

Development of a real-time monitoring infrastructure for critical traffic situations for use at a research intersection

Bachelorarbeit

Author:

Ben Kantwerk

Betreut durch:

Prof. Dr. habil. Rudolf Kruse – Otto-von-Guericke-Universität Magdeburg

M.Sc. Chakajkla Jesdabodi – Deutsches Zentrum für Luft- und Raumfahrt e.V

Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Abbildungsverzeichnis.....	III
Abkürzungsverzeichnis.....	IV
Abstract.....	V
1 Einleitung.....	1
1.1 Motivation.....	1
1.2 Ziele der Arbeit.....	2
1.3 Übersicht der Kapitel.....	3
2 Theoretische Grundlagen.....	4
2.1 Erkennung kritischer Verkehrssituationen.....	4
2.2 Anwendungsplattform Intelligente Mobilität.....	6
2.3 Aufbau der Forschungskreuzung Braunschweig.....	8
3 Entwurf.....	10
3.1 Beschaffenheit der Daten.....	10
3.2 Anforderungen an das System.....	12
3.3 Das IPC-Framework.....	13
3.4 Konzeptueller Systemablauf.....	15
4 Implementation.....	19
4.1 Verwendete Software und Bibliotheken.....	19
Visual Studio 2010 / gedit.....	19
Jenoptik IPC-Framework.....	19
Boost.....	19
IPC Offline-Entwicklungsumgebung.....	19
4.2 Implementationsdetails.....	20
4.2.1 Organisation der Datenmengen.....	20
4.2.2 Anwenden von Metriken.....	22
4.2.3 Auswertung der Ergebnisse.....	25
4.2.4 Speicherung der Ergebnisse.....	26
4.2.5 Bereits implementierte Erkennungsmodule und Metriken.....	27
4.2.6. Teilweise implementierte Metriken.....	28
4.3 Systemtests.....	29
4.4 Systemgrenzen.....	29
5 Zusammenfassung und Auswertung.....	32
6 Ausblick.....	34

Inhaltsverzeichnis

6.1 Einbindung von Ampel­daten.....	34
6.2 Nutzung neuer Daten­quellen.....	34
6.3 Erweiterung durch weitere Kern­module.....	35
6.4 Unterscheidung anhand des Fahrzeug­typs.....	36
6.5 Implementation weiterer Metriken.....	36
6.6 Weiterleitung der Analyse an Verkehrsteilnehmer.....	37
Literaturverzeichnis.....	39

Abbildungsverzeichnis

Abbildung 1: Unfalltodstatistik 2012, aus [Statistik 2012].....	1
Abbildung 2: Hierarchische Einteilung von Verkehrseignissen, aus [Hyden 1987].....	4
Abbildung 3: Fünf Aspekte der AIM, aus [Lemmer 2013].....	6
Abbildung 4: Technik der Forschungskreuzung. Bestehend aus Kameras (rot), Verteiler (grün) und Infrarot-Beleuchtung (blau). Radar an der Vorderseite des Mastes ist nicht sichtbar. Aus [DLR Fokr 2014].....	8
Abbildung 5: Bildhafte Funktionsweise eines IPC-Systems, aus [IPC-Reference 2011].....	14
Abbildung 6: Sequenzdiagramm eines Systemdurchlaufes.....	18
Abbildung 7: Berechnung der PET, aus [Laureshyn 2010].....	37
Abbildung 8: Darstellung der C2X Kommunikation. Dargestellt ist die Kommunikation zwischen den Fahrzeugen (gelb), alternative Fahrrouten (grün), sowie eine zurzeit blockierte Route (rot). Aus [C2X 2014].....	38

Abkürzungsverzeichnis

AIM	Anwendungsplattform Intelligente Mobilität
C2X	Car to X (beinhaltet Fahrzeug zu Fahrzeug und Fahrzeug zu Infrastruktur)
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V
Fokr	Forschungskreuzung
FSRM	Fusion Server Receiver Module (Fusion Server Empfängermodul)
IAST	Inner Area Stay Time (Innen-Aufenthaltszeit)
IDE	integrated development environment (Integrierte Entwicklungsumgebung)
IPC	inter-process communication (inter-prozess Kommunikation)
LSA	Lichtsignalanlage
MCCM	Metric Central Control Module
MSS	Multi-Sensor-System
PET	Post Encroachment Time (Post Beeinträchtigungs Zeit)
RRM	Result Report Module
TTC	Time to Collision (Zeit bis zu einer Kollision)

Abstract

In dieser Arbeit geht es um die Entwicklung und Implementierungen eines Systems zur Erkennung kritischer Verkehrssituationen aufgrund von Trajektorien- und Positionsdaten des Teilprojekts „Forschungskreuzung“ des Deutschen Zentrums für Luft- und Raumfahrt.

Aufgrund von theoretischen Grundlagen zur Situationserkennung mithilfe von Metriken wird ein System entwickelt, welches Datensätze der Forschungskreuzung in Echtzeit auswertet und für spätere Auswertungen speichert. Die Systemarchitektur und die für die Arbeit verwendeten Algorithmen werden dabei im Detail erklärt und weitere zukünftige Ansätze für die Erweiterung des Systems werden vorgeschlagen.

1 Einleitung

1.1 Motivation

Durch die zunehmende Weiterentwicklung und Verbesserung des Autos hat der Begriff der Mobilität ein wichtiges Thema der Verkehrsforschung geworden. Der zunehmend höheren Mobilität der Verkehrsteilnehmer folgt ein höheres Verkehrsaufkommen, wodurch neue Ansätze gefunden werden müssen, um den Verkehrsablauf sicher für alle, sowohl motorisiert als auch nicht-motorisiert, zu gestalten.

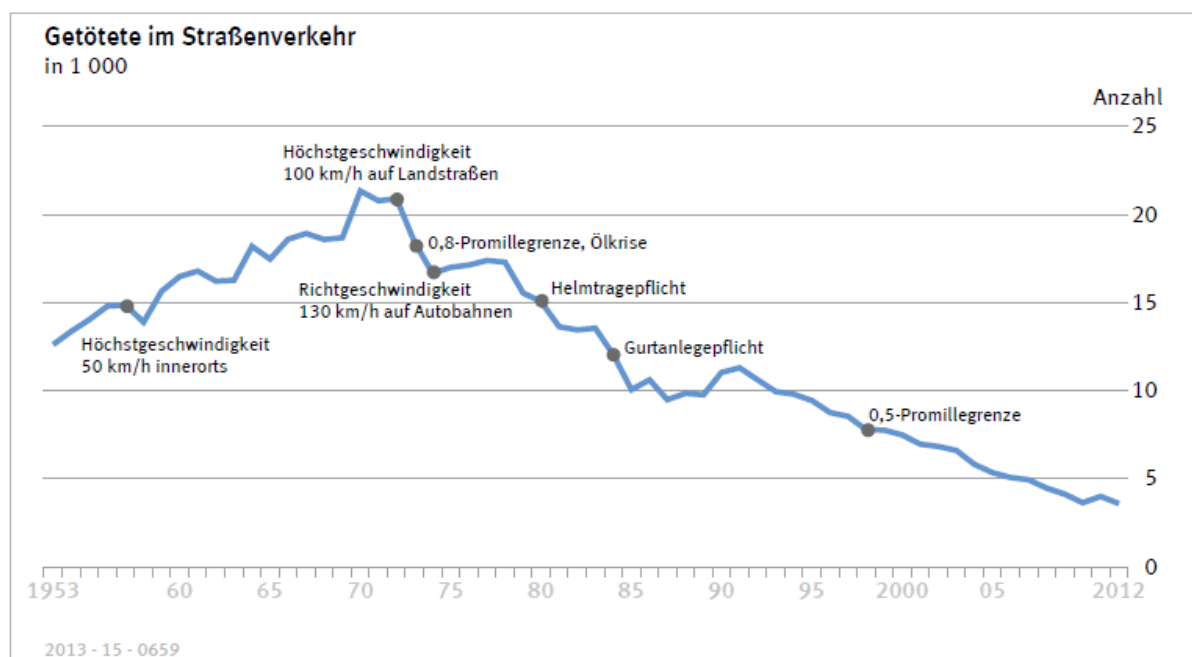


Abbildung 1: Unfalltodstatistik 2012, aus [Statistik 2012]

Die Tendenz der Unfalltoten im Straßenverkehr, sowie die Beeinflussung dieser Tendenz aufgrund der Einführung gesetzlicher Verkehrsmaßnahmen, ist in Abbildung 1 sichtbar. Nach einer Statistik des Statistischen Bundesamtes sank die Zahl der Unfalltoten im Jahr 2012 um 2,2% im Vergleich zum Vorjahr. Die Zahl der Gesamtunfälle stieg jedoch um 1,7%, siehe [Statistik 2012]. Obwohl nicht direkt in der Abbildung vermerkt, zeigt die Abbildung

außerdem den Einfluss von technischer Neuerungen auf das Vorkommen von Personenschäden bei Unfällen. Wesentliche Beispiele in dieser Hinsicht sind die Einführung des Tempomat im Jahr 1958, sowie die Einführung des Anti-Blockiersystems im Jahr 1978.

Zur Vermeidung leichter Unfälle können moderne Fahrassistenzsysteme ansetzen, indem sie fehlerhaftes Fahrverhalten korrigieren oder den Fahrer über bevorstehende Verkehrereignisse informieren (z.B. bremsenden Verkehr oder bevorstehende Baustellen). Diese Informationen können an nachfolgende Verkehrsteilnehmer weitergeleitet werden, um so das Unfallrisiko zu vermindern.

Als Basis für die Entwicklung neuer Verkehrssysteme ist es entscheidend herauszufinden, welches Fahrerverhalten kritische Verkehrssituationen verursacht und anhand welcher Kriterien diese frühzeitig erkannt werden können. Die Forschungskreuzung (Fokr) des Deutschen Zentrums für Luft- und Raumfahrt, als Teilprojekt der Anwendungsplattform Intelligente Mobilität (AIM), erlaubt eben diese Art der Forschung, indem sie die benötigte Infrastruktur zur Verfügung stellt, um den realen Straßenverkehr der Stadt Braunschweig zu beobachten und Informationen über das Verkehrsgeschehen zu sammeln.

1.2 Ziele der Arbeit

Das Ziel dieser Arbeit ist der Entwurf und die Implementation eines Systems zur Erkennung kritischer Verkehrssituationen anhand von Trajektorien- und Positiondaten diverser Verkehrsteilnehmer. Die Forschungskreuzung der Stadt Braunschweig dient als Quelle für die auszuwertenden Daten. Ein wichtiges Anforderungskriterium dieser Arbeit ist hierbei, dass die Situationserkennung des implementierten Systems in Echtzeit stattfindet soll. Auf diese Weise soll eine möglichst frühzeitige Erkennung von kritischen Verkehrssituationen gewährleistet und eine Grundlage für weitere Entwicklungen auf dem Gebiet der Verkehrsforschung geschaffen werden.

Weiterhin soll die Architektur des Systems es ermöglichen, neue Erkennungsalgorithmen, Datenquellen und Verkehrsmetriken möglichst einfach in den bestehenden Systemablauf zu integrieren. Durch diese Erweiterbarkeit des Systems soll eine Basis geschaffen werden,

welche als Grundlage für weitere Arbeiten und Forschungen innerhalb des AIM-Projektes dient.

1.3 Übersicht der Kapitel

Das nachfolgende Kapitel beschäftigt sich zunächst mit den theoretischen Grundlagen dieser Arbeit. Dabei wird besonderer Wert auf grundlegende Definitionen von kritischen Situationen, als auch systematische Methoden zur Erkennung und Klassifizierung solcher eingegangen. Desweiteren werden bisherige Arbeiten des Deutschen Luft- und Raumfahrtzentrums auf dem Gebiet der Verkehrsforschung, sowie der grundlegende Aufbau der Forschungskreuzung beleuchtet.

