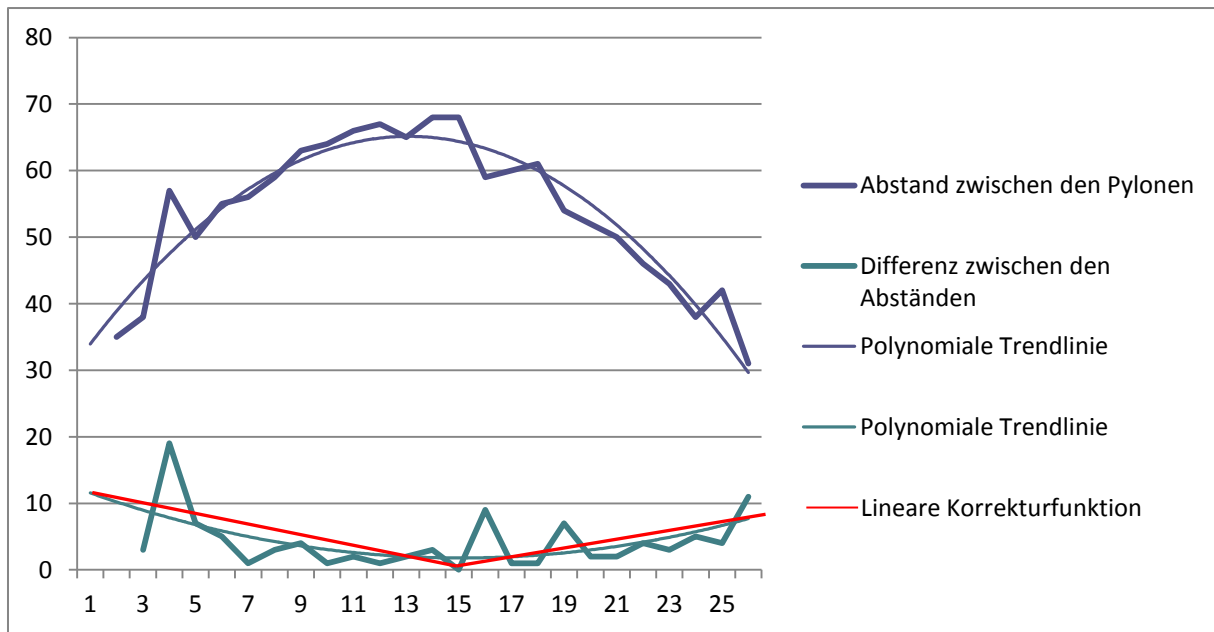


ABBILDUNG 36: PYLONENABSTAND AUSWERTUNG

Daraus wird ersichtlich, dass es nicht mehr möglich ist, eine Strecke wie zuvor auf eine feste Anzahl Pixel abzubilden.

In einem ersten Schritt wurde versucht, mittels der Werte, die durch die Vermessung der Pylonen entstanden sind, eine Funktion abzubilden, mit welcher die Pixel-pro-Meter-Werte korrigiert wurden (siehe rote Linie in Abbildung 36: Pylonenabstand Auswertung), was aber nicht zu dem gewünschten Erfolg führte.

In einem zweiten Schritt wurde eine Referenzfahrt mit konstanter Geschwindigkeit von links nach rechts getrackt und die dadurch erhaltenen x-Koordinaten des Fahrzeugs analysiert.

Angenommen das Bild wäre verzeichnungsfrei, hätte das Fahrzeug zwischen jedem Bild gleich viele Pixel zurückgelegt. Aufgrund der Verzeichnung ist dies aber nicht der Fall, wie folgendes Diagramm zeigt.

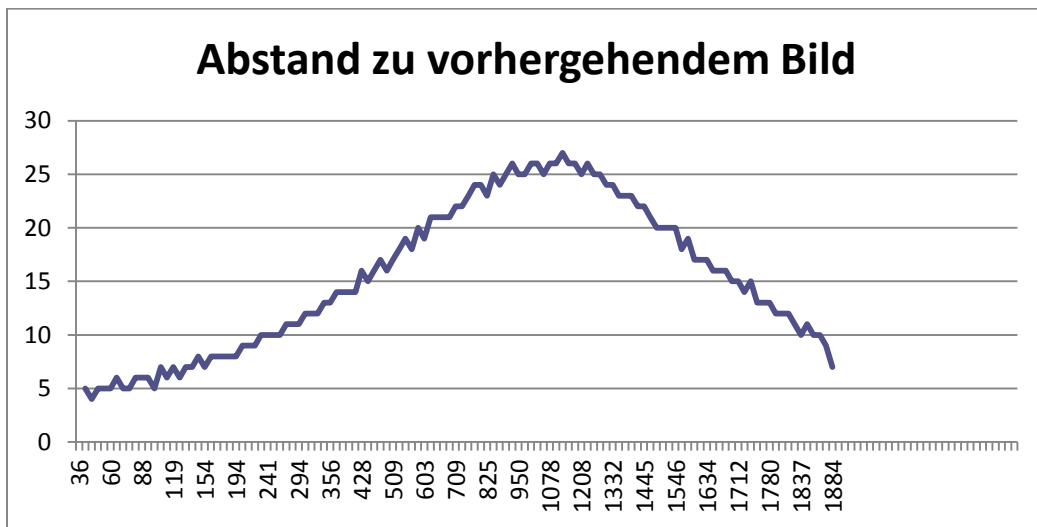


ABBILDUNG 37: ABSTAND ZU VORHERGEHENDEM BILD

In Wirklichkeit sollte die Distanz, welche das Fahrzeug zwischen den einzelnen Bildern zurückgelegt hat, immer bei ca. 27 Pixeln liegen, wie in der Mitte des Bildes. Durch dieses Wissen konnte für jeden Abstand ein Faktor berechnet werden, mit welcher der Abstand korrigiert werden muss, um auf die 27 Pixel zu kommen.

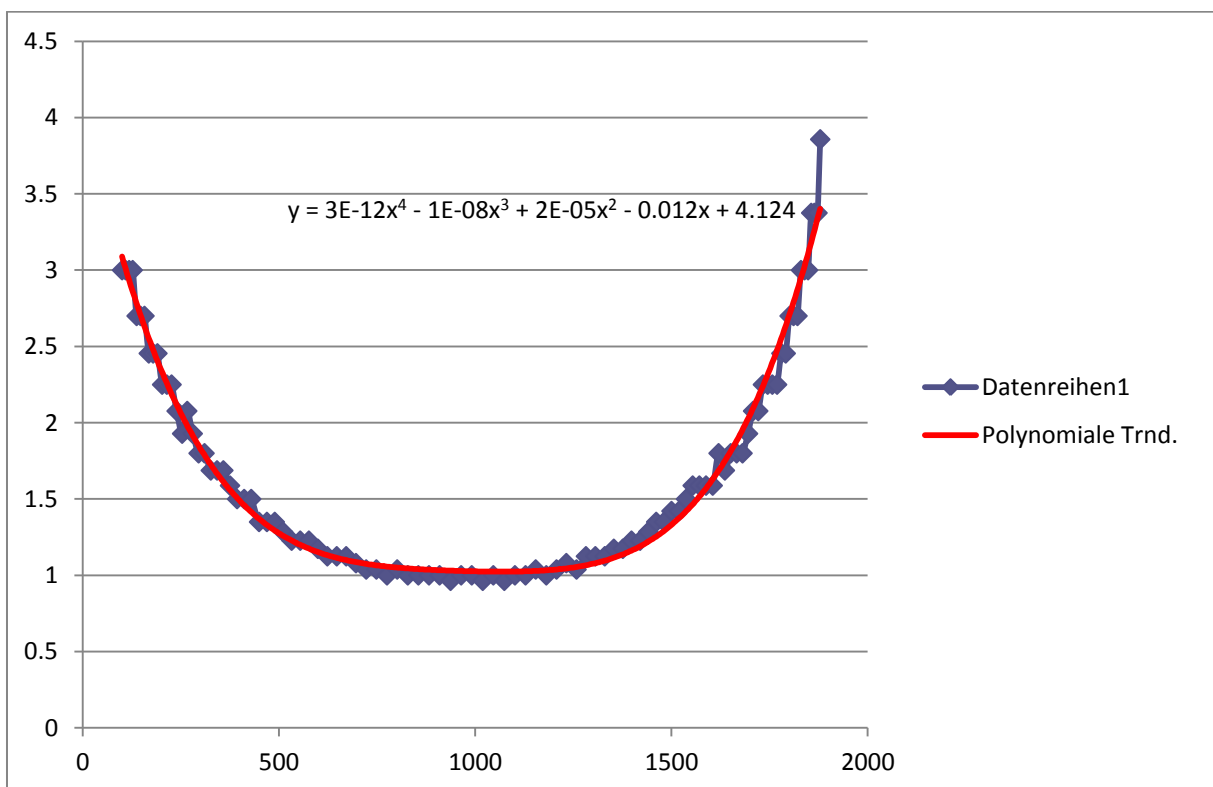


ABBILDUNG 38: KORREKTURFAKTOR

Mit den Werten dieser Fahrt konnte eine polynomiale Trendlinie des 4. Grades erzeugt werden.

$$F(x) = 3E - 12x^4 - 1E - 08x^3 - 2E.05x^2 - 0.012x + 4.124$$

Mit diesem Polynom wäre es möglich, dass man den Korrekturfaktor zu jedem beliebigen Punkt auf der x-Achse berechnen lässt. Leider ist Microsoft Excel fehlerhaft und berechnet falsche Polynome für Trendlinien oberhalb des dritten Grades.¹¹ Die polynomialen Trendlinien unterhalb des vierten Grades waren aber zu ungenau, weshalb diese nicht verwendet werden konnten.

Durch die tonnenförmige Verzeichnung tritt noch ein weiterer Effekt auf. Zusätzlich zu den zu kurzen Distanzen am Bildrand werden Distanzen in der Bildmitte zu lang dargestellt. Dies ist zu sehen an den roten Flächen in Abbildung 39: Tonnenförmige Verzeichnung, welche die Differenz zwischen einem verzeichnungsfreien Bild und einem tonnenförmig verzeichnetem Bild zeigt.

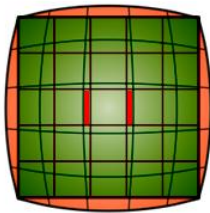


ABBILDUNG 39: TONNENFÖRMIGE VERZEICHNUNG

Durch die anfänglich berechneten Korrekturfaktoren werden nun zwar die Fehler an den Rändern des Bilder korrigiert, in der Mitte wird der Verzeichnungseffekt jedoch noch mehr verstärkt.

Um richtig zu korrigieren, müsste in der Mitte mit einem Korrekturfaktor <1 gerechnet werden.

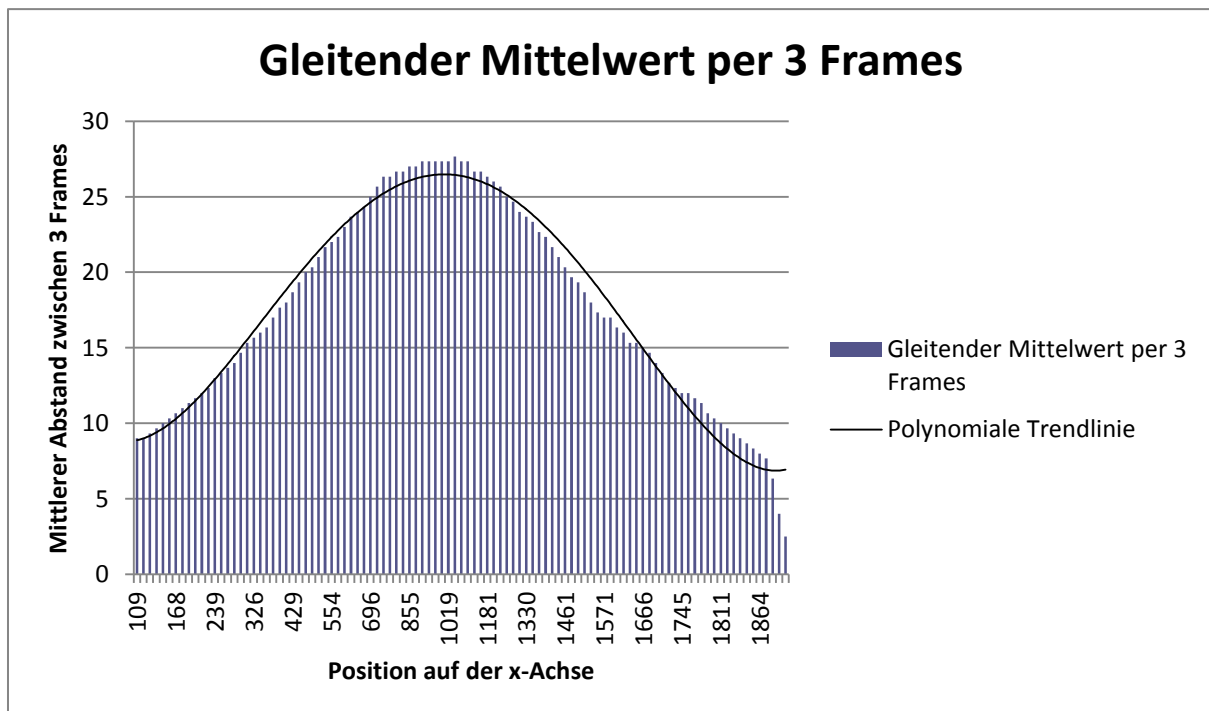


ABBILDUNG 40: VERZEICHNETE DARSTELLUNG EINER KONSTANTEN GESCHWINDIGKEIT

Nun muss also statt der 27 Pixel Abstand ein tieferer Wert gewählt werden, um in der Mitte den Verzug ebenfalls aufzuheben. Tests haben gezeigt, dass 26.33 ein guter Wert ist (das Maximum der Trendlinie), womit man auf folgende Korrekturfaktoren kommt.

