Multi-Modal Aggression Detection in Trains

Multi-Modal Aggression Detection in Trains

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J. T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 29 oktober 2009 om 15:00 uur
door

Zhenke YANG

Informatica Ingenieur
geboren te Guangdong, China

Dit proefschrift is goedgekeurd door de promotoren:

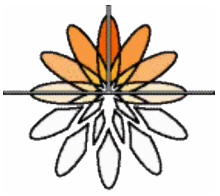
Prof. dr. H. Koppelaar

Prof. dr. drs. L. J. M. Rothkrantz

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. H. Koppelaar,	Technische Universiteit Delft, promotor
Prof. dr. drs. L. J. M. Rothkrantz,	Technische Universiteit Delft, promotor
Prof. dr. M. A. Neerincx,	Technische Universiteit Delft
Prof. dr. C. Witteveen,	Technische Universiteit Delft
Prof. dr. M. Novak,	Czech Technical University
Prof. Ing. V. Matoušek, CSc.,	University of West Bohemia, Plzeň

TU Delft
Delft University of Technology



This research was supported by BSIK MultimediaN: Professional's Dashboard (N6) project.

Publisher: TU Delft Mediamatica

ISBN 978-90-813811-6-1

© Copyright 2009, Zhenke Yang. All rights reserved.

Preface

This research is part of the MultimediaN N6 Professional's Dashboard project, to develop prototypes that support the professional user in his daily practice. The thesis work itself was carried out from July 2005 to October 2009 at the Man-Machine Interaction research group of the department of Mediamatics at the Delft University of Technology. Many parts of this work are thanks to the contributions of the Master and Bachelor students I have had the pleasure to work with during this period. I have very fond memories of the many discussions and exchanges of ideas and points of view.

For me, the PhD period has been a constant source of excitement that never seemed to weaken. I guess that is thanks to Leon Rothkrantz, who guided me through the whole period. His knowledge and inspiration helped me to conquer many challenges. At the same time, he gave me the freedom to "create something beautiful". His kindness to other people, and his willingness and devotion to help others, often at the expense of his own precious time, invariably left me with awe and admiration. I would also like to thank him for giving me the opportunity to continue work in the Netherlands Defence Academy. My special thanks to Henk Koppelaar for being a wonderful promotor and for being a source for support and reassurance.

Much gratitude is also owed to John Dietz and Henk Bulthuis from Pro-Rail and the NS for arranging the trains and the equipment to do the experiments, and also for facilitating the meetings with surveillance professionals.

This thesis would not have been possible without the support I have received from many people. I want to take this opportunity to express my gratitude to them. First, I would like to thank my colleagues of the MMI group (who I long since have considered to be friends) for proofreading and supporting me and giving joy and fun. I enjoyed the many "colaquiums" we had together. Alin, Alina, Bogdan, Coen, Dragos, Hani, Hantao, Harold, Nike, Pascal, Ramon, Siska, Wietske... Thanks. Also many thanks to my incidental proofreaders Sharon, Tracy and Ann.

Last, but certainly not least, I would like to thank my parents, in Suriname, who supported me emotionally, mentally (and financially). Thank you everyone from the bottom of my heart.

Zhenke Yang

Delft, October 2009

Contents

Preface **iii**

1 Introduction **1**
1.1 Objective . . . 3
1.2 Significance . . . 4
1.3 Methodology . . . 5
1.4 Outline of this thesis . . . 6

I Background **9**

2 Aggression in train compartments **11**
2.1 What is aggression? . . . 11
2.1.1 Types of aggression . . . 12
2.1.2 Components of human aggression . . . 12
2.1.3 Groups and aggressive behavior . . . 14
2.2 Aggression assessment . . . 15
2.2.1 Self-report questionnaires . . . 15
2.2.2 Expert assessment . . . 16
2.3 Aggressive behavior in trains . . . 17
2.3.1 Factors related to aggression . . . 17
2.3.2 Frequently occurring incidents . . . 17
2.4 Conclusions . . . 19

3 Data fusion for aggression detection **21**
3.1 Motivation for audio-visual fusion . . . 21
3.2 Problem analysis . . . 22
3.2.1 Challenges . . . 23
3.2.2 Requirements . . . 24
3.3 Data fusion . . . 24
3.3.1 Overview . . . 25
3.3.2 The JDL data fusion model . . . 26
3.3.3 Refinements to the JDL model . . . 27
3.3.4 Design of the aggression detection system . . . 28
3.4 Summary and conclusions . . . 29

4 Domain analysis **31**
4.1 An analysis of the train compartment . . . 31

4.1.1 Overview . . . 32
4.1.2 Compartment characteristics . . . 32
4.1.3 Objects in the train compartment . . . 33
4.1.4 Train compartment states . . . 36
4.2 Passenger behavior . . . 38
4.2.1 Groups . . . 38
4.2.2 Individuals . . . 39
4.2.3 Actions and interactions . . . 40
4.2.4 Emotion . . . 43
4.3 Aggression model . . . 44
4.3.1 Knowledge representation language . . . 45
4.3.2 The core ontology . . . 46
4.3.3 Aggression perception model . . . 49
4.4 Summary and discussion . . . 56

II Design **57**

5 A detection model for aggressive situations **59**
5.1 Overview . . . 59
5.1.1 Detection model for affective aggression . . . 61
5.1.2 Detection model for instrumental aggression . . . 61
5.2 Environment modeling . . . 64
5.2.1 Background subtraction . . . 65
5.2.2 Motion segmentation . . . 68
5.2.3 Sound source separation . . . 68
5.3 Object detection . . . 73
5.3.1 Human detection . . . 74
5.3.2 Unusual object detection . . . 75
5.3.3 Recognition of environmental sounds . . . 75
5.4 Tracking . . . 76
5.4.1 Target representation . . . 77
5.4.2 Data association . . . 78
5.4.3 Prediction . . . 79
5.5 Behavior recognition . . . 80
5.5.1 Behavior representation . . . 81
5.5.2 Activity recognition . . . 81
5.5.3 Affect recognition . . . 83
5.6 Discussion and conclusions . . . 84

6 Computational framework and architecture **85**
6.1 Motivation and design goals . . . 85
6.2 Requirements . . . 86
6.3 Framework overview . . . 87
6.3.1 Data distribution . . . 87
6.3.2 Communication between components . . . 90
6.3.3 Service discovery . . . 90
6.3.4 Memory . . . 91

6.4 System architecture 92
6.4.1 Intercompartment architecture 92
6.4.2 Intracompartment architecture 92
6.4.3 Sensor nodes 93
6.4.4 Processing nodes 94
6.4.5 Fission nodes 95
6.5 Discussion 95

7 A rule-based approach to reasoning about aggression 97
7.1 Model 97
7.1.1 Theory 98
7.1.2 Competing scripts 98
7.2 Designing the rule base 99
7.2.1 Influence matrix 100
7.2.2 Specifying scripts 102
7.3 Implementation overview 103
7.3.1 Program flow 104
7.3.2 Hypothesis evolution in time 104
7.4 Evaluation 105
7.4.1 Evaluation process 105
7.4.2 Results 106
7.5 Conclusion 108

8 A Bayesian-based approach to reasoning about aggression 109
8.1 Bayesian networks 109
8.1.1 Bayesian network components 110
8.1.2 Bayesian learning 111
8.1.3 Bayesian inference 111
8.2 Bayesian network of the proposed system 112
8.2.1 Implementation 114
8.2.2 Testing 114
8.2.3 Dynamic Bayesian extension 115
8.3 Evaluation 116
8.3.1 The footage 116
8.3.2 Activating the features 117
8.3.3 Evaluation 117
8.4 Conclusion 118

9 Aggression detection using an emergent based reasoning approach 119
9.1 Overview 119
9.2 The emergent model 120
9.2.1 The link matrix 120
9.3 Implementation 121
9.3.1 Fitness function 122
9.3.2 Time decay function 124
9.4 Results 124
9.5 Discussion 125

III Results **127**

10 Experiments **129**
10.1 Data acquisition . . . 129
10.1.1 Requirements . . . 130
10.1.2 Experiment setup . . . 131
10.1.3 Data products . . . 134
10.2 Data preparation . . . 134
10.2.1 Data annotation . . . 135
10.2.2 Annotation tool . . . 136
10.2.3 Camera alignment . . . 137
10.3 Evaluation . . . 139

11 Results **141**
11.1 Environment modeling . . . 141
11.1.1 Background subtraction . . . 141
11.1.2 Sound localization . . . 143
11.1.3 Motion estimation . . . 145
11.1.4 Energy signatures . . . 145
11.2 Object detection . . . 147
11.2.1 Human detection . . . 147
11.2.2 Emotions . . . 148
11.3 Tracking . . . 149
11.4 Activity recognition . . . 150
11.4.1 Unusual behavior . . . 150
11.4.2 Template-based classifier . . . 151
11.4.3 Abandoned luggage . . . 153
11.5 Results of reasoning methods . . . 154
11.5.1 Rule-based system . . . 155
11.5.2 Discussion . . . 155

12 Evaluation and conclusions **157**
12.1 Summary of results . . . 157
12.2 System evaluation . . . 159
12.3 Future work . . . 160

Appendix

A Annotation schema **163**

B Scenarios **169**

Chapter 1

Introduction

Aggressiveness and problematic behavior in public places can cause great distress on the part of innocent bystanders. When left untreated, the symptoms of this behavior may escalate and lead to physical aggression, destruction of property, and defiance which evoke even more frustration and anguish. According to a 2004 report on social safety[1] by the Dutch Ministry of Transport, Public Works and Water Management [1], 20% of all train travelers in 2004 have been the victim of an incident, the highest level in four years. The report also shows an increase in serious incidents such as abuse and theft.

Nederlandse Spoorwegen (Dutch Railways), or NS, is the principal passenger railway operating company in the Netherlands. The NS spends around 45 million euros per year to improve social safety in and around trains and train stations[2]. Some of the money is spend on improving the technical infrastructure. Currently, all international trains and some local trains (e.g. those on the Zoetermeer line) are equipped with cameras and microphones. To make effective use of the cameras, the images have to be monitored in order to detect and respond to aggression in a timely fashion. Currently, humans are called upon to perform this task (see Figure 1.1). However, the human ability to intensively and continuously observe dynamic changing scenery for lengthy periods of time is limited and unreliable. Added to this is the continual increase of the number cameras and other surveillance devices causing a data explosion and information overload for human operators.

The ability of computers to perform repetitive tasks for lengthy periods without performance loss make them very suitable to replace humans for the aggression detection task. In order to achieve this, however, the computer have to be taught how to deal with problems that human operators are trained to solve. By observing the way humans detect aggression, we can establish that information from the aural and visual modalities are necessary to accomplish this task. For example, in order to detect

[1] The term social safety includes both the subjective feeling of safety as well as the actual number of incidents e.g. fights, robberies, vandalism.

[2] In a 2004 survey, 84% of travelers and employees indicated a feeling of safety during the day, however this percentage dropped to 45% at night.



Figure 1.1: Analysis of real-time video is bound by the limitations of humans who are often required to monitor several cameras simultaneously to detect security threats.

who is shouting in a group of people, follow the movement of the mouths. This suggests that an automated aggression system based on data from just a single modality (an unimodal system) will not be sufficient.

Our approach therefore is to imitate the successful example set by humans and focus on a multi-modal (audio-video) surveillance system in which one modality can complement, support, trigger, assist and disambiguate data from another modality. Focusing on the humans in addition limits the range of sensors to normal cameras and microphones. Special sensors used by some researchers such as infrared or vibrometers [2] are essentially beyond the scope of this research. In this thesis we propose a data fusion system that combines sensor data from video as well as audio in order to automatically detect aggression and aggressive situations in a Dutch train compartment. Using ideas and techniques developed in the data fusion domain gives us decades of research experience to build upon. Figure 1.2 shows an overview of the proposed system.

This research was done as part of the MultimediaN N6 Professional's Dashboard project to develop prototypes that support the professional user in his daily practice. As industrial partners of the project, the NS and Pro-Rail were interested in achieving innovative and durable solutions for the aggression problem. To limit the scope of the project, it was decided to focus primarily on aggression inside a train compartment. At the same time in the Man Machine Interaction group at the Delft University of Technology, two other projects on multi-modal fusion were in progress [3, 4]. Both projects were focused on the algorithmic level of fusion. This provided opportunities for cooperation and symbiosis: on the one hand state of the art fusion algorithms can be applied in this research, while on the other hand, practical issues and new insights from this research provides feedback to the theory.

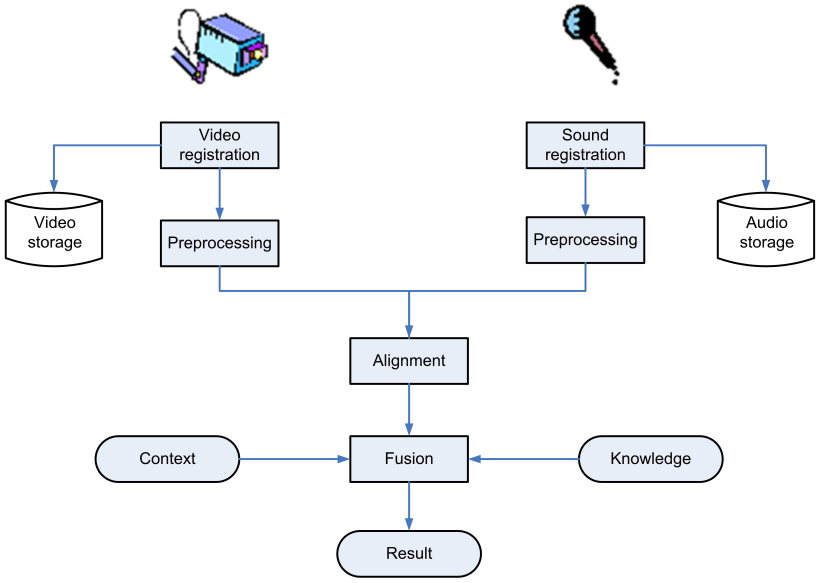


Figure 1.2: Overview of an Intelligent Multi-modal Surveillance System.

1.1 Objective

Following the above discussion, the goal of the project can be summarized as follows:

build an automatic, context aware aggression detection system that uses audio and video data to achieve a timely assessment of situations and their significance in train compartments in order to detect aggression and support human operators.

The specific aims of the project include the following:

Aggression detection. Not all aggression is directly identifiable. Some forms of aggression can only be detected by considering features over time. Others require combination of features into higher level concepts and reasoning about these higher level concepts. In addition aggression always needs to be interpreted in the proper context. Thus, domain knowledge of possible aggressive situations is necessary. Moreover, a process of knowledge acquisition from human experts is required to obtain domain and behavior models that describe how passengers behave in the train compartment. Furthermore a reasoning mechanism is needed that can draw relevant conclusions based on the current data and the models.

Multi-modal fusion. Attempts to build automatic surveillance systems are traditionally primarily focused on video. The inclusion of audio information is argued to have significant benefits for the performance of the surveillance system. Therefore our approach differs from others in that we include fusion of video and audio information from the start. Multi-modal fusion deals with the association, correlation, and combination of data and information coming from

multiple sources and multiple modalities. The result is a value judgment of the amount of aggression in a compartment that can be presented to the human operator.

Software framework. The goal of the software framework is to provide the underlying functionality (such as error handling mechanisms, communication infrastructure) to facilitate aggression detection. Another important property of the software framework is that it allows for a structured decomposition of the application into algorithms, models and prior knowledge at different levels of abstraction. This makes it possible to adapt or replace components without altering the remainder of the application, making the system flexible to be applied with different combinations of algorithms.

1.2 Significance

The scientific and technical significance of our research lies in the novel application of existing technologies and development of new concepts to tackle the problem of data fusion for the surveillance in a train compartment.

Several attempts have been made in the past to achieve automated surveillance for improving passenger safety in public transport systems. In the PRISMATICA project [5] a surveillance system for railway stations was designed to integrate different intelligent detection devices. In the system, detection of events is done in spatially distributed devices and using different algorithms. Bayesian networks are used for presenting and fusing the data of these devices. In the ADVISOR project [6], a surveillance system was developed for detecting specific behavior (such as fighting or vandalism) in metro stations. Just as in the PRISMATICA project, different methods and algorithms are used to recognize context specific types of behaviors under different situations. Compared to previous work, we adopt a more human centered approach by extracting knowledge and rules of thumb from security experts. In addition we aim to solve the issues identified in previous projects such as difficulty with testing and comparing such systems by means of a software data fusion framework designed specifically for multi-sensor, multi-modal and multi-algorithm systems.

The results of this research will have social and economical benefits as well. With a successful working aggression detection system, less human operators will be required to monitor the same amount of compartments. For the Dutch Railways, the potential savings in costs of human power will hugely outweigh additional costs to install extra hardware and software, especially considering that computer systems can work all day with no loss of quality and performance. The social benefits are that having constant aggression detection results in safer trains for passengers and NS personnel and a higher willingness of people to travel by public transport, resulting in more economical and environmental benefits.

1.3 Methodology

To achieve automated aggression detection, a successful integration and application of ideas and results from various fields of research, from psychology to signal processing and AI, is required. This thesis reports on the tasks that resulted in the final automated aggression detection system. The tasks are described below in chronological order.

Aggression survey. We start the research by first examining the concept of aggression from the psychological perspective as well as from the perspective of the NS and from a train passenger. Next we conduct a data and environment analysis by examining the train and identifying the objects in the train and their characteristics. In addition we hold interviews with security experts to discover the process and the cues that trigger them to classify a situation as aggressive. The interviews with experts also result in a list of features[3] that play an important role and can contribute to the automatic aggression detection process.

Data fusion survey. Given the list of features associated with aggression, we examine the feasibility of existing algorithms to detect these features. This results in a list of feature detection algorithms, a characterization of their complexity, performance and availability. Next we examine data fusion models for combining and fusing multi-modal data. We focus on models that allow for a structured decomposition of the data fusion system into independent functional units, each performing their part in the total process. As good communication between different data fusion components is of great importance for a good overall performance, we create a train aggression ontology to provide a formal specification of the communication language between all components. The ontology also serves as the semantical grounding for higher level reasoning.