Im dritten Kapitel geht es um den theoretischen Entwurf der Systemarchitektur. Es wird auf die Beschaffenheit der Trajektorien Daten eingegangen und Anforderungen formuliert, denen das System gerecht werden muss. Abschließend wird mithilfe von UML Diagrammen der Aufbau, sowie der allgemeine Ablauf des Systems dargestellt und verdeutlicht. Desweiteren wird auf das IPC-Framework eingegangen, welches als Grundlage für die Entwicklung des Systems dient.

Kapitel Vier befasst sich mit der Implementierung des Systems. Insbesondere werden verwendete Algorithmen im Detail erklärt und begründet, in wieweit diese Ansätze die Anforderungen aus dem dritten Kapitel erfüllen. Außerdem werden die im Rahmen dieser Arbeit entwickelten Module zur Situationserkennung, sowie eventuelle Probleme bei der Entwicklung beschrieben.

Kapitel fünf dient der abschließenden Zusammenfassung der Arbeit, sowie eine Beurteilung der Ergebnisse.

Das sechste und letzte Kapitel soll als Ausgangspunkt für mögliche Erweiterungen des Systems dienen. Es werden mehrere potentielle Ansätze beschrieben und erklärt, auf welche Art und Weise sie in das existierende System integriert werden können.

2 Theoretische Grundlagen

2.1 Erkennung kritischer Verkehrssituationen

Ein Verkehrskonflikt, nach [Amundsen/Hyden 1977], ist „an observable situation in which two or more road users approach each other in space and time to such an extent that a collision is imminent if their movements remain unchanged“, also eine beobachtbare Situation in der sich mindestens zwei Verkehrsteilnehmer derart annähern, das sie miteinander kollidieren, wenn ihre Bewegungen unverändert bleiben.

In Abbildung 2 wird die allgemeine Häufigkeit von Verkehrskonflikten und Unfällen hierarchisch dargestellt. Im Sinne dieser Arbeit sind eben jene Situationen von Bedeutung, welche durch ein verändertes Fahrerverhalten hätten verhindert werden können. Das betrifft sowohl tatsächlich geschehene Unfälle, als auch Situationen in denen ein Unfall nur knapp verhindert worden ist. Die frühzeitige Erkennung dieser Situationen ist für die Verkehrssicherheit von großer Wichtigkeit.

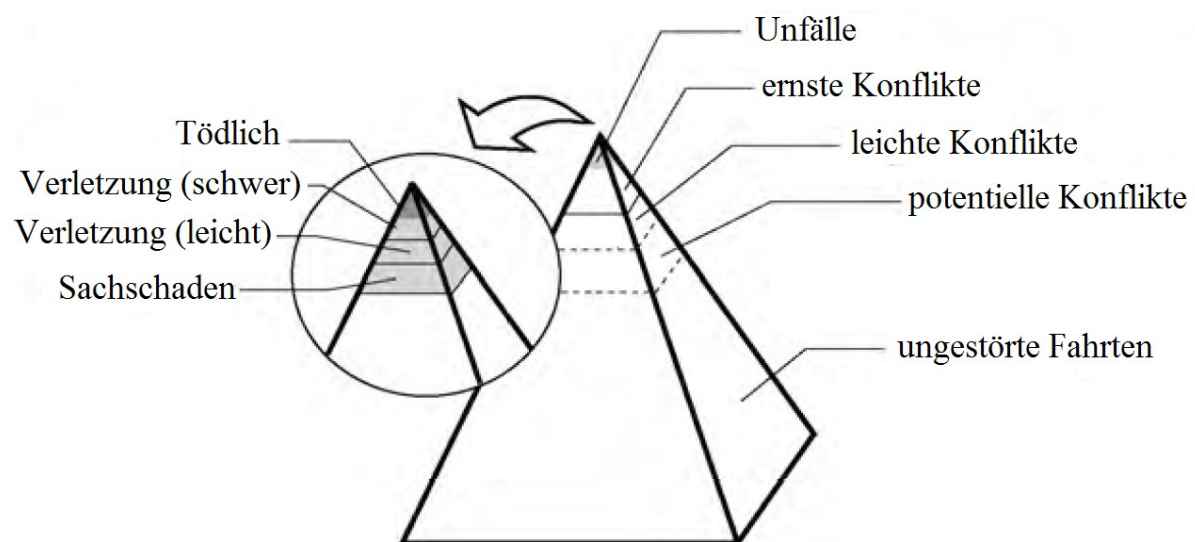


Abbildung 2: Hierarchische Einteilung von Verkehrssereignissen, aus [Hyden 1987]

Catalá-Prat teilt Erkennungsverfahren in zwei wesentliche Ansätze: den regelbasierten Ansatz, sowie den Ansatz über Atypische Ereignisse (siehe [Catalá-Prat 2010]). Diese sollen im folgenden kurz erklärt werden:

Regelbasierter Ansatz

Bei diesem Ansatz geht es hauptsächlich um das Ableiten von Regeln, anhand derer eine kritische Situation erkannt werden kann. Hierzu gehören insbesondere diverse Fahrzeugassistenzsysteme, welche jeweils auf bestimmte Gefahrensituationen spezialisiert sind. Ein Beispiel hierfür sind Systeme zur Kollisionswarnung. Anhand einer Kamera kann ein solches Warnungssystem Metriken wie Time-to-Collision berechnen, anhand welcher die Kritikalität einer Situation beurteilt werden kann (siehe [Dagan et al. 2004]).

Die Auswertung dieser Metriken findet meist über statische Schwellenwerte statt, anhand welcher die Daten ausgewertet werden. Auch das im Rahmen dieser Arbeit entstandene System arbeitet ebenfalls über einen Ansatz mit Metriken und statischen Schwellenwerten. Allerdings gibt es weitere Strategien zur Auswertung, wie die dynamische Berechnung von Schwellenwerten über eine Monte-Carlo-Simulation (siehe [Yang et al. 2003]).

Atypische Ereignisse

Dieser Ansatz beschäftigt sich insbesondere auf Abweichung von der Norm. Anders als der regelbasierte Ansatz werden also die entsprechenden Situationen nicht direkt definiert. Eine Möglichkeit der Erkennung ist, gesammelte Fahrzeugtrajektorien zu clustern. Trajektorien, welche nicht in eines der Cluster passen, werden als atypisch klassifiziert (siehe [Fu et al. 2005]). Nachteil dieser Methode ist, dass das bloße Vorkommen eines atypischen Ereignisses keine Aussage über die Kritikalität einer Situation gibt, d.h ein atypisches Ereignis muss nicht unbedingt gefährlich für andere Verkehrsteilnehmer sein.

2.2 Anwendungsplattform Intelligente Mobilität

Die Anwendungsplattform Intelligente Mobilität (AIM) ist ein Großforschungsprojekt innerhalb der Stadt Braunschweig und deren näherer Umgebung. Durch diese Kooperation bietet das AIM-Projekt wesentliche Vorteile gegenüber konventionellen Ansätzen der Verkehrsforschung. Durch die großräumige Forschungsumgebung, sowie die Forschung am realen Straßenverkehr ermöglichen es, Forschungsergebnisse leicht anhand von Realsituationen zu testen. Der Einsatz von Fahrsimulationen ermöglicht die frühzeitige Validierung von Prototypen, sowie die Prognose über zukünftiges Fahrverhalten. Das Projekt teilt sich in fünf wesentliche Aspekte, welche in Abbildung 3 dargestellt und im folgenden näher beschrieben werden sollen.

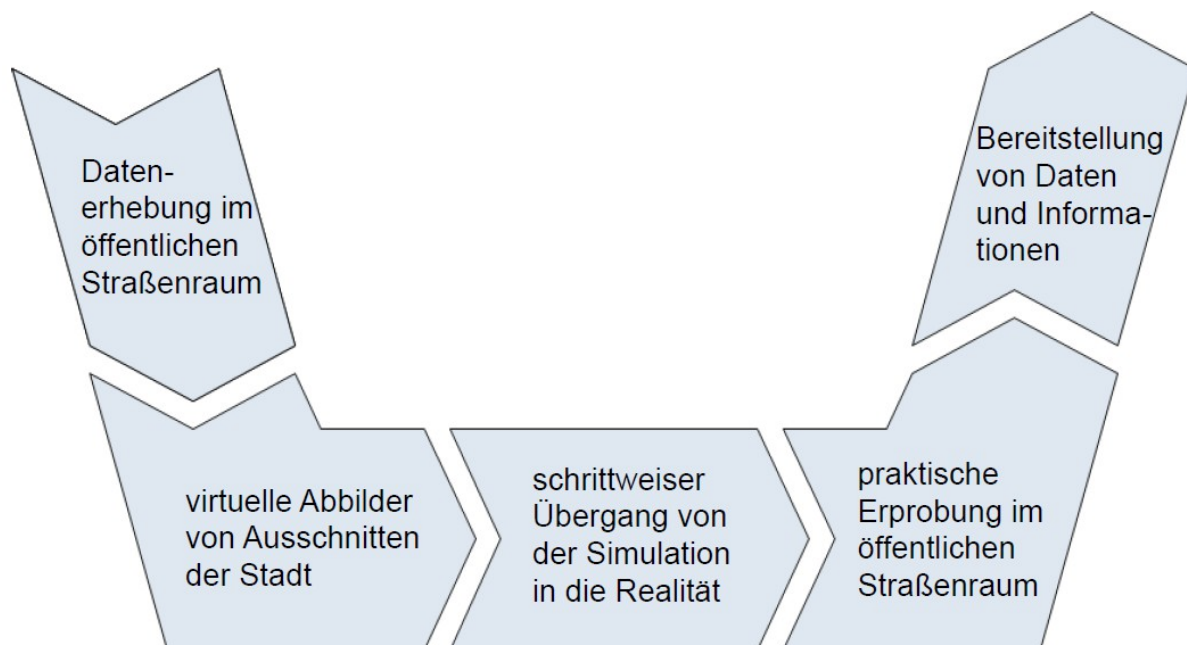


Abbildung 3: Fünf Aspekte der AIM, aus [Lemmer 2013]

1. Datenerhebung im öffentlichen Verkehrsraum

Ein zentrales Beispiel für die Datenerhebung von AIM ist die Forschungskreuzung Braunschweigs, welche als Ausgangspunkt für diese Arbeit dient. Durch die Beobachtung des Straßenverkehrs können reale Daten über das Fahrverhalten gesammelt werden. Diese Daten ermöglichen es, sowohl Systeme zur Identifikation von kritischen Verkehrssituationen zu entwickeln, als auch Fahrsituationen zu rekonstruieren, um realitätsnahe Simulationen zu entwickeln.

2. Virtuelle Abbilder von Abschnitten der Stadt

Diese Simulationen können wiederum für die Erstellung von Propandenstudien und Verkehrsmodellen der Stadt Braunschweig genutzt werden.

3. Schrittweiser Übergang von der Simulation in die Realität

Eine Anforderungsanalyse für Prototypen neuer Assistenzsysteme können innerhalb von Fahrsimulatoren erstellt werden, bevor ein realer Test im Straßenverkehr durchgeführt wird. Ein Beispiel solcher Simulatoren ist der MoSAIC-Fahrsimulator des AIM-Projektes, welcher es erlaubt, mehrere Probanden an einer Simulation teilnehmen zu lassen und so die Entwicklung kooperativer Assistenzsysteme zu erleichtern.