¹¹ <http://support.microsoft.com/kb/114629/en-us>, 07.06.2012

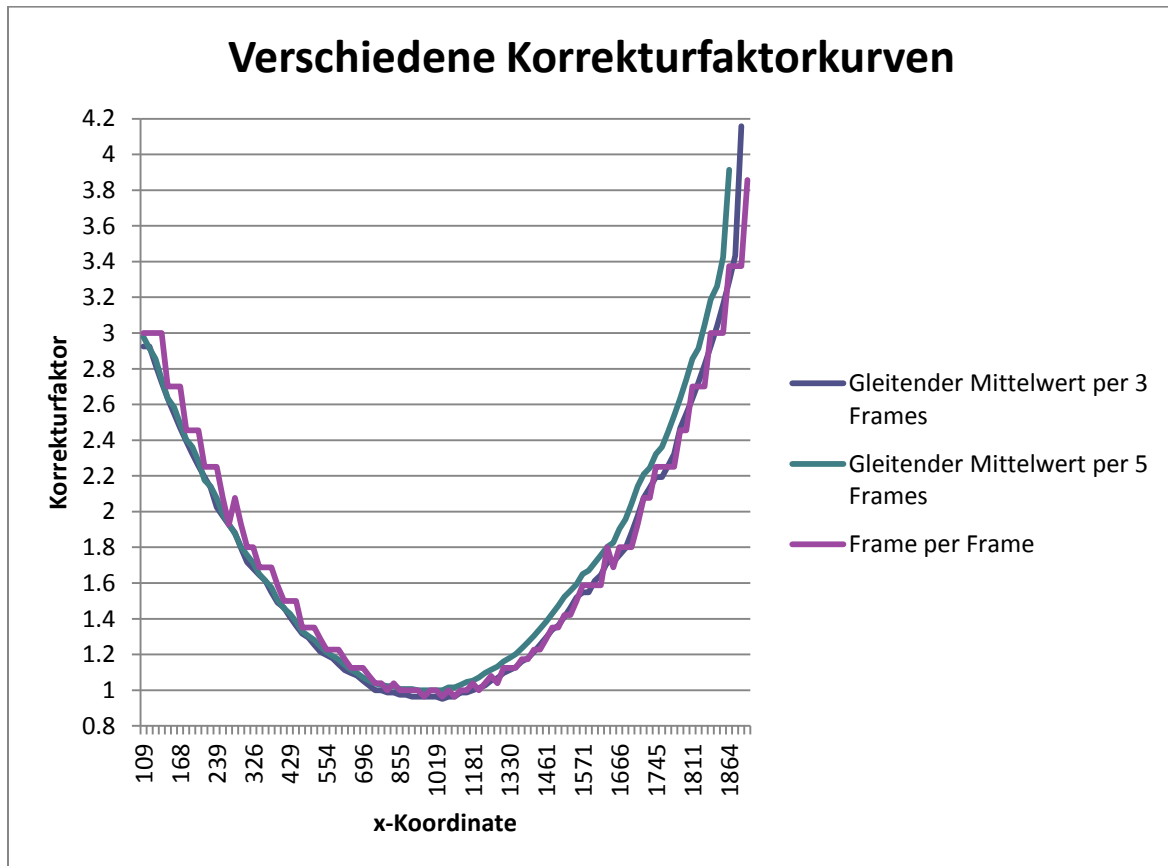


ABBILDUNG 41: UNTERSCHIEDLICHE KORREKTURFAKTOREN

In der Applikation werden die Werte der Kurve *Gleitender Mittelwert per 3 Frames* verwendet, da diese am besten den Verzeichnungseffekt in der Mitte aufheben.

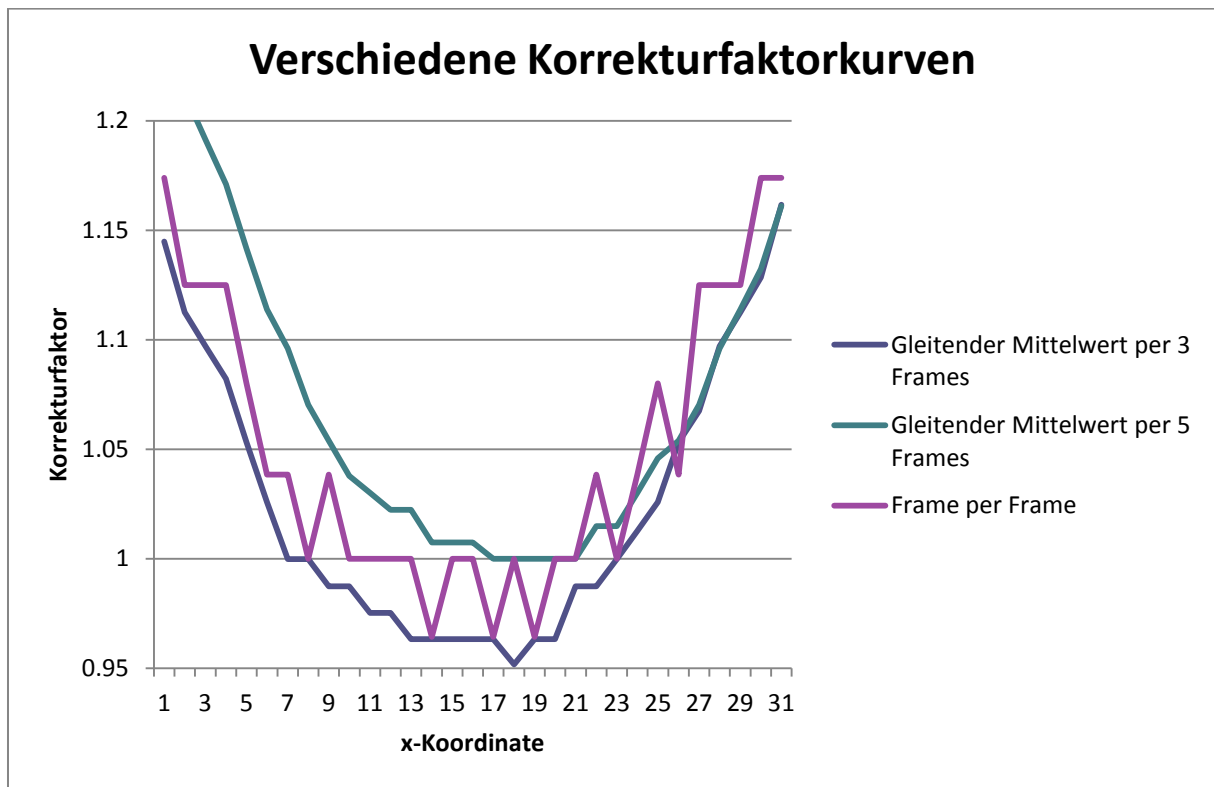


ABBILDUNG 42: KORREKTURFAKTOREN IN DER MITTE DES BILDES

Die jeweils errechneten Korrekturfaktoren sind immer abhängig von der Kameraposition und deren Ausrichtungswinkel zur Strecke. Am idealsten ist es, wenn die Kamera in der Mitte der Strecke und rechtwinklig dazu steht. Die Positionierung in der Mitte ist nicht zwingend, macht aber am meisten Sinn, wenn gleichviel An- und Wegfahrstrecke auf dem Video zu sehen sein soll. Die rechtwinklige Ausrichtung der Kamera zur Strecke ist zwingend um eine optimale Genauigkeit der Endresultate einer Videanalyse zu erreichen.

KALKULATIONEN

DISTANZEN

In einem ersten Schritt müssen die Distanzen zwischen den Frames in Metern ermittelt werden. Dazu wird zuerst der Pixel pro Meter-Wert auf Höhe des Fahrzeugs ermittelt. Danach teilt man die Pixeldistanz zwischen jedem Framepaar durch den Pixel pro Meter-Wert und erhält die Distanz in Meter. Dies wäre richtig bei einer perfekten Linse, wie sie bei einer Lochkamera vorhanden ist. Bei einer Linse mit Verzeichnung muss der ermittelte Wert noch mit dem im Kapitel „Weitwinkel und daraus entstehende Probleme“ erwähnten Korrekturfaktor multipliziert werden um die korrekte Meterdistanz zu erhalten.

GESCHWINDIGKEIT

Nun, da man die Distanzen zwischen den Frames kennt, kann man die Geschwindigkeit berechnen. Dazu wird die Distanz über 5 Frames betrachtet. Diese Distanz geteilt durch die vergangene Zeit entspricht der Geschwindigkeit in Pixel. Der Grund, dass man eine Zeitspanne von 5 Frames betrachtet und nicht nur die Distanz von zwei aufeinander folgenden Frames ist, dass durch das visuelle Tracking Fehler entstehen, die zu einem Rauschen führen. Wenn man hingegen immer 5 Frames betrachtet erhält man einen gleitenden Durchschnitt, welcher dieses Rauschen entfernt (siehe Abbildung 43: Vergleich Geschwindigkeitsberechnung).

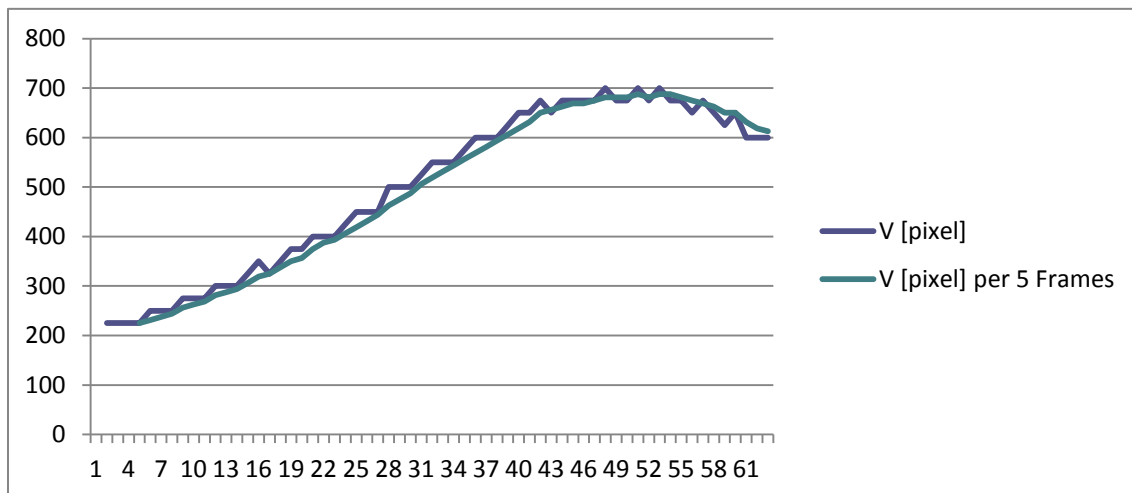


ABBILDUNG 43: VERGLEICH GESCHWINDIGKEITSBERECHNUNG

BESCHLEUNIGUNG

Die Beschleunigung ist relevant, um festzustellen, wann der Fahrer die Vollbremsung einleitet. Die Beschleunigung errechnet sich aus dem Geschwindigkeitsunterschied im Verhältnis zur vergangenen Zeit.

REAKTIONSZEIT UND REAKTIONSWEG

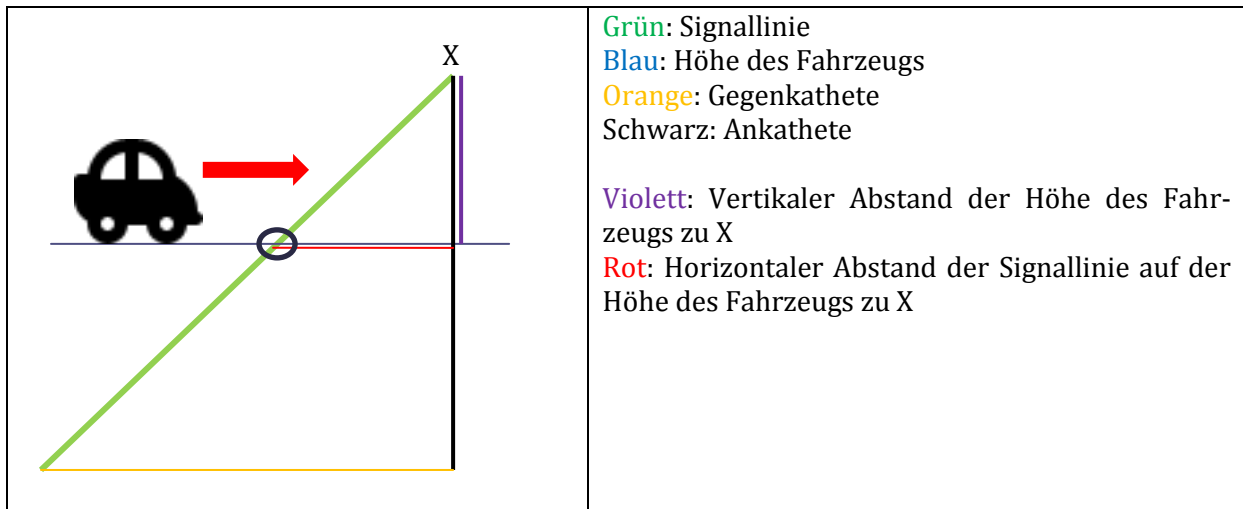
Um Reaktionszeit und -weg zu bestimmen sind zwei Dinge ausschlaggebend.

1. Es muss das Videoframe ermittelt werden, bei welchem das Fahrzeug die Signallinie überquert.

2. Und es muss detektiert werden, wann der Fahrer die Vollbremsung einleitet.

SIGNALLINIEÜBERQUERUNG

Ersteres ist einfach zu bewerkstelligen. Mittels Trigonometrie kann ermittelt werden, wann die x-Koordinate auf der rechten Seite der Signallinie ist.



Mittels der schwarzen Distanz und der orangenen Distanz, welche durch die Signallinie gegeben sind, kann der Winkel berechnet werden

$$a = \tan^{-1} \left(\frac{\text{Schwarz}}{\text{Orange}} \right)$$

Wurde der Winkel berechnet, kann damit wiederum der horizontale Abstand der Signallinie auf der Höhe des Fahrzeugs zu X berechnet werden (rot).

$$\text{Rot} = \tan(a) * \text{Violett}$$

X minus der roten Strecke gibt dann der Punkt, auf welcher das Fahrzeug die Signallinie überquert.

Nun, da man die X-Koordinate des Punktes bestimmt hat, kann man diesen mit den Fahrzeugpositionen vergleichen und das erste Frame bestimmen, bei welchem das Fahrzeug auf der rechten Seite liegt.