Modeling and conceptual design. Analysis of the results of the aggression survey produced a list of relevant semantic concepts and their relationships. In addition, a perception model is developed that consists of (1) an observation part that describes how sensor data can be processed to yield features and concepts that can be reasoned with and (2) a reasoning part that captures the process that security experts perform in they daily job when they reason with observed concepts and draw conclusions about the level of aggression. Finally a conceptual design of the aggression detection system is made based on the requirements of the observation model set against the technical possibilities and limitations identified in the data fusion survey.

Data fusion framework. A problem associated with developing systems that combine different features using different algorithms, is that for some feature there

[3] Humans/security experts tend to reason with high level concepts instead of low level features. For each high level concept, the low level features that are associated with and constitute the concept must be identified.

may be many detection algorithms available (or under development). It requires a framework that makes it possible to replace or modify a component without altering the remainder of the application. For the communication between the components a language is used based on a train aggression ontology mentioned earlier.

Design and implementation of the system. On top of the framework, the actual aggression detection system is built based on the conceptual design. The system is meant to function as a warning system working alongside a human operator (by indicating the compartments with the highest probability of aggression). Since humans are still the most reliable aggression detectors, we will focus on a system that mimics human perception and reasons with human knowledge. This knowledge (i.e. the underlying semantics of concepts in the vocabulary) is expressed in the ontology mentioned previously. Several reasoning models will be explored including the rule-based approach, the (Dynamic) Bayesian approach and the emergent behavior approach.

Testing and evaluation. To test and evaluate the system we are in need of data of aggression taking place in the train compartment. Because of privacy and legal issues it is not appropriate to use real live data. In data acquisition experiments in a train compartment, we used storyboards to illustrate to actors the aggressive scenarios they have to perform. As the actors performed the scenarios, audio visual data of their actions were captured with the cameras and microphones located in the train. We used the raw data to test and evaluate the aggression detection system. The aggression detection system consists of a discrete sound event recognizer and several discrete object recognizers to detect objects and events in separate samples and frames. Simultaneously, human attributes such as faces, movement and movement change are extracted from the data. In addition, crowd attributes such as crowd size, inter-distance and total movement, are calculated. A tracking algorithm keeps track of human motion, while an activity recognizer tries to assign labels to these motion patterns. Finally, high level inference engines give a judgment of the situation based on the available data. The captured data was also annotated using an annotation language derived from the train aggression ontology.

1.4 Outline of this thesis

The dissertation is divided into three parts. Part I contains an overview of the problem domain to present the background and the context for the research. A generic data fusion framework forming the basis of the contribution is presented in part II. This part also contains the discussion of the detection model and presents three different approaches to the automatic aggression detection problem: the rule-based approach, the Bayesian approach, and the emergent behavior approach. Part III presents the experiments and the results.

Part I is constituted of Chapter 2 to Chapter 4. Chapter 2 gives an introduction to the aggression problem and the relevance of a proper solution. The foundations for a model for aggression will be presented in this chapter, which will be further elaborated into a detection model in Chapter 5. Chapter 3 argues the case for audio-video fusion and presents relevant background from the field of data fusion. Chapter 4 focuses on the train compartment and analyzes the actions and activities that take place in it. The ontology that formalizes this analysis is also presented in this chapter.

Part II describes the software foundation of the aggression detection system. It first presents the computational model to detect relevant low level features in Chapter 5, followed by a presentation of a generic multi-modal data fusion framework that provides the basic functionality on which data fusion systems can be built (Chapter 6). Finally, several reasoning models will be explored, namely the rule-based approach (Chapter 7), the (Dynamic) Bayesian approach (Chapter 8) and the emergent behavior approach (Chapter 9).

Part III and final part of this thesis presents the experiments and the results to validate the aggression detection model and system. It starts with a discussion of the data collection experiments, and a description of the data that was captured (Chapter 10). Chapter 11 discusses the results of the separate algorithms. The validation and evaluation of the developed system is presented in Chapter 12. Chapter 12 finishes with overall conclusions, outlook and suggestions for future work.

Part I

Background

Chapter 2

Aggression in train compartments

Although The Netherlands is considered to be a safe place according to international standards, many people in the Netherlands will confirm that aggression still exists in their society. According to a 2004 report on social safety by the Dutch Ministry of Transport, Public Works and Water Management [1], 20% of all train travelers in 2004 have been the victim of an incident. Even though the majority of interactions do not involve violence, they are nevertheless the source of a great deal of physical pain and psychological damage. The broad way the word aggression can be used, has caused all attempts to understand aggression at a generic level to be criticized as inadequate. In our discussion on aggression we will focus on human (and crowd) behavior with respect to observable features. This section discusses research in this area and creates the foundations of a model based on data from observation. We will present the different types of aggression and discuss methods and techniques to detect these various forms of aggression.

2.1 What is aggression?

Aggression seems to be a real and important part of the human condition. Most people, when they hear the word aggression immediately think of a fight, verbal harassment, or any other form of intense action in the conflict between two people. However, aggression covers far more than this image: it encompasses any behavior with an intent to harm another person. Despite this, there seems to be general agreement regarding the principle characteristic of aggressive behavior: *the infliction of partial or complete destruction on a human victim or other entity*. Disagreements arise in the interpretation of what precisely constitutes destruction (e.g. injury may but need not be only physical), who or what constitutes the target of aggression and if aggressive behavior needs to be intended or not [7].

2.1.1 Types of aggression

Because of the broad definition of aggression, many types of aggression are encountered in literature. The principle distinctions of interest concern (a) the infliction of injury versus the infliction of harm in more general terms, (b) the intention to inflict injury or harm versus the expression of anger or annoyance, and (c) the communication of threat. The current consensus in the scientific community is for at least two broad categories of aggression [8, 9]. The main pairs are affective versus instrumental, impulsive versus premeditated and reactive versus proactive. Empirical research indicates that there is a critical difference, both psychologically and physiologically.

Instrumental, premeditated and proactive aggression. The two important characteristics of this group are goal-directedness and planning or deliberation. The instrumental aggressor acts to obtain a readily apparent goal for example pick-pocketing to get money. In theory, this group of aggression does not need a stimulus or provocation to occur. However, security and surveillance experts will concur that it mostly occurs when the situation and the environment permits. Being able to identify and detect these so called "high risk" environments forms one part of the detection process. As a result of the premeditated nature of the aggressor, it can be expected that in some cases an attempt will be made to fool or mislead the observer. Proactive aggression is sometimes accompanied by pleasure or satisfaction (from the aggressor).

Affective, impulsive and reactive aggression. Affective aggression is associated with strong emotional feelings. Anger is usually the dominant emotion in this type of aggression. Anger as an emotion is relatively short lived for most people, however, the retaliatory aspect of anger can come long after provocation. Reactive aggression often follows as a defensive response (that is accompanied by anger) to a perceived or actual provocation. If this happens, the aggression can be detected as a cycle of provocations and responses. The type of provocation and response can be both verbal and physical. Anger can be recognized from body language such as poses, facial expressions and speech.

Table 2.1 shows a short summary of types we encountered and ideas for identification of the type of aggression.

2.1.2 Components of human aggression

The origins of violent and aggressive behavior have been studied for many years. Researchers have wondered why some people exhibit different forms of behavior under identical circumstances. Psychologists have sought the cause in different factors

Table 2.1: An overview of aggression types and means of its identification.

Aggression type	Instrumental
Characteristics	Goal-directedness and planning to obtain a readily apparent goal.
Recognition	Movement and gesture tracking. Matching against known aggressive behavior patterns. Detect risky, provocative environments such as crowded compartments, train delays, ticket checking[1].

Aggression type	Affective
Characteristics	Strong emotional feelings. Anger is usually the preeminent emotion.
Recognition	Anger (and other emotions) can be recognized from faces, speech, gestures. Cycles of provocation and responses may occur before escalation.

such as culture and upbringing and genetic inheritance [10].It is generally accepted that there is no single cause of aggression, but that it results from a complex interaction of physiological, psychological, and environmental variables. According to Bandura [11], most aggressive behaviors are acquired through observational learning (modeling or imitation). The fact that aggressive behavior can be learned, suggests that there must be certain relevant features in the environments and in the behavior of people and crowds, that allow for detection of aggression. In addition, Bandura has suggested that there are three essential components of aggression.

Origins are the factors related to acquisition or development of aggression. Important sources are the family, the subculture, and media events.

Instigators are the events that activate or provoke aggression. The instigators of aggression include observation of aggressive acts, aversive treatment, expectations of positive outcomes of aggression, instruction, and bizarre symbolic control (e.g. delusions or hallucinations).

Reinforcers reward aggressive behaviors. Aggressive and violent behavior can be rewarded through external reinforcement (e.g. getting money or objects of value, gaining status), vicarious reinforcement (seeing others being rewarded for aggressive behavior), or self-reinforcement (enhancing self-esteem or blaming/dehumanizing the victim).

Other studies show that hostile cognitions and negative affect are positively correlated with aggression, with anger acting as a strong mediator between hostility and physical aggression [12]. Hostility was also found to be the best predictor of being the target of physical aggression. Anger is even theorized to represent the emotional component of aggression, whereas hostility is regarded as a cognitive component [13]. It should be noted however, that aggression can take place without the

presence of anger and the presence of anger does not automatically lead to aggression. So, anger is neither a necessary nor sufficient component of aggression.

During the time aggression exist, it generally passes through a number of stages.

Trigger phase. The trigger phase or initiation is characterized by explosions of anxiety.

Escalation phase. Accumulated anxieties lead into the escalation phase, characterized by mounting anxiety, changing body language (e.g. staring, head down, frowning, clenched teeth, breathing fast, facial coloring) and the way we communicate with each other (e.g. swearing, verbal venting, intimidation).

Crisis phase. If the situation is unable to de-escalate, we may enter a crisis phase of uncontrollable violence.

De-escalation. De-escalation may occur naturally but often needs external intervention. De-escalation leads back to a situations of peace and calm.

2.1.3 Groups and aggressive behavior

A group or crowd is a number of people temporarily congregated. Generally, the members of a crowd think and act as individuals and are without organization. Recent studies on aggression have also focused on the behavior of individuals in a group and the influence of the group on the individual. Four main crowd types have been identified, with differing behavioral tendencies[2].

Casual crowds have no unity of purpose and no leadership, e.g. shoppers, on-lookers, or watchers who come and go, and will usually respond to direction by police.

Cohesive crowds usually assemble for a common purpose without leadership, e.g. spectators at sporting events, along parade routes, and other activities. They have a common interest but behave and think as individuals.

Expressive crowds are assembled for some purpose. They have leadership and the intention of expressing an attitude for or against some person or idea, e.g. political rallies and picket lines.

Aggressive crowds are usually assembled for some purpose. They have positive leadership, are determined to accomplish a specific end, and move actively toward their objective. Usually high emotional tension is present. Consequently, this type of crowd presents an acute police problem.

[2] http://www.tcleose.state.tx.us/GuideInst/HTML/1000/31CRO.htm

Aggressive behavior results when the group develops into a mob when all or most members are intent to carry out their purpose. As a result it is important to recognize when it becomes necessary to control a crowd. Factors that play a role in crowd control are knowledge to identify the legality of a crowd (i.e. spectators going to a sporting event) and knowledge to recognize and determine the leaders. The front and center of the mob is usually the highest point of tension. Excited individuals acting as leaders can usually be located there.

Psychological and computational models of crowd behavior have mainly focused on crowd animation and simulation. Examples can be found in [14, 15, 16, 17]. Automatic control of groups has been a subject of studies for the last twenty years. It involves many difficult problems that have to be solved: conflict detection, modeling of uncertainties on trajectories, clustering etc.

2.2 Aggression assessment

Research in the psychological background and the classification of different types of aggression gives a better insight in the fundamentals of aggression. However, it still remains difficult to assess the effects of aggressive behaviors. This is caused by the lack of suitable methods for measuring the severity of aggressive acts. A classification in types of aggression can be useful in determining the best way to detect a certain type of aggression, while a measure of the severity of the aggressive situation can be used to determine the impact and the best actions to take.

2.2.1 Self-report questionnaires

In the past several attempts have been made to create aggression scales in order to assess overt components of aggression (e.g. arguing, threatening, hitting, fighting). Many of the scales designed are self-report questionnaires of angry feelings, violent thoughts or reactions to anger provoking situations. As a result, these scales represent subjective judgments of aggression. There are a few objective scales that may be used in rating aggressive behaviors, but most of them are too specific of a certain area e.g. The Nurses' Observation Scale of Inpatient Evaluation (NOSIE), and the Brief Psychiatric Rating Scale (BPRS).

As an aggression scale for aggressive situations does not exist for trains in the Netherlands, we designed a survey [18] to assess the impact of events in a train on social safety and passenger satisfaction. Adopting the approach taken in [19], we divided the questionnaire of the survey into aggression categories: verbal aggression, physical aggression against objects, physical aggression against self, and physical aggression against others. Within each category a scale is created by listing examples of aggressive behavior in order of increasing severity. The participants of the survey were daily train commuters from the university. The participants had to rate each

situation in terms of danger and own experience. The results of the survey provide a scale for actual (and experienced) severity and frequency of aggressive situations in the train.

2.2.2 Expert assessment

At all train stations in the Netherlands, surveillance cameras are installed and in most of the major stations there is a team of security personnel that monitor the surveillance images. All occurrences of major and minor incidents that have been observed or reported by passengers are put in an incident database. In an effort to obtain statistics of behaviors that occur most frequently in train compartments, we interviewed security experts from the NS. The interviews were also meant as a way to get insights in the way human experts detect aggression. During the interviews, the experts explained their thinking process while walking through several archived videos from security cameras. After several such session it was clear that training and experience plays an important role in the detection process: observation of crowd behavior takes into consideration the influence of the behavior of one member (e.g. instigator) on the entire crowd, cultural awareness and cognitive psychological factors, and identification of the presence of weapons, size of crowd, use of alcohol and drugs etc. Furthermore, analysis from the videos showed the stages in the dynamics of aggression. For example a pickpocket will start to find a suitable victim. When an opportunity presents itself e.g. when the train arrives and the victim is distracted, he will re-examine the immediate surrounding environment, make a plan (increment) and finally execute the plan and do the actual stealing (climax). After this he tries to exit the scene unobserved (dispersal). A detailed report of these interviews can be found in [20].

The interviews with the experts resulted in the stepwise procedure for aggression detection and control summarized below.

1. Trigger: while watching the video streams attention can be directed towards a specific stream by a trigger. The trigger could be the detection of something unusual, experience, an external warning call, a hunch, etc.
2. Orient: find the most salient cues in the scene and create one or more hypotheses. The hypothesis could be that everything is normal. In this case it is a false alarm otherwise go to the next step.
3. Observe: find other cues that support or refute the hypotheses. The more supporting cues the more plausible the hypothesis becomes.
4. Conclude: when a threshold of plausibility is reached the hypothesis is concluded. The value of the threshold is a combination of training, experience and personal judgment.
5. Act: take the appropriate actions to solve the problem.

2.3 Aggressive behavior in trains

The NS employ specially trained surveillance personnel to monitor the images of the surveillance cameras. The agreement is to work in shifts, such that at any moment at least three operators are in the control center analyzing the camera images, answering telephones etc. The adopted aggression assessment techniques are based on the judgment of the operators.

2.3.1 Factors related to aggression

From interviews with experts we compiled a list of factors related to aggression and violence in trains. This list differs from the scientific studies in that it represents a more pragmatic approach to aggression (prevention).

Early predictors. e.g. aggression history of an individual, increased verbal aggression.

Opportunity factors. e.g. absence of supervision, presence of potential victim, availability of a weapon, presence of a distraction.

Precipitating events. e.g. substance abuse, invasion of personal space, instruction from another, frustration.

Contributing factors. e.g. family and cultural influences, gang/cult/group influences.

Escalation conditions. e.g. crowding, irritating noise, boredom, unsocialized group members, rewards for aggression.

Based on these factors, the expert adopts heuristics to relate observations to the conclusion that the situation in the compartment is indeed aggressive. From interviews we gathered some of the heuristics summarized in Table 2.2.

Table 2.2 is still very generic, in the domain analysis in Chapter 4 we will refine the list with concrete cues that are relevant and specific for aggression in the train environment.

2.3.2 Frequently occurring incidents

The incident database of the NS stores information about all incidents that have occurred around the rail. Whereas the scientific perspective is interested in pinpointing the underlying mechanisms that cause the aggression to manifest itself, the NS is more interested in the cost of resulting physical or emotional damage by the incident. For example, painting graffiti in trains is considered an act of aggression. Due to this

Table 2.2: Heuristics used by security personnel of the NS to spot possible aggression.

Heuristic	Description
Time of day	The chance of aggression increases when trains are leaving or entering the train station.
Place	Certain places pose increased risk e.g. near ticket vending machines, near the entrance of the train.
Objects	Knifes, beer bottles, but also seemingly abandoned luggage.
Targets	Tourists are frequent targets of pickpockets.
History	Special attention is given to people who have a history of aggressive behavior.
Provocation	Actions such as shouting, begging or singing may provoke aggression.
Evasion	The conductor or other people of authority may cause evasive behavior such as running or walking away.
Timing	When people are waiting too long for the train (normal passengers spend 10 to 30 minutes in the station), or if their actions are too fast.
Moving against flow	People who walk opposite the general flow of the crowd hinder other people. It might indicate a robbery attempt.
Attention	Passengers who pay too much attention to their luggage find the contents valuable or dangerous.
Direction of attention	The direction of attention can be a reason for alertness. If all people look in the same direction they may be looking at something very unusual.

Table 2.3: Incident categorization used by the NS.

Category	Description
A	Suspicious behavior
B	Robbery and theft
C	Violence
D	Serious public inconveniences
E	Small public inconveniences
F	Vandalism
G	Accident
H	Fire

viewpoint, the aggression categorization by the NS is tailored towards the procedures that should be taken when an incident of a certain category occurs (see Table 2.3).