4. Praktische Erprobung im öffentlichen Straßenraum

Für den Realtest ist eine Referenzstrecke der Stadt Braunschweig mit der nötigen Infrastruktur ausgestattet, um Informationen an Testfahrzeuge weiterzuleiten und den Verkehrsablauf zu steuern. Der Status der Lichtsignalanlagen kann ebenfalls abgerufen werden und an Fahrzeuge übermittelt werden, um zu hohes Verkehrsaufkommen und Staus an Kreuzungen zu vermindern.

5. Bereitstellung von Daten und Informationen

Die Bereitstellung der Informationen ermöglicht es, z.B öffentliche Verkehrsmittel in das Kommunikationsnetz zu integrieren. Durch die Untersuchung von Priorisierungskonzepten

für diese Fahrzeuge können Verlustzeiten reduziert werden. Insbesondere für Rettungsfahrzeuge kann diese Art der Verkehrssteuerung von großen Nutzen sein.

2.3 Aufbau der Forschungskreuzung Braunschweig

Die Begriff Forschungskreuzung (Fokr) bezeichnet speziell die Verkehrskreuzung Rebenring / Brucknerstraße / Hagenring / Hans-Sommer-Straße der Stadt Braunschweig. Sie ist ein AIM-Teilprojekt und dient der Sammlung von Daten der Verkehrsteilnehmer an dieser Kreuzung. Die Fokr ist mit der nötigen Infrastruktur zur Sammlung von Bild- und Trajektorien ausgestattet. Insbesondere diese Trajektoriendaten dienen als Ausgangspunkt für die Auswertung von Verkehrssituationen in dieser Arbeit. Abbildung 4 zeigt die Forschungskreuzung, sowie die Infrastruktur zur Sammlung der Daten.

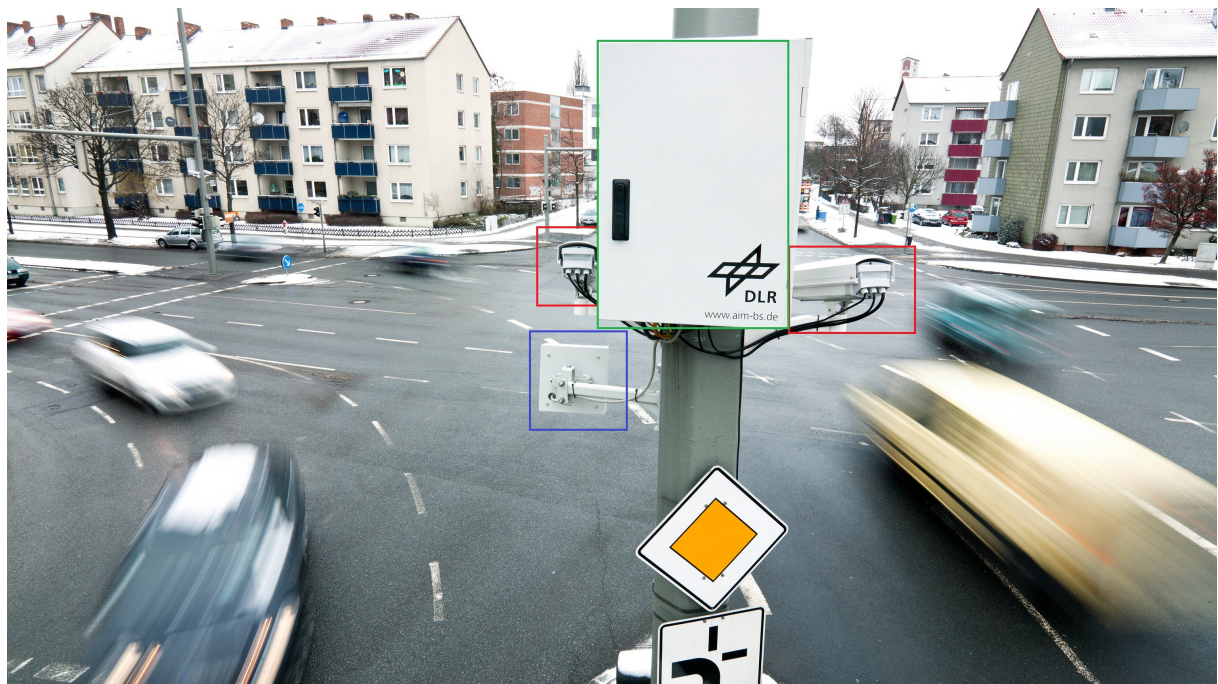


Abbildung 4: Technik der Forschungskreuzung. Bestehend aus Kameras (rot), Verteiler (grün) und Infrarot-Beleuchtung (blau). Radar an der Vorderseite des Mastes ist nicht sichtbar. Aus [DLR Fokr 2014]

An jedem der vier Kreuzungsarme steht ein Mast welcher mit jeweils einem Radar-Sensor, sowie einem paar Kameras ausgestattet ist. Ein Kamerapaar dient dabei jeweils der Aufnahme von hochauflösenden Detailaufnahmen, sowie der Aufnahme einer allgemeinen Übersicht über die Kreuzung. Durch den Einsatz von vier Masten kann somit die vollständige Ausleuchtung der Kreuzung gewährleistet werden. Für schlechte Sichtbedingungen und den Nachtverkehr ist jeder Mast außerdem mit einem Infrarot-Blitz ausgestattet.

Alle Masten sind miteinander vernetzt und übersenden die gesammelten Daten an einen zentralen Rechner. Dieser ist für die Zusammenführung und Speicherung der Daten verantwortlich. Diese fusionierten Datensätze dienen als Input für das in Rahmen dieser Arbeit entwickelte System. Für den Empfang dieser Daten ermöglicht die Systemarchitektur der Fokres, Daten aus einer bereitgestellten Datenbank auszulesen, sowie die Daten in Echtzeit vom Zentralrechner zu empfangen. Die Gesamtheit dieses Systemaufbaus beschreibt das Multi-Sensor-System der Forschungskreuzung.

3 Entwurf

3.1 Beschaffenheit der Daten

Zur Erstellung eines Systems zur Erkennung von Verkehrssituationen muss zunächst der inhaltliche Aufbau einer gesendeten Nachricht von Fahrzeugtrajektorien geklärt werden. Das Multi-Sensor-System (MSS) der Forschungskreuzung, welches für die Datensammlung verantwortlich ist, sendet eine gebündelte Nachricht von Trajektorien in jedem Zeitschritt (ungefähr alle 40 Millisekunden). Diese enthält alle innerhalb dieser Zeitspanne gesammelten Trajektorienpunkte auf der Forschungskreuzung. Dabei wurden die Aufnahmen aller 4 Aufnahme-Masten berücksichtigt, sodass Daten wie die Position eines Fahrzeuges bereits in Weltkoordinaten vorliegen. Im folgenden sollen den wesentlichen Inhalte der Nachricht beschrieben werden:

Zeitstempel

Jede versandte Nachricht enthält einen Zeitstempel, welcher die Sendezeit in Sekunden (in Unixzeit) und Mikrosekunden angibt.

Absender

Zur korrekten Verarbeitung der Daten enthält eine Nachricht eine eindeutige ID des Absendermoduls. Wichtig zu beachten ist, dass die nicht-fusionierten Daten das gleiche Nachrichtenformat verwenden, weshalb der fusionierte Datensatz lediglich anhand der ID des für die Fusion zuständigen Moduls erkannt werden kann. Diese ID ist konstant und bleibt auch bei Neustarts des MSS unverändert.

Objekt IDs

Alle in der Nachricht enthaltenen Fahrzeug-IDs werden in einem Array von Integerwerten versandt. Sobald ein Fahrzeug den Aufnahmebereich der Kreuzung betritt, wird ihm vom MSS eine eindeutige ID zugeordnet, welche genutzt werden kann, um einem Fahrzeug die

jeweiligen Trajektorien zuzuordnen. Da in einem Zeitschritt mehrere Trajektorienpunkte eines Fahrzeuges gesammelt werden können, kann es vorkommen, dass die Gesamtgröße dieses Arrays größer ist, als die Anzahl an Fahrzeugen auf der Forschungskreuzung. In diesem Fall sind gleiche Fahrzeug-IDs nacheinander im Array angeordnet. Eine weitere wichtige Eigenschaft der Nachricht ist, dass gleiche Array Indizes zu gleichen Objekten gehören. Das bedeutet, dass zu einer ID mit Index i alle Daten mit Index i in anderen Arrays gehören.

Trajektorienpunkte

Geschwindigkeit, Position und Beschleunigung eines erkannten Fahrzeuges werden jeweils in X- und Y-Komponenten zerlegt übermittelt. Da das IPC-Framework nur Zahlenwerte vom Typ Integer senden kann, werden diese Werte mit einem Skalierungsfaktor multipliziert, welcher ebenfalls in einer Nachricht enthalten ist. Jede neu versendete Nachricht enthält nur Daten des aktuellen Zeitschrittes. Um also Aussagen über den Fahrverlauf eines Fahrzeuges treffen zu können, müssen ältere Daten gespeichert werden. Für den Zusammenhang von Geschwindigkeit und Beschleunigung enthält eine Nachricht die Werte der Kovarianzmatrix eines jeden Objektes.

Boundingbox

Anhand der Kamerabilder ermittelt das System die Ausmaße erkannter Fahrzeuge mithilfe von Boundingboxen. Deren Längen-, Breiten- und Höhenwerte sind ebenfalls in der Nachricht enthalten. Diese Werte sind für genauere Berechnungen nützlich, da ansonsten Fahrzeuge anhand ihrer aktuellen Position zu Punkten vereinfacht werden müssen.

Fahrzeugklasse

Neben der Ermittlung einer Boundingbox wird jedes Fahrzeug außerdem in eine von vier Fahrzeugklassen („motorcycle“, „car“, „van“ und „truck“) eingeteilt. Diese Zuordnung wird in Form eines Wahrscheinlichkeitswertes für jeden Fahrzeugtyp versandt.

Konfidenzwerte

Neben den bereits erwähnten Daten werden ebenfalls Konfidenzwerte übersendet. Die Nachricht enthält Konfidenzwerte für die Ausmaße der Boundingbox, sowie die Wahrscheinlichkeit der Fahrzeugklassen-Zuordnung.

3.2 Anforderungen an das System

Für einen potenziellen Einsatz des Erkennungssystems an realen Kreuzungen muss das System folgende Voraussetzungen erfüllen:

Verarbeitung in Echtzeit

Zur frühzeitigen Situationserkennung sollen die ermittelten Daten in Echtzeit verarbeitet werden, anstatt aufgezeichnete Daten aus einer Datenbank auszulesen. Für einen vollständigen Überblick über die Kreuzungssituation muss deshalb ein Algorithmus entworfen werden, um Trajektorien für alle aktiven Fahrzeuge auf der Kreuzung zu speichern und aktualisieren. Durch die regelmäßige Einspeisung neuer Daten in das System muss dieser Algorithmus möglichst performant sein.

Erweiterbarkeit des Systems

Zur eventuellen Verbesserung der Situationserkennung und Anpassung an reale Kreuzungsverhältnisse soll das System den Einbau neuer Erkennungsalgorithmen und Datenquellen erlauben. Aus diesem Grund werden Verkehrsmetriken und die zu untersuchenden Situationen voneinander getrennt, um Algorithmen zur Berechnung der Metriken wiederverwendbar für neue Situationen zu machen.

Speicherung der Ergebnisse

Zur Auswertung und Kontrolle der Analyse, muss das System eine Möglichkeit bereitstellen, um erzielte Ergebnisse zu speichern. Um die Erweiterbarkeit des Systems zu garantieren muss also folglich das Speicher-Modul selbstständig auf neuartige Verkehrssituationen reagieren und diese verarbeiten können.