BESTIMMUNG DES VOLLBREMSUNGSZEITPUNKTES

Der Zeitpunkt bei dem die Vollbremsung eingeleitet wird, ermittelt man mit der Beschleunigung. Während von hinten nach vorne über alle Bilder iteriert wird, berechnet man den gleitenden Durchschnitt über jeweils fünf Bilder. Sobald dieser Wert positiv wird, ist man beim Bild angekommen, bei dem der Fahrer mit der Vollbremsung begonnen hat.

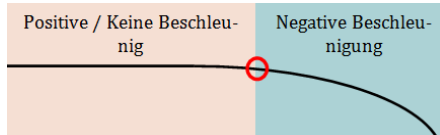


ABBILDUNG 44: BESTIMMUNG DES BREMSZEITPUNKTES ANHAND DER BESCHLEUNIGUNG

Die genaue Bestimmung des Bremszeitpunktes ist im Kapitel Rohdatenanalyse beschrieben.

Nun, da Signalüberquerungs- und Bremspunkt bestimmt sind, kann man auf Grund der verstrichenen Zeit, welche über die Bildrate ermittelt wird, die Reaktionszeit und die Distanz bestimmen.

QUALITÄTSMANGEMENT

ROHDATENANALYSE

Um die physikalischen Grundlagen aller Berechnungen von der Implementation zu abstrahieren, wurden in Excel auf Basis der XY-Koordinaten des Trackingalgorithmus verschiedene Rohdatenanalysen durchgeführt.

GESCHWINDIGKEIT

In einer ersten Version des Programms wies der Verlauf der Geschwindigkeit grosse Sprünge auf, die nicht der Realität entsprechen konnten.

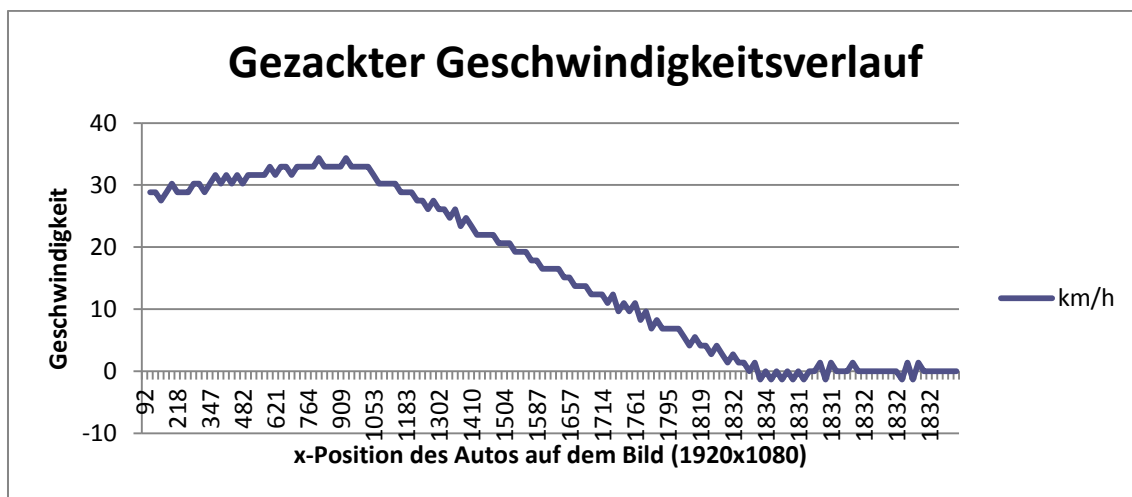


ABBILDUNG 45: GESCHWINDIGKEIT AUS ROHDATEN

Die deutlich sichtbaren Spitzen im Verlauf entstehen durch das Tracking, welches jedes einzelne Bild berücksichtigt. Da aber die Genauigkeit des Trackings von Bild zu Bild um einige Pixel variieren kann, entstehen die Spitzen, wenn bei der Geschwindigkeitsberechnung jedes einzelne Bild einbezogen wird.

Um dies zu verhindern, wurde ein gleitender Durchschnitt der Geschwindigkeit über jeweils fünf Bilder berechnet und als Geschwindigkeitsverlauf verwendet, wodurch das Resultat schon viel besser wurde.

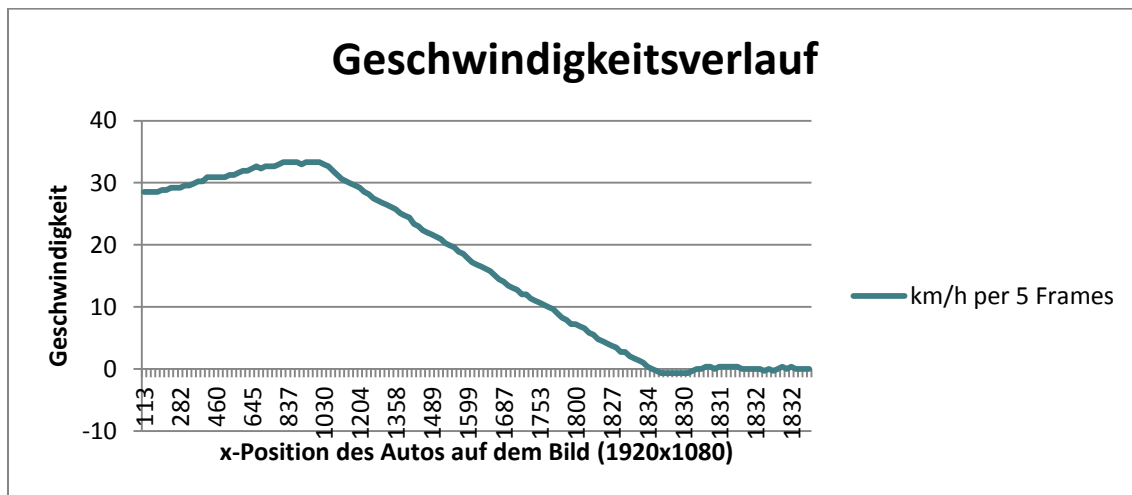


ABBILDUNG 46: GEMITTELTE GESCHWINDIGKEIT

Als Versuch wurde noch ein gleitender Mittelwert über jeweils zehn Bilder berechnet, welcher allerdings nicht brauchbar war, da dadurch eine Geschwindigkeit erst nach zehn Bildern verfügbar war, was zu spät ist.

BESCHLEUNIGUNG

Auch die Beschleunigung musste berechnet werden, zur korrekten Ermittlung des Bremszeitpunktes. Wie schon bei der Geschwindigkeit zeigte sich auch hier, dass eine Berechnung der Beschleunigungswerte aus den direkt aufeinanderfolgenden Geschwindigkeitswerten des gleitenden Durchschnitts über fünf Bilder, nicht konstant genug ist. Eine Besserung brachte die Berechnung der Beschleunigung aus zwei Geschwindigkeitswerten, die immer einen Abstand von fünf Bildern haben.

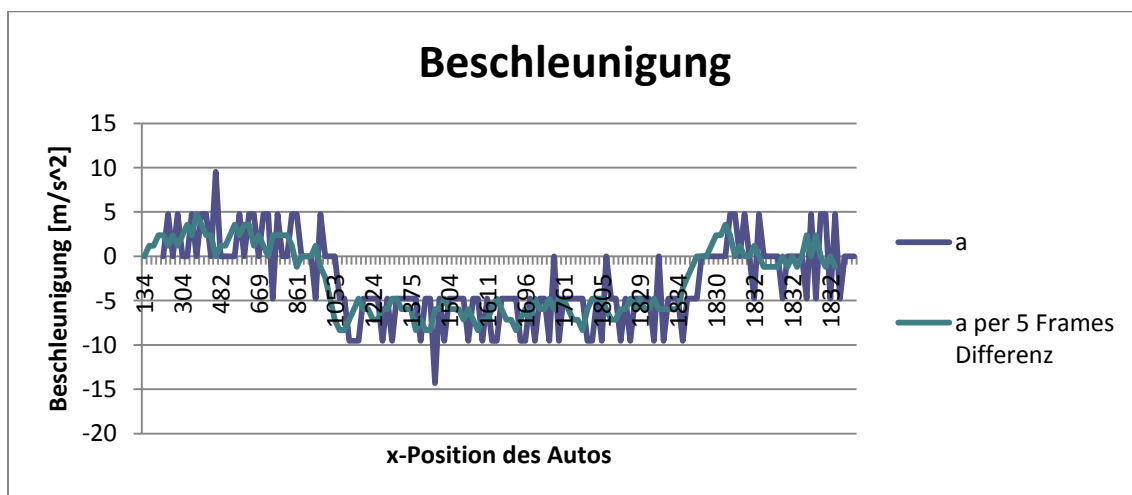


ABBILDUNG 47: BESCHLEUNIGUNG

Obwohl die Werte, die mit fünf Bildern Differenz berechnet wurden, schon viel besser sind, ist das Ergebnis trotzdem noch nicht zufriedenstellend. Denn die Kurve macht bereits einen kleinen Sprung ins Negative, bevor der tatsächliche Bremsvorgang stattfindet.

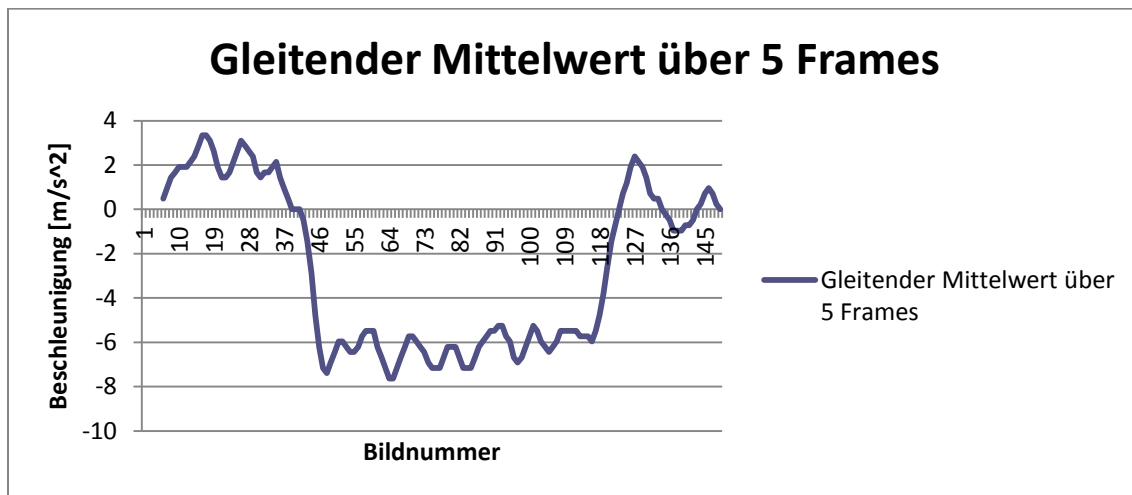


ABBILDUNG 48: GEMITTELTE BESCHLEUNIGUNG

Nun kann man den Bremszeitpunkt bei Bild Nummer 42 feststellen, bei dem die Beschleunigung erstmals unter null fällt.

BILDVERZUG

Der Bildverzug, dessen Auswirkungen und der Umgang damit wird im Absatz *Weitwinkel und daraus entstehende Probleme* im Kapitel *Analyse* beschrieben.

TESTING

Um sicherzustellen, dass das System richtig funktioniert oder Fehler zu lokalisieren, wurde Testing auf verschiedenen Ebenen betrieben.

EXPLORATIVES TESTING

Es wurde viel exploratives Testing betrieben. Dazu wurde eine *IVideoSource* implementiert, die eine abgespeicherte Videoaufnahme einliest und sequenziell jedes Frame von Anfang bis zum Ende des Videos liefert. Dadurch bestand anfangs die Möglichkeit, ohne Kamera zu arbeiten.

Die Videos wurden von Rolf Bader während eines Kurses, bei dem die Teilnehmer Vollbremsungen durchführten, aufgenommen.

Auf dem Testgelände in Gossau SG wurden weitere Tests, vor allem gegen Ende des Projekts, durchgeführt. Mit Fahrten, die über die gesamte Strecke eine konstante Geschwindigkeit hielten, wurden weitere Tests durchgeführt. So konnten die Konstanz und Stabilität der errechneten Werte kontrolliert werden.

Da auf dem Testgelände ein Geschwindigkeitsradar installiert ist und das Fahrzeug zusätzlich zum normalen Digitaltachometer ein zweites Tachometer besitzt, das per GPS arbeitet, war es auch möglich, die angezeigten Geschwindigkeiten zu verifizieren. Bei konstant gefahrener Geschwindigkeit (per Tempomat) zeigten sogar alle drei Geräte dieselbe Geschwindigkeit an.

Videos der Kamera, die dann schlussendlich eingesetzt wurde und die im Gegensatz zu den anderen Kameras über ein Weitwinkelobjekt verfügte, waren erst ab der dritten Wochen vor Pro-

jektende verfügbar. Dies führte bei der Auswertung zu grossen Problemen (siehe Kapitel „Weitwinkel und daraus entstehende Probleme“)

FELDTTESTS

Gegen Ende der Arbeit erhielten wir Zugang zur eingesetzten Kamera und konnten in Gossau (SG) auf dem Gelände unser System testen.

Als die endgültige Kamera verfügbar war, konnte das System auf dem Gelände in Gossau SG ausführlich getestet werden.

Dabei wurden einige Probleme entdeckt:

Test Nr.	Datum	Problem	Ursache	Massnahme
1	18. Mai 2012	Stream stockt	Der Implementierte Stream mit dem Videoformat mjpeg ist zu langsam für die Live Verarbeitung	Umstellung auf Real Time Streaming Protocol (RTSP)
2	24. Mai 2012	Bilder kommen sehr spät oder es entstehen Arbeitsspeicherprobleme	Die Bilder der Kamera kamen schneller als sie vom System verarbeitet werden konnten	Der Laptop, welcher zur Verfügung gestellt wurde, verfügte nicht über die nötige Leistung und wurde ausgewechselt
3	30. Mai 2012	Geschwindigkeit steigt anfangs an und fällt gegen Ende wieder ab.	Die Ursache für diesen Effekt ist die Verzeichnung	Das Bild wurde mittels Pylonen vermessen und daraus ein Korrekturfunktion abgeleitet → Ohne Erfolg (mehr dazu im Kapitel: Weitwinkel und daraus entstehende Probleme)
4	31. Mai 2012	Geschwindigkeit steigt anfangs an und fällt gegen Ende wieder ab.	Die Ursache für diesen Effekt ist die Verzeichnung	Es wurde eine Referenzfahrt mit konstanter Geschwindigkeit aufgenommen und daraus Korrekturfaktoren abgeleitet (mehr dazu im Kapitel: Weitwinkel und daraus entstehende Probleme)

5	1. Juni 2012	Distanzen(2-5m) und Geschwindigkeiten(3-6km/h) weichen von der Realität ab.	Bug in der Berechnung	Anpassung von Korrekturfaktoren. Weitere Rohdatenanalyse Aufnahme von Videos für weitere Analyse
6	8. Juni 2012	Boden des Fahrzeugs kann nicht genau genug erkannt werden.	Das Fahrzeug ist nicht immer auf derselben Höhe. Spiegelung im Wasser	Die Spurhöhen müssen definiert werden in der Bildkalibration (Siehe Kapitel: Kalibration)
7	12. Juni 2012	Die letzte Testsession mussten wir leider auf Grund von Hochwasser ausfallen lassen.		