Another important reason to study the kind of aggressive acts the NS is interested in is that results of the detection system can be matched more easily against the expectations of the NS. The list of incident types in Table 2.3 will therefore be considered the output of our aggression detection system. If the detection system is able to output

the type of aggression and its severity, the result can be used directly in the information systems the NS already has. In addition, the human operators monitoring the systems will be confronted with familiar terms.

2.4 Conclusions

Aggression is such a widely used and broad concept that there is no clear and concise definition for it. For a long time researchers have tried to categorize aggression. In this thesis we distinguish between two main types: aggression with a goal (instrumental aggression) and aggression from strong emotional feelings (affective aggression).

An important observation is that instrumental aggression is a result of cognitive processes that map perceptions through decision processes to produce actions oriented towards goals. Firstly, this means that the environment, in particular the perception of it, influences the behavior of an aggressive individual and secondly, since aggression is a cognitive decision process, aggressive actions can be explained by reasoning back from the behavior to the goal (while taking into account the circumstances). The aggression detection model presented in Chapter 5 is primarily founded on these two conclusions.

In affective aggression, emotions are involved (with anger being the predominant one). Emotions can be conveyed in different ways such as text, speech, facial expressions and gestures. Compared to instrumental aggression, affective aggression detection involves relatively low level emotion recognition algorithms. However, reliable conclusions about aggression can only be achieved by taking into account high level context information. Consequently, affective aggression detection is best combined with other high level detection models such as instrumental aggression.

The current problem of automatically assessing aggression boils down to a lack of objective severity measures: it takes expert knowledge to judge when to take action and how urgent the situation is and how urgent it can become (as aggression goes through several stages). The measure of severity is crucial in aggression management and is a requirement to be able to take actions at the correct moment. Human experts rely on training and experience to aid them to make this judgment. It is therefore too ambitious to aim to build a system that handles aggressive situations automatically. Instead, we aim to develop a computer model that can infer potential aggressive situations and leave judgments regarding severity and urgency to a human operator. Thus the focus of this thesis is on detection, not assessment.

The goal of the aggression detection system is to support a human operator in monitoring the safety of the Dutch train system. Its main task is to help the human operator to sift through huge amounts of incoming data from the sensors and calculate the potential of aggression. When this potential is high enough a warning is given to the

human operator. To achieve this, we analyzed the human expertise and extracted the way experts recognize aggressive situations through interviews. More specifically, we identified the factors in the environment that influence human behavior and gathered the heuristics used by humans to spot upcoming aggression. In Chapter 4, the expert knowledge extracted in this chapter will be elaborated with environment cues to form the aggression model. But first, we will argue the case for multiple modalities and discuss the technical approach to achieve multi-modal aggression detection.

Chapter 3

Data fusion for aggression detection

Automatic surveillance systems have traditionally primarily focused on video as input source, while audio as a source of information has largely remained untapped. Audio data however, contains a mountain of potentially useful information that is not present in video data. Therefore the inclusion of audio information in the surveillance process will have significant benefits to the quality of the surveillance system.

To assist in this endeavor, there are many data fusion techniques that combine and assess information from multiple sources in an efficient and effective manner. This chapter gives an overview of data fusion and discusses the most important and common data fusion models. The purpose is to create a common reference framework and unified terminology that can be used throughout this thesis. First a definition of data fusion is given along with the issues concerning fusion, and an overview of the benefits.

3.1 Motivation for audio-visual fusion

To make effective use of the sensors installed in the train, the data from these sensors have to be monitored continuously in order to detect and adequately respond to aggression and aggressive situations. Currently, trained humans are called upon to perform this task. However, the human ability to intensively and continuously observe dynamic changing scenery for lengthy periods of time is limited and unreliable. Added to this is the continual increase of cameras and other surveillance devices causing a data explosion and information overload for human operators. Given the ability of computers to perform repetitive tasks for lengthy periods without performance loss, the obvious solution is to implement an intelligent system to take over this task from humans.

Cameras are currently the main sensing devices employed in the surveillance of train compartments, consequently visual inspection is the main device to detect aggression. This unimodal approach was born out of human limitation. People are able to monitor several video streams, but not several audio streams simultaneously. The inclusion of audio information in the automated surveillance process will have significant benefits to the quality of the surveillance system:

- The main argument of our decision for audio and video data is to imitate the successful example set by humans. Humans are very good in detecting and assessing aggression in their direct environment. When we observe the way humans detect aggression, we can conclude that the primary senses for the detection are the sense of sight (e.g. emotion, gestures, events, actions) and the sense of hearing (e.g. stress in the voices, gunshots).
- New findings from the area of speech recognition [21, 22] and facial expression recognition [23] indicate that humans convey their emotional state through both modalities (voices and expression). By considering only one modality, valuable information is lost.
- Studies have shown that the security personnel monitoring the surveillance cameras, find it very difficult to detect aggression using only the image material from the cameras. Often they depend on warnings from people on site (once a warning is given, they have no problems following the flow of aggression using focused image data). This finding suggests that video is a necessary but not a sufficient criterion to spot aggression.
- Audio startups (e.g. Sound Intelligence, BioMimetic Systems) are specialized in detecting sounds of violence. This indicates that some aspects that indicate aggression are contained within audio information.
- Audio and video complement each other in the sense that one does not suffer from the problems of the other: problems in video such as occlusion do not exist in audio, while objects can be detected in video even if they are completely silent.

3.2 Problem analysis

Recent years have seen increased attempts to apply data fusion for surveillance of public spaces. The information overload currently experienced by human operators has the potential to worsen significantly because the amount of data is steadily rising, due to the increased number of cameras and other monitoring devices. Multimodal data fusion has addressed in the military domain the issues of information overload [24].

3.2.1 Challenges

A prerequisite for an automated aggression detection system is the ability to detect abnormal behavior or patterns in the data stream. In the field of pattern recognition, techniques for abnormal pattern detection have already been developed. For example, in quality control in production processes smart camera's can accurately detect and classify the good products from faulty ones in real time.

The main challenge however is the unpredictability and intricacies of human behavior (aggressive behavior being no exception). Section 4.2.3 presents some cues for detecting aggressive behavior, but detecting the cues and interpreting them in the correct context is still a big challenge. Apart from this main challenge, several other technical challenges exist. Most of them stem from the unpredictable world of the train environment in which environment parameters are just partly under our control.

Lighting conditions may change rapidly in the train. During the day, the compartment is illuminated by sunlight, which depends on time of day, season and weather. In addition, rapid change may occur when the train passes through a tunnel. At night there is relatively stable lamplight, but this may change as the train passes a bright light source located outside the train (e.g. when the train passes through a tunnel).

Background noise in the train is not stable. The type of noise may change depending on the state of the train e.g. moving, stopped, braking. In addition, the volume of the noise may vary, leading to unpredictable signal to noise ratios.

Train movement , especially bends in the railroad track, cause shaking in the compartment inducing involuntary movement of passengers and sensors in the train. Such movement also occurs when the train is braking or accelerating.

Occlusions. The confined space and arrangement of objects in the train compartment cause occlusions by seats and by other passengers.

Echoes and reverberations are also caused by the confined space and arrangement of objects in the train compartment.

Even more challenges arise as the result of the multi-modality of the system: multiple cameras are installed in each train compartment and each camera has panning, tilting and zooming capabilities. To add the audio to the detection system, microphones have to be installed in the compartment too. The cameras and microphones in the train produce video and audio signals which, at a certain moment in the process, have to be combined. The combining of this non-commensurate information is the topic dealt with in Section 3.3.

3.2.2 Requirements

If one algorithm would be able to assess the degree of aggression in a compartment, the aggression detection system is simply a matter of running this algorithm in all compartments and comparing the output of each compartment. Compartments can exchange information, so that incidents can be tracked over several neighboring compartments as passengers move from one compartment to another. The operator can use this comparison to select the most aggressive compartments to monitor. In addition, when the aggression level of a certain compartment exceeds a threshold, an alarm can be given.

From the above description, the global functional requirements of the system can be summarized as follows (see also Figure 3.1).

- The system consists of detection units and each detection unit is able to detect different types of aggression and assess the degree of aggressiveness in a train compartment.
- A detection unit is able to convey the results of the assessment of a compartment in real time to a human operator.
- The system is modular and scalable in that detection units can be installed in every train compartment. These detection units should be able to communicate and exchange information with neighboring units.
- The system should be able to compare and rank aggression detection results of different detection units.
- The system should be able to issue a warning when the aggression level of a detection unit exceeds a threshold.

As to the evaluation of the classification system, achieving performance levels like that of humans is often held as the ultimate goal. The truth is that current classification systems invariably fall short in comparison with the flexibility to changing environments that the human can deal with. Therefore, we will make a ranking of aggressive situations based on impact and threat level. Our goal is to detect the top 25% of most critical of aggressive situations.

3.3 Data fusion

As the aggression detection system should use data from both audio and video sources, we need a method to combine the information from these different sources in an efficient and effective manner. The research area of data fusion deals with this problem.

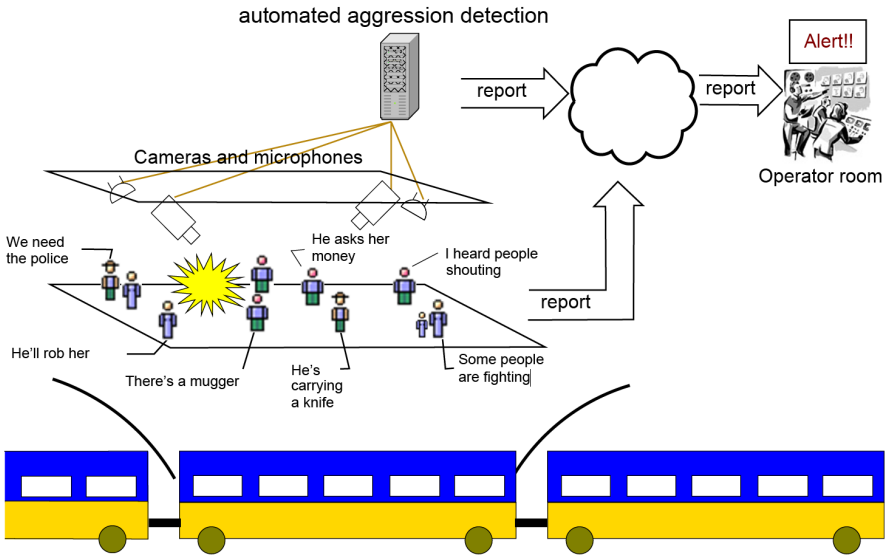


Figure 3.1: A schematic overview of the proposed solution with a modular aggression detection unit per compartment. A detection unit is able to convey the results of multi-modal (audio-visual) assessment of a compartment to a human operator.

3.3.1 Overview

Data fusion originated from the military domain where the volume of information presented to army intelligence analysts significantly exceeds their capabilities to fully analyze and interpret it in a timely manner. The initial Joint Directors of Laboratories (JDL) Data Fusion Group[1] defined data fusion as [25]:

A process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats, and their significance. The process is characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results.

Because this definition is very broad, the research area is best illustrated by listing the specific problems data fusion researchers typically deal with:

- Combining information from various types of sensors, ranging from numeric measurements to linguistic reports and maintaining the enormous number of ways to associate and interpret large numbers of observations of multiple entities (sub-object assessment).
- Estimation, identification and prediction of the states of entities both external and internal to the system of interest (object assessment).

[1]The JDL Data Fusion Group is now re-charted as the Data and Information Fusion Group within the Deputy Director for Research and Engineering's Information System Technology Panel at the U.S. Department of Defense.

- Estimation of the context, by taking into account the meaning of the observed entities, the interrelations among the entities and the relations between entities and the environment (situation assessment).
- Estimation of the consequences of the current estimated situation (impact assessment).
- Refinement of the data fusion process to improve the processing results (process refinement).
- Refinement of the fusion process to improve interaction with the human decision maker (cognitive refinement).

In practice, data fusion applications heavily depend on the application area and the demands of the specific problem. Thus, there exists no generally accepted model for fusion. It is also unlikely that one technique or one architecture will provide the perfect solution. As a result, numerous architectures and models for fusion can be found in the literature. The JDL data fusion model, developed in 1985, with subsequent revisions [26], is currently the most widely used system for categorizing data fusion-related functions [27].

3.3.2 The JDL data fusion model

Since there is unlikely to be a clear cut solution for any given data fusion problem, the JDL model was developed to provide a common frame of reference for fusion discussions and to facilitate understanding and recognition of the types of problems for which data fusion techniques are applicable. The model helps researchers to recognize similarities among data fusion problems and select (tried and tested) candidate solutions. The JDL data fusion model consists of several processing levels: levels 0 through 4 (see Figure 3.2). The fusion levels are not meant to be processed in a strict order and can also be executed concurrently.

Level 0 deals with sub-object data assessment i.e. the estimation and prediction of signal- or object-observable states on the basis of signal-level data.

Level 1 deals with object assessment i.e. the estimation and prediction of entity states on the basis of inferences from observations.

Level 2 deals with situation assessment i.e. the estimation and prediction of context on the basis of inferred relations among entities.

Level 3 deals with impact assessment i.e. the estimation and prediction of effects on situations of planned or estimated actions by the participants (e.g. assessing susceptibilities and vulnerabilities to estimated threat actions, given one's own planned actions).

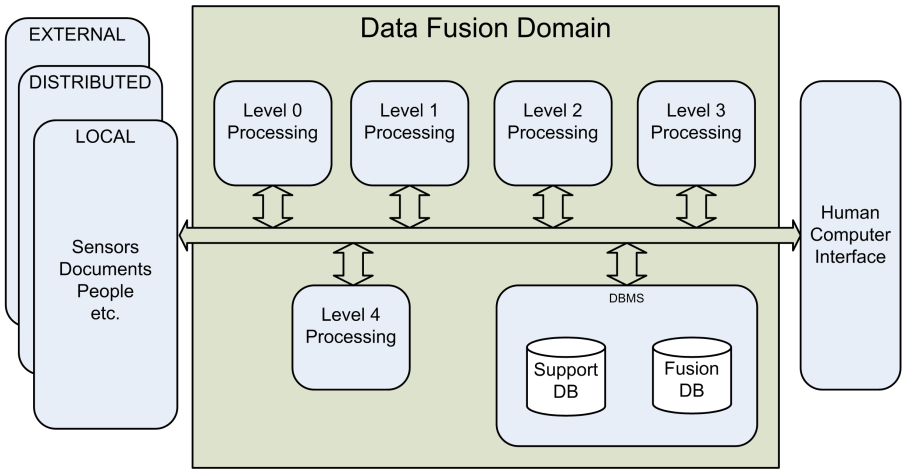


Figure 3.2: The data fusion model developed by the JDL Data Fusion Group consists of several levels of processing that work together to transform sensor data to usable information for humans.

Level 4 deals with process refinement i.e. adaptation to support goals and mission objectives.

3.3.3 Refinements to the JDL model

Although the original data fusion model has gained great popularity, there exists the concern that this model definition is heavily influenced by military operations thinking patterns. Steinberg et al. [26] tried to broaden its implication using more general terms. In a subsequent paper, Llinas et al. [28] suggest ways to further revise and extend the JDL model. Proposals are made regarding (a) improvements in the understanding of internal processing within a fusion node and (b) extending the model to include (1) remarks on issues related to quality control, reliability, and consistency in data fusion processing, (2) assertions about the need for co-processing of abductive/inductive and deductive inferencing processes, (3) remarks about the need for and exploitation of an ontology-based approach to data fusion process design, and (4) extensions to account for the case of distributed data fusion. Meanwhile Blasch et al. [29] address the issue that the JDL model is only intended for automatic processing of a machine and does not account for human processing, although a fusion architecture typically supports human users. Consequently, a level 5 process, called user refinement, was proposed to delineate the human from the machine in the process refinement. Thus level 5 processing deals with the adaptive determination of who queries information and who has access to information, to include knowledge management. Elmenreich [30] identified the following drawbacks of the model:

- It is a data-centered or information-centered model, which makes it difficult to extend or reuse applications built with this model.

- The model is very abstract, which makes it difficult to properly interpret its parts and to appropriately apply it to specific problems.
- The model is helpful for common understanding, but does not guide a developer in identifying the methods that should be used. Thus, the model does not help in developing an architecture for a real system.

Even though we have chosen the JDL model in this thesis, it is a good idea to study different existing models to derive general principles for developing and evaluating data fusion processes. Some other data fusion models are the waterfall model [31, 32], the blackboard model [33, 34, 35, 36, 37], the Dasarathy model [38], the Omnibus process model [39], and the Data Fusion Systems Reference Model [40]. We conducted a review of these models specifically for aggression detection. The results can be found in the literature survey [41].

3.3.4 Design of the aggression detection system

Given the requirements, the main function of the aggression detection system boils down to a classification algorithm that, given aural and visual input of a train compartment, can distinguish between an aggressive and a non-aggressive compartment. Figure 3.3 shows the various stages followed for the design of such a classification system. Some of the fundamental questions that need to be answered before classifier design are: which features to use and how are they generated. As the influence of these choices is only visible during the system evaluation stage, one may go back to redesign earlier stages in order to improve overall performance. Therefore, all the stages in the process are interrelated (hence the feedback arrows in Figure 3.3).

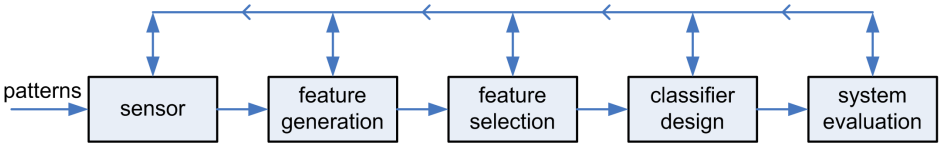


Figure 3.3: The basic stages involved in the design of a classification system (from Theodoridis and Koutroumbas [42]).