3.3 Das IPC-Framework

Die interne Kommunikation des MSS der Forschungskreuzung wird zu großen Teilen über ein Netzwerk zur Interprozess-Kommunikation (IPC) geregelt. Da auch das im Rahmen dieser Arbeit entwickelte System auf diesem Framework aufbaut, soll im nachfolgenden die allgemeine Funktionsweise des Frameworks beschrieben werden.

Zentraler Mittelpunkt ist dabei ein Nachrichten-Verteilungs-Server. Dieser ist für den Empfang und Versand aller Nachrichten innerhalb des lokalen IPC-Systems verantwortlich. Ein IPC-Modul kann in das System integriert werden, indem es Nachrichten eines bestimmten Typs abonniert und diese vom Zentral-Server zugeteilt bekommt. Auf diese Weise wird sichergestellt, dass ein Modul des IPC-Systems stets aktuelle und relevante des gewünschten Typs erhält. Desweiteren kann jedes Modul den von ihm gesendeten Nachrichtentyp deklarieren. Diese werden an den Zentral-Server gesendet welcher sie an andere, diesen Typ abonnierende, Module weiterleitet. Dieses Funktionsprinzip ist in Abbildung 5 bildhaft verdeutlicht.

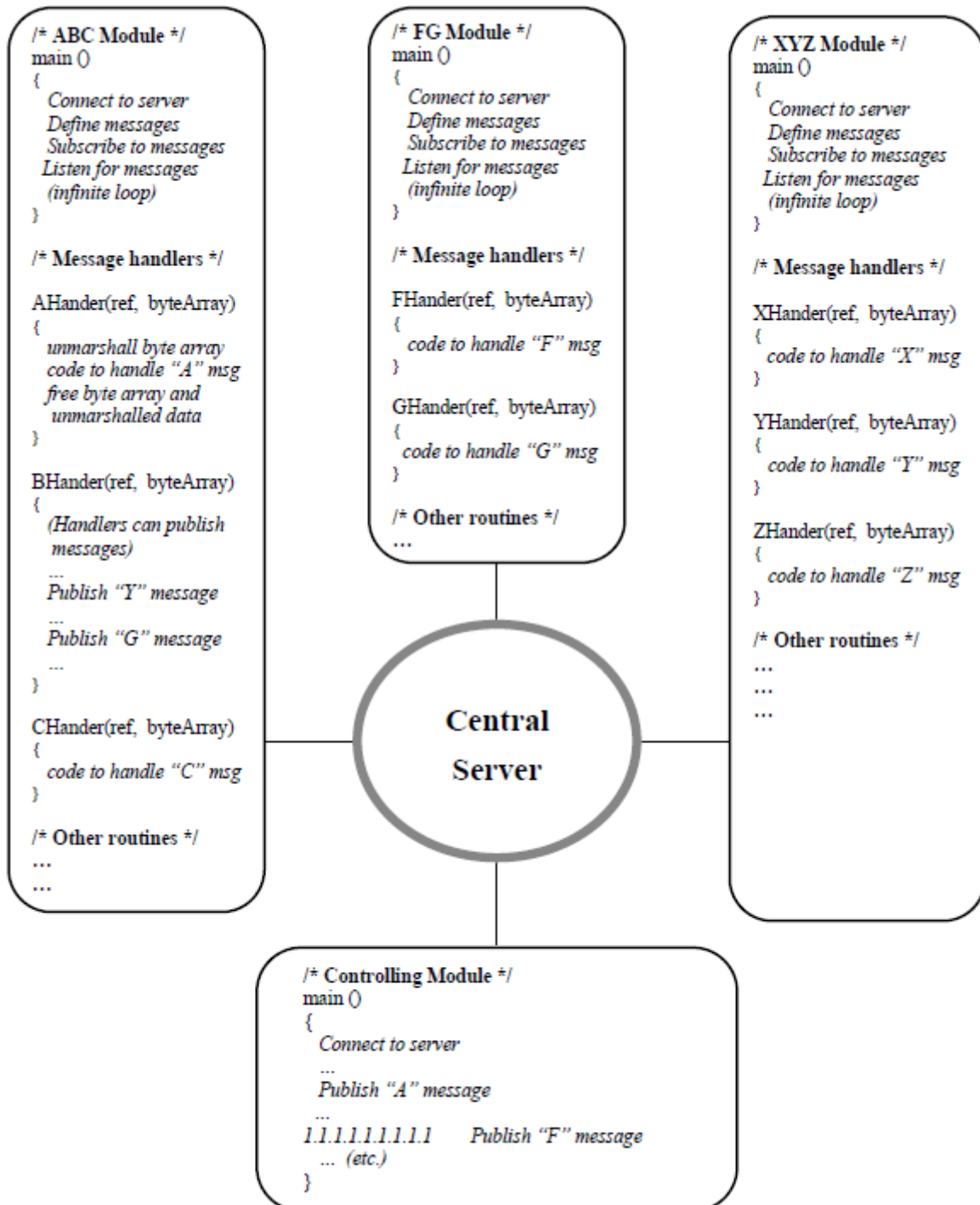


Abbildung 5: Bildhafte Funktionsweise eines IPC-Systems, aus [IPC-Reference 2011]

Der Inhalt einer IPC-Nachricht ist weitestgehend von der Implementation des Entwicklers abhängig. Vom Framework wird dabei lediglich vorausgesetzt, dass eine Nachricht den Absender (in Form einer eindeutig zuordbarer ID-Nummer) und einen Zeitstempel (sowohl für Sekunden in Unix-Zeit, als auch Mikrosekunden) enthält. Außerdem müssen das Format, sowie der Name einer Nachricht im voraus definiert werden, damit sie vom Zentral-Server korrekt erkannt und weitergeleitet werden kann. Ansonsten kann eine Nachricht beliebig groß sein und beliebig viele Daten versenden. Dabei ist jedoch zu beachten, dass eine Nachricht nur die elementaren Datentypen String (in Form von Char-Arrays) , signed und unsigned Integer enthalten kann. Dezimal-Werte müssen folglich mit einem Integer-Wert skaliert werden, bevor sie versendet werden.

Das für diese Arbeit verwendete IPC-Framework kennt im wesentlichen zwei Arten von Modulen. Basis-Module, welche Nachrichten empfangen, diese verarbeiten und anschließend neue Nachrichten verschicken, und Adaptermodule. Diese unterscheiden sich von Basismodulen in der Hinsicht, das sie zusätzlich eine beliebige Anzahl von Threads kontrollieren, welche zur Kommunikation mit Gerätetreibern eingesetzt werden können. Dies ermöglicht es einem Adapter-Modul auch dann Nachrichten zu verschicken, wenn sie keine neuen Daten vom Zentral-Server erhalten. Auf der Forschungskreuzung werden Adapter-Module eingesetzt, um Kamera und Radardaten zu empfangen.

3.4 Konzeptueller Systemablauf

Für den kompletten Ablauf des Systems lässt sich der Prozess zur Situationserkennung in 4 wesentliche Arbeitsschritte zerlegen. Jeder dieser Arbeitsschritte soll dabei jeweils in Form von separaten IPC-Modulen implementiert werden:

Sammlung und Organisation der Trajektorien

Durch Konfiguration über Kommandozeilenparameter kann ein IPC-Modul die fusionierten Trajektorien Nachrichten aus dem MSS empfangen und für eigene Algorithmen verwenden. Eine auf diese Weise empfangene Nachricht enthält jedoch nur Daten aus einem einzigen

Zeitschritt. Um einen kompletten Überblick über die Kreuzungssituation zu erhalten ist eine einzelne Nachricht deshalb nicht ausreichend. Das Modul für den Datenempfang soll stattdessen mehrere in Folge empfangene Nachrichten sammeln und in einem festen Zeitabstand von t Sekunden alle gesammelten Nachrichten gebündelt verschicken. Diese Zeitkonstante t sollte zur späteren Anpassung des Systems konfigurierbar sein. Versandte Daten werden jedoch nicht gelöscht, damit jede versandte Nachricht sämtliche Trajektorien aller auf der Forschungskreuzung aktiven Fahrzeuge komplett enthält. Das bedeutet, dass eine zum Zeitpunkt T_2 versandte Nachricht alle Daten aus T_1 , sowie die im Zeitabstand t gesammelten Daten enthält.

Desweiteren sollte das Empfangsmodul erkennen, wenn ein Fahrzeug die Kreuzung verlassen hat und die dazugehörigen Daten löschen. Dies verringert sowohl den Speicherverbrauch der versandten Nachrichten, als auch die Laufzeit der Auswertungsalgorithmen, da Fahrzeugdaten nur dann ausgewertet werden, wenn das Fahrzeug die Kreuzung noch nicht verlassen hat. Besonders wichtig ist dabei, zwischen einer tatsächlichen Abwesenheit eines Fahrzeuges und einem fehlenden Datensatz aufgrund von technischen Fehlern zu unterscheiden. Beispielsweise kann es vorkommen, dass ein Fahrzeug in einem einzelnen Zeitschritt nicht vom MSS erkannt wird und deshalb Daten über dieses Fahrzeug nicht in einer Nachricht enthalten sind. Das Modul soll auf diese Fehler Rücksicht nehmen und nur bei mehrmaligem Fehlen eines Datensatzes ein Fahrzeug als abwesend betrachten.

Berechnung von Metriken anhand der Daten

Anhand der gesammelten Trajektorien-Daten wird die aktuelle Verkehrssituation der Kreuzung durch Berechnung verschiedener Verkehrsmetriken ermittelt. Durch diese Berechnung kann ein Überblick über das Verhältnis der Fahrzeuge untereinander gewonnen werden, beispielsweise ob und wann Fahrzeuge miteinander kollidieren, falls keine der beteiligten Verkehrsteilnehmer eine Bremsung durchführt.

Dieser Arbeitsschritt ist im besonders Hinblick auf die Erweiterbarkeit des Systems entscheidend, da durch die Implementation von neuen Metriken auch neuartige kritische

Verkehrssituationen erkannt werden können. Ein Beispiel ist die Post-Encroachment-Time Metrik, welche zur Auswertung von Situationen mit Linksabbiegern besser geeignet ist als z.B. eine Time-to-Collision (TTC) Metrik, da die unabhängig von der Geschwindigkeit der jeweiligen Fahrzeuge ist und auch dann berechnet werden kann, wenn Fahrzeuge sich nur knapp verfehlen (TTC setzt einen Kollisionskurs voraus). Deshalb soll das Metrik-Modul das Einfügen neuer Metrik-Algorithmen durch den Nutzer ermöglichen. Damit das System trotz möglicher Erweiterungen weiterhin fehlerfrei arbeitet, muss ein Nachrichtentyp speziell für diese versandten Metrikauswertungen festgelegt werden.

Auswertung der Metrik-Analyse

Nach der Berechnung der einzelnen Metrikgrößen wird die Kritikalität einer Situation in separaten Auswertungsmodulen bewertet. Diese Trennung von Situation und Metrikberechnung ist wichtig für die Erweiterbarkeit des Systems, damit die Algorithmen zur Metrikberechnung unabhängig bleiben und für neue Situationen wiederverwendet werden können. Dies ermöglicht, dass eine bestimmte Metrik für verschiedene Situationen unterschiedliche Schwellenwerte haben kann. Die Implementierungen von neuen Erkennungsmodulen ist der zweite wesentliche Arbeitsschritt, in dem das System erweitert werden kann. Deshalb ist es auch hier wichtig, dass ein einheitlicher Nachrichtentyp für erkannte kritische Situationen benutzt wird, um einen fehlerfreien Ablauf des Speicher-Moduls ermöglichen zu können.