UNITTESTING

Für Klassen im Namespace "CarTracker.Analyzation" wurden in einem separatem Testprojekt Unittests geschrieben, die sicherstellen, dass Berechnungen wie Distanz, Geschwindigkeit oder Meter zu Pixel richtig durchgeführt werden.

SCHLUSSFOLGERUNG

Das Resultat der Arbeit zeigt, dass es möglich ist, mittels Videoanalyse ein System bereitzustellen, womit Fahrten vollautomatisch erkannt und in kurzer Zeit ausgewertet werden können. Es ist ideal, dass keine Änderungen oder Erweiterungen an Strecke oder Fahrzeug vorgenommen werden müssen. Dadurch ist maximale Mobilität gewährleistet, weshalb dieses System ohne grossen Aufwand auch an anderen Orten in Betrieb genommen werden kann.

Da dieses Projekt zeitlich auf 17 Wochen begrenzt war, konnten nicht alle ursprünglich geplanten Features implementiert werden, wie beispielsweise die Analyse einer Kurvenfahrt.

Bei einer Weiterentwicklung des Systems stehen folgende Punkte auf der Liste der zu implementierenden Features:

- Robustheit: Keine falschen Bewegungen mehr detektieren
- Analyse einer Kurvenfahrt
- Partielle Autokalibration des Systems durch Kalibrationsfahrt mit konstanter Geschwindigkeit, anstelle einer manuellen Kalibration
- Unterstützung anderer Streamingformate

Der Industriepartner ist an einer Weiterentwicklung der Anwendung interessiert und ist davon überzeugt, dass auch andere Fahrtrainingszentren Interesse an diesem System haben.

LITERATURVERZEICHNIS

Nathan, A., & Lehenbauer, D. (2007). *Windows Presentation Foundation Unleashed*. Sams Publishing.

ABBILDUNGSVERZEICHNIS

Abbildung 1: Resultat	1
Abbildung 2: Statistik der beratungsstelle für unfallverhütung (bfu)	6
Abbildung 3: Ablaufdiagramm	6
Abbildung 4: Ausschnitt aus der Endanwendung mit Beschreibung der Werte	7
Abbildung 5: Statistik des bfu zum Anhalteweg.....	8
Abbildung 6: Beschriftete Endanwendung.....	9
Abbildung 7: Reaktions- und Bremsweg bei verschiedenen Geschwindigkeiten auf trockener Fahrbahn.....	13
Abbildung 8: Screenshot	14
Abbildung 9: Architektur	18
Abbildung 10: Domainklassen.....	19
Abbildung 11: Videohandlergrobkonzept	21
Abbildung 12: VideoHandling.....	22
Abbildung 13: Sequenzdiagramm Videohandler	24
Abbildung 14: Verschiedene Models.....	26
Abbildung 15: navigationsleiste	26
Abbildung 16: Durchgangsnummer erhöhen.....	26
Abbildung 17: Konfigurierte Filter	27
Abbildung 18: Webcam ein- und ausschalten	27
Abbildung 19: SpurAuswahl	27
Abbildung 20: abspielen zweier Fahrten	28
Abbildung 21: Auflistung der fahrten des aktuellen Tages	28
Abbildung 22: Filter konfigurieren.....	29
Abbildung 23: Globale Einstellungen.....	30
Abbildung 24: Bildkalibration	30
Abbildung 25: Bewegungserkennung	31
Abbildung 26: Schnelle Bewegung und resultierende Abweichung der Bewegungserkennung...	31
Abbildung 27: Langsame Bewegung.....	32
Abbildung 28: Muster erstellen.....	33
Abbildung 29: Muster suchen	34
Abbildung 30: Suchfenster neu ausrichten	34
Abbildung 31: Musterneuerstellung	35
Abbildung 32: Bildkalibration	36
Abbildung 33: Verzeichnung.....	38
Abbildung 34: Geraden sind durch das Weitwinkelobjektiv gebogen.....	38
Abbildung 35: Vermessung mit Pylonen	39
Abbildung 36: Pylonenabstand Auswertung.....	39
Abbildung 37: Abstand zu vorhergehendem Bild	40
Abbildung 38: Korrekturfaktor	40
Abbildung 39: Tonnenförmige Verzeichnung.....	41

Abbildung 40: Verzeichnete darstellung einer konstanten Geschwindigkeit **Fehler! Textmarke nicht definiert.**

Abbildung 41: Unterschiedliche Korrekturfaktoren	42
Abbildung 42: Korrekturfaktoren in der Mitte des Bildes	43
Abbildung 43: Vergleich Geschwindigkeitsberechnung	44
Abbildung 44: Bestimmung des Bremszeitpunktes anhand der Beschleunigung	46
Abbildung 45: Geschwindigkeit aus Rohdaten	47
Abbildung 46: Gemittelte Geschwindigkeit.....	48
Abbildung 47: Beschleunigung.....	48
Abbildung 48: Gemittelte Beschleunigung	49
Abbildung 49: Klassendiagramm GUI.....	63
Abbildung 50: Design model	64
Abbildung 51: Finaler Paperprototype	72
Abbildung 52: Paperprototype Vollbremsung.....	73
Abbildung 53: Paperprototype Kurvenfahrt	73
Abbildung 54: Paperprototype Fahrtenvergleich Version 1.....	73
Abbildung 55: Paperprototype Fahrtenvergleich Version 2.....	74
Abbildung 56: Paperprototype Kurvenfahrtvergleich	74
Abbildung 57: Paperprototype Fahrtenvergleich mit video.....	74
Abbildung 58: Paperprototype Fahrtenvergleich mit Video finale version	75
Abbildung 59: Scrum Taskboard	77
Abbildung 60: Projektplan Seite 1	78
Abbildung 61: Projektplan Seite 2	79
Abbildung 62: Projektplan Seite 3	80

GLOSSAR

Begriff	Bedeutung
Action	Datentyp in C#, der eine Methode kapselt, die über einen einzelnen Parameter verfügt und keinen Wert zurückgibt.
ActiveX	Bezeichnet ein Softwarekomponenten-Modell von Microsoft für aktive Inhalte.
Bildrate	Anzahl Bilder die eine Kamera pro Sekunde liefert.
Binding	Durch Datenbindung wird eine Verbindung zwischen der UI der Anwendung und der Geschäftslogik hergestellt. Wenn die Bindung über die korrekten Einstellungen verfügt und die Daten die richtigen Benachrichtigungen bereitstellen, ändern sich die an die Daten gebundenen Elemente automatisch bei jeder Änderung des Werts der Daten, sodass die Änderungen entsprechend wiedergegeben werden. Datenbindung kann auch bedeuten, dass bei einer Änderung der äußeren Darstellung von Daten in einem Element die zugrunde liegenden Daten automatisch aktualisiert werden können, um die Änderung wiederzugeben. Wenn der Benutzer beispielsweise den Wert in einem TextBox-Element bearbeitet, wird der zugrunde liegende Datenwert automatisch aktualisiert, um diese Änderung anzuzeigen.
Delegate	Eine Datenstruktur, die auf eine statische Methode oder auf eine Klasseninstanz und eine Instanzenmethode dieser Klasse verweist.
Durchgang	Ein Durchgang ist eine Gruppe von Fahrten
Exception	Stellt einen Fehler dar, der beim ausführen einer Anwendung auftritt.
Fahrt	Eine Fahrt ist eine Durchführung eines Manövers (Vollbremse)
Frame	Einzelnes Bild aus einem Video einer Kamera

GUI	Graphical User Interface – Grafische Benutzeroberfläche
Livestream	Echtzeitübertragung des Kamerabildes
Netzwerkamera	Kamera die über das Netzwerk angesteuert werden kann
Passives System	Unter einem passiven System wird in unserem Fall ein System verstanden, welches funktioniert ohne dass Vorbereitungen oder Anpassungen nötig sind
Pattern (Muster)	Ein Muster welches in anderen Bildern gesucht wird
Patternmatching	Algorithmus zum finden eines vorgegebenen Musters auf einem Bild
RTSP	RealTime Streaming Protokoll
SCRUM	Scrum ist ein Vorgehensmodell der Softwaretechnik. Der Ansatz von Scrum ist empirisch, inkrementell und iterativ. Er beruht auf der Ansicht, dass die meisten modernen Entwicklungsprojekte zu komplex sind, um durchgängig planbar zu sein.
SDK	Software Development Kit
Singleton	Ein in der Softwareentwicklung eingesetztes Entwurfsmuster und gehört zur Kategorie der Erzeugungsmuster (engl. <i>Creational Patterns</i>). Es stellt sicher, dass von einer Klasse genau ein Objekt existiert.
Thread	Bezeichnet in der Informatik einen Ausführungsstrang oder eine Ausführungsreihenfolge in der Abarbeitung eines Programms.
Tracking (Verfolgen)	Das Verfolgen eines Objekts. In unserem Fall innerhalb eines Videos.
Trackingpoint (Verfolgungspunkt)	Der genaue Punkt, welcher verfolgt wird
URL	Uniform Resource Locator, identifiziert und lokalisiert eine Ressource wie z.B. eine Website.
UserControl	Steuerelement auf der Benutzeroberfläche.
WAB-Kurse	Weiterausbildungskurse, die während der dreijährigen Probezeit absolviert werden müssen, bevor der endgültige Ausweis ausgestellt wird.
Weitwinkelkamera	Kamera mit einem besonders breiten Sichtbereich
WMV	Windows Media Video (Vidoeformat)
WPF	Windows Presentation Foundation (kurz WPF), auch bekannt unter dem Codenamen Avalon, ist ein Grafik-Framework und Teil des .NET Frameworksvon Microsoft, das mit Windows Vista und Windows 7 ausgeliefert wird, sich aber auf Windows XP und Server 2003 nachinstallieren lässt. WPF stellt ein umfangreiches Modell für den Programmierer bereit. Dabei werden die Präsentation und die Geschäftslogik getrennt, dies wird vor allem durch die Auszeichnungssprache XAML (basierend auf XML) unterstützt.
XAML	Extensible Application Markup Language (XAML) ist eine von der Software-Firma Microsoft entwickelte allgemeine Beschreibungssprache für die Oberflächengestaltung von Anwendungen sowie zur Definition von Workflows in WF

PERSONALIEN

Martin Boos

Rümikerstrasse 34
8409 Winterthur

mboos@hsr.ch

Peter Heinzmann

Obere Bahnhofstrasse 32b
CH-8640 Rapperswil

pheinzma@hsr.ch

Jonas Hofer

Rümikerstrasse 34
8409 Winterthur

jhofer@bluewin.ch

Rolf Bader

Sekretariat
A-Z Verkehrsschule
Industriestr. 135 c
9200 Gossau

robader@hispeed.ch

ANHANG

PERSÖNLICHE BERICHTE

JONAS HOFER

PROJEKTVERLAUF

Das Projekt ist sehr gut angelaufen. Wir konnten schnell erste Erfolge erzielen und schon früh das Fahrzeug erkennen und mitverfolgen. Und schnell stand eine erste Version. Trotzdem konnten wir unsere Software nicht verifizieren, da wir nicht wussten wie lange die Distanzen auf den Videos waren und auch nicht mit welchen Geschwindigkeiten die Fahrer die Vollbremsung durchführten. Wir haben uns dann eine Zeit lang auf alles andere was neben dem eigentlichen Tracking noch nötig war konzentriert.

Als wir dann ca. in der Mitte der Arbeit die Zwischenpräsentation mit unserem Gegenleser Herr Augenstein hatten, mussten wir feststellen, dass wir unsere Daten genauer betrachten und verifizieren müssen. Entsprechend wurden Rohdatenanalysen durchgeführt und daraus viele Erkenntnisse gezogen

Gegen Ende bekamen wir dann Zugang zur eingesetzten Kamera und konnten unsere Software in Gossau unter verschiedenen Bedingungen testen. Leider funktionierte es nicht auf Anhieb und wir mussten jedes Mal wieder mit einem neuen Problem nachhause. Und dieses zum Teil bereits bis zum nächsten Tag lösen.

Beim letzten Test konnten wir dann aber glücklicherweise Erfolge verzeichnen und das System funktionierte wie gewünscht. Dies freute uns genauso wie unser Industriepartner und wir konnten das Projekt so erfolgreich abschliessen.

RÜCKBLICK

Wie in jedem Projekt lief auch bei uns nicht alles optimal. Die Tatsache, dass wir erst ca. vier Wochen vor Abgabe Zugang zum richtigen Bildmaterial bekamen, hatte seine Konsequenzen. Durch das neu eingesetzte Weitwinkelobjektiv traten Probleme auf die man anfangs unterschätzte. Sowohl von unserer Seite, als auch von Seiten des Betreuers und Industriepartners. Es traten neue Probleme auf, wie der Bildverzug. Aber auch andere Konzepte, die verwendet wurden funktionierten nicht mehr einwandfrei, was sich natürlich auf die Motivation auswirkte. Hätte man von Anfang an, oder viel früher auf die echten Gegebenheiten zu arbeiten können, hätte man Probleme früher erkannt und besser lösen können. Trotz allem sind wir aber stolz darauf, dass diese Software zum jetzigen Zeitpunkt einsatzfähig ist und ein gutes Grundgerüst für eine Weiterentwicklung bildet.