Because of the broad definition of aggression and the many objective and subjective factors that contribute to or effect the emergence and perception of aggression, it is unlikely that all the factors can be cramped into one single algorithm. Instead, several classification subsystems (all following the same structure as in Figure 3.3) are necessary. Each subsystem performs a detection or classification of a separate contributing factor while all the detected contributing factors combine into the final classification system to yield the final aggression classification.

Concluding from the above discussion, the final solution has the following properties:

Independent detection units. The proposed system will include heterogeneous and distributed sensing sources (multiple cameras and microphones) that function in a dynamic environment. A modular detection unit per compartment will assess the level of aggression within that compartment. Each compartment has a separate autonomous unit. Communication between units from neighboring compartments will take place to inform them of inter-compartment activities (e.g. fires, conductor checking tickets, aggressive person moving to another compartment).

Multiple algorithms. A detection unit consists of several classification algorithms that run in parallel. Each algorithm is focused on a different aspect of the aggression spectrum: one algorithm might perform face recognition, another gesture recognition or speech recognition etc.

Use off-the-shelf algorithms. For many classification tasks, algorithms already exist. Given the time constraints of the project it is more efficient to use these algorithms instead of inventing and implementing new ones. Algorithms for each classification task in the system will be presented and discussed in Chapter 5.

Facilitating framework. As many different algorithms will have to work together to yield the final result, an overall framework is needed to facilitate this. The framework provides a collection of services and capabilities that all algorithms have access to. Chapter 6 presents the framework in more detail.

The following assumptions about the system have been made:

Fixed sensors. Even though the cameras in the train compartment have zooming, panning and tilting capabilities, their settings are assumed to be fixed for simplicity. Also the positions of the microphones are assumed to be fixed.

Central computer. No processing will be done in the sensors, all sensors will provide their data directly to a computer located at a safe place inside the compartment where the processing will take place.

Wi-fi connection to control center. Each computer has a Wi-fi connection to a control center where a human operator is available.

3.4 Summary and conclusions

The main argument of the decision to use audio and video data is to imitate the successful example set by the human. This also means that modalities adopted by other

researches, such as infrared- and vibrometers, fall out of the scope of the project described in this thesis. Despite this, many challenges still remain. The confined and unpredictable world inside the train compartment poses challenges usually ignored or deemed irrelevant in laboratory conditions.

The multi-modality of the system requires a data fusion model that describes where and when data is fused. The data fusion model highlights the component functions which the system has but makes no statement regarding their software implementation or physical instantiation. Many data fusion models exist, each with its strong points and its weaknesses. Despite the shortcomings of the JDL model, it is the best option in that it fulfills the most important two requirements for models namely (1) a data fusion model should create a common reference framework and terminology and (2) a data fusion model should admit a process of gradual, piece-wise refinement which eventually leads to the deployable system.

The task of the aggression detection system is to process the audio and video input from the sensors in the train compartment and to distinguish between an aggressive and a non-aggressive compartment. Internally each system will consist of many off-the-shelf algorithms running simultaneously on a different aspect of the aggression spectrum. Each train compartment will independently run such a system on a computer inside the compartment. All sensors in the compartment will be directly connected to this computer and will provide data directly to it. The amount of communication of the computer to computers in neighboring compartments will be limited to messages to inform them of inter-compartment activities e.g. upcoming trouble. Finally, each compartment will have a wi-fi connection to a central computer in the Netherlands, where the information of all the detection systems is gathered and presented to human operators.

Chapter 4

Domain analysis

Since humans play the main role in aggression, human emotions and the complex interplay of human behavior towards each other and towards objects in their environment all contribute to the final aggression assessment. As such, aggression detection in a train compartment is a typical example of a domain in which complex reasoning is required. Reasoning itself requires a clear understanding of exactly what knowledge is available (in the knowledge base) and what this knowledge actually means (precision of meaning). Given that these requirements are fulfilled, reasoning consists of a process that tries to figure out what must be true, given what is known, by combining available knowledge according to a reasoning model.

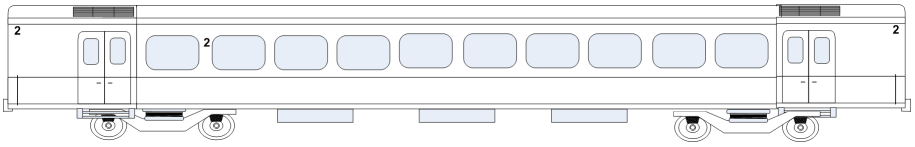
This chapter starts with an analysis of the physical train compartment in which an overview of the entities in the train compartment, that are relevant to aggression, are presented. This is followed by an elaboration about the activities that take place (aggressive as well as non aggressive) in the compartment. The domain analysis is compiled into a formal domain model which contains the concepts of the domain. Finally a brief overview is given of reasoning models that can reason with the domain model.

4.1 An analysis of the train compartment

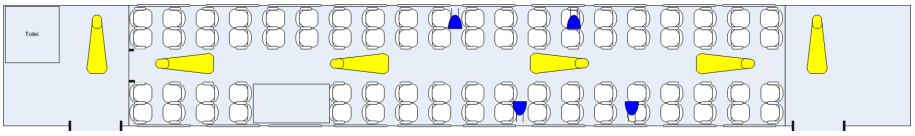
The goal of this analysis is to assemble appropriate information that will define the terms used formally to describe things in the train compartment. These definitions must be collected so that they can be expressed in the language selected for the aggression ontology. The aggression ontology consists of the objects active in the train compartment and presents the formal definition of these objects. The ontology will be used later in reasoning about aggression (Chapter 7 to Chapter 9). But it will also be used as the language in which experiment data (Chapter 10) will be annotated, therefore the analysis is done in more detail than is strictly necessary for reasoning.

4.1.1 Overview

A train typically consists of two machine rooms (or locomotives), one at the front and one at the back of the train. Between the two machine rooms, there are several connected train compartments. As the machine rooms are closed to the public, we will ignore them and focus on a single train compartment. The assumption is that once aggression detection in a single compartment is possible, it is possible to extend it to an entire train. More specifically, we will focus on a Benelux train compartment type IC600 (Figure 4.1).



(a)



(b)

Figure 4.1: Side view (a) and top view of the interior (b) of a Benelux train compartment type IC600. The positions of the surveillance cameras are indicated as cones in figure (b).

A train compartment consists of three sections: one main section and two side sections as illustrated in Figure 4.1 (Figure 4.2 presents a schematic view). The side sections have a sliding door to the outside world. These doors are the only means by which non-train-native objects (people, etc.) can enter and exit the compartment. The side section also contains a toilet. The main section is the largest of the three sections. It contains the seats and a door on both ends that connect the main section to the two side sections.

4.1.2 Compartment characteristics

Focusing on the main section (Figure 4.3), there are double glass windows on both sides of the wall in the middle section. Each window is positioned about half a meter above the compartment floor. Each window is also divided in a bottom and a top part. The bottom part is closed while the top part (which is about 30 cm high) contains sliding windows (that can be opened by passengers). In addition the train compartment contains two rows of seats. Each seat has sitting space for two adult passengers. Passengers sometimes have luggage. They can keep their luggage on them, put it next to them on the seat (if the compartment is not crowded) or temporarily store it in the overhead bins. When a passenger leaves he takes his entire

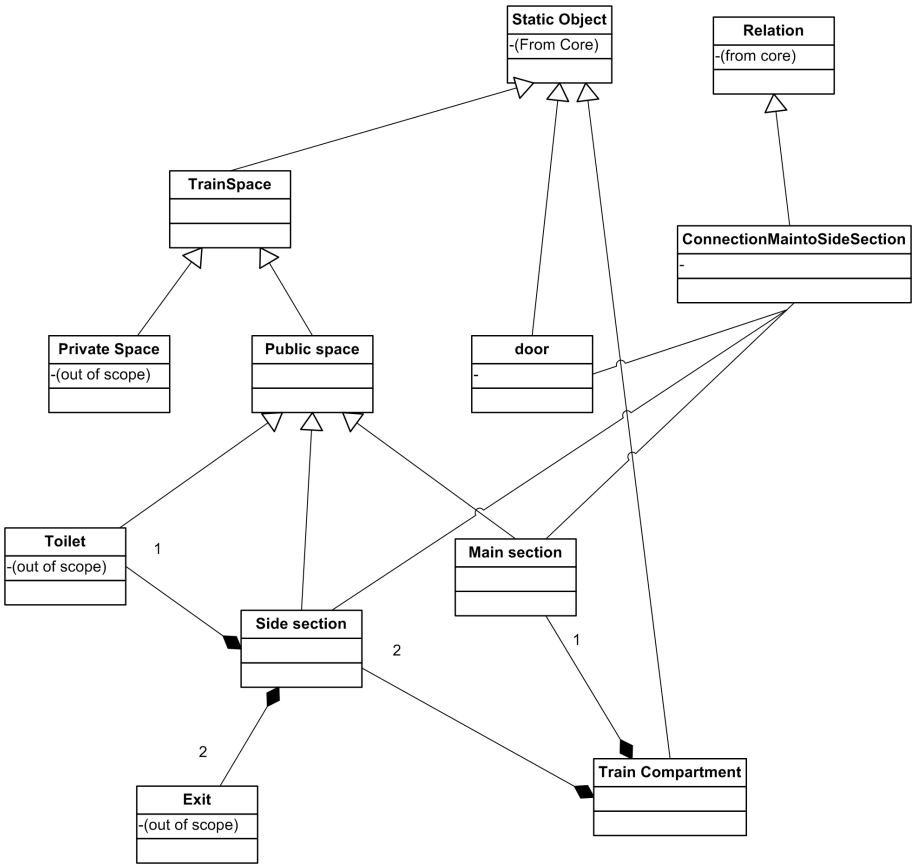


Figure 4.2: Overview of the main spaces in the compartment and their relationships.

luggage with him. Between the two rows of seats there is a way path. If it is very crowded in the compartment and all seats are taken, people can stand in the corridor. Smoking in the train is not allowed.

4.1.3 Objects in the train compartment

In this section, we will further refine the objects inside the train compartment (although the exterior of a train is also subject to acts of aggression, e.g. graffiti we will not focus on that). The following enumeration identifies objects in a train compartment. For every object, we identify its intended use and also the use that is considered aggressive. All objects can be subject to vandalism and mutilation. We will therefore not mention this for each object explicitly.

Doors are found at the entrances/exits of a train compartment. Intended use: opening, walking through, closing. Aggressive use: keeping doors closed so other people cannot enter, blocking the door, keeping doors open for a longer time, opening or closing aggressively, closing the door when somebody is passing. Keeping doors open for other people is allowed.

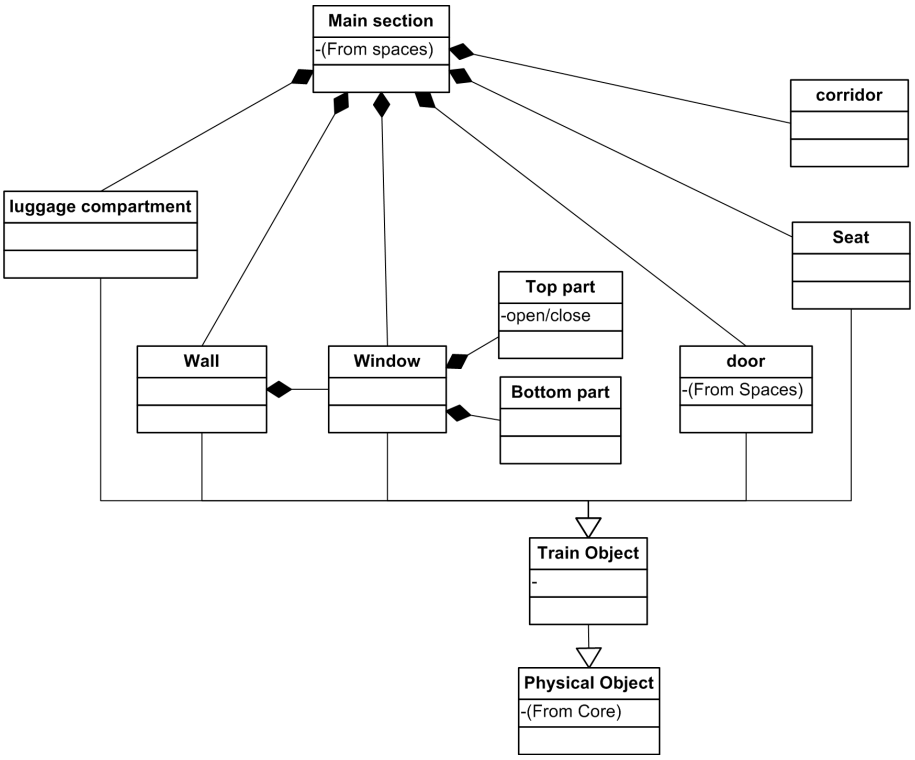


Figure 4.3: Overview of the main section of the train compartment.

Seats are found ordered on both sides of the compartment. Intended use: sitting upon. Aggressive use: drawing upon, putting feet on the bench, occupying more then one seat, putting objects on the bench when there are not many seats available. Exceptions: putting feet on the bench may be allowed when using a paper or so to cover the bench. Small children may use the bench in a way that would be considered aggressive when done by adults.

Trash bins are located along the compartment walls, usually a small bin with a lid. Intended use: disposing of small objects. Aggressive use: stuffing, setting on fire, turning upside down, pouring liquids in it.

Tables are found in compartments, on the wall or on the back of a bench. Intended use: to put objects on. Should be cleaned after use. Aggressive use: befoulment, drawing upon, mutilation, occupying the whole table when it is a shared table.

Lights are found on the ceiling throughout the train, and above seats. They can not be switched on or off by passengers. Intended use: turns on or off at a fixed time. Aggressive use: be on or off when it is not supposed to, turning on and off repeatedly.

Windows are found in compartments. Some windows can be opened. Intended use: looking through, opening when it's hot, closing. Aggressive use: throwing objects through open windows, opening window when it's cold.

Overhead bins are located above the seats, near the ceiling. They are used to store objects on. Intended use: store objects, taking objects off. May be held on to by passengers standing to avoid falling over. Aggressive use: hanging into the rack, occupying all space.

In general, for all native objects it is not allowed to vandalize them (e.g. breaking, writing on etc.). Apart from native objects, the train compartment also has foreign objects that can be brought in by travelers. The range of objects that fit in this category is very large, the most important one of which is of course the passenger itself.

Passengers can move around freely in the train compartment. They usually sit on a seat if one is available or remain standing if there is none. Passengers entering as a group tend to stay close together, otherwise close proximity of unrelated passengers can be considered intrusion of personal space. Proximity also differs depending on the number of people in the compartment. Because passengers are the most important entities in aggression, possible (aggressive) actions and interaction between passengers are discussed separately in Section 4.2.

Bags and suitcases should be kept on passengers' laps, next to him, or stored in the overhead bins. Storing bags on a seat occupies sitting space and may be considered an act of aggression when the compartment is very crowded. Intended use: holding while walking or standing in corridor, storing in overhead bin, keeping on lap, storing in corridor or on seat when the object is too large to fit or too heavy to put in the overhead bin. Aggressive use: keeping on seat in crowded train, storing in corridor, throwing, touching other passengers, leaving it behind.

Walking aids such as wheelchairs and crutches, by necessity, are kept close to the owner. This may hinder other passengers but this should not be considered a problem as the person is really dependent on these aids. Intended use: walking aid for the owner while walking through the train, storing close to owner when owner is seated. Aggressive use: taking away from owner.

Bicycles (folding or non-folding) should be stored in the side section in a location where they do not hinder passengers coming through. Intended use: storing as stated. Aggressive use: storing elsewhere, hindering passengers, throwing.

Foods and drinks can be brought in bags or separately. Foods and drinks can be put on the table, and containers should be disposed of after use, either by putting them in the trash bin or taking them outside the train. Intended use: consuming by owner, being given away/handed out by owner. Aggressive use: throwing, spilling or dropping and not cleaning up, leaving somewhere, not properly disposed of.

Papers, magazines or books can be put on the table, on seats or on the luggage rack. Although leaving papers and magazines in the train after leaving the train creates a mess, it can also be considered polite as it allows the papers to be read by

other passengers. Intended use: storing in location as stated, reading, laying on an empty seat to put feet upon, thus not dirtying the seat. Aggressive use: putting on fire, reading in a way hindering other passengers.

Electronic devices such as phones and wearable audio, e.g. mp3-players can be put on the table or on the listener's lap or in his pockets. Turning the volume up too loud irritates other passengers. Intended use: listening to music, storing on location as stated, making phone calls. Aggressive use: storing elsewhere, turning volume up too high, throwing, loud conversation

4.1.4 Train compartment states

During the day, the train compartment goes through several states and cycles. Each state exhibits characteristics that can be observed through a camera or a microphone. It is important to distinguish these states and know their characteristics, because it is necessary for an appropriate context sensitive interpretation of observations: aggressive behavior in one state may be considered normal behavior in the other.

If we consider the long term, we can distinguish the annual cycle with the coming and going of seasons or the weekly cycle with alternating weekends and weekdays. In this thesis we will focus on the daily cycle of an ordinary workday. In particular we will not consider special days such as weekends and holidays. Figure 4.4 shows that the occupation level is more or less the same on any weekday, so we do not distinguish them.

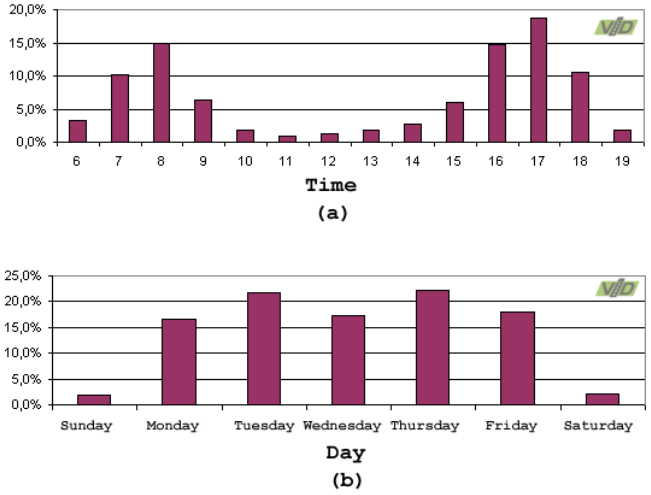


Figure 4.4: Distribution of traffic congestion during the day (this is correlated with the occupation in trains). The occupation is highest during rush hour (a), while (b) shows that there is little difference in occupation between weekdays (source: *http://nl.wikipedia.org/wiki/Spits_(verkeer)* (date: 21-01-2009)).