Speicherung der Daten ermittelter kritischer Situationen

Der vierte und letzte Schritt ist der Empfang von Nachrichten, welche kritische Situationen enthalten und deren Speicherung in einer Datenbank. Durch die Erweiterbarkeit des Systems muss das Speicher-Modul selbstständig neue in das System eingebaute Situationen und Metriken erkennen und neue Datenbanken für diese anlegen. Dies muss beim Entwurf des Nachrichtentyps für die Situationsauswertung berücksichtigt werden, damit die Erweiterbarkeit gewährleistet bleibt.

Das in Abbildung 6 dargestellte Sequenzdiagramm soll abschließende die Kommunikation der einzelnen Systemaspekte untereinander verdeutlichen.

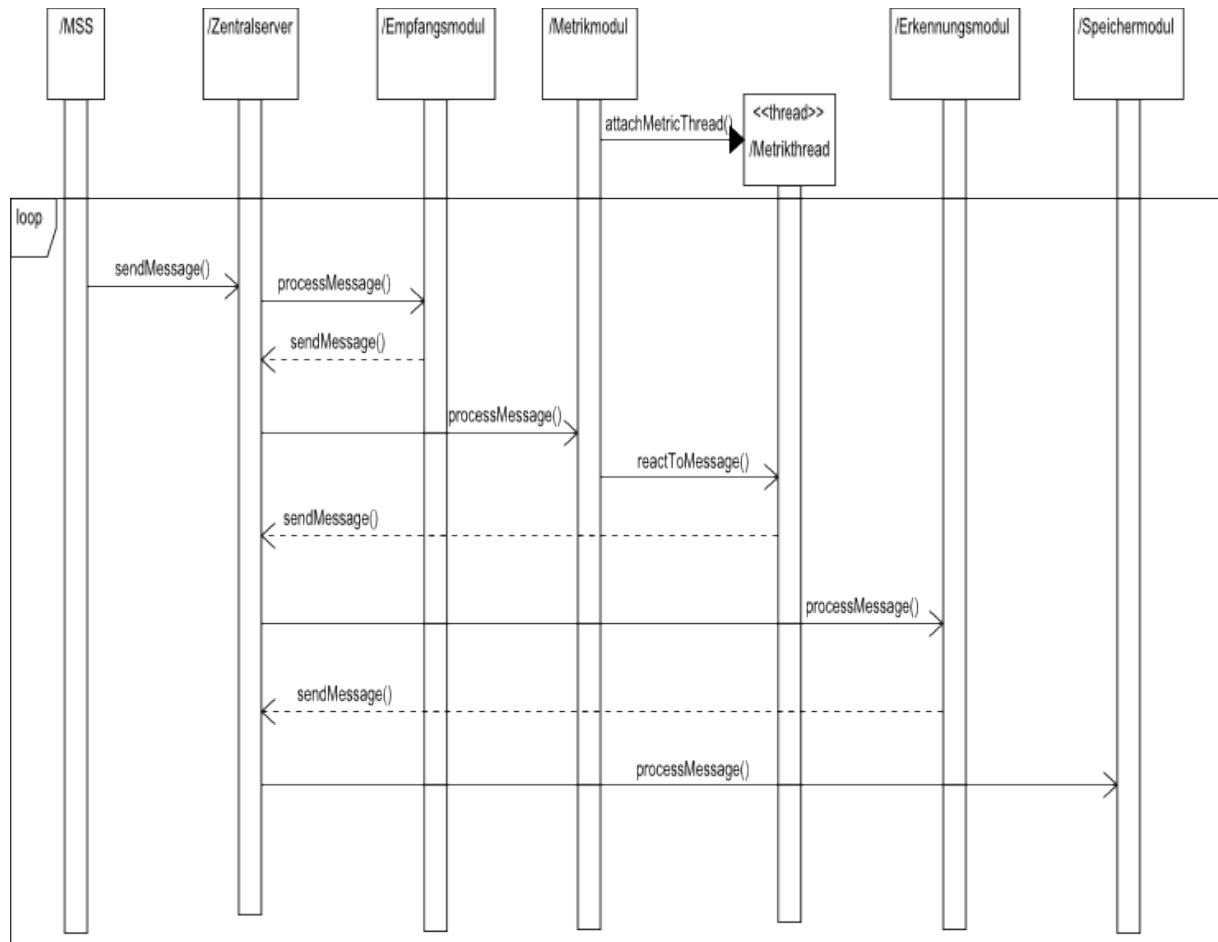


Abbildung 6: Sequenzdiagramm eines Systemdurchlaufes

4 Implementation

4.1 Verwendete Software und Bibliotheken

Visual Studio 2010 / gedit

Obwohl das System für ein Linux System entwickelt wurde, fand ein Großteil der Entwicklung auf einem Windows-System mit Visual Studio 2010 statt. Kleine Änderungen am Code während des Testens wurden mithilfe des Texteditors gedit unter Linux durchgeführt.

Jenoptik IPC-Framework

Das IPC-Framework dient als Grundlage für alle im Rahmen dieser Arbeit entstandenen Module. Es ermöglicht einen einfachen Datenaustausch zwischen Modulen, sowie eine einfache Integration in die bestehende Systemarchitektur der Forschungskreuzung. Das Framework basiert auf C++, weshalb auch das im Rahmen dieser Arbeit entwickelte System mit C++ implementiert wurde.

Boost

C++ Boost wird für das IPC-Framework verwendet und wird lediglich der Vollständigkeit halber erwähnt.

IPC Offline-Entwicklungsumgebung

Die IPC Offline-Entwicklungsumgebung dient zur Entwicklung von IPC Modulen ohne den Ablauf des Live-Systems zu unterbrechen. Mithilfe von Rohdaten können alle Verarbeitungsrechner der Forschungskreuzung, sowie deren IPC-Systeme simuliert werden. Außerdem werden CMake-Files zur Verfügung gestellt, um neue IPC-Module für diverse Systemkonfigurationen zu kompilieren.

4.2 Implementationsdetails

4.2.1 Organisation der Datenmengen

Für die Verwaltung der Trajektorien aus dem MSS ist das Fusion Server Receiver Module (FSRM) zuständig. Das FSRM ist als IPC-Basis Modul implementiert und empfängt die IPC-Nachrichten des Typs „ipcm_STrackObjectList“ als einzigen Input. Weiterhin ist das Modul so konfiguriert, dass nur die fusionierten Trajektorien des „multi_sensor_object_fusion_module“ aus dem MSS empfangen werden. Im folgenden wird der Algorithmus zur Verwaltung dieser Daten im Detail erklärt:

Update der internen Objekt Liste

Nach dem Empfang einer neuen Nachricht wird zunächst eine interne Objekt Liste überarbeitet um zu überprüfen, ob Fahrzeuge die Forschungskreuzung verlassen oder neu betreten haben. Umgesetzt wurde diese Liste als eine Hashtabelle, welche die (vom MSS eindeutig zugeordnete) Fahrzeug ID auf einen Integerwert mappt, welcher die Abwesenheit einer ID in Folge zählt.

Da es aufgrund von potentiellen Messfehlern möglich sein kann, dass ein Fahrzeug in einem Zeitschritt nicht vom MSS erkannt wird, reicht es nicht aus, einen Datensatz bei einmaliger Abwesenheit einer ID zu löschen. Stattdessen wird für jede ID, welche in der Hashtabelle vorkommt, aber nicht in der aktuellen Nachricht enthalten ist, der interne Zähler in der Hashtabelle hochgezählt. Der Zähler aller IDs die sowohl in der empfangenen Nachricht, als auch in der Hashtabelle vorkommen, wird stattdessen auf 0 gesetzt.

Über eine Konstante (OBJECT_ABSENCE_COUNT) kann der Nutzer festlegen, wie oft eine ID in Folge abwesend sein darf, bevor das System annimmt, dass das zugehörige Fahrzeug die Kreuzung verlassen hat.

Für den Sonderfall, dass eine ID in einer Nachricht, nicht aber in der Hashtabelle vorkommt, wird ein neue Eintrag in der Tabelle erstellt. Da keine bisherigen Einträge vorliegen, hat

dieses Fahrzeug aus Sicht des Systems die Kreuzung neu betreten.

Einfügen der neuen Trajektorien

Nachdem veraltete Einträge entfernt wurden, werden im nächsten Schritt die empfangenen Daten der Nachricht gespeichert. Die Speicherung dieser Daten erfolgt ebenfalls über eine Hashtabelle, allerdings wird hier die ID des Fahrzeugs auf einen Vektor gemappt, welcher alle Trajektorien eines Fahrzeuges enthält. Die neuen Daten werden ans Ende des Vektors eingefügt. Auf diese Weise enthält jeder Vektor alle Information eines Fahrzeuges in chronologischer Reihenfolge. Neben den eigentlichen Trajektorieninformationen werden außerdem alle Zeitstempel empfangener Nachrichten gespeichert.

Update der internen Zeitmessung

Die interne Zeitmessung des FSRM entscheidet, in welchem Abstand die gesammelten Daten ausgesendet werden. Abhängig von einer Konstante (`SAMPLE_TIME_IN_SECONDS`) wird der Inhalt der gesamten Hashtabelle in regelmäßigen Abständen in eine IPC kompatible Datenstruktur umgewandelt und versandt. Obwohl es technisch möglich ist, eine Nachricht zu versenden so bald neue Input Nachrichten empfangen werden (indem `SAMPLE_TIME_IN_SECONDS` auf 0 festgesetzt wird), bietet sich diese Vorgehensweise nicht an, da durch die Sammlung der Daten über einen längeren Zeitraum eine genauere Bestimmung der Fahrzeug Trajektorien möglich ist.

Wichtig zu beachten ist, dass Nachrichten zwar im Abstand von t Sekunden verschickt werden, die Nachricht aber dennoch jene Daten enthält, welche vor dem aktuellen Zeitintervall gesammelt wurden. Unabhängig von der Sendezeit werden Datensätze nur gelöscht, wenn ein Fahrzeug die Kreuzung verlassen hat, damit zu jeder Zeit ein vollständiger Überblick über sämtliche Fahrzeugtrajektorien gewährleistet ist.

Versenden der Nachricht

Das FSRM versendet Nachrichten vom Typ „`dfMessages_AggregateTrajectoryMessage`“. Da innerhalb des FSRM die empfangenen Daten selbst nicht verändert werden, ist dieser

Nachrichtentyp strukturell nahezu identisch mit der „ipcm_STrackObjectList“ Nachricht des MSS, wurde jedoch durch einige weitere Informationen erweitert. Zum einen wurde ein Array hinzugefügt, welches die individuellen Längen der Fahrzeugvektoren enthält. Da Informationen wie die Geschwindigkeit aller Fahrzeuge sequentiell in einem Array vorliegen, ist diese Erweiterung hilfreich, um später leichter über den gesammelten Datensatz zu iterieren. Außerdem enthält die Nachricht ein Array, welches jedem Trajektorienpunkt in der Nachricht den dazugehörigen Zeitstempel zuweist. Dies ist hilfreich, um den gesamten Fahrverlauf eines Fahrzeuges zeitgemäß zu rekonstruieren.

4.2.2 Anwenden von Metriken

Das Metric Central Control Module (MCCM) ist als IPC Adaptermodul implementiert. Eine Besonderheit dieses Modultyps ist es, dass ein Adaptermodul über eine beliebige Anzahl von „Device Control Threads“ verfügen kann. Diese dienen innerhalb des IPC Frameworks dazu, um Verbindung mit Kameras oder ähnlichen Geräten Verbindung aufzunehmen und diese zu steuern. Außerdem können Device Control Threads konfiguriert werden, um Nachrichten selbstständig zu verschicken, ohne dabei auf eine vorhergehende Input Nachricht zu reagieren.