GELERNTES

Ich hatte viel Freude daran diese Arbeit zu schreiben und konnte viel für weitere Projekte lernen. Das tolle war, dass die Arbeit sehr vielschichtig war und wir so mit vielen Themen in Berührung kamen. Durch die Videoanalyse erhielt ich einen Überblick über die gängigen Videoanalyse Bibliotheken. Und konnte so erfahren, wie mächtig diese sind und was alles nötig ist, um in diesem Bereich ein gestecktes Ziel zu erreichen. Generell fand ich das Themengebiet sehr interessant und werde wohl auch im Privaten dieses Thema im Auge behalten.

Auch war es wichtig, dass die Informationen, die wir durch die Bibliotheken erhielten genau analysierten und so Mittel und Wege fanden die richtigen Schlüsse zu ziehen. Ich habe dadurch, und auch durch Anstoss von Herrn Augenstein, gemerkt dass es wichtig ist, dass man die Werte kritisch betrachtet und sich nicht gleich mit dem ersten Resultat zufrieden gibt.

Die Sprache .Net C# war mir bis anhin nicht unbekannt, jedoch hatte ich bis zum jetzigen Zeitpunkt noch keine Projekte mit C# .Net umgesetzt. Für mich war es sehr lehrreich einmal einen tieferen Einblick in die .Net Landschaft zu erhalten und damit zu Arbeiten. Glücklicherweise verfügte Martin bereits ein grosses Wissen darüber und konnte mir neue Konzepte schnell erklären.

Zudem hat das Projekt gezeigt, dass es wichtig ist, so früh wie möglich im realen Umfeld zu testen. Damit man Probleme frühzeitig erkennen kann.

FAZIT

Ich bin zufrieden mit der Arbeit. Es war eine anstrengende aber gute Zeit und ich bin froh, dass wir schlussendlich alle unsere gesteckten Ziele erreichen konnten.

MARTIN BOOS

PROJEKTVERLAUF

Die ersten zwei bis drei Wochen verbrachten wir damit, die Produktevaluation durchzuführen und beispielsweise einen Prototypen für GPS Zugriff zu programmieren. Später war dann klar, dass das Resultat mittels Videoanalyse erreicht werden muss, da dies die einzige Möglichkeit war, allen Anforderungen, vor allem der des passiven Systems (keine Modifikationen oder Erweiterungen an den Fahrzeugen), gerecht wird.

Die nächste Zeit suchten wir dann nach einer guten Möglichkeit Bewegungen zuerst zu erkennen und dann zu verfolgen. Ersteres war anfangs noch einfacher, doch beim Tracking kamen wir vorerst auf keinen grünen Zweig. Irgendwann entdeckte ich in den Tiefen des Internets jedoch ein OpenSource Projekt in C#, das eine Videoanalyse Tool war. Nach einigen Tests damit zeigte sich schnell, dass das Tracking, das die Programmierer verwendeten, sehr gut mit den Videoaufnahmen, die wir von Rolf Bader erhalten hatten, funktioniert. Es stellte sich heraus, dass zum Tracking ein Patternmatching Algorithmus verwendet wird, wodurch wir dann wussten, was zu tun ist.

Die Entwicklung ging zügig voran und wir machten gute Fortschritte. In den regelmässigen Sitzungen die wir mit Peter Heinzmann und Rolf Bader hatten, zeigten wir die gemachten Fortschritte, besprachen das Ergebnis und wie weiter fortzufahren ist.

Irgendwann waren wir soweit, dass wir nicht wussten, ob unsere errechneten Werte stimmen oder nicht. Wir erhielten deshalb von Rolf Bader neue Videos, worauf er jeweils nach der Vollbremsung sagte, mit welcher Geschwindigkeit diese durchgeführt wurde.

Während einer Sitzung kurz darauf, an der Peter Heinzmann und unser Gegenleser Oliver Augenstein anwesend waren, stellte sich heraus, dass wir etwas ziemlich schlechtes im Programm machten. Da nämlich durch das Bildweise Tracking (damals noch mit Videos die 50 Bilder pro Sekunde lieferten), entstanden bei der Berechnung Werte, die so nicht sein konnten. Dies hatten wir erkannt, nur war unsere Problemlösung die Sprünge in den Werten einfach „glattzubügeln“ eher suboptimal. Herr Augenstein hat uns deshalb geraten, in Excel eine Rohdatenanalyse durchzuführen. Als wir das taten stellten wir fest, dass uns das massiv weiterbringt, da wir nun wussten, wo und weshalb seltsame Werte entstanden und wie wir dies vermeiden konnten.

Wenige Wochen vor Projektende erhielt Rolf Bader die neue Kamera, die endgültig eingesetzt werden soll. Da diese über ein Weitwinkelobjektiv mit 110° Blickfeld in der Horizontalen besitzt, hatten wir nun plötzlich ein massives Problem. Denn durch den Weitwinkel wurden die Bilder so sehr verzogen, dass die Resultat unserer Berechnungen wieder völlig falsch waren.

Viele Stunden, Berechnungen und Diagramme später (Snickers hatten wir leider keine, obwohl alles ziemlich lange gedauert hat), hatten wir eine Möglichkeit gefunden den Bildverzug rechnerisch zu korrigieren. Und plötzlich nahmen die Resultate wieder Werte an, die uns Freude bereiteten. Allerdings hatten wir bis kurz vor Projektende das Problem, dass die Werte konstant um ca. 5 km/h abwichen. Den Grund dafür konnten wir einfach nicht finden. Als wir schon ziemlich demotiviert waren, da es aussah, als ob wir ein nicht richtig funktionierendes Projekt abgeben müssten, fand Jonas den kleinen Fehler mit der grossen Auswirkung.

Einige Tage darauf konnten wir nochmals Tests in Gossau SG durchführen. Es goss wie aus Kübeln und ich fürchtete, dass dadurch nichts mehr funktionieren wird. Zu meiner Überraschung war das allerdings nicht der Fall und es zeigte sich, dass unser Programm die Geschwindigkeit

der Fahrzeuge auf +/- 1 km/h richtig berechnet. Das ist genauer als der auf der Anlage installierte Radar, der bei Regen völlig falsche Resultate lieferte.

Danach mussten wir noch die Dokumentation zu Ende schreiben und alles für die Abgabe vorbereiten.

RÜCKBLICK

Anfangs hielt ich die zwei bis drei Wochen die wir mit der Produktevaluation verbrachten, für reine Zeitverschwendung. Wir sollten uns doch einfach für GPS Sensoren entscheiden. Als sich die Produktevaluation ergab, dass Videoanalyse verwendet werden soll, wollte ich trotzdem lieber mit GPS Sensoren arbeiten, da mir das spannender erschien. Dass ich falsch lag zeigte sich in der darauf folgenden Zeit, als wir mit Videoanalyse begannen zu arbeiten. Es machte doch sehr viel Spass Bewegungen in einem Video zu erkennen oder einen bestimmten Punkt programmatisch zu verfolgen, auch wenn es nicht immer ganz einfach war. Das Endresultat mit den Videos, auf die die Resultate gezeichnet werden ist auch viel besser, als wenn man stattdessen einfach eine Animation aufgrund GPS Daten gemacht hätte.

Problematisch war für uns, dass wir erst sehr spät Videomaterial der endgültigen Kamera erhielten, da uns die Probleme mit dem Bildverzug beinahe einen Strich durch die Rechnung machten. Die Beschaffung einer geeigneten Kamera hätte viel früher im Verlauf der Arbeit angetrieben werden müssen.

Das mit der Dokumentation war während des Studiums bei Übungsprojekten, die wir manchmal in verschiedenen Modulen durchführten, auch so eine Sache. Programmieren macht mehr Spass, warum soll man dann früher als nötig mit der Dokumentation beginnen? Deshalb fürchtete ich ein wenig, dass uns das auch bei der Bachelorarbeit passieren wird. Allerdings war es dann eher so, dass wir etwas zu früh mit der Dokumentation begannen. Wir mussten doch noch einige Kapitel ein paar Mal überarbeiten, als wir noch Änderungen an der Implementation vornahmen. Nun ist die Dokumentation so gut wie fertig und sie muss nurnoch gedruckt und geheftet werden, was wir noch heute, Donnerstag, machen werden. Morgen werden wir an der HSR noch das Plakat für die Ausstellung am Abend drucken, worauf ich mich sehr freue.

GELERNTES

Zum Glück arbeitete ich vor der Bachelorarbeit ein halbes Jahr mit einem Pensum von 100% als Softwareentwickler mit .NET Technologien. Ich habe dabei sehr viel gelernt und konnte deshalb während der Bachelorarbeit das Gelernte anwenden und mich auf Dinge wie Videoanalyse konzentrieren, anstatt hauptsächlich C# zu lernen.

Video- und Bildanalyse finde ich nun ziemlich spannend und ich habe einiges in Sachen Bewegungserkennung und Objektverfolgung gelernt.

Auch war es toll, vieles von dem was ich im Studium gelernt habe, auch endlich mal praktisch umzusetzen und nicht bloss mit Schulbeispielen, die meistens keinen praktischen Bezug hatten.

FAZIT

Es war ein tolles Projekt, doch bin ich auch froh, es nun abgeben zu können. Die letzten Wochen waren für mich und dadurch oft auch für meine Freundin anstrengend.

Zu wissen, dass unser Projekt von Rolf Bader nun produktiv verwendet und zusätzlich weiterentwickelt wird, fühlt sich gut an. Wir haben etwas geschaffen, das tatsächlich einen Nutzen bringt und zudem noch viel Potenzial birgt. Möglicherweise werden Jonas und ich selbst auch an der Weiterentwicklung beteiligt sein, was schon toll wäre. Es passiert mir schliesslich nicht jeden Tag, dass ich eine Bachelorarbeit durchführe, die danach noch weiterverwendet wird.

ENTWICKLUNGSUMGEBUNG

Als Entwicklungsumgebung wurde Microsoft Visual Studio 2010 Ultimate¹² in Kombination mit dem ReSharper Plugin von JetBrains¹³ benutzt.

Zur Versionskontrolle während der Entwicklung wurde der Microsoft Visual Studio Team Foundation Server 2010 benutzt. Der Team Foundation Server wurde auf einem virtuellen Windows 2008 Server installiert, der von der Hochschule Rapperswil im Rahmen der Bachelorarbeit zur Verfügung gestellt wurde.

DATENTRANSFER

Office Dokumente, Bilder und ähnliches wurden per Dropbox¹⁴ zwischen den einzelnen Parteien ausgetauscht.

VERWENDETE BIBLIOTHEKEN

Titel	Beschreibung	URL
AForge	C# Videoanalyse Bibliothek	http://www.aforogenet.com/
Emgu CV	C# Wrapper für Open CV (C++ Videoanalyse Bibliothek)	http://www.emgu.com/wiki/index.php/Main_Page
MediaSuite	Kostenpflichtige C# Bibliothek im Bereich Multimedia	http://streamcoders.com/
DB4O	C# Bibliothek um Objekte in einer Objektorientierter Datenbank abzulegen	http://db4o.com/
Microsoft Ribbon for WPF	Diese Software muss installiert werden, damit die Funktionsleiste in	http://www.microsoft.com/en-us/download/details.aspx?id=11877

¹²<http://www.microsoft.com/germany/visualstudio/products/team/visual-studio-ultimate.aspx> 13.06.12