In the daily cycle, the compartment goes through a day and a night state. During the day state illumination of the compartment is provided by the internal lamps in the compartment as well as the sunlight through the window. At night, the illumination is dominated by the internal light sources. The the day state can be further subdivided into off-hours and two rush hours. The rush hours, between 7-9 in the morning and 4-6 in the afternoon, are characterized by an overall increase of occupation of the compartment (see Figure 4.4). For the sake of simplicity, we will not consider special events such as concerts and football matches that may cause a one-time increase in the occupation level of the compartment. The effects of these events are local (limited to at most a small number of trains) and can be considered abnormal situations which should cause heightened alertness anyway.

Within this daily cycle, the compartments goes through a functional cycle, in which the train moves between stations to let passengers in and out (Figure 4.5). The characteristics of each state are summarized below:

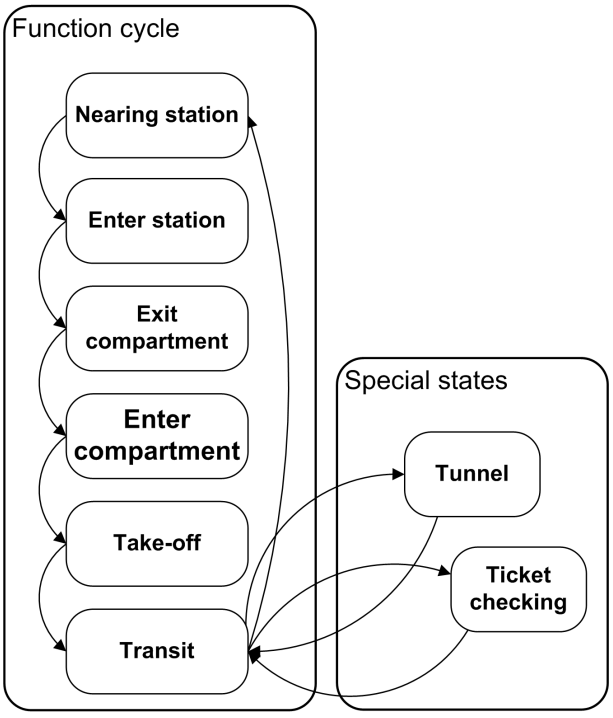


Figure 4.5: Functional cycle of the compartment.

Nearing station. This states starts when the announcer announces the name of the next station. Usually there is an increased movement, some people stand up and start to gather their personal belongings. The number of people in the compartment stays the same.

Enter station. The train enters the station and starts to brake. Usually the light/illumination in the compartment changes because the train station blocks sunlight coming from the window. Some passengers walk towards the exit. The number of people in the compartment stays the same.

Exit compartment. The train has stopped in the station and some passengers have reached their destination. They are leaving. The number of people in the compartment decreases.

Enter compartment. The train has stopped in the station and most people that want to get out have already left. New passengers now enter the train. The number of people in the compartment increases, and some (or all) empty seats become occupied by passengers. Luggage is stored properly and coats are hanged if preferred.

Take off. The train takes off from the station, the characteristic sound of the train gaining speed can be heard and the rocking motion of the train can be felt. The number of people in the train stays the same.

Transit. The compartment is in transit to the next station. The characteristic sound of the train is heard and stable. The rocking motion of the train can also be felt. The number of people in the train stays the same.

When the compartment is in transit state, two additional states can occur:

Ticket checking. A train conductor moves along the compartment and asks each passenger to show his ticket.

Tunnel. The train passes through a tunnel and illumination temporarily changes.

We have discussed the normal states of the compartment and the transitions. As mentioned earlier, any other state e.g. concerts, football matches, are considered abnormal situations and are cause for heightened alertness.

4.2 Passenger behavior

The train compartment is a place where passengers reside when traveling to their destination. Aggression is committed by people. On the same token, people may be the subject of aggression as well. Often people that commit aggression show potential aggressive behavior before they actually commit deeds of aggression. For an automated aggression detection system it is the most important task to detect people. The relationship of a person with other entities in the train compartment is described in Figure 4.6.

4.2.1 Groups

It is important to distinguish if a person belongs to a group for several reasons: detection of theft and detection of possible future aggressor. If a person belongs to a group

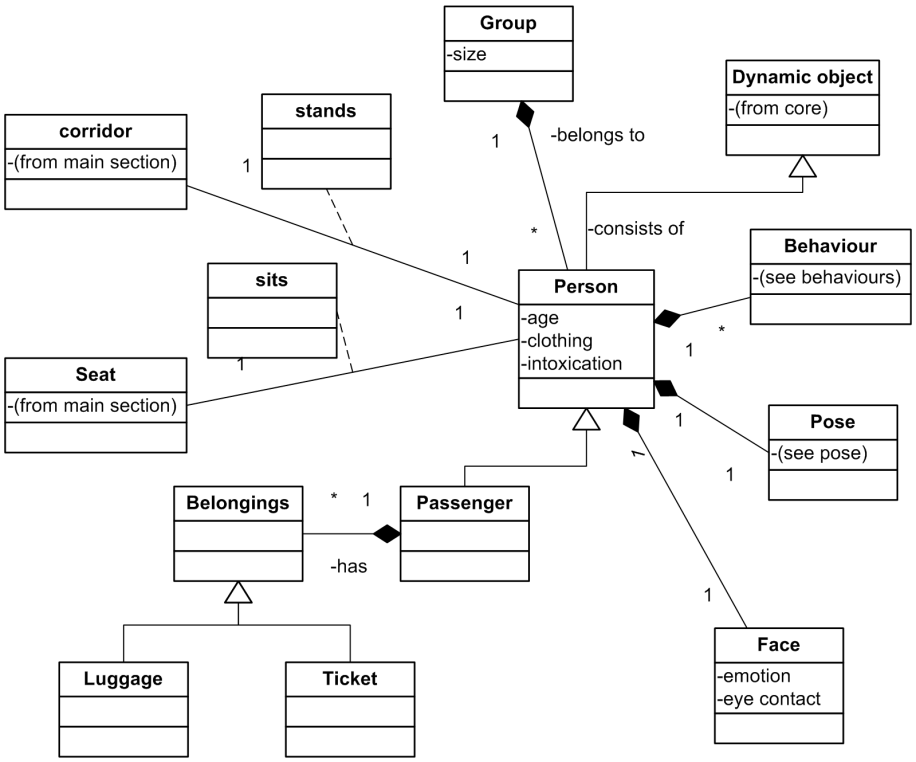


Figure 4.6: The relationship of a person with other entities in the train compartment.

of people in which one is committing aggression, then the person himself is likely to become a committer of aggression as well. The level of suspicion for this person should be heightened along with an increase in overall potential of aggression. Detection of a group is hard and needs constant attention. A group can be identified when it enters the train. The people in the group will be more closely together, have eye contact and may be talking. In rush hour this may not be effective because of the large amount of people entering a train. There may be no group of seats available for the group as a whole, so the people of the group will be spread over one or even more compartments. They can be identified as a group by showing more frequent eye contact and conversation than strangers. The nature of the conversation of strangers will also be different than that of acquaintances.

4.2.2 Individuals

Focusing on the level of a single individual, the appearance of a person is the first cue that may raise awareness to their aggressive potential. Sports supporters are an example. When the camera observes a person of a group of persons dressed as football supporters, it should raise awareness for this group because the group is more likely to commit aggressive behavior than a group of businessmen or an elderly couple. The following appearance based features may be significant:

Age. Aggression is most typically committed by people between approximately twelve and forty years old.

Clothes. The clothing style of a person may give an indication of his aggressive potential. A group of sports supporters all wearing their club's clothes generate a raised awareness for aggressive behavior. Clothing styles of typical aggressive groups can be obtained from heuristics of experienced railway personnel and railway police.

Alcohol/drugs. The camera must be able to detect drunk or otherwise intoxicated people. Using drugs on a train is not allowed as it is forbidden by law and therefore always considered aggressive behavior. Alcohol is permitted however and the drinking, which may lead to aggression, raises awareness.

Belongings. Passenger's belongings should be associated to the person. This should be done in order to detect theft and abandoned luggage. In addition, aggressive and forbidden belongings such as weapons and alcohol should raise awareness.

More important than the appearance is the behavior of a person as indicator of his aggression potential. Behavior is something very difficult to detect as it is a label that we give a person's actions and interactions over time. The next section is entirely focused on behavior.

4.2.3 Actions and interactions

A behavior is a label given to a sequence of actions and interactions of a person over time. A person exhibiting the actions that match a behavior is said to show that behavior. Some behaviors can be flexible in that it does not matter in which order the actions take place. Others have a strictly fixed global pattern with minor deviations allowed. The behaviors with a fixed global pattern are especially useful for aggression detection as it allows for us to make predictions and calculate expectations of what may happen given a preliminary list of observations. Table 4.1 shows a list of normal actions and behaviors and indicates in which state of the compartment they are likely to occur.

The purpose of specifying the normal behaviors (and the states of the train) is that normal behavior in one state can be considered abnormal if the compartment is not in the correct state. Apart from these kinds of abnormal behavior there are a number of behavior patterns that can be considered abnormal at all times. Now we will describe this behavior and discuss how to detect it in terms of the objects discussed in Section 4.1.3. For each type of abnormal behavior we discuss ways in which the behavior could be automatically detected. In general, whenever actions occur, movement will occur (which can be observed as local changes in pixel/energy values in consecutive video frames) or sound is produced (which can be observed as changes

Table 4.1: Normal actions and behaviors and indication in which state of the compartment they are likely to occur (A_r: rush hour transit, A_l: low hour transit, A_n: normal transit, B: nearing station, C: stopped, D: exit station, +: highly likely, *o*: likely, −: unlikely).

Behavior	A_r	A_l	A_n	B	C	D
enter	-	-	-	-	+	-
exit	-	-	-	o	+	-
sit	+	+	+	o	o	+
stand	+	-	o	o	o	o
talk	o	o	o	o	o	o
eat	o	o	o	o	o	o
ticket check	-	o	o	o	-	-

in volume). For a correct judgment of the action it is important to pinpoint the position/place of the source of the action.

Shouting. Shouting is closely related to aggression. Physical violence is often preceded and/or accompanied by shouting. Shouting in itself is very irritating to other passengers and as such should be considered aggressive behavior. However, sometimes passengers shout but are not behaving aggressively. Optionally, whenever shouting is taking place, the system could detect if any swearing or cursing is present. If that is the case, the shouting is aggressive behavior.

Theft. Theft is an obvious deed of aggression but may not be easy to detect when people travel in groups. Consider a passenger traveling alone with a suitcase. When somebody else takes the suitcase, this is an obvious theft. However, if the owner entered the train with a fellow passenger and brings in two bags. When leaving, the other passenger may help him carry one bag. This should be distinguished from stealing, where the bag is taken by another passenger. Whenever an object is taken by a person not the owner, the camera should check the reaction of the owner. The owner should approve of the taking of the object. The approval can be implicit, with the person noticing the other taking his bag and not responding, or explicitly, where the owner explicitly, possibly verbally, allows the other to take the suitcase. The camera can check this by looking at the body language of the owner and his facial expressions. The approval may be implicit, because the owner and his fellow passenger have agreed upon the taking of the bag. In that case, the owner may show no reaction. This can hardly be distinguished from stealing the object as the owner might not notice the bag is being taken. Therefore it should be detected whether the owner notices the object being taken away or not. If he does not, it's a possible act of aggression.

Verbal aggression and offensive gestures. Swearing and cursing are forms of aggression, especially when directed to someone. Many people swear when something bad happens to them, or add curses when telling a story. Other people

can find this offensive behavior and consider it aggressive. In order to recognize swearing and cursing, the system must know which words are terms of abuse. Additionally, it should recognize when these words are directed to somebody in particular, as that is a clear indication of aggression. Certain signs are aggressive behavior, e.g. obscene gestures as showing the middle finger, mimicking a gun or making a gesture of slitting a throat. Signs like this must be detected by the system as well. When these gestures are directed to someone in particular, this is a clear indication of aggression. Aggressive or angry facial expressions are also a clear indication of aggression. When observing a compartment the system will detect what is said and who makes which gestures and will therefore detect these forms of aggression.

Intimidation. Intimidation is a subtle form of aggression with little or no physical contact. The aggressor threatens someone with words, may stand very close to him and touching him. He may be shouting, but can also be talking softly in a threatening way or staring. It can be identified by examining the body language and speech of the aggressor, and the reactions of the addressed. Threatening speech and aggressive body language may be intimidating. The addressed can be frightened or show aggressive behavior in return. He may also try to ignore the aggressor in which case the reactions of the intimidated cannot be used for the diagnosis. The system will then still detect possible aggression is going on and heighten its level of suspicion for this situation. The ignoring will mostly not last for long: either the aggressor gets more aggressive or he will back off, or the victim will show some reaction after a while.

Physical aggression. Some forms of physical aggression are easy to detect. Fighting is an example that can not be mistaken. Near-physical aggression and touching, which is more intimidation, is sometimes also aggressive behavior and should therefore be detected. Physical aggression will often be accompanied with verbal aggression. Physical aggression can be detected by looking for unwanted physical contact. Unwanted contact can be determined by analyzing the reactions of the victim. Both the aggressor and the victim will possibly be shouting or talking loudly and aggressively. Other people in the train will react to the situation as well, either being frightened or angry. These people may show aggressive behavior themselves in reaction to the aggression taking place. When no other people are present, the camera can only use the input of the aggressor and victim.

Vandalism. We call it vandalism when objects are being demolished. This can be in many forms: breaking, tearing, shattering, anything that will irreversibly damage an object. Some objects are demolished on purpose and this mutilation should not be considered an act of aggression. An example is peeling an orange. In general, food is subject to mutilation that is not considered aggression. The reactions and body language from other passengers indicate whenever mutilation is going on. When no other passengers are around or nobody

is looking, the camera has to decide for itself. The sound of breaking, tearing, etc. will heighten the system's suspicion of the aggression taking place.

Sexual harassment. Sexual harassment occurs when someone is unwantedly touched, most notably in private areas of the body. This differs from physical aggression in that the touching is not meant to hurt. In general touching somebody that is not in the group is an indication of possible sexual harassment. The system should detect this and check the victim's facial expressions. When these show fear, anger or rejection, sexual harassment is taking place. When the touching is happening between two people in the same group, the same criteria apply, but the touching itself is less an indication of possible sexual harassment then when the people are not in the same group.

Abandoned luggage. Since the recent bombings in the London underground and in trains in Madrid, abandoned luggage in the worst case can be thought of as a form of aggression. In addition, luggage forgotten by accident can cause great distress among other worried passengers. It is therefore important to be able to detect luggage that is (accidentally) left behind.

Other deeds of aggression. Certain behavior is forbidden in a train and can therefore always be considered an act of aggression. An example of this is smoking. Smoking in the train should always be considered an act of aggression. Other behavior may be forbidden more generally by law. For all objects we identified in the train in Section 4.1.3, we also discussed normal usage and misuse that constitutes to aggressive behavior.

Finally there are some unusual situations that can occur and need increased attention. These unusual situations such as checking for tickets, begging in the train, medical emergency, people entering the compartment when the train is moving, standing in the aisle if the compartment is nearly empty or standing up as the train is moving.

4.2.4 Emotion

In case of aggression, specifically in affective aggression, often emotions are involved. Moreover, the aggressor as well as the victim may be influenced by his current emotional state. For example, bystanders tend to behave differently (less emotionally) than people who feel threatened or are themselves the victim of the incident they are reporting about [43]. In addition, the urgency and impact of the situation may be conveyed in the emotional qualities of the people involved. A person who has just witnessed a major bomb explosion has different emotions than a person experiencing a small unexpected explosion as loud as a fire cracker. Therefore, an important aspect in the training of operators is to sense emotional qualities of the reports, since the urgency and impact of a observation is often conveyed by the emotions of the reporter. Thus, the emotion, combined with the contents of the observation, may

give an indication of the size of the crisis and the actions to take. In order to capture this aspect in a automatic surveillance system, detecting emotions becomes an important issue.

Emotions can be conveyed in different modalities such as text, speech, facial expressions and gestures. Researchers have tried to extract emotions from other modalities such as facial expressions [44, 23, 45, 46], body poses, postures and gestures [47, 48, 49], and speech and intonation [50, 51, 52, 53, 22, 54]. In recent years, several works on human emotion recognition have been published. All current work assumes some simplifications of the nature of emotions, for example by working on artificial emotions performed by actors, or by restricting the emotion discrimination to few classes only [55].

The subject of emotions in speech has been widely studied by psychologists and psycho-linguists to determine which acoustic features (e.g. the relative loudness and pitch of each sound change, intensity, speaking rate) encode the emotional state of a speaker. Usually emotion is represented in a three-dimensional emotion space. Every emotion to be recognized consists of three components: Valence, Activation, and Dominance. Valence describes how positive or negative an emotion is felt e.g. negative emotion class might include “anger”, “frustration”, and “boredom”, and non-negative class would have “neutral”, “happiness” and other positive emotions. The second dimension, Activation expresses the degree of excitation. Dominance refers to how strong or weak the speaker appears. A commonly used emotion categorization is Ekman’s [44] six basic emotions for the classification of facial expressions: happiness, surprise, fear, sadness, disgust, and anger. For the classification of emotions in speech (especially in the case of crisis situations), the five emotional states happiness, sadness, anger, fear and neutral are gaining popularity [56, 57]. Studies show that the most dominant emotion associated with aggression is anger.