Device Control Threads dienen als Basis für die Algorithmen zur Berechnung der im verwendeten Verkehrsmetriken. Für jede separate Metrik wird ein Thread zur Berechnung gestartet. Ein Vorteil dieser Trennung ist, dass neben den Trajektorien ein Thread mit weiteren Geräten Verbindung aufnehmen kann, was einen potentiellen Ansatzpunkt für Erweiterungen des Systems liefert. Das MCCM selbst dient dabei lediglich zur Instanziierung der Threads, sowie dem Versenden der Ergebnismeldungen. Da die Implementation der Metriken vom jeweiligen Thread abhängt, wird deshalb im folgenden genauer auf den versandten Nachrichtentyp, welcher identisch für alle Threads des MCCM ist. Weitere Details zur Implementation einzelner Threads, siehe 4.2.5.

Die Threads des MCCM verschicken Nachrichten des Typs

„dfMessages_MetricReportMessage“. Diese enthält die folgenden Daten:

Namen der versandten Metrik

Da alle Threads den selben Nachrichtentyp verschicken, enthält jede Nachricht einen String, welcher zur Identifikation der Metrik dient. Dies ist erforderlich, damit Erkennungsmodule nur diejenigen Metriknachrichten verarbeiten, welche sie auch zur Auswertung benötigen. Die Wahl des Namens in Form eines Strings anstelle eines Integers erhöht den Speicherverbrauch der Nachricht, bietet jedoch einige Vorteile. Da dieser Namensstring in nachfolgenden Modulen bis hin zur Speicherung der Daten weiterversandt wird, kann er eingesetzt werden, um die gespeicherten Logdateien menschenlesbar zu machen. Diese Lesbarkeit kann die Implementation neuer Metriken erleichtern, indem die Gefahr von Namenskonflikten reduziert wird.

Datensätze

Fahrzeug IDs und Metrikwerte sind jeweils in einem Array gespeichert, in welchem alle Werte sequentiell aufeinander folgen. Die genaue Reihenfolge in welcher die Daten vorliegen hängt dabei vom Format der Nachricht ab, welches mithilfe von versandten Booleans definitert wird.

Booleans

Zur genauen Definition des Formates einer Metriknachricht werden neben den eigentlichen Daten zwei Booleans bestimmt, deren Wahrheitswert jeweils von der jeweiligen Implementation einer Metrik abhängt. Ein wesentliches Unterscheidungsmerkmal von Metriken ist, ob sie für jedes Fahrzeug einzeln (z.B im Falle eine Berechnung der Durchschnittsgeschwindigkeit) oder zwischen Fahrzeugpaaren (z.B eine Berechnung des Abstandes von Fahrzeugen untereinander) berechnet werden. Mithilfe des Booleans „singleCarMetric“ wird für Erkennungsmodule verdeutlicht, von welchem Typ eine empfangene Metrik ist.

Für den Fall, dass eine paarweise Metrikberechnung zwischen Fahrzeugen erfolgt, kann ein

weitere Unterscheidung getroffen werden, indem definitert wird, ob die Reihenfolge der Fahrzeugpaarung eine Rolle spielt. Für das Fahrzeugpaar A-B wird in diesem Fall ein anderer Metrikwert ermittelt als für das Paar B-A. Dieser Boolean dient im wesentlichen der Einsparung von Speicher, da doppelte Paarungen nicht übermittelt werden müssen.

Feldgrößen

Zur Ermittlung der Feldgrößen werden zwei Kenngrößen versandt. Zum einen die Länge des Array der Fahrzeug IDs (`vehicleCount`), also die Gesamtzahl aller analysierten Fahrzeuge. Zum anderen die Anzahl an Werten, die pro Fahrzeug errechnet werden (`metricCount`). Dies ist nützlich, wenn neben dem eigentlichen Metrikwert weitere Informationen, wie Konfidenzintervalle, versandt werden sollen. Aus den bereits beschriebenen Booleans und dieser Kenngröße kann implizit die Größe des Arrays für Metrikwerte ermittelt werden, sowie die genaue Reihenfolge, in der Werte im Speicher liegen. Im folgenden werden die drei möglichen Formate einer Nachricht kurz beschrieben:

- **keine paarweise Berechnung**

Für den Fall, dass eine Metrik unabhängig von anderen Fahrzeugen berechnet wird, ermittelt sich die Gesamtlänge des Metrikarrays aus der Multiplikation der einzelnen Feldgrößen (`vehicleCount * metricCount`). Für den Spezialfall, dass nur ein einzelner Wert berechnet wird, ist der Index eines Fahrzeuges aus dem Fahrzeugarray gleich dem Wert aus dem Metrikarray.

- **Paarweise Berechnung, ungleiche Paare**

Bei ungleichen Paaren werden für jedes Fahrzeug genau `vehicleCount - 1` Werte berechnet. Die Größe des Arrays für Metrikwerte ist also demnach `metricCount * vehicleCount * (vehicleCount - 1)`. Die Reihenfolge der einzelnen Paare hängt dabei von deren Reihenfolge im Fahrzeugarray ab.

- **Paarweise Berechnung, gleiche Paare**

Dieses Format ist ähnlich dem Format für ungleiche Fahrzeugpaare, durch die Entfernung doppelter Paare kann jedoch sowohl die Arraygröße, als auch der verbrauchte Speicher einer Nachricht halbiert werden.

4.2.3 Auswertung der Ergebnisse

Die eigentliche Auswertung und Interpretation der Ergebnisse findet in separaten Modulen statt. Dies ermöglicht es, die Algorithmen zur Metrikberechnung für mehrere kritische Situationen wiederzuverwenden ohne dabei Erkennungsmerkmale (z.B in Form von Schwellenwerten) von diesen Algorithmen abhängig zu machen.

Alle Erkennungsmodule werden als separate IPC-Module implementiert. Das System selbst stellt dabei keine Anforderungen an die Implementation der Erkennungsmethoden dieser Module. Die einzige Voraussetzung für eine Integration des Systemablaues ist, dass versandte Nachrichten vom Typ „DatabaseReportMessage“ sind. Alle im Rahmen dieser Arbeit implementierten Erkennungsmodule nutzen fest vorgegebene Schwellenwerte zur Situationserkennung. Andere Strategien zur Erweiterung des Systems (beispielsweise in Form von dynamischen Schwellenwerten) sind jedoch ebenfalls möglich. Weitere Details zur Implementation dieser Module, siehe 4.2.5

Der von diesen Modulen versandte Nachrichtentyp („DatabaseReportMessage“) ähnelt dem von Metrikmodulen versandten Nachrichtentyp („MetricReportMessage“), ist jedoch vereinfacht, da lediglich Informationen über kritische Situationen weiterverschickt werden müssen. In folgenden werden der Aufbau dieses Nachrichtentypes beschrieben werden:

Namen von Metriken und Situation

Außer einem String zur Identifikation der Metriken wird außerdem ein String zur Identifikation der Situation selbst benötigt. Mithilfe dieser Informationen kann das Modul zur Ergebnisspeicherung die jeweiligen Daten korrekt zuordnen.

Booleans

Aufgrund der unterschiedlichen Formate für Metriknachrichten müssen die verwendeten Booleans weitergeleitet werden, damit Paarinformationen nicht verlorengehen.

Datensätze und Feldgrößen

Die einzige Unterschied zu Metriknachrichten ist, dass eine Datenbanknachricht nur Fahrzeuge und Metrikwerte enthält, welche eindeutig einer kritischen Situation zugeordnet wurden.

4.2.4 Speicherung der Ergebnisse

Im letzten Schritt eines Systemablaufes werden kritische Daten vom „ResultReportModule“ gespeichert. Das Modul empfängt alle Nachrichten des Typs „DatabaseReportMessage“ und ist als IPC-Basismodul implementiert. Im Gegensatz zu anderen IPC-Modulen des Systems werden vom RRM jedoch keine Nachrichten verschickt.

Ausgehend von den Namen der empfangenen Situationen wird eine Ordnerstruktur erstellt, welche für jede Art von kritischer Situation eine Logdatei mit allen empfangenen Informationen enthält. Falls für eine Situation keine Logdatei existiert, erstellt das Modul selbstständig die dazugehörige Ordnerstruktur und Logdatei. Dies ermöglicht es, neue Metriken und Situationerkennungsmodule in das System einzubinden, ohne das RRM selbst für diese anpassen zu müssen.

Eine weitere Möglichkeit der Archivierung, ist die Speicherung der Ergebnisse in einer Datenbank. Das Auslesen der Datenbank- und Accountinformationen aus einer Konfigurationsdatei wird von Seiten des IPC-Frameworks bereits unterstützt. Diese Funktionalität konnte jedoch aus Zeitgründen nicht getestet werden.

4.2.5 Bereits implementierte Erkennungsmodule und Metriken

Im Rahmen dieser Arbeit wurden die folgenden IPC-Module und Verkehrsmetriken bereits implementiert:

Inner Area Stay Time Metric

Dieses Metrik ermittelt für jede Fahrzeug ID einer Nachricht die Aufenthaltszeit auf der Kreuzung. Eine Ermittlung zu hoher Aufenthaltszeit auf der Kreuzung ist als Metrik nützlich, da ein zu langer Aufenthalt ein erster Hinweis auf Unfälle oder Verkehrsstaus innerhalb der Kreuzung ist.

Berechnet wird die Metrik durch Betrachtung der einzelnen Zeitstempel eines Fahrzeuges. Da in einer gebündelten Nachricht des FSRM alle Trajektorienpunkte in chronologischer Reihenfolge vorliegen, kann durch die Differenz des ersten und letzten Zeitstempel eines Fahrzeuges berechnet werden. Da IPC Zeitstempel in Linuxzeit vorliegen, ist der errechnete Wert die aktuelle Aufenthaltszeit jedes Fahrzeuges in Sekunden. Dieser Wert wird für alle IDs der empfangenen Nachricht berechnet und als Metrik des Typs „InnerAreaStayTime“ weiterversandt.

Inner Area Detection Module

Anhand der Metriken des Typs „InnerAreaStayTime“ ermittelt dieses Modul, ob ein Fahrzeug sich für eine ungewöhnlich lange Zeit auf der Kreuzung aufhält. Alle in der Nachrichten enthaltenen Aufenthaltszeiten werden mit einer vorher festgelegten Konstante, METRIC_REPORT_THRESHOLD verglichen. Dieser Wert gibt an, ab welcher Aufenthaltszeit (in Sekunden) die Fahrzeug ID an das Result Report Module gemeldet werden soll. Dieser Wert ist standardmäßig auf 30 Sekunden gesetzt. Alle Fahrzeug IDs, welche diesen Schwellenwert überschreiten, werden mitsamt der dazugehörigen Metrik-Werte weiter an der Result Report Module zur Speicherung der Ergebnisse gesendet.

4.2.6. Teilweise implementierte Metriken

Die folgenden Metriken wurden bereits teilweise implementiert, konnten jedoch aus Zeitgründen jedoch nicht vervollständigt werden, weshalb im folgenden deren Ansatz erklärt wird:

ExcessSpeed

Diese Metrik dient der Ermittlung der Durchschnittsgeschwindigkeit eines Fahrzeuges. Für den Einsatz als Metrik ist die Geschwindigkeit nützlich, da eine Überschreitung des Geschwindigkeitslimits die Gefahr eines Unfalls erhöht.

Errechnet wird diese Metrik indem zunächst das arithmetrische Mittel aus den Geschwindigkeitswerten eines Fahrzeuges ermittelt wird. Da die Geschwindigkeit in X- und Y- Komponenten zerlegt ist, müssen aus diesen mithilfe des Satz des Pythagoras die eigentliche Geschwindigkeit berechnet werden.