¹³<http://www.jetbrains.com/resharper/> 13.06.12

¹⁴<https://www.dropbox.com/> 13.06.12

GRAPHICAL USERINTERFACE (GUI) KLASSEN

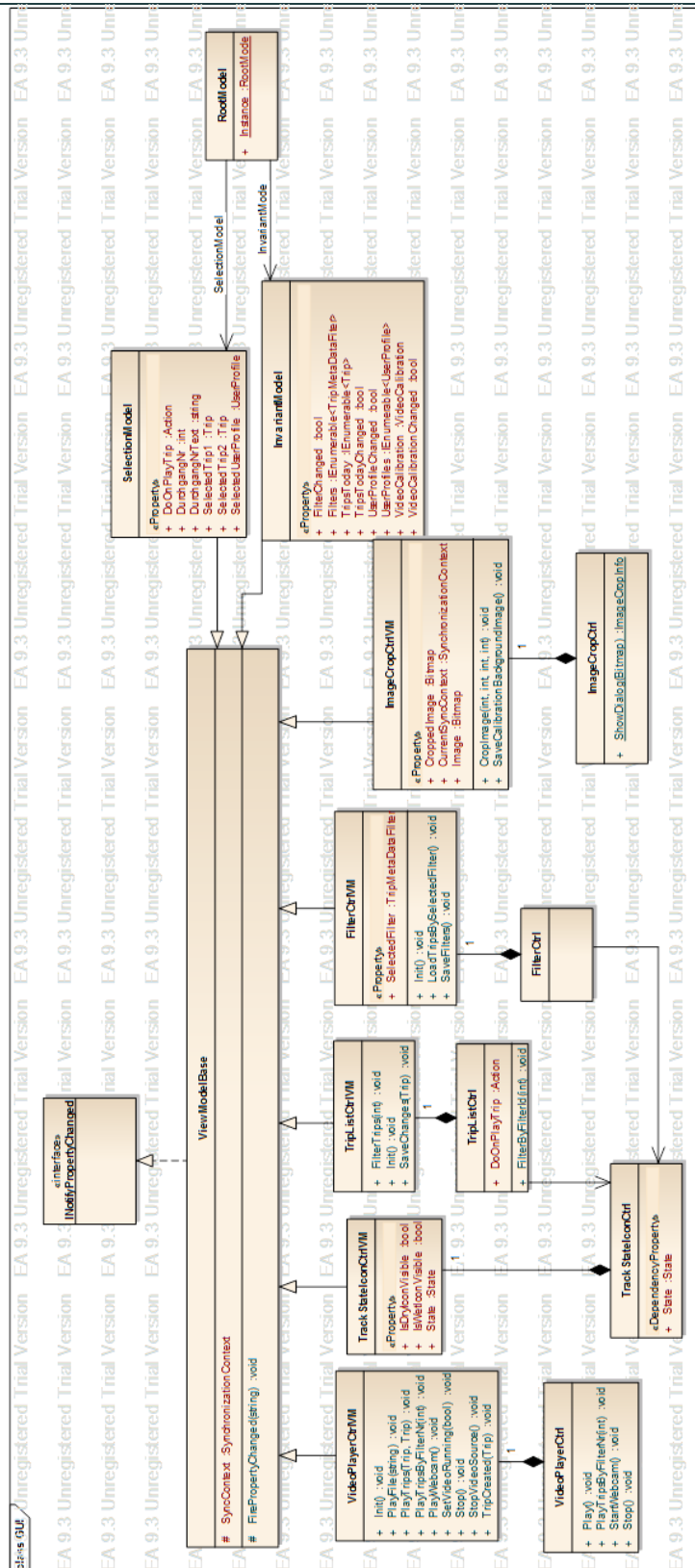
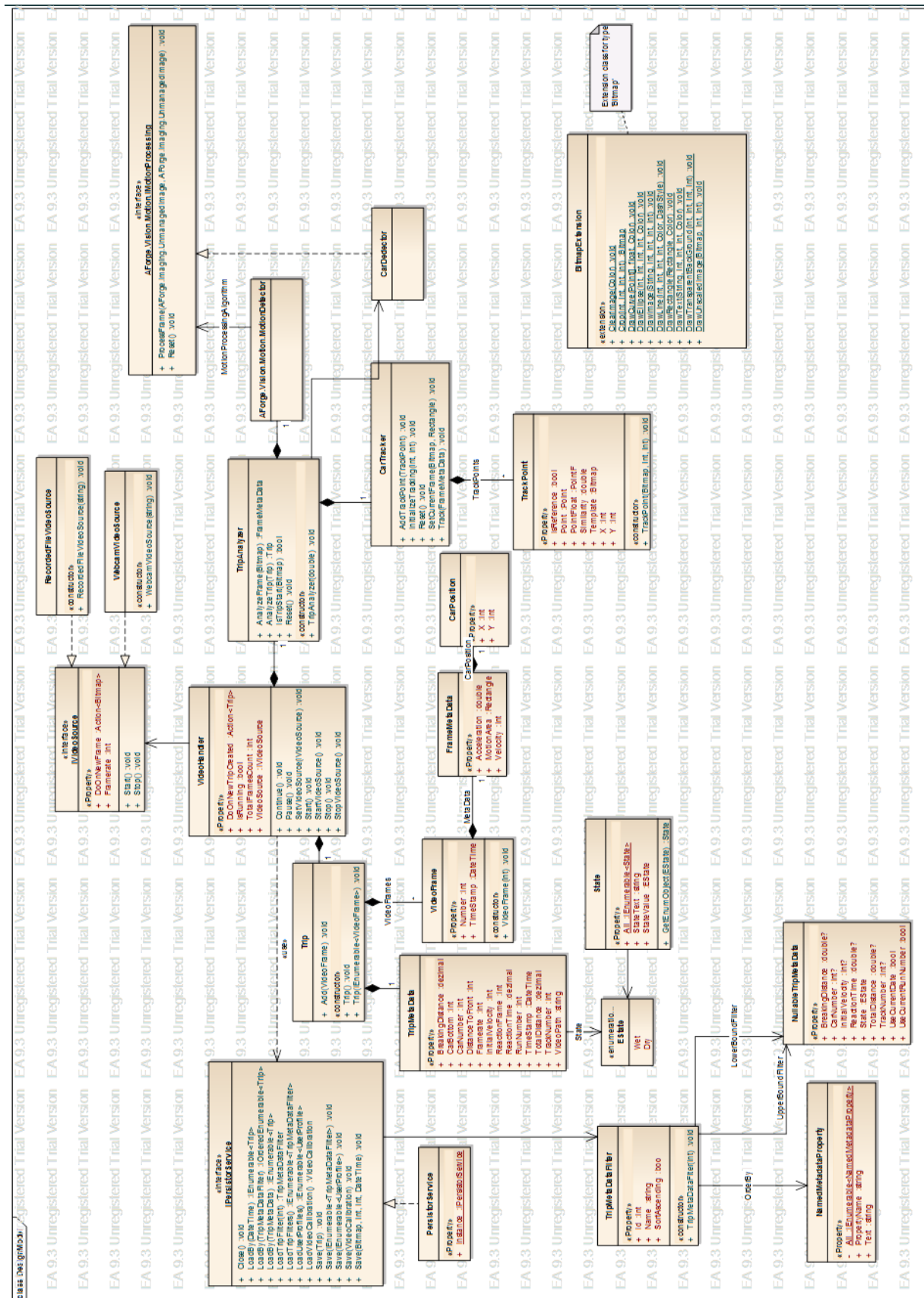


ABBILDUNG 49: KLASSENDIAGRAMM GUI

ABBILDUNG 50: DESIGN MODEL



EVALUATIONS DETAILS

ZIEL DER EVALUATION

Das Ziel dieser Evaluation war herauszufinden, welche Technologie am besten dafür geeignet ist ein fahrendes Auto in Echtzeit zu erkennen. Dabei müssen Position, Geschwindigkeit und Beschleunigung des Autos erfasst, verarbeitet und innerhalb einer vordefinierten Zeitspanne (ca. 30 – 60 Sekunden) dargestellt werden.

TECHNOLOGIEN

Folgende Technologien wurden der Analyse und Bewertung unterzogen.

Ubisense Ultra Wide Band Real-Time Location System (UWB RTLS)

Technologie zur präzisen Echtzeitortung von bewegenden Objekten auf der Basis von Ultra-breitband Technologie.

Dartfish

Software zur Videoanalyse in Echtzeit.

Emgu CV

Programmbibliothek zur Bildanalyse in Echtzeit.

Qstarz BT-Q1000eX

Hochgeschwindigkeits GPS Rundenzeitmesser.

Smartphone GPS

Der GPS Empfänger eines Smartphones wird zur Positionsbestimmung benutzt.

Oxford Technical Solutions RT 3000

Hochpräzises Trägheitsnavigationssystem zur Messung der Bewegung eines frei im Raum befindlichen Objektes.

BEWERTUNGSKRITERIEN

Die aufgezählten Technologien wurden nach folgenden Kriterien bewertet. Jedes der Bewertungskriterien wurde eingestuft als ‚sehr wichtig‘, ‚wichtig‘ oder ‚normal‘. So wirkt sich beispielsweise eine negative Bewertung in der als ‚wichtig‘ eingestuften Kategorie ‚Kosten‘ weniger stark auf die Gesamtbewertung einer Technologie aus, als eine negative Bewertung in der ‚sehr wichtig‘ eingestuften Kategorie ‚Datenrate‘.

Folgende Werte wurden zur Berechnung der Punktzahl jeder Kategorie verwendet

Wertung	Punkte
hervorragend	5

sehr gut	4
gut	3
genügend	2
ungenügend	1

Folgende Faktoren wurden zur Gewichtung der Kategorien verwendet

Gewichtung	Faktor
sehr wichtig	1.5
wichtig	1.25
normal	1

Die Gesamtpunktzahl einer Technologie in der jeweiligen Kategorie berechnet sich aus der Wertung der Technologie multipliziert mit dem Gewichtungsfaktor der Kategorie.

Daten- / Aktualisierungsrate

Die Geschwindigkeit mit welcher die bewertete Technologie Messwerte zur Analyse erfassen kann.

Diese Kategorie wurde als ‚sehr wichtig‘ eingestuft und hat daher den Gewichtungsfaktor 1.5.

Präzision

Die Position eines fahrenden Autos zu erfassen, dessen Geschwindigkeit im Minimum 20 Kilometer pro Stunde beträgt, bedeutet, dass eine bestimmte Zeitdauer zwischen den Messungen nicht überschritten werden darf, da sonst die erforderte Genauigkeit nicht erreicht wird.

Diese Kategorie wurde als ‚sehr wichtig‘ eingestuft und hat daher den Gewichtungsfaktor 1.5.

Kosten

Die Kostenspanne der bewerteten Technologien reicht von um die CHF 200.- für einen Qstarz BT-Q1000eX GPS Sensor¹⁵ bis ca. CHF 25'000.- für ein Ubisense UWB Research Package¹⁶. Aufgrund dieser Differenzen ist der Preis ein nicht zu vernachlässigendes Bewertungskriterium.

Diese Kategorie wurde als ‚wichtig‘ eingestuft und hat daher den Gewichtungsfaktor 1.25.

Installationsaufwand

Einen GPS Sensor in der Grössenordnung einer Zigarettenschachtel zu transportieren bedeutet weniger Aufwand, als die Installation einer bis mehreren Kameras inklusive Verkabelung zur Stromversorgung und Datenübertragung.

Diese Kategorie wurde als ‚wichtig‘ eingestuft und hat daher den Gewichtungsfaktor 1.25.

Datenzugänglichkeit

¹⁵ <http://www.k55.ch/qstarz-bt-q1000ex-10hz-xtreme-recorder-a-gps-10hz-high-speed-log-modus-bluetooth.html>, 02.03.12

¹⁶ http://mpg.ndlab.net/wp-content/uploads/2009/11/mocap_ubisense_locationresearchpackage.pdf, 02.03.12

Da die Verarbeitung und Darstellung der Daten innerhalb von 30 – 60 Sekunden nach deren Aufnahme abgeschlossen sein soll, ist ein möglichst einfacher und schneller Zugriff auf die Daten nötig.

Diese Kategorie wurde als ‚sehr wichtig‘ eingestuft und hat daher den Gewichtungsfaktor 1.5.

Entwicklungsschnittstellen

Neben den Anforderungen des Kunden ist es auch für die Entwickler der Endlösung wichtig, dass die verwendete Technologie gute Schnittstellen zur Entwicklung bietet und nicht vollkommen in sich geschlossen ist.

Diese Kategorie wurde als ‚sehr wichtig‘ eingestuft und hat daher den Gewichtungsfaktor 1.5.

Erweiterbarkeit

Zusätzlich zu den anderen Kriterien spielt auch die Erweiterbarkeit eine Rolle. Allerdings ist eine mögliche Erweiterbarkeit der Lösung an neue Muster (beispielsweise ein Ball der in die Fahrbahn geworfen wird), eher ein nice to have Feature, als eine unumgängliche geforderte Funktionalität.

Diese Kategorie wurde als ‚normal‘ eingestuft und hat daher den Gewichtungsfaktor 1.

AUSWERTUNG

Datenrate

Gewichtungsfaktor 1.5

Technologie	Einstufung	Gewichtete Bewertung	Bewertung
Ubisense UWB	hervorragend (5)	7.5	
Dartfish	hervorragend (5)	7.5	
Emgu CV	hervorragend (5)	7.5	
Qstarz BT-Q1000eX	sehr gut (4)	6.0	
Smartphone GPS	genügend (2)	3.0	
Oxford Technical Solutions RT 3000	hervorragend (5)	7.5	

Die meisten der evaluierten Technologien überzeugen durch ihre hervorragenden Datenraten. An der Spitze steht Oxford Technical Solutions RT 3000 mit einer Datenrate von 100 Hz (= 100 Updates / Sekunde), worauf Dartfish und Emgu CV folgen mit 60 Bildern pro Sekunde, welche von den Kameras geliefert werden (= 60 Updates / Sekunde), Ubisense UWB mit 39 Hz, Qstarz BT-Q1000eX mit 10 Hz und das Smartphone GPS welches als einziges mit seinen 5 Hz nicht überzeugen konnte.

Präzision

Gewichtungsfaktor 1.5

Technologie	Einstufung	Gewichtete Bewertung	Bewertung
Ubisense UWB	hervorragend (5)	7.5	

Dartfish	hervorragend (5)	7.5
Emgu CV	hervorragend (5)	7.5
Qstarz BT-Q1000eX	gut (3)	4.5
Smartphone GPS	ungenügend (1)	1.5
Oxford Technical Solutions RT 3000	hervorragend (5)	7.5

Oxford Technical Solutions RT 3000 überzeugt mit einer völlig ungeschlagenen Präzision in sämtlichen Bereichen¹⁷: Position (2cm), Geschwindigkeit (0.05 km/h), Richtung (0.1°), Beschleunigung (0.01 m/s²).

Ubisense UWB, Dartfish und Emgu CV haben ebenfalls hervorragende Präzision da sie ebenfalls im Minimum 39 (Ubisense) bis 60 (Dartfish) Messwerte / Sekunde erhalten. Qstarz BT-Q1000eX weist mit 10 Messwerten / Sekunde ebenfalls noch eine genügende Präzision auf (Position 2.5-3m). Lediglich das Smartphone GPS hat mit einer Präzision von 2-10m ungenügend abgeschnitten.

Kosten

Gewichtungsfaktor 1.25

Technologie	Einstufung	Gewichtete Bewertung
Ubisense UWB	ungenügend (1)	1.25
Dartfish	gut (3)	3.75
Emgu CV	sehr gut (4)	5.0
Qstarz BT-Q1000eX	hervorragend (5)	6.25
Smartphone GPS	sehr gut (4)	5.0
Oxford Technical Solutions RT 3000	ungenügend (1)	1.25

Wie man hier deutlich sieht, schlägt sich die hohe Präzision sowie Datenrate von Ubisense UWB (ca. CHF 25'000.-) und Oxford Technical Solutions RT 3000 (ca. 60'000 €) in einem sehr hohen Preis nieder, was sich entsprechend negativ auf die Bewertung auswirkt.

Installationsaufwand

Gewichtungsfaktor 1.25

Technologie	Einstufung	Gewichtete Bewertung
Ubisense UWB	genügend (2)	2.5
Dartfish	gut (3)	3.75
Emgu CV	gut (3)	3.75
Qstarz BT-Q1000eX	hervorragend (5)	6.25
Smartphone GPS	hervorragend (5)	6.25
Oxford Technical Solutions RT 3000	gut (3)	3.75

¹⁷ <http://www.oxts.com/default.asp?pageRef=7>, 02.03.12

Beim Installationsaufwand gewinnen klar die kleinen GPS Geräte, die leicht und einfach zu verstauen sind. Am schlechtesten schneidet hier Ubisense UWB ab, da dafür vier Empfangsstationen, sowie vier Sender am Auto installiert (und die Sender jeweils gewechselt) werden müssen.