4.3 Aggression model

The correct interpretation of the behavior from observations and the subsequent assessment of the aggressive potential of a situation needs some kind of reasoning to put the observations and domain knowledge in the correct context. Reasoning means, at least in part, combining available knowledge to draw new conclusions. This requires a clear understanding of exactly what the knowledge being reasoned with actually means. The fundamental goal of knowledge representation is to represent knowledge in a manner to achieve this precision of meaning. In this section, we identify the common concepts within our problem domain and precisely define these concepts. Then we formulate rules about those common concepts and discuss techniques to exploit these rules.

4.3.1 Knowledge representation language

Knowledge representation needs a knowledge representation language that allows us to express all relevant knowledge in our problem domain in a clear and unambiguous way. Candidate representation languages range from natural languages (such as English or Dutch) to logic-based languages to object-oriented programming languages. One issue in the choice of representations is expressiveness.

Natural language obviously is very expressive. But this leads to the problem that the same meaning can be expressed in many different ways. This gives problems when we want to reason with the represented knowledge. Another issue in the choice of representation language is ambiguity. Natural language is highly ambiguous as the same expression can have different meanings.

On the other hand, programming languages normally used in computer science (such as object oriented languages) have the property of being unambiguous. However, they lack the expressiveness of natural language, since these languages are designed for fact storage rather than knowledge storage. For example, the information stored in objects in OO languages is only accessible through the methods that have been implemented separately. Therefore, usage of the information is restricted to what the programmer or designer has intended and foreseen. As a result, only part of what that information can express is accessible and much of the implicit meaning of the object is lost.

In the field of artificial intelligence, special languages for knowledge representation have been developed e.g. CycL, IKL, KIF, Loom, OWL and KM. These languages usually offer the encoding of knowledge in the form of ontologies. An ontology consists of a set of concepts, a set of relationships and a set of axioms that describe a domain of interest, complemented with two partial orders defining the concept hierarchy and relation hierarchy respectively. With an ontology we can precisely define the concepts we use. This allows us to relate the appropriate rules with their respective concepts. First, we identify the common concepts, for example, the relation “x caused y”. Then, we use logical relationships to formulate rules about those common concepts. For example, “if x caused y, then x temporally precedes y”. An ontology representation gives us enough expressiveness, and facilitates the reasoning as well.

In semantic annotation, the ontology is used to describe the meaning of objects, relationships and hierarchy structure of given resources. Based on the ontology we can add machine understandable annotated information to sensor recordings according to the knowledge described by the ontology. The knowledge representation language of choice is the web ontology language OWL DL[1]

[1]OWL DL, where DL stands for “Description Logic”, was designed to provide a language subset that has desirable computational properties for reasoning systems (http://www.w3.org/TR/owl-ref).

4.3.2 The core ontology

The ontology will be used by the detection system for semantic interpretation of complex situations. It also provides a vocabulary for linking a wide range of aggression related information. For example, the face detection algorithm may publish some information about having detected a face at location x, y, z. Because the semantics of a face and of a location are defined in the ontology, other algorithms can proceed with it. The aggression ontology provides a vocabulary to express information like: *"there are four people in the compartment, three are sitting, one is standing"*.

The ontology is divided in three levels of expressiveness (corresponding to the three lowest levels of the JDL fusion model):

Level 0 aims at providing a vocabulary for simple low level description of features of the data (e.g. blobs, sound sources). This level is not so relevant to (human) reasoning but provides a way to annotate the ground truths of the data to compare the performance of low level algorithms.

Level 1 or static context aims at providing a vocabulary for expressing static objects and actions e.g. person x is sitting.

Level 2 or dynamic context aims at providing a vocabulary for complex situation and event decomposition, to express, for example, what happened during a particular scenario, what are sequences of actions and what are the motives of a certain aggressive act.

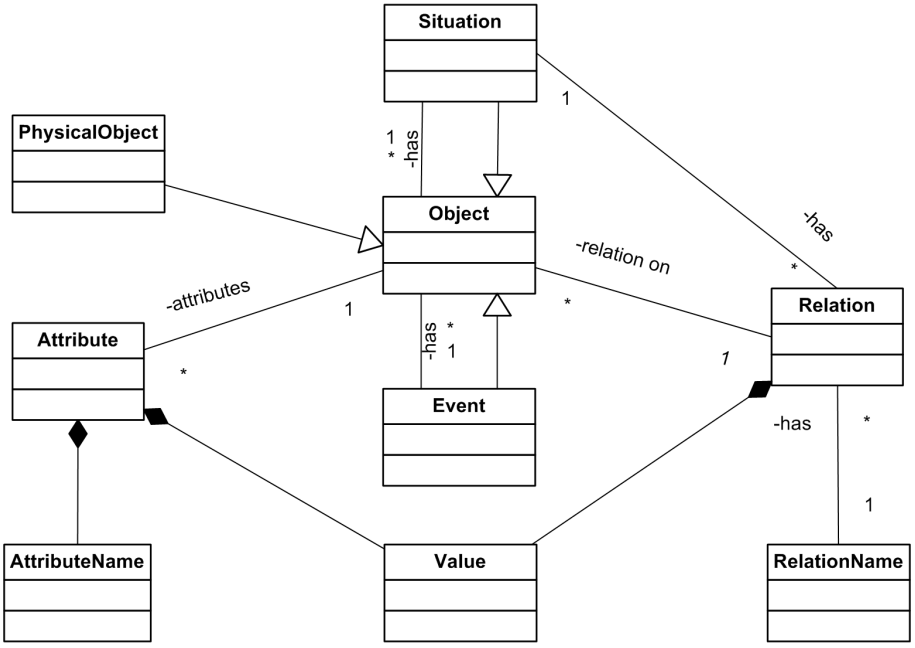


Figure 4.7: The core ontology captures the basic dynamic concepts of situations and their dynamics over time and space.

In this thesis we will focus on the last two levels. We start the conceptualization first with a definition of the core ontology shown in Figure 4.7. In brief, the figure shows that in this core ontology, objects have attributes (e.g. visible, audible), that they can be physical, have relations with other objects (object 1 belongs in a group with object 2) and that it can play a role in a situation (when there is a fight, there should be fighting relations between objects). `Relation` define the relations between ordered sets of objects e.g. inRange(x, y). `Value` has a mandatory part and an optional part. The mandatory part specifies the actual value and the optional part describes the certainty. The core ontology captures the relevant basic properties of abstract concepts and their relationships.

The core ontology is extended with two domain specific extensions: (a) the sensing environment and all the objects of relevance in it (static context) and (b) a specification of the typical (as well as unusual) scenarios, represented by a specification of the essential actors and a specification of the flow of events in time from the start of the scenario to the end including the alternative endings (dynamic context). Each of the extensions is discussed in more detail below.

4.3.2.1 The static context

The static context contains all the objects that are active in the environment and presents the formal definition of these objects in the ontology. Figure 4.8 shows a global view of the static context of a train compartment. The context describes all of the objects that are in a train compartment (`TrainObjects` class). To contain the size of the ontology, only objects that play a role in what needs to be observed should be included. Seats and tables are included as `TrainObjects` because they could be damaged by graffiti, fire, etc. For each native train object identified in Section 4.1.3 and non-native object identified in Section 4.1.3 a concrete concept of `TrainObjects` is created. The possible uses of each object can be represented as a `Relation` with a `HumanObject` in the ontology. The qualification 'intended use' or 'aggressive use' in the listing can be represented as attributes of `Relation`. This attribute does not matter for the annotation vocabulary but may be essential in automatic reasoning.

An important entity in the dynamic context is the `HumanObject`. The `HumanObject` is capable of manipulating (generating events in) the dynamic context (by actions and behaviors). For example, aggressive behavior involves a human aggressor and one or more human or non-human victims. A chain of situations that occurs by a combination of an aggressor and possibly one or more `Object` will be a scenario. An action or event caused by a `HumanObject` and possibly using other `HumanObjects` could result in another situation being triggered and possibly switching the scenario to maybe an even more dangerous one.

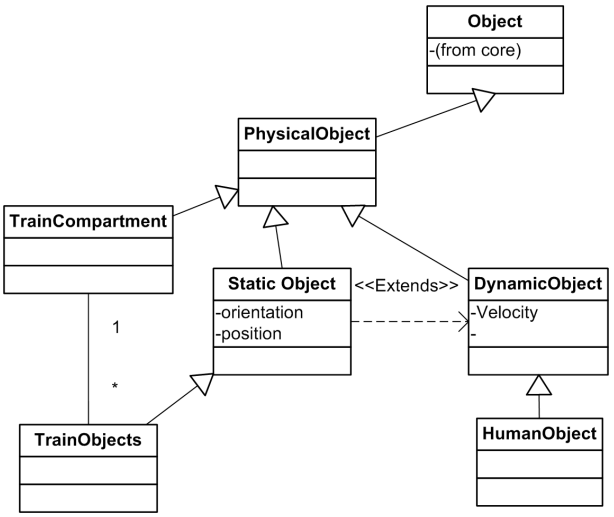


Figure 4.8: Overview of the static context.

4.3.2.2 Dynamic context

The dynamic context describes how objects can interact within the environment. The central classes are `Situation` and `Event` (Figure 4.9). A `Situation` includes some objects and relationships among them (in the form of the `Relation` elements shown in Figure 4.7). As `Situation` is a subclass of `Object`, every instance of `Situation` can have attributes like any other `Object`. The same holds for `Event`. `Events` are objects that contain information about events in the real world observed at a specific time that affects the value of `Attributes` and `Relations`. `Scenarios` describe situations that can occur in the environment, more specifically it summarizes the events and actions that are usually observed and their order of occurrence when such a situation takes place. A `Scenario` consists of a sequence of `Situations` and a `timeline` on which `Events` can be placed. An `Event` that leads to a transition of one situation into another is represented by a `SituationSwitch`.

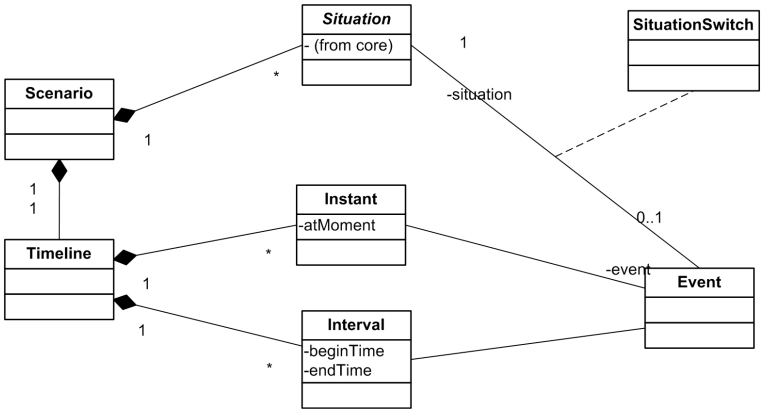


Figure 4.9: Overview of the scenario sub ontology.

The ontology defines the `Timeline` concept, to address temporal information. A number of `Intervals` and `instants` can be defined by which `Events` can be associated to the `Timeline`.

An `Event` concept represents events that may happen in the train (e.g. passenger 1 is fighting with passenger 2). It represents a way by which we can classify certain useful and relevant patterns of change. An event has a location, a time and optional relations to actors, factors and products, as depicted in Figure 4.10. `Actors` are the objects that caused the event (e.g. passenger sits on seat), `Products` are the possible appearing or disappearing objects as a result of the event (e.g. train stops in station results in passengers exiting), while `factors` are objects that also play a role in the event but do not belong in the aforementioned categories. In addition, an event can consist of a number of subevents. The `Event` concept forms the connection between the objects in the static context and situations and scenarios in the dynamic context.

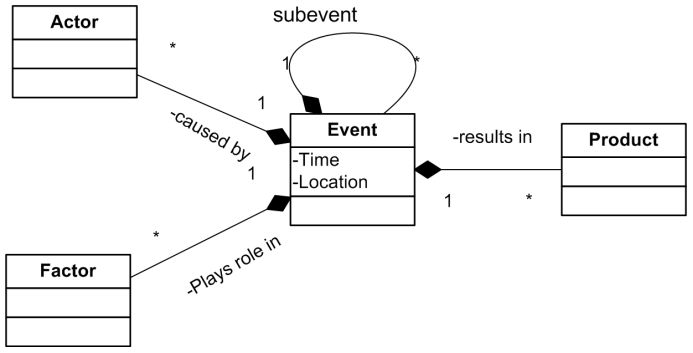


Figure 4.10: Overview of the event sub ontology.

The RDF/N3[2] code in Listing 4.1 deals with the event that the first seat on the second row is occupied by a person who is drawing on it.

4.3.3 Aggression perception model

From interviews with experts, it became clear that human aggression detection reasons in terms of high level concepts (e.g. person x is walking too fast). Even though the input data may be a number of pixels on a screen, humans are largely unaware of the process that makes a bunch of pixels a person or a bag etc.

To imitate the human system, our aggression detection approach is essentially divided into two models: (1) the observation model which describes how low level features from observations are combined into high level concepts (as described in the static context in Section 4.3.2.1) and (2) the reasoning model in which high level concepts are reasoned with in order to infer the presence of aggression. In the observation model, classic feature extraction and classification techniques will be used to

[2]Resource Description Framework (RDF) is a specification produced by the World Wide Web Consortium (W3C) to describe various kinds of things and their inter-relationships using a simple 3-part-sentence framework.

```
a event:Event;
    event:location 2, 1;
    event:time "2007-10-15T12:00:00"^^xsd:dateTime;
    event:duration "PT1H"^^xsd:duration;
    event:actor [
      a passenger: HumanObject;
      passenger:id 007;
      passenger:location 2, 1;
      passenger:visible true;
      passenger:audible: true;
    ]
    event:actor [
        a seat: TrainObject;
        seat:location 2, 1;
        seat:visible true;
        seat:audible: false;
        seat:sit_relation [
          a sitter:HumanObject;
          sitter:id 007;
        ];
        seat:drawn_upon_relation [
          a sitter:HumanObject;
          sitter:id 007;
        ];
    ]
```

Listing 4.1: RDF code of an event specifying that the first seat on the second row is occupied by a person who is drawing on it

transform audio and video signals into semantic concepts. In the reasoning model knowledge of the human expert is used to infer the presence of aggression.

4.3.3.1 Human perception

This thesis concentrates on imitation of human experts in particular types of recognition of concepts and phenomena in sensor data, and as such has many general aims in common with perception. An appropriate step therefore is to examine the biomechanical and physiological processes leading to human perception.

Human perception can be seen as a concatenation of mathematical mappings [58]. A specification of concrete mathematical mappings is still subject to investigation. Each sense organ only detects certain components of a vector containing all stimuli affecting the human ($s_i(t)$). For example, in visual perception, the retina is responsible for the reception of visual stimuli. It picks up photons of light and simply act as a messenger to a particular stimulus that is seen.

Most of the sensors can be modeled as detectors for when specific stimuli go above a certain threshold or are in a certain frequency band. Information is directly transmitted to both the spinal cord and the brain. The complex processes of convergence, divergence, lateral inhibition and excitation within the neural net of the sensory pathways accomplish a pre-processing of neural information, e.g. contrast enhancement and feature extraction [59]. According to the above description each sensory system accomplishes a non-linear dynamic mapping $s_i(t) \rightarrow a_i(t)$ of the input vector $s_i(t)$ to an output vector $a_i(t)$ of neural activity arriving at the cortex, see Figure 4.11.

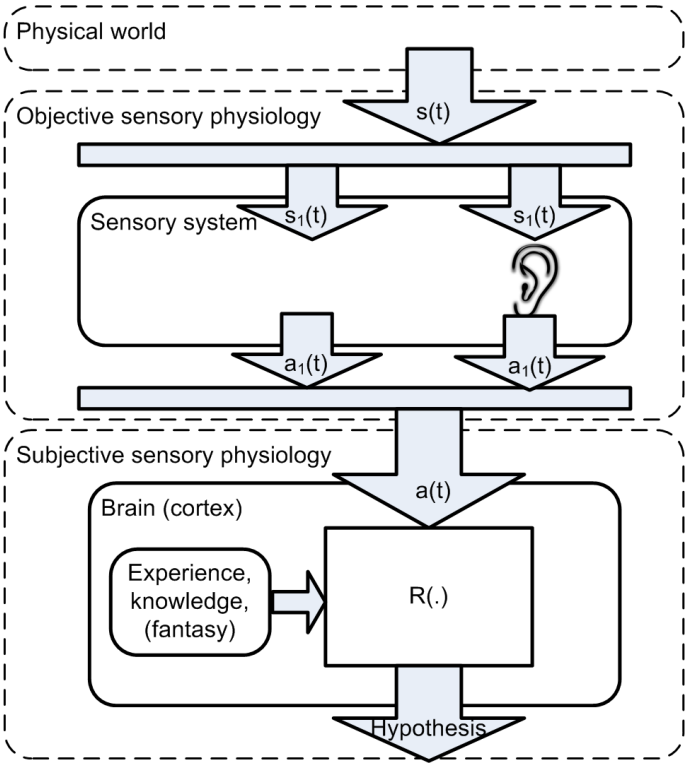


Figure 4.11: Overview of the model for human perception [58].

The brain is the place where external stimuli are actually perceived. In the brain, percepts are created: representations in the mind of categories that can change and shift over time. This process can be viewed as part of an inferential reasoning process ($R(.)$): from seeing a face the process infers a person.

As for the question how inference leads to the recognition of aggressive situations, the Recognition Primed Decision making model (RPD) developed by Klein [60] suggests an answer. The RPD model describes how people make quick, effective decisions when faced with complex situations. According to this model, experienced people will generally be able to come up with quicker recognitions and decision because the situation may match a prototypical situation they have encountered before. Novices, lacking this experience, must cycle through different possibilities, and have the tendency of use trial and error through their imagination. The RPD model is used widely to train officers to make decisions in organizations such as the navy, fire fighters, police, and other emergency services. The rule-based reasoning model described below in this section is based on the notion of prototypical situations of the RPD model.