Allein das arithmetische Mittel ist für eine statistische Auswertung nicht ausreichend. Aus diesem Zweck muss zusätzlich die Standardabweichung des Geschwindigkeitswerte berechnet werden und zusammen mit den Durchschnittswerten versandt werden.

Time-to-Collision (TTC)

Die Time-to-Collision Metrik gibt Auskunft darüber, in welcher Zeit t zwei Fahrzeuge miteinander kollidieren, wenn ihre Geschwindigkeit unverändert bleibt. Als Verkehrsmetrik kann die TTC eingesetzt werden, um Auffahrunfälle und andere Situationen zu erkennen, in welche mehrere Fahrzeuge die gleiche Fahrspur benutzen.

Die TTC wird folgendermaßen berechnet:

$$TTC = \text{Fahrzeugabstand} / |v_1 - v_2|$$

Ein Nachteil dieser Metrik ist die Abhängigkeit von einem Kollisionskurs. Wenn also Fahrzeuge sich knapp verfehlen kann die TTC nicht berechnet werden.

4.3 Systemtests

Für den Einsatz des Systems für reale Verkehrssituationen muss die korrekte Funktionsweise des Systems getestet werden. Diese Tests wurden anhand eines Offline-Entwicklungssystems durchgeführt. Der Zweck dieses Entwicklungssystems ist die korrekte Simulation der Forschungskreuzung. Anhand von vorher bereitgestellten Rohdaten aus einer vergangenen Aufzeichnung können die vier Sensor-PCs und der zentrale Fusions-PC, sowie deren IPC-Systeme simuliert werden. Sowohl die Bilddaten der Kameras, als auch die Trajektorien zum Zeitpunkt der Rohdatenaufzeichnung werden anhand der originalen Zeitstempel versandt und können so von Modulen des Systems zu Testzwecken empfangen werden.

Durch die Berechenbarkeit der eingehenden Daten konnte das System insbesondere im Hinblick auf die Performance und die korrekte Verarbeitung und Auswertung der Daten untersucht werden. Außerdem konnte sichergestellt werden, dass alle Module des entwickelten Systems miteinander korrekt kommunizieren.

4.4 Systemgrenzen

Da das System allein auf Basis der Fahrzeugtrajektorien arbeitet und die Kameraaufnahmen der Forschungskreuzung nicht berücksichtigt, muss sichergestellt sein, dass die Datensammlung möglichst fehlerfrei ist, damit die korrekte Verarbeitung der Daten gewährleistet werden kann. Während der Systemtests sind zwei Probleme im besonderen aufgefallen, welche für den realen Einsatz des Systems hinderlich sind:

Fehlende Trajektorienpunkte

Dieses Problem wurde bereits in der Implementierung des Receivermoduls kurz angeschnitten (siehe 4.2.1). Manchmal kann es vorkommen, dass das MSS der Forschungskreuzung ein Fahrzeug in einem Zeitschritt nicht erkennt. In diesem Fall sind keine Daten über das Fahrzeug in der von der Fokr versandten Nachricht enthalten.

Aus der Sicht des Systems ist dies ein Problem, da allein die Abwesenheit von Daten keine Aussage darüber gibt, ob ein Fahrzeug die Kreuzung verlassen hat. Gerade für die Laufzeit und den Speicherverbrauch ist diese Erkennung jedoch wichtig, da ein Fahrzeug das die Kreuzung verlassen hat nicht in die Berechnungen miteinbezogen werden sollte.

Die aktuelle Lösung des Problems ist die Definition einer Konstante, welche ein Fahrzeug löscht, sofern eine gewisse Anzahl an Datensätzen in Folge nicht auftreten. Diese Lösung ist jedoch nicht optimal, da diese Schwellenwert-Konstante zur Laufzeit nicht änderbar ist und nur ungefähr abgeschätzt werden kann. Falls Daten über ein Fahrzeug gelöscht werden wenn es sich noch auf der Kreuzung befindet wird es als „neues“ Fahrzeug vom System erkannt. In diesem Fall gehen alle vor dem gesammelten Daten über das Fahrzeug verloren.

Eine alternative Lösung ist es, anhand der Daten zu entscheiden, ob ein Verlassen der Kreuzung wahrscheinlich ist. Wenn also ein Fahrzeug nicht erkannt wird, aber im letzten Zeitschritt im Zentrum der Kreuzung war, ist ein Verlassen der Kreuzung unwahrscheinlich. In diesem Fall würden die Daten nur dann gelöscht werden, wenn Daten über ein Fahrzeug fehlen, während es sich in der Nähe der Kreuzungsarme aufhält. Diese zusätzliche Auswertung würde jedoch eine erhöhte Laufzeit für einen kompletten Systemdurchlauf bedeuten.

Falsche ID Zuweisung

Ein weiteres Problem der Datensammlung ist, dass einem Fahrzeug unterschiedliche IDs in unterschiedlichen Zeitschritten zugeordnet werden können.

Dieser Fehler ist für das System besonders problematisch, da die ID zur eindeutigen Identifikation eines Fahrzeuges während des gesamten Systemablaufes genutzt wird. Folglich kann nicht garantiert werden, dass die von der Forschungskreuzung übermittelten Trajektorienpunkte zu den angegebenen IDs der Fahrzeuge gehören. Ein weiteres Problem ist, dass dieser Fehler im Gegensatz zur manuellen Auswertung der Bilddaten nur schwer anhand der Trajektorien Daten zu erkennen ist.

Eine Möglichkeit zur frühzeitigen Erkennung ist, neu empfangene Daten mit den bereits vorhandenen Daten zu vergleichen und eventuelle Unstimmigkeiten festzustellen (z.B. plötzlicher Positionswechsel oder stark veränderte Geschwindigkeitswerte). Dieser Ansatz würde jedoch nur funktionieren, wenn die falschen Daten den alten Daten nicht zu sehr ähneln. Außerdem ist die bloße Erkennung des Fehlers noch keine konkrete Lösung des Problems, da nun die korrekte ID gefunden werden muss, zu dem der Datensatz gehört. Dies setzt einen Vergleich mit bereits gesammelten Daten aller potenzieller korrekten IDs voraus, was eine erhöhte Laufzeit des Systems nach sich ziehen würde.

5 Zusammenfassung und Auswertung

Das Thema dieser Arbeit ist der Entwurf und die Implementation eines Systems zur Erkennung kritischer Verkehrssituationen anhand von Trajektorien, für den Einsatz an der Forschungskreuzung des DLR.

Zu diesem Zweck mussten zunächst sowohl die bestehende Infrastruktur an der Forschungskreuzung, als auch die Rolle der Fokr im AIM-Projekt geklärt werden. Der Großteil der Literaturrecherche bezog sich dabei auf das Gebiet der Situationserkennung, welches zusammen mit dem bestehenden Aufbau der Fokr eine theoretische Grundlage für die Implementation des Systems liefert.

Aufbauend auf diesen Grundlagen ging es in Kapitel drei um den eigentlichen Entwurf des Systems. Zunächst mussten das von der Forschungskreuzung verwendete Nachrichtenformat untersucht werden, um festzustellen, welche Art der Analyse für den Einsatz an der Fokr geeignet ist. Außerdem wurden notwendige Anforderungen an das System aufgestellt und die allgemeine Funktionsweise des IPC-Framework beschrieben, welches bereits an der Forschungskreuzung zur Datensammlung eingesetzt wird. Aufgrund der einfachen Erweiterbarkeit eines IPC-Systems wurde das Framework als Grundlage für das System ausgewählt.

Kapitel vier dient der ausführlichen Beschreibung der Implementation. Sowohl die verwendete Software, als auch die im einzelnen verwendeten Algorithmen wurden beschrieben und untersucht, in wie weit das Gesamtsystem die aufgestellten Anforderungen erfüllt.

Die untersuchten Systemgrenzen zeigen, dass das System zum gegenwärtigen Zeitpunkt für einen Einsatz im realen Verkehrs noch nicht ausgereift ist, da Fehler in der Datensammlung zu Problemen beim nachfolgenden Systemdurchlauf führen können. Dennoch erfüllt das System die aufgestellten Anforderungen und kann durch zusätzliche Erweiterungen als reales Erkennungssystem genutzt werden.

Im nachfolgenden Kapitel soll es um einen kurzen Ausblick gehen. Es werden Anstöße für Erweiterungen des Systems gegeben und deren Nutzen erklärt.

6 Ausblick

6.1 Einbindung von Ampeldata

Das im Rahmen dieser Arbeit beschriebene System ist lediglich in der Lage, kritische Situationen anhand der aus den Fahrzeugtrajektorien ermittelten Metriken zu erkennen. Dabei werden jedoch die für den Verkehrsablauf notwendigen Lichtsignalanlagen nicht berücksichtigt.

Durch die Einspeisung dieser LSA Daten in das System können sowohl bestehende Erkennungsalgorithmen verbessert, als auch neuartige Metriken errechnet werden.

Beispielsweise wäre es dadurch möglich, das Überqueren einer roten Ampel innerhalb des Systemablaufs festzustellen und zu melden. Durch die Kenntnis über die aktuelle Ampelschaltung und eine Abschätzung über den Wechsel des Ampelzustandes können Aussagen erstellt werden, wann eventueller Gegenverkehr einsetzen könnte oder Vorderfahrzeuge den Bremsvorgang aufgrund von roten Ampeln einleiten werden. Dies ermöglicht es, potentielle Situationen frühzeitiger zu erkennen und bereits vorhandene Metriken statistisch abzusichern.

6.2 Nutzung neuer Datenquellen

Da das System auf Basis des IPC-Frameworks implementiert wurde ist es möglich, neben den bestehenden Trajektorien weitere von der Forschungskreuzung versandte Nachrichtentypen in den bestehenden Systemablauf zu integrieren.

Neben dem bereits verwendeten Nachrichtentyp für Trajektorien bietet das IPC Framework außerdem Nachrichtenprotokolle für den Empfang der Kamerabilder, sowie des Vektorenflusses dieser Bilddaten. Dadurch bietet sich die Möglichkeit, das System durch den Einsatz von bildbasierten Erkennungsverfahren zu erweitern.

Die Integration von neuen Nachrichtentypen setzt jedoch die Erweiterung diverser Kernmodule, insbesondere des ResultReportModule voraus. Da das System die Ergebnisse einer Auswertung von Trajektorienbasierten Metriken erwartet, muss das Modul und der empfangene DatabaseReportMessage Nachrichtentyp entsprechend angepasst werden, damit die Erweiterbarkeit des Systems weiterhin gewährleistet werden kann.

6.3 Erweiterung durch weitere Kernmodule

Eine weitere Möglichkeit die Qualität und Laufzeit des Systems zu verbessern ist es, weitere Kernmodule vor der Auswertung der Daten in den Systemablauf zu integrieren.

In dieser Hinsicht besonders interessant wäre ein Modul, welches jeder Fahrzeug ID aufgrund der Positionskoordinaten eine zugehörige Verkehrsspur zuordnet. Eine solche Spuruordnung hätte erhebliche Vorteile, sowohl für die Optimierung von Mehr-Fahrzeug Metriken (z.B TTC und PET), als auch für die Erkennung neuer kritischer Situationen.

Insbesondere wird es aufgrund einer Spuruordnung möglich, zu große Spurabweichungen eines Fahrzeuges von der zugeordneten Fahrspur festzustellen, was je nach der Größe der Abweichung als kritische Situation gewertet werden kann. Weiterhin wird der Algorithmus der anderen Metriken vereinfacht, da Berechnungen zwischen Fahrzeugen ignoriert werden können, deren Spuren sich nicht miteinander kreuzen und somit keine Unfallmöglichkeit besteht, solange diese Spuren eingehalten werden, was positiven Einfluss der Laufzeit dieser Module nach sich zieht.