Datenzugänglichkeit

Gewichtungsfaktor 1.5

Technologie	Einstufung	Gewichtete Bewertung
Ubisense UWB	hervorragend (5)	7.5
Dartfish	hervorragend (5)	7.5
Emgu CV	hervorragend (5)	7.5
Qstarz BT-Q1000eX	sehr gut (4)	6.0
Smartphone GPS	hervorragend (5)	7.5
Oxford Technical Solutions RT 3000	ungenügend (1)	1.5

Bei allen Technologien ausser der Oxford Technical Solutions RT 3000 können die Daten entweder per Ethernet, Wireless LAN oder Bluetooth gelesen werden. Da das Oxford Technical Solutions RT 3000 im Auto mitgeführt werden muss und nicht über eine kabellose Schnittstelle verfügt, führt das zu einer ungenügenden Bewertung.

Entwicklungsschnittstellen

Gewichtungsfaktor 1.5

Technologie	Einstufung	Gewichtete Bewertung
Ubisense UWB	hervorragend (5)	7.5
Dartfish	gut (3)	4.5
Emgu CV	hervorragend (5)	7.5
Qstarz BT-Q1000eX	hervorragend (5)	7.5
Smartphone GPS	hervorragend (5)	7.5
Oxford Technical Solutions RT 3000	ungenügend (1)	1.5

Die meisten Technologien verfügen über gute bis hervorragende Schnittstellen zur Entwicklung, wodurch eine Programmentwicklung auf Basis der gewählten Technologie kein Problem darstellen dürfte. Lediglich bei Oxford Technical Solutions RT 3000 ist unbekannt, ob Entwicklungsschnittstellen vorhanden sind und falls ja, wie diese genutzt werden können, was zu einer ungenügenden Bewertung führt.

Erweiterbarkeit

Gewichtungsfaktor 1.0

Technologie	Einstufung	Gewichtete Bewertung
Ubisense UWB	gut (3)	3.0
Dartfish	hervorragend (5)	5.0
Emgu CV	hervorragend (5)	5.0
Qstarz BT-Q1000eX	sehr gut (4)	4.0
Smartphone GPS	gut (3)	3.0
Oxford Technical Solutions RT 3000	ungenügend (1)	1.0

3000

Als Erweiterung gilt das Erkennen eines Balls, der vor das fahrende Auto geworfen wird.

Mit den meisten der evaluierten Technologien ist eine Erweiterung der Lösung gut machbar. Lediglich Oxford Technical Solutions RT 3000 erhält eine ungenügende Bewertung, da eine Erweiterung nur durch ein zusätzliches Gerät realisierbar wäre (der Ball kann sonst nicht erkannt werden), was wegen des hohen Anschaffungspreises nicht vertretbar ist.

ANFORDERUNGEN

FUNKTIONALE ANFORDERUNGEN

FAHRTEN AUFNAHME

- Das System erkennt automatisch neue Fahrten und zeichnet diese via Video auf.
- Fahrten werden dem aktuellen Durchgang zugewiesen.

FAHRTEN AUSWAHL

- Standard Auswahl ist immer die letzte aufgenommene Fahrt.
- Der Nutzer hat die Möglichkeit spezifisch Fahrten aus einer Liste auszuwählen. Dabei werden folgende Attribute dargestellt.
 - Zeitpunkt der Fahrt (Beginn der Videoaufzeichnung, Datum, Zeit; Zeit in 1/10-Sekunden Auflösung) |
 - Durchgang (fortlaufend nummeriert, jeden Tag neu beginnend) |
 - Nummer des Autos (Nummer 01 ... 99) |
 - Anfahrtsgeschwindigkeit (gemessen beim Aufleuchten des Rotlichts bzw. Auslösepunkts) [in km/h, ohne Kommastellen] |
 - Reaktionszeit [in Sekunden, mit einer Kommastelle bzw. 1/10-Sekunden Auflösung]
 - Bremsweg [in Metern, mit einer Kommastelle bzw. Dezimeter-Auflösung]
 - Anhalte strecke (Distanz Fahrzeugfront im Auslösepunkt bis Fahrzeugfront bei $v=0$ (angehalten) in Metern, mit einer Kommastelle bzw. Dezimeter-Auflösung
 - Fahrspurzustand (z.B. bewässert, trocken)
 - Fahrspurnummer
- Die Tabelle ist standardmässig nach dem neustem Durchgang gefiltert und nach Zeit Sortiert

Zeitpunkt	Durchgang	Nummer	Geschwindigkeit (km/h)	Anhalttestrecke (m)	Bremsweg (m)	Reaktionszeit (s)	Fahrspurzustand	Fahrspurnummer
10:03 19.03.12	1	1	40	23,0		0.5	nass	1

FAHRTEN VISUALISIERUNG

- Fahrten werden als Video und die daraus berechneten Informationen mit graphischem Overlay dargestellt und einem Graphen dargestellt.
- Es können zwei Fahrten miteinander verglichen werden.
- Es können bis zu zwei Fahrten aus einer Liste aller Fahrten gewählt werden
- Die Liste kann sortiert und gefiltert werden
- Optional ist es möglich den Trackingalgorithmus zu visualisieren.

Weitere Details können aus dem Papierprototyp entnommen werden

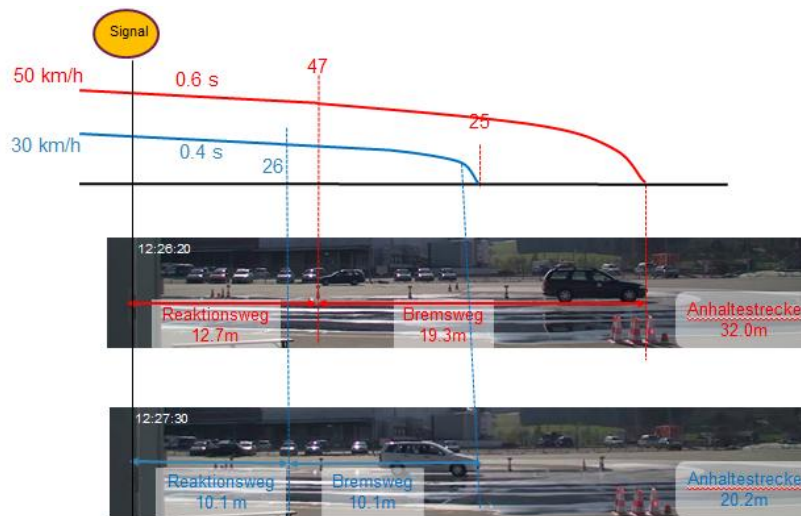


ABBILDUNG 51: FINALER PAPERPROTOTYPE

DURCHGANG

- Für die erste Fahrt am Tag wird automatisch ein neuer Durchgang erstellt.
- Der Beginn eines neuen Durchgangs wird manuell über das GUI gestartet.

KONFIGURATION

- Referenzpunkte die für die Konfiguration der Applikation nötig sind können visuell auf einem Referenzbild festgelegt werden.
- Standard Filterung und Sortierung kann angepasst werden.
- Ein-/Ausblenden der Videoanalyse möglich(für Demonstrationszwecke)

NICHTFUNKTIONALE ANFORDERUNGEN

- Der Bremsweg muss auf 20cm genau bestimmt werden können.
- Die Videoanalyse muss nach 30 Sekunden abgeschlossen sein.
- Einfache Bedienung (ohne Konsultation eines Handbuchs, sollen Fahrlehrer nach einer halbstündigen Einführung das System bedienen können)
- Genau eine Fahrt kann zu einem Zeitpunkt aufgenommen und verarbeitet werden
- Keine Datenschutzrelevanten Daten (z.B. Nummernschild, Personendaten) werden erfasst
 - Personen werden von der Seite nicht erkannt
 - Nummernschild wird von der Seite nicht erfasst

PAPIERPROTOTYPEN

Die folgenden Paperprototypes sind chronologisch aufgelistet.

Anfangs ging man von einer Animation aus, um die Fahrten zu visualisieren.

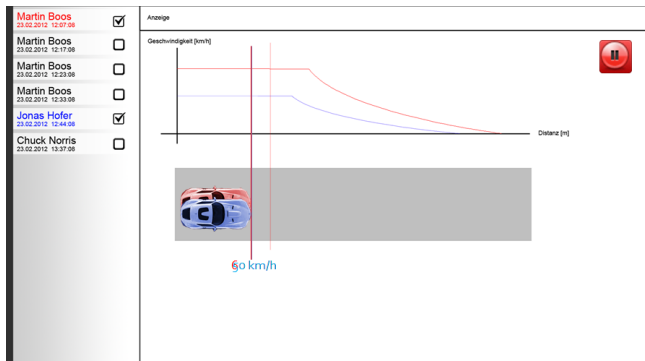


ABBILDUNG 52: PAPERPROTOTYPE VOLLBREMSUNG

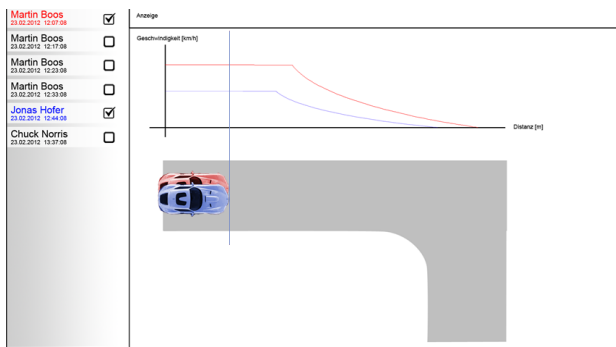


ABBILDUNG 53: PAPERPROTOTYPE KURVENFAHRT

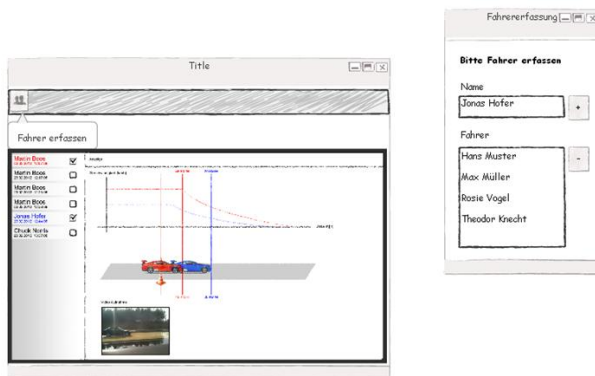


ABBILDUNG 54: PAPERPROTOTYPE FAHRTENVERGLEICH VERSION 1

Man kam dann zum Entschluss, dass es wichtig ist, dass Kennzahlen wie Reaktionsweg eingeblendet werden.

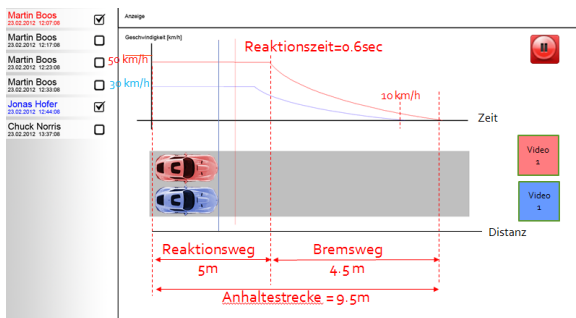


ABBILDUNG 55: PAPERPROTOTYPE FAHRTENVERGLEICH VERSION 2

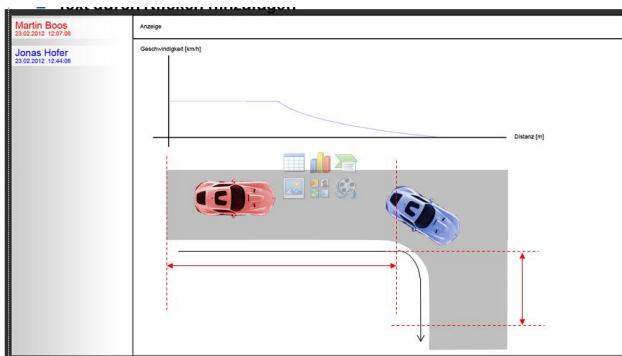


ABBILDUNG 56: PAPERPROTOTYPE KURVENFAHRTVERGLEICH

Weitere Gespräche mit dem Kunden ergaben, dass es wichtig ist, dass sich der Fahrer selber wieder erkennt. Eine Animation ist also nicht ausreichend. Man hat sich deshalb dafür entschieden, dass man das die Aufnahme anstatt der Animation verwendet und darauf die Informationen einblendet.

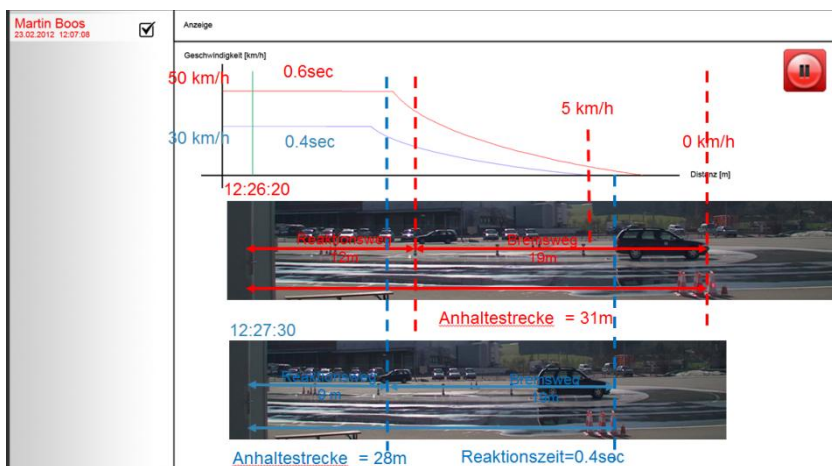


ABBILDUNG 57: PAPERPROTOTYPE FAHRTENVERGLEICH MIT VIDEO

Später hat man noch gemerkt, dass die Kurve nicht der Realität entspricht. Diese wurden zum besseren Verständnis angepasst.