Several researchers (e.g [61] and [59]) have published approaches for perceptual organization of knowledge (e.g. to store prototypical situations). A particularly interesting approach is the Schema theory [62]. In this theory, schemas are cognitive structures that link declarative (percepts) and procedural (e.g. experience) knowledge together in patterns that facilitate comprehension of the situation (hypothesis) within

a context. The resulting hypothesis itself thus depends on the individual knowledge, training, etc. The interviews we conducted with human experts support the schema theory (see Section 2.2.2). Section 7.1.1 discusses the Schema theory in more detail.

Following the human perception model, the aggression detection model also contains two parts similar in function to the objective sensory and the subjective sensory parts of the human perception model: the observation model and the reasoning model.

4.3.3.2 Observation model

The aggression detection model emulates the properties of the human perception system. Since humans reason in terms of high level concepts, a pre-processing step is necessary that fills the gap between low-level features and high-level concepts. Low-level features are features that can be extracted directly from digital representations whereas high level concepts are the symbols that are reasoned with in the reasoning model. The observation model describes how low level features are extracted and combined into high level concepts (see Figure 4.12).

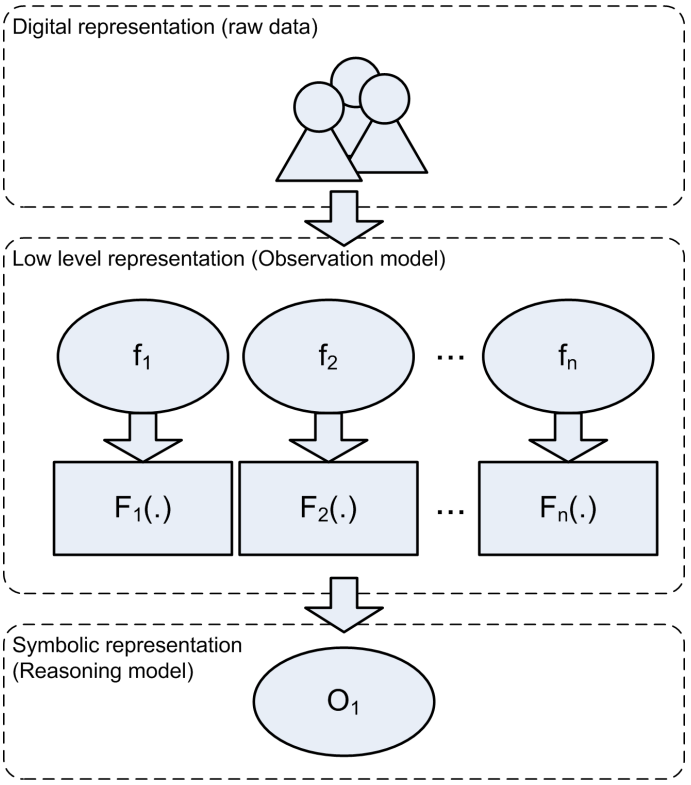


Figure 4.12: The observation model fills the gap between low-level features and high-level meanings of concepts.

The development of the observation model is done in a top down approach by first specifying the high level concepts (e.g. person) that we want to detect and then identifying the low level observable features (e.g. movement, pitch) that need to be detected (i.e. the feature has to be a certain value or be in a certain range or exceed a certain threshold, etc.) in order to conclude the high level concepts. Finally a suitable (off the shelf) algorithm is chosen to realize the detection. In Figure 4.12, low level features are denoted by $f_1, f_2, ..., f_n$ and the detection algorithms by $F_i(.)$.

Table 4.2: High level concepts can be inferred by the detection of low level features.

Concept	Features and detection method
Static object	*Location, color, shape.* Static objects such as doors, seats and windows are easy to detect because they have a fixed location, color and shape.
Passenger	*Face, voice.* Face detection is the most suitable given the chances of other body part occlusion in the confined space. If the passenger is talking he can be detected by speech activity recognition (and sound source localization).
Other objects	*Color, sound.* Other objects that are not native to the compartment such as bags, suitcases etc. can be detected by differencing with the background (this leads to detection of a foreign object (or blob) but does not identify the class of the object). Salient objects need custom detection algorithms e.g. fire, smoke, gun detection. Some concepts can be detected by their characteristic sound e.g. brakes of the train.
Shouting	*Audio threshold, anger emotion, open mouth.* The speech level is the most dominant cue which can be enhanced by the presence of the anger emotion and open mouth(s).
Physical cues	*Body language, movement, hitting, pushing.* Distance between people and their orientation (facing each other), rapid movements and body parts touching. Physical cues are potentially combined with swearing, verbal aggression, preceded by verbal aggression.

Table 4.2 relates some high level concepts to their constituting low level features. To preserve space only the video and audio related features are listed. Other significant concepts from other modalities, e.g. time of day, place are omitted here. In reality concepts are usually more complex then presented in Table 4.2. For example, *physical aggression* needs consideration of features over time (tracking). Chapter 5 will discuss in more detail the available algorithms for the observation model.

4.3.3.3 Reasoning model

Current computer perception applications, especially those that require a higher semantic interpretation of the sensor data, often also use some kind of domain knowledge to guide the perception process. The conflicting views on the precise way the human perception process handles background knowledge is reflected in the myriad of ways that domain knowledge is handled in these applications. Depending on the characteristics and requirements of the problem domain (e.g. complexity, required precision of the results) many reasoning models such as expert system, case based reasoning, logic, Bayesian nets have been proposed during the years. If the problem domain is fixed and small, and the concepts and the relationships between these concepts in the domain are generally static, then domain knowledge and decision rules are sometimes hard coded in a knowledge base (for example [63]). However, if the problem is dynamic and observations are probabilistic, then hard coded decision rules will soon become unmanageable. In this case more complex reasoning such as Bayesian nets and emergence are necessary.

In the recognition phase, algorithms detect motions and events based on the movement and sounds of people in the environment. In the reasoning phase a high level reasoning system infers scenarios from high level concepts. In essence this boils down to combining the results of the recognition phase to a description of the current scenario, and comparing this to known scenarios. If, given the correct context, the current scenario is similar to a known unwanted scenario or if the current scenario deviates too much from a known normal scenario, an alarm situation may be announced. There are a number of different approaches to accomplish this inference.

In general, a data fusion processes have to deal with uncertainty and imprecise data. In fact, representing the uncertainty in observation and in models of the phenomena that generate these observations is part of the data fusion problem itself. This section covers the widely used tools and techniques used to handle the different forms of inexactness of data and knowledge. This section contains a general discussion of some well known models from a knowledge engineering perspective. A more detailed discussion of three specific models focused on the surveillance task can be found further in this thesis.

Rule-based reasoning In the rule-based approach a set of rules describes which conclusion to draw given a certain input. The major component in a rule-based reasoning system is the rule base, which consists of a number of “Rules”. In general, a rule consists of a condition and a consequence which are expressed in the form of an IF-THEN statement. The IF part or condition is a logical statement and the THEN part or consequence specifies the hypothesis if the condition is true. Let U be the universe of discourse for condition and consequence of the rules, then the rule IF x

$is\ A$, THEN $y\ is\ B$, represents a relation between A and B in U. Extension to multiple rules can easily be done by specifying both composition and inference methods.

The rule-based systems are constructed using the knowledge about the correlation between input features and other concepts such as conclusion, hypothesis and other features. This means that rule-based systems require a-priori knowledge of the problem domain. As a result, rules are usually extracted from interviews with domain experts in a so called knowledge acquisition phase. Apart from correlations, rules can also be based on stereotypes and rules of thumb.

Bayesian reasoning Probability theory can be defined as the study of how knowledge affects belief. Belief in some proposition, p, can be measured in terms of a number between 0 and 1. The probability p is 0 means that p is believed to be definitely false (no new evidence will shift that belief), and a probability of 1 means that p is believed to be definitely true. Statistics of what has happened in the past is knowledge that is conditioned on and used to update belief [64].

The conditional probability $P(A|B)$ states the probability of A given that B is true. Much reasoning in data fusion systems can be seen as evidential reasoning: reasoning from observations towards one or more hypotheses to explain the observations. Using the conditional probability notation, this can be more precisely written as $P(H_i|E_1,E_2,...,E_m)$ where H_i represents a hypothesis and E_j represents observed evidences where $1 \leq j \leq m$.

$$P(H_i|E) = \frac{P(E|H_i)P(H_i)}{\sum_{k=1}^{n} P(E|H_k)P(H_k)} \tag{4.1}$$

Bayes' theorem (see Equation 4.1) provides a way of computing this probability given the probabilities with which the evidence follows from the hypothesis ($P(E|H_i)$) and the frequency of the hypothesis ($P(H_i)$).

Emergent reasoning Most automated classification systems require a full list of hypotheses with a specified list of events that occur for each hypothesis. An incoming event increases the probability of the associated hypothesis. The rule-based systems and Bayesian reasoning systems both fall under this category. In emergent reasoning systems adopt a different approach. Here a network of linked events or concepts is built from an incoming stream of events and triggers.

Clusters of related features are formed based on the link strength between the features. The starting point is a matrix of correlation coefficients between pairs of events: given two events, the correlation coefficient tells us how well the events fit together. The challenge is to compose one or more networks out of incoming events. For each network, the system calculates its fitness using the fitness function. This function is based on the size of the network (since bigger networks are fitter), but also on the

average link-strength in the network. The average salience value of the network can also be incorporated into this function. The largest network of highly salient events is the prevailing hypothesis at a given moment. In this way the system is able to generate a hypotheses not defined at start.

4.4 Summary and discussion

To distinguish groups we need to recognize eye contact between people, distinguish conversations between acquaintances from conversations between strangers, and identify a relationship between people by looking at their body language.

Our approach is essentially divided into two models: the observation model describes how low level features from observations are combined into high level concepts and the reasoning model in which high level concepts are reasoned with in order to infer the presence of aggression. The main advantage of this division is that performance of signal processing algorithms can be separated from logical inference algorithms. Since we have decided to use third party signal processing algorithms the division is essential to ensure that the performance of the (domain specific) reasoning can be independently measured from the chosen third party solutions. In fact, the validation of the reasoning models, as we shall see in Section 11.5, will be done using manually annotated ground truth values.

Depending on the characteristics and requirements of the problem domain (e.g. complexity, required precision of the results) several different types of reasoning models with ranging complexities can be employed. In this chapter we briefly discussed rule-based reasoning, Bayesian nets and emergent reasoning. Many other reasoning schemes exist such as Belief theory, templates, neural networks, fuzzy theory and voting logic fusion. However, we have chosen these three reasoning schemes for several different reasons. Rule-based reasoning serves very well as a first system to test the essence of our reasoning strategy, but since the problem is dynamic and observations are probabilistic, the hard coded decision rules will soon become unmanageable. In this case more complex reasoning such as Bayesian nets and emergence are necessary. Emergent systems have the added flexibility of not requiring predefined hypotheses. However, they currently still lack the proven track record and formal mathematical footing of the Bayesian nets.

Part II

Design

Chapter 5

A detection model for aggressive situations

Representing, associating, interpreting and learning behavioral patterns and rules from observation is always a challenging research problem. This problem becomes even more difficult in complex environments with many unexpected behavioral patterns from objects, individuals or small groups of people acting under certain plans at the same time in different locations. This chapter presents techniques that can be used to realize the aggression detection function described in the previous chapters. The following sections go into more detail in each function group and give suggestions for possible algorithms. This chapter serves as a consolidation of the analysis in the previous chapters into a working model for aggression detection. We have broadly divided multi-sensor fusion algorithms into four classes: estimation methods, classification methods, tracking methods and inference methods. This chapter deals with the first three methods and gives a brief overview of the inference methods. Each inference method will be discussed in more detail in separate chapters (Chapter 7 to Chapter 9).

5.1 Overview

With the availability of inexpensive sensors and the ever increasing processing power of computers, the number of surveillance and surveillance related research projects has increased. The most commonly used modalities for this purpose are video [65, 6, 66, 67, 68], audio [69, 70, 71, 72] or a combination of both [73, 74]. We observe that in complex surveillance environments, such as in public transport systems, the combination of multiple modalities is more common, e.g. PRISMATICA for railways [5, 73] and ADVISOR for metro stations [6].

Currently, commercially employed surveillance applications have very simple goals, e.g. to detect trespassers or crowding in areas. Our research and the research direction in the last few years tries to go beyond that. We aim to infer the behavior of people in the surveyed area in order to detect what is going on at a high abstract level and to predict the level and potential of aggression. One approach to this is to view the separate events (e.g. arm motions, gestures) as related parts of a bigger scenario e.g. fighting, ticket checking. A scenario thus is a combination of objects, states, events or sub scenarios ordered in time and space. In the representation of the scenario, the influence of the separate events on the outcome of the scenario is also included e.g. the occurrence of shouting might cause the ticket checking scenario to trigger.

For the scenario representation to be useful, a number of basic assumptions are imposed:

- scenarios can be uniquely identified by its elements (object, states, events etc.). Thus, the observation of particular object or event makes the scenarios more plausible,
- the elements are related to each other and have a logical significance in time (time logic, causality) and place (location, motion) which can be defined a priori.

The above assumptions allow us to create prototype scenarios for many aggressive situations. The scenarios can also be used as a guideline for data acquisition. The creation and validation of scenarios is discussed in Section 10.1.

The goal of our surveillance systems is to recognize the scenario based on the recognition of the separate events and objects and their properties from features extracted from the input data. The basic idea is that specific scenarios (and their probabilities of occurring) are defined a priori. During runtime, the surveillance system tries to infer the consequences of the activity/scene recognized based on this a priori knowledge. Bayesian networks can be used for the inference (e.g. in [5, 75]), but other approaches could be used, including multi-layered hidden Markov models [76, 77] and coupled hidden Markov models [78]. For activity recognition in normal environments (e.g. rooms, offices) data can be collected quite easily. Härmä et al. [70] collected all interesting acoustic events in an office over a duration of more than two months. Similar research was conducted by Stauffer and Grimson [79].

Our work differs from others by the fact that our system has to work under more problematic settings (e.g. no stable lighting conditions). Moreover, the scenarios we are interested in are somewhat unusual. For ethical and privacy reasons it is unsuitable to use real world recordings of aggressive situations, while the emotional undercurrent that often needs to be present in aggressive situations makes it very difficult to set up a controlled experiment for data collection.

5.1.1 Detection model for affective aggression

Emotions can be conveyed in different modalities such as text, speech, facial expressions and gestures. Detecting affective aggression is essentially a process of trying to come to a conclusion by selecting the most probable emotion the currently available features can resemble. One approach is to recognize the intensity and valence of affects or emotions in the data. The predominant emotion associated with aggression is anger. Sadness, fear and disgust to a lesser extent can also indicate aggression.

Compared to instrumental aggression detection, affective aggression detection (e.g. angry shouting) can be detected using relatively low level processing. However, reliable conclusions about aggression can only be achieved by taking into account high level context information. Consequently, conclusions drawn from the affective aggression detection model might be unreliable even if the underlying algorithm to recognize the emotion is very robust and accurate.

The main justification for affect recognition therefore is that the results can be used to support the other detection models or as an indicator for the system or human operators to focus attention on the particular situation. As a result, the detection model for affective aggression will be presented in Section 5.5.3 as part of the instrumental detection model.

5.1.2 Detection model for instrumental aggression

Instrumental aggression (see Section 2.1) has a certain obvious goal. Enabling instrumental aggression detection is a two step process. First identify the goal of the most common aggressive behaviors people can have in the train. Next, for each behavior, find the features that identify this behavior (a feature in this sense is for example an action that occurs in the behavior). Also find the arrangement of the features in time (chronology) and the relationship of one feature with other features (co-occurrence and association). The advantage with instrumental aggression is that once aggression has been identified, it is instantly known what the cause and the possible consequences of the aggression are.

In the detection model, behaviors are represented as hypotheses and the detected features can express their support for a given hypothesis. The algorithms by which the features express their support will be discussed in Chapter 7 to Chapter 9. This section discusses the transformation from raw data into relevant features.

Figure 5.1 shows the detection model for affective aggression and instrumental aggression in one diagram. In terms of the observation model discussed in Section 4.3.3, behaviors as well as emotions can be seen as high level concepts which can be detected using a number of detectors for low level features.

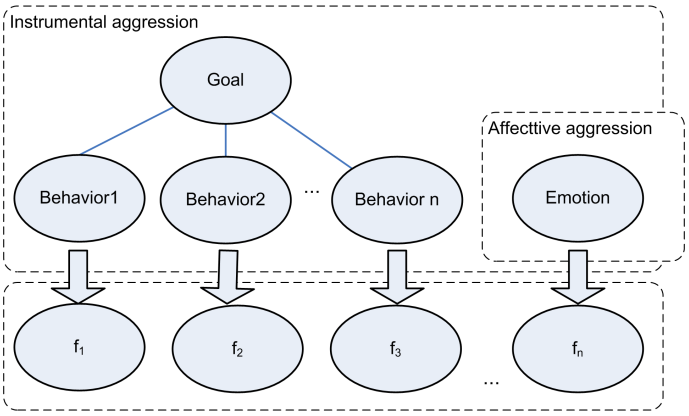


Figure 5.1: Overview of the detection model for aggression.

5.1.2.1 General workflow

In the general workflow of an audio and video based surveillance system, the system receives audio and video streams to discover important features in it. Without going into too much detail, Figure 5.2 depicts the general components and work flow for the video case (a similar work flow applies also for the audio case).

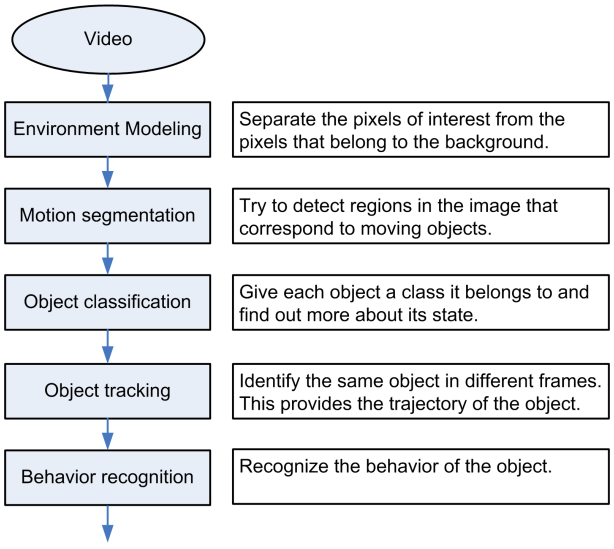


Figure 5.2: The general workflow of a video surveillance system.