Allerdings ist dabei zu berücksichtigen, dass ein solches Modul auf die Forschungskreuzung selbst zugeschnitten wäre und deshalb für den Einsatz an weiteren Kreuzungen angepasst werden muss.

6.4 Unterscheidung anhand des Fahrzeugtyps

Da die gebündelte Nachricht des IPC-Frameworks neben den eigentlichen Trajektorien Informationen auch eine Zuordnung zu diversen Fahrzeugtypen aufweist (siehe 3.1) , bietet sich an dieser Stelle eine Möglichkeit, anhand des erkannten Fahrzeugtypes die Erkennung kritischer Situationen weiter zu verfeinern. Da beispielsweise ein frontaler Zusammenstoß eines Motorrads mit einem Truck ein hohes Gefahrenrisiko für den Motorradfahrer birgt, kann das System angepasst werden, indem die Bedingungen zur Detektion kritischer Situationen innerhalb der Erkennungsmodule angepasst und frühzeitiger gemeldet werden, wenn solche Fahrzeugtypen involviert sind.

In der Zukunft wäre eine solche Erweiterung besonders sinnvoll, falls das MSS erweitert wird, um Fußgänger innerhalb der Kreuzung zu erkennen.

6.5 Implementation weiterer Metriken

Für den potentiellen realen Einsatz des Systems an Verkehrskreuzungen ist es sinnvoll, weitere Metriken zu implementieren, um den Funktionsumfang und die Auswertungsfähigkeit des Systems zu erweitern.

Ein Problem der Time-to-Collision Metrik ist, dass sie nicht berechnet werden kann, wenn kein Kollisionskurs zwischen Fahrzeugen berechnet werden kann (siehe [Laureshyn 2010]). Dies ist ein Problem falls sich zwei Fahrzeuge sehr knapp verfehlen, was zwar keinen Unfall erzeugt, aber dennoch als gefährliche Situation zu interpretieren ist.

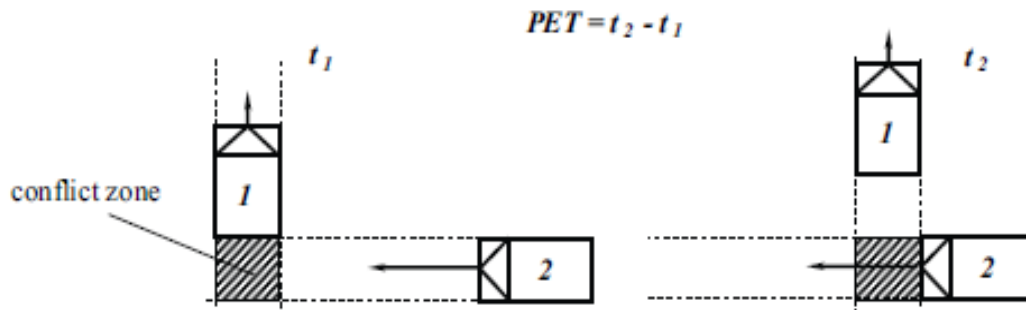


Abbildung 7: Berechnung der PET, aus [Laureshyn 2010]

Deshalb bietet sich die PET, Post Encroachment Time, als mögliche zusätzliche Metrik an. Sie beschreibt die Zeitdifferenz, in welcher ein Fahrzeug eine vorher bestimmte Gefahrenzone verlässt und ein zweites Fahrzeug diese erreicht. Die Größe der Gefahrenzone ist dabei abhängig von den Ausmaßen der Fahrzeuge und deren Fahrtrichtung. Abbildung 7 veranschaulicht die Berechnung dieser Metrik. Da für eine PET-Berechnung kein rechnerischer Kollisionskurs vorhanden sein muss, eignet sich diese Metrik für die Untersuchung von Links-Abbieger Situationen (siehe [Peesapati et. al 2013]).

6.6 Weiterleitung der Analyse an Verkehrsteilnehmer

Aus der Analyse gewonnene Daten können wiederum genutzt werden, um aktive Verkehrsteilnehmer zu unterstützen und zu warnen. Im Falle eines Unfalls könnte beispielsweise eine Warnmeldung an Fahrzeuge in Unfallnähe ausgesandt werden. Deren Fahrassistenzsysteme können wiederum diese Informationen an weitere Fahrzeuge schicken. Diese Art der Kommunikation, allgemein als Car-to-X (C2X) bezeichnet, wird bereits als Teil des AIM Projektes in Testfahrzeugen verwendet. Dies würde es erfordern, alle von Erkennungsmodulen des Systems identifizierten Situationen in eine C2X Nachricht zu konvertieren und zu versenden. Eine Verdeutlichung für die Kommunikation von Fahrzeugen mit C2X-Kommunikation wird in Abbildung 8 dargestellt.

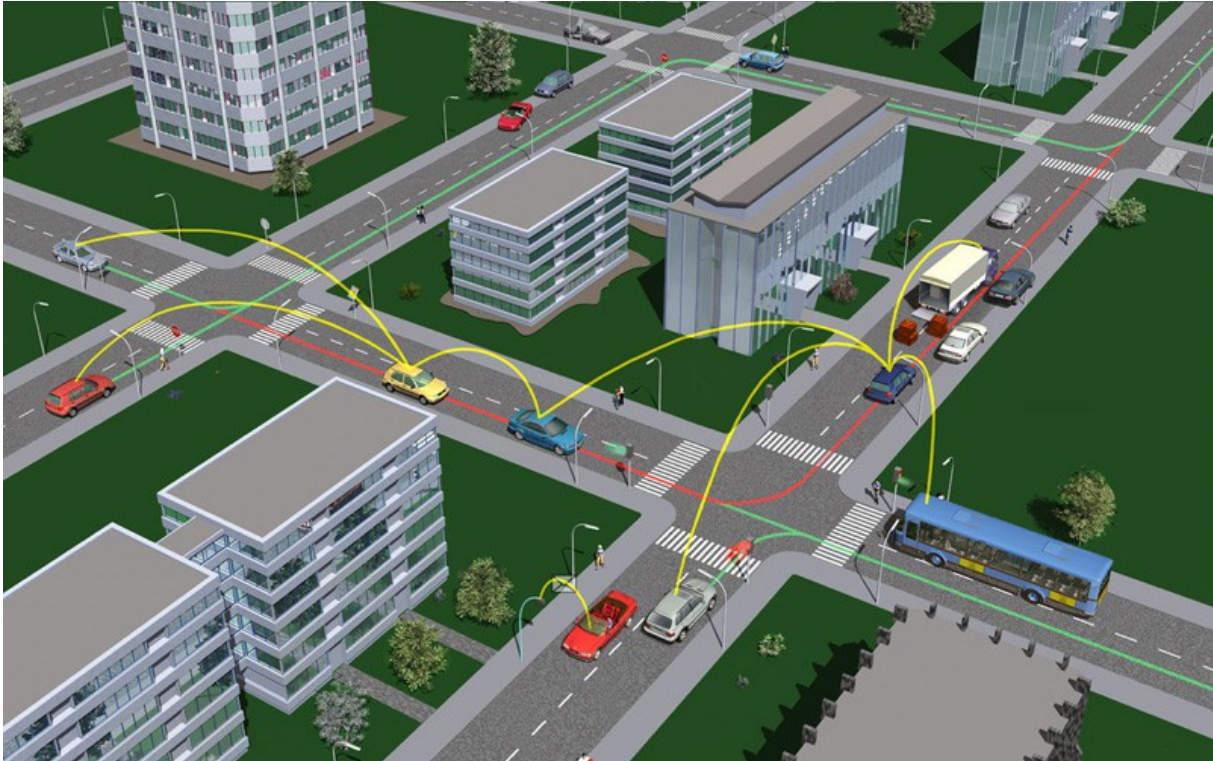


Abbildung 8: Darstellung der C2X Kommunikation. Dargestellt ist die Kommunikation zwischen den Fahrzeugen (gelb), alternative Fahrrouten (grün), sowie eine zurzeit blockierte Route (rot). Aus [C2X 2014]

Literaturverzeichnis

- [Amundsen/Hyden 1977] Amundsen, F.H., Hydén, C. (1977): Proceedings, First Workshop on Traffic Conflicts: Oslo 77. Norwegian Council for Scientific and Industrial Research. ISBN 8271331957.
- [C2X 2014] Car 2 Car Communication Consortium (2014) : Mission & Objectives. <http://www.car-to-car.org/index.php?id=5>.
Stand 13.04.2014
- [Catalá-Prat 2010] Catalá-Prat, Á (2010): Sensordatenfusion und Bildverarbeitung zur Objekt- und Gefahrenerkennung. Dissertation, Technische Universität Braunschweig.
- [Dagan et al. 2004] Dagan, E., O. Mano, G. P. Stein, A. Shashua, M. Ltd und I. Jerusalem (2004): Forward collision warning with a single camera. In IEEE Intelligent Vehicles Symposium, S. 37–42, Piscataway, NJ. IEEE Operations Center.
- [DLR Fokr 2014] Deutsches Zentrum für Luft- und Raumfahrt e.V (2005): Eine Kometenlandung, ein Deutscher im All und alternative Kraftstoffe im Luftverkehr.
http://www.dlr.de/dlr/presse/desktopdefault.aspx/tabid-10308/471_read-9315/year-all/#gallery/13496.
Stand 13.04.2014.
- [Fu et al. 2005] Fu, Z., W. Hu und T. Tan (2005): Similarity Based Vehicle Trajectory Clustering and Anomaly Detection. In IEEE International Conference on Image Processing, Vol. 2, S. 2029–2032.

- [Hyden 1987] Hydén C (1987): The development of a method for traffic safety evaluation: The Swedish Traffic Conflicts Technique. Department of Traffic Planning and Engineering, Lund University.
- [IPC-Reference 2011] Reid Simmons und Dale James (2011): Inter Process Communication – A Reference Manual. Carnegie Mellon University, School of Computer Science / Robotics Institute.
- [Laureshyn 2010] Aliaksei Laureshyn. 2010. Application of automated video analysis to road user behaviour.
http://www.tft.lth.se/fileadmin/tft/video_in_traffic/Publications/Laureshyn_2010.pdf.
Stand 8.4.2014.
- [Lemmer 2013] Lemmer, K. (2013): Was bewegt uns morgen? Forschung für die Mobilität der Zukunft. VDI-GI-Vortragsreihe, 20. Juni 2013, Braunschweig.
- [Peesapati et al. 2013] Peesapati, L. N; Hunter, M. P.; Rodgers, M. O. (2013): Evaluation of Postencroachment Time as Surrogate for Opposing Left-Turn Crashes. In Transportation Research Record: Journal of the Transportation Research Board Vol. 2386, S. 42-51.
- [Statistik 2012] Statistisches Bundesamt Wiesbaden (2013): Unfallentwicklung auf deutschen Strassen 2012.
https://www.destatis.de/DE/PresseService/Presse/Pressekonferenzen/2013/Unfallentwicklung_2012/begleitheft_Unfallentwicklung_2012.pdf.
Stand 8.04.2014.

- [Yang et al. 2003] Yang, L., J. H. Yang, E. Feron und V. Kulkarni (2003):
Development of a performance-based approach for a rear-end
collision warning and avoidance system for automobiles. In
IEEE Intelligent Vehicles Symposium, S. 316–321. Columbus,
OH.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit eigenständig angefertigt habe und keine anderen Hilfsmittel als die angegeben verwendet habe.

Ben Kantwerk

Magdeburg, 21.04.2014