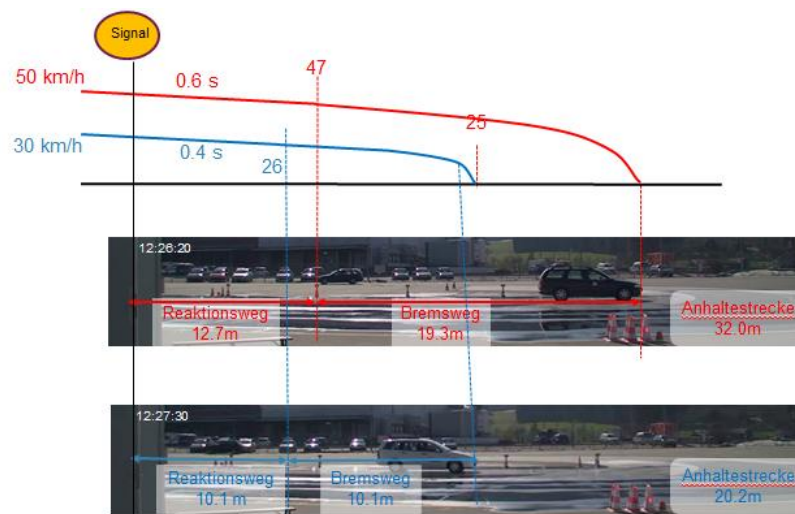


ABBILDUNG 58: PAPERPROTOTYPE FAHRTENVERGLEICH MIT VIDEO FINALE VERSION

POSTER

Aufnahme- und Visualisierungssystem für Fahrtrainingszentren

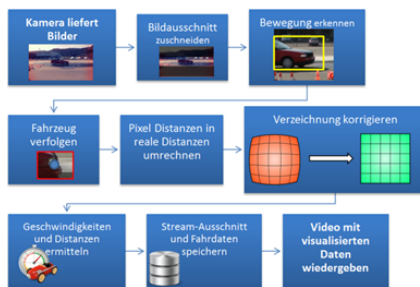
Bachelorarbeit Frühjahrssemester 2012
 Themengebiet: Videoanalyse



Betreuer: Prof. Dr. P. Heinzmann
 Experte: Dr. Th. Siegenthaler,
 Projektpartner: Rolf Bader, A-Z Verkehrsschule

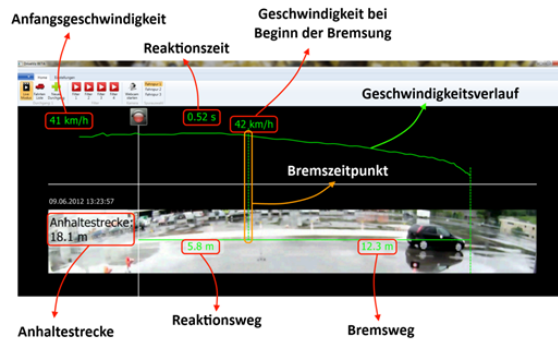
Ziel

Ein System zur Visualisierung von Vollbremsungen an Fahrtrainingszentren.



Funktionsumfang

- Reaktionszeit und –Weg ermitteln
- Geschwindigkeit ermitteln
- Bremszeitpunkt ermitteln
- Fahrten vergleichen
- Fahrten Filtern und sortieren



Technologien



PROJEKTMANAGEMENT

SCRUM¹⁸

Unseren Arbeitsprozess haben wir an Scrum angelehnt. Im Wohnzimmer der WG wurde ein Taskboard aufgehängt, mit welchem wir unsere Task koordinierten. Die Sprints wurden auf die Länge zwischen den Meetings mit dem Betreuer und Industriepartner gesetzt, was im Schnitt zwischen 1- 2 Wochen andauerte. So hatte man immer ein überschaubares Arbeitspaket.



ABBILDUNG 59: SCRUM TASKBOARD

PROJEKTPLAN

Anfang des Projekts wurde ein grober Zeitplan erstellt an welchen man sich grösste Zeit halten konnte. Der Entwicklungsprozess dauerte aber länger als geplant, das hing damit zusammen, dass wir bis ca. 2 Wochen vor Schluss mit der Verzeichnung des Bildes zu kämpfen hatten.

¹⁸ <http://de.wikipedia.org/wiki/Scrum>

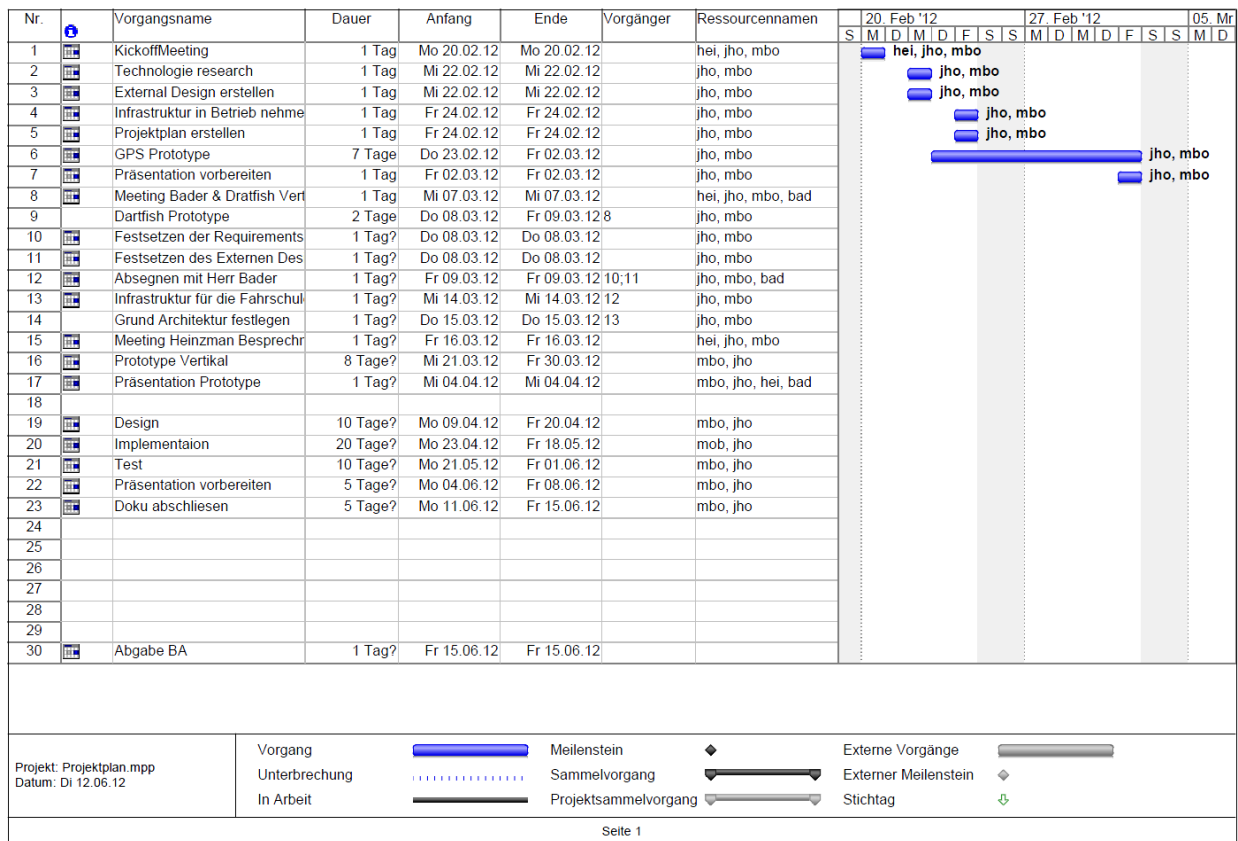


ABBILDUNG 60: PROJEKTPLAN SEITE 1

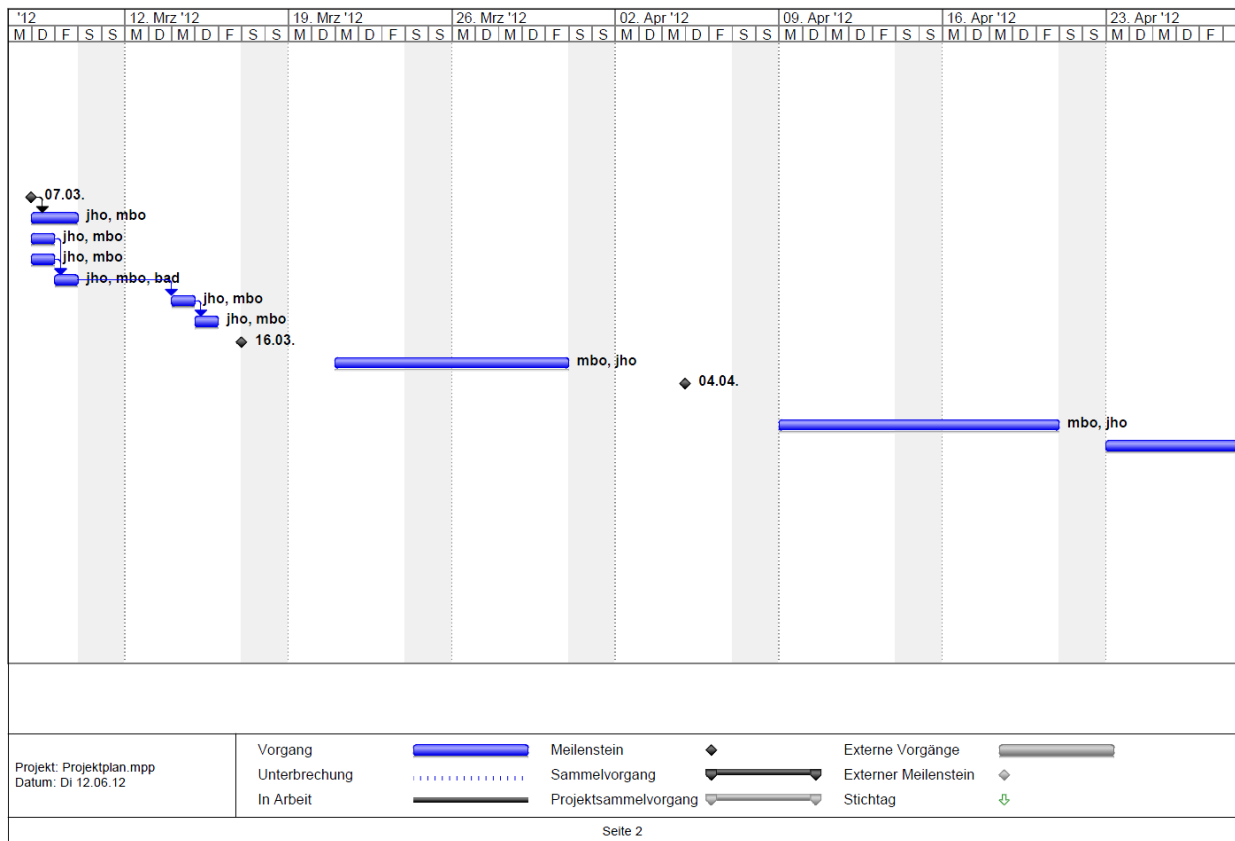


ABBILDUNG 61: PROJEKTPLAN SEITE 2

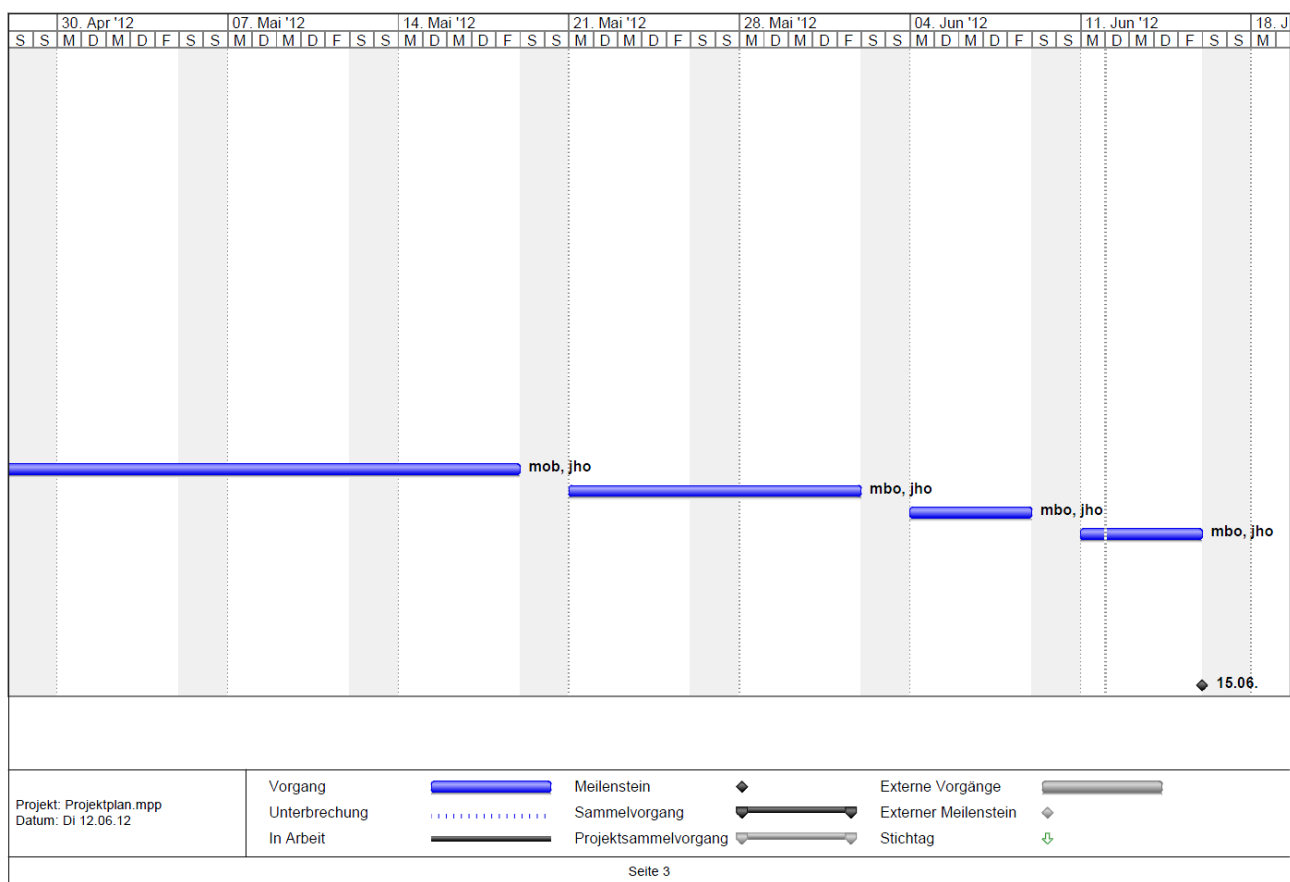


ABBILDUNG 62: PROJEKTPLAN SEITE 3