The general structure is not mandatory, depending on the goal of the surveillance system, some items can be omitted. In literature, environment modeling is considered as part of motion segmentation or motion estimation. To avoid confusion with analogous methods in the audio domain, we will group all these methods into the term estimation methods. This way, these methods correspond better with level 0

processes in the JDL model (as the JDL model does not have an environment modeling level). The other correspondences are more apparent: object classification algorithms and object tracking algorithms correspond to level 1 processes, while behavior recognition algorithms fall into level 2 processes.

Even though the global work flow is relatively well defined, the separate components in the flow require careful analysis of the problem the system is to solve. In their review of work done in audio-video fusion Abdeljaoued et al. [80] conclude that there are still many open questions related to multi-modal analysis which have to be taken into account when designing or developing a multi-modal system:

- Which classifiers are the most efficient ones for the application?
- Which features are the most appropriate for the classifiers?
- How many classifiers are necessary?
- Given a set of classifiers, which combination scheme may be the one that improves the overall performance of the system?
- How is the information or the outputs of each separate classifier fused?

The first two questions are inherited from classical pattern recognition, whereas the last three are specific for multi-modal fusion analysis. Hall and McMullen [24] offer a comprehensive introduction to the concepts of multi-sensor data fusion. They present key algorithms, provide advice on their utilization, and raise issues associated with their implementation. The choice of a suitable fusion algorithm depends on the available sensor type and the fusion level, for example, in sound localization with multiple microphones, physical characteristics of the signals, such as signal strength, are more important than the meaning of the sound.

Abdeljaoued et al. however do not mention a very decisive factor in real live applications: the data. In real life applications the location and the quality of the sensors may not be in the hands of the researcher. Sensors may need to be placed in suboptimal locations so that they are unobtrusive. In addition, for economic reasons not the best quality sensors may be employed. Also environment factors pose a bottleneck on the quality of the data the surveillance system is able to get. In addition, most existing techniques for classifier training require a large number of labeled samples to learn correctly and suffer from the problems of context and concept uncertainty when only a limited number of labeled samples are available. Careful analysis of the data reveals which classifiers are possible.

The separate components of the general work flow will be discussed next.

5.1.2.2 Dealing with uncertain dynamic environment

Before continuing it is essential to point out the two components that we are dealing with: the *environment* and the *observer*. We consider the environment as the complex entity of the train compartment that can be observed (as presented earlier in Chapter 4). We have to make a clear distinction between this (true) environment and the *environment representation* of the aggression detection system. The environment representation (ER) only contains information that is relevant to the observer. It is typically composed of discrete and continuous variables and hypotheses, representing the knowledge of the observer. As the environment and sensors in real life applications are often of sub-optimal quality, the information in the ER is rarely fully reliable and is contaminated with uncertainties.

As stated above, the nature and quality of the input data forms the first mechanism by which classification algorithms should be chosen. For the sake of clarity, we will describe the detection models in this chapter under the assumption that uncertainty is negligible. Chapter 7 to Chapter 9 present different approaches to deal with this uncertainty. In the reasoning model in Chapter 7 we acknowledge the presence of uncertainty and try to overcome it by a process of weighing evidences. The assumption is that, even though the observations may not always be correct, they are so most of the time. As a result, in the long run, the evidence for the correct hypothesis will outweigh all other hypotheses. The reasoning model just selects the hypothesis with the most supporting evidence. A similar idea is employed in Chapter 9. In Chapter 8 we express uncertainty in probability functions over variables and hypotheses.

Another property of the environment representation is its temporality: sensor observations only capture a snapshot of an aspect of the environment. As the environment is dynamic and evolves over time, so too, must the variables in the environment representation that correspond to the changed entities. Whereas the environment is innately up-to-date, the environment representation may not be. Second, the changes do not occur at random, but are governed by certain spatial and temporal constraints. The environment representation should also maintain a representation of these constraints, either as part of the knowledge representation of the domain or as part of the internal state representation of the observer.

5.2 Environment modeling

In a frame of a video sequence, not all pixels are of interest. The uninteresting parts of a frame are called the background. For example, in a scene where two people are walking in front of a tree, the tree may be considered to be background. By removing the background, the calculations of the two people walking can be much more accurate. In the audio case, environment modeling methods try to filter speech and other interesting sounds from background noise, while motion estimation tries to detect the motion of a sound source.

5.2.1 Background subtraction

The goal of background subtraction is to detect all the foreground objects given a frame sequence from a camera. The detected foreground objects are called blobs. Blobs can be used in subsequent processing steps such as tracking and behavior recognition. In principle, background subtraction can be achieved by comparing the current frame with an image of the scene's static background. The first consequent problem is: how to automatically obtain the image of the scene's background? This problem is even more complex when taking into account gradual as well as sudden illumination changes, motion changes (as a result of camera oscillations) and high-frequency background objects (such as tree branches, sea waves, and similar). Several methods for background subtraction will be discussed here.

5.2.1.1 The basic method

In the most basic method, if the difference between a pixel of the current frame $F(x,y)$ and the corresponding background pixel $B(x,y)$ is greater than a certain threshold T, then $F(x,y)$ is said to be a foreground pixel. In its simplest form, the previous frame is taken as the background i.e. $B_i(x,y) = F_{i-1}(x,y)$. Other commonly used measures for the background are the average or the median of the previous n frames or the running average (see Equation 5.1).

$$B_{t+1} = \alpha F_t + (1-\alpha)B_t \tag{5.1}$$

The learning rate α is typically 0.05. The main assumption in the basic method is that each pixel location is based on the pixel's recent history. Therefore the basic method evidently works only in particular conditions of object speed and frame rate. In addition, the performance of this method is very sensitive to the threshold T. In addition, the method does not take into account the spatial correlation between different (neighboring) pixel locations.

5.2.1.2 Running Gaussian average

A more advanced approach to the basic method is adopted in Pfinder [81] where a Gaussian distribution (μ,σ) is fit over each point in the texture surface. This gives the background probability distribution. μ_0 is defined as the mean of the pixel on the texture surface (expressed in the YUV color space over the last n values) and σ_0 the covariance of that pixel's distribution. The background probability distribution is updated using the running average according to Equation 5.2 and Equation 5.3.

$$\mu_{t+1} = \alpha F_t + (1-\alpha)\mu_t \tag{5.2}$$

$$\sigma_{t+1}^2 = \alpha(F_t - \mu_t)^2 + (1-\alpha)\sigma_t^2 \tag{5.3}$$

5.2.1.3 Gaussian Mixtures

In practice, each pixel in a frame rarely results from a single surface under fixed lighting. Instead, multiple surfaces often appear in the view frustum of a particular pixel and the lighting conditions change in time. As a result Stauffer et al. [79] use a mixture of Gaussian distributions to reflect the changing nature of the scene's background. The model considers the values of a particular pixel over time. Furthermore, the recent history of each pixel, $\{X_1, ..., X_t\}$, is modeled as a mixture of K Gaussian distributions. Thus, the probability of observing the current pixel value is

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{5.4}$$

where K is the number of distributions determined by the available memory and computational power, $\omega_{i,t}$ is an estimate of the weight (the portion of the data accounted for by this Gaussian) of the ith Gaussian in the mixture at time t, $\mu_{i,t}$ and $\Sigma_{i,t}$ are the mean value and covariance matrix of the ith Gaussian in the mixture at time t, and where η is a Gaussian probability density function.

At every new frame, every new pixel value X_t is evaluated against the existing Gaussian mixture model. A simple heuristic (e.g. those within 2.5 σ) is used to hypothesize which are most likely to be part of the "background process". Pixel values that do not match one of the pixel's "background" Gaussians are grouped into blobs. The μ_i and σ^2 of matched distributions are updated by a running average with a learning factor for adapting current distributions. The μ and σ parameters for unmatched distributions remain the same.

The prior weights of the K distributions at time t are adjusted as follows:

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{5.5}$$

where α is the learning rate and $M_{k,t}$ is 1 for the model which matched and 0 for the remaining models. After this approximation, the weights are renormalized. If none of the K distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance, and low prior weight.

One of the significant advantages of this method is that, when something is allowed to become part of the background, it does not destroy the existing model of the background. The original background color remains in the mixture until it becomes the K-th most probable and a new color is observed. Therefore, if an object is stationary

just long enough to become part of the background and then it moves, the distribution describing the previous background still exists with the same μ and σ^2, but a lower ω, and will be quickly reincorporated into the background.

5.2.1.4 Other methods

A range of other methods are found throughout literature. Some well known methods are mentioned here in brief, for an extended discussion we refer to the literature survey [41].

Kernel density background estimation. Elgammal et al. [82] propose a generalization of the Gaussian mixture model where each single sample of the N previous samples is considered to be a Gaussian distribution by itself. The method consists of a kernel density estimator to estimate the density function of the intensity distribution at any moment of time given only very recent history information.

Mean shift based background estimation. Most methods such as kernel density background estimation are extremely sensitive to the kernel bandwidth, leading in many practical cases to a poor estimate of the true background distribution. A technique with lower dependency on the bandwidth parameter is the mean shift based background estimation [83]. The mean shift technique is an iterative gradient ascent method with convergence properties that allow it to detect the modes of a multivariate distribution and their covariance matrix.

The eigenspace model. To detect moving objects, Oliver et al. [78] adaptively build an eigenspace that models the background. Previous approaches model the static background of each separate pixel and do not take into account the substantial degree of correlation between neighboring pixels. The eigenspace model describes the range of appearances (e.g. lighting variations) that have been observed. In the eigenspace model, the static background image is formulated as an appearance model where any image instance may be represented as a linear deformation from an average background image.

The codebook algorithm The codebook algorithm [84] adopts a quantization/clustering technique, inspired by Kohonen to construct a background model from long observation sequences. For each pixel, it builds a codebook consisting of one or more codewords. Samples at each pixel are clustered into the set of codewords based on a color distortion metric together with brightness bounds.

Table 5.1 shows a summary and comparison of the various background subtraction techniques discussed in this section.

The accuracy score of the methods in Table 5.1 is gathered from results reported in literature. It is not possible to select the best background subtraction method for the

Table 5.1: Comparison of background subtraction methods.

	Speed	Memory Req.	Accuracy
The basic method	+	-	-
Running Gaussian Average	+	+	-
Gaussian Mixture	o	o	o
Kernel density	o	-	+
Mean-shift	-	-	+
Eigen-background	o	o	o
Codebook	+	o	o

train compartment based on these results. This can only be done after an objective comparison with recordings of the actual train compartment. The result of this comparison is presented in Section 11.1.1. In the end, the codebook method was chosen because of the balance between accuracy and performance.

5.2.2 Motion segmentation

Motion segmentation in image sequences aims at detecting regions corresponding to moving objects such as vehicles and humans. The background subtraction techniques discussed in the previous section are popular methods for motion segmentation. Moving regions in an image are obtained by taking the difference between the current image and the reference background image in a pixel-by-pixel fashion.

Alternatively, optical-flow-based methods can be used to detect independently moving objects even in the presence of camera motion. Optical-flow-based motion segmentation uses characteristics of flow vectors of moving objects over time to detect moving regions in an image sequence. Barron et al. [85] review the performance of several optical flow techniques. However, most flow computation methods are computationally complex and very sensitive to noise, and cannot be applied to video streams in real time without specialized hardware.

5.2.3 Sound source separation

The counterpart of background subtraction in the audio domain is sound source separation. Sound separation covers techniques to segment the part of the audio signal that is not noise. The main problem is to decide which part of the signal is meaningful without having to recognize the signal. This problem becomes apparent in the example of a lively train compartment.

The first problem is to separate the speech part of the signal from non speech signals such as the background noise of the engines. The second problem is to separate the different speech sources when multiple passengers are talking simultaneously. This is known as the cocktail party problem. Recently, a particular method that can

be applied to solve these problems, called independent component analysis (ICA), has gained wide-spread attention [86]. As the name implies, the basic goal is to find a transformation in which the sound sources are statistically as independent from each other as possible.

5.2.3.1 Speech detection

Speech activity detection (SAD) tries to discriminate between speech and silence. It is one of the fundamental issues in many speech processing tasks but becomes very difficult in the presence of noise. The most common SAD algorithms are HMM-based, energy based or speech enhancement based.

To create automatic speech detectors to determine if a signal contains speech we have to examine how speech sounds are produced. Speech is produced in the vocal tract. Its energy stems from the flow of air around constrictions in the vocal tract and vocal folds, which leads to a signal in which all frequencies contribute [87]. This restricts the signal to a discrete set of frequencies that are periodic with the main period of the vocal fold movement. The resulting sound is changed by the resonant properties of the vocal tract. The excitation of the vocal folds determines the pitch and the loudness, which carry intonation and information about speaker identity [88]. Moreover, the excitation is a set of harmonics. This discrete set of frequency components is filtered by the vocal tract. This leads to a relative enhancement of some of the harmonics and to a relative attenuation of others. Frequency regions where harmonics are enhanced are called formants.

The typical design of a SAD algorithm is as follows:

- Noise reduction stage. There may be some feedback in this sequence, in which the SAD decision is used to improve the noise estimate. Typically, the first few frames are assumed to be noise and a threshold is computed from it. The feedback method relies on the ability to track the noise characteristics and use them to set a speech threshold adaptively.

- Then some features or quantities are calculated from a section of the input signal. The objective of the feature extraction process is to compute discriminative speech features suitable for detection. A number of robust speech features have been studied in this context. The different approaches include: full-band and sub-band energies, spectrum divergence measures between speech and background noise, pitch estimation, zero crossing rate, and higher-order statistics.

- A classification rule determines a sound section as speech or non-speech. Often the classification criterion is whether the calculated values exceed a certain threshold. Any frame with more energy than the threshold is marked as speech. The threshold is continually updated using the energy level of the non-speech

frames. The different measures as used in SAD methods include spectral slope, correlation coefficients, log likelihood ratio, cepstral, weighted cepstral, and modified distance measures. Segmentation of the input waveform has also been treated as an edge-detection problem. Instead of estimating the energy levels of noise and speech, the slope of the energy is convolved with a predetermined matched filter. These methods are not inherently robust to noise bursts, telephone tones, and varying noise powers, and require additional heuristics in real applications.

5.2.3.2 Blind source separation

The cocktail party problem is one of the topics of interest in blind source separation (BSS). Blind source separation deals with obtaining estimates of the unknown source signals using a number of mixed observations, where the mixing process is also unknown. A special case of BSS called Independent Components Analysis (ICA) is the problem of finding directions in the data space such that the independence of the projections is maximized [89, 86]. Assuming that the speakers are independent, the ICA assumption is valid in the train compartment.

In ICA the observation signal x is a linear mixture of its sources s given by $x = Hs$, where H represents the unknown mixing matrix. Limiting the discussion to the square mixture case where the number of observations equals the number of sources, the square matrix H is assumed to be invertible. The separation is achieved by finding a matrix A that optimizes some criterion that measures the independence of the outputs, which are given by $y = Ax$.

A criterion for measuring independence is Shannon's mutual information. For n random variables $Y_1, ..., Y_n$, whose joint pdf is $f_Y(y)$ and marginal pdfs are $f_1(y^1), ..., f_n(y^n)$, respectively, this mutual information can be written in terms of the marginal and joint entropies:

$$I(Y) = \sum_{o=1}^{n} H(Y_o) - H(Y) \tag{5.6}$$

where Shannon's entropy is defined as

$$H(Y) = -\int_{-\infty}^{\infty} f_Y(y) log\, f_Y(y) dy \tag{5.7}$$

$$H(Y_o) = -\int_{-\infty}^{\infty} f_o(y^o) \log f_o(y^o) dy^o \tag{5.8}$$

for marginal and joint entropies, respectively. The difficulty in using information theoretic measures lies in estimating the density of the underlying data. Successful algorithms must use robust probability density function estimators in order to improve convergence to asymptotic performance and to minimize sensitivity to outliers. A common approach is polynomial expansions. Truncation of these expansions introduce inherent errors in the estimation of the information theoretic criteria. Alternative density estimation approaches include Parzen windowing, orthonormal basis functions and kernel estimates.

5.2.3.3 Sound source localization

An optional final step in the sound separation process is the localization of the single (separated) sound sources in space. The methods used in sound source localization can roughly be divided into binaural cues and monaural cues.

Binaural cues Binaural localization relies on the comparison of auditory input from two separate detectors; most evolved auditory systems feature two ears, one on each side of the head. The primary biological binaural cue is the delay between the time when sound from a single source reaches the near ear and when it reaches the far ear. A less biologically important binaural cue is the reduction in loudness when the sound reaches the far ear.

Monaural (filtering) cues Monaural localization mostly depends on the filtering effects of external structures such as the head, shoulders, torso, etc. Sounds are frequency filtered specifically depending on the angle from which they strike the various external filters. In order to enhance filtering information, many animals have evolved large, specially shaped outer ears.

The most direct cue to distance is sound amplitude, which decays with increasing distance. However, this is not a reliable cue, because in general it is not known how strong the sound source is. In case of familiar sounds, such as speech, there is an implicit knowledge of how strong the sound source should be, which enables a rough distance judgment to be made.

Monaural localization algorithms are less accurate and more complex then binaural ones. In addition, we already have multiple microphones in the train compartment. Therefore, this thesis focuses on binaural methods for sound source localization (SSL). Two different approaches for binaural SSL exist. The first approach uses the time delay reception between microphones located at different distances from a sound source (e.g. [90]). The second approach is based on the intensity or energy decrease of a sound signal with the distance to the microphone. A detailed investigation of the performance of both methods has been conducted in [91]. The investigation was performed in a real train compartment, thus taking into account the effect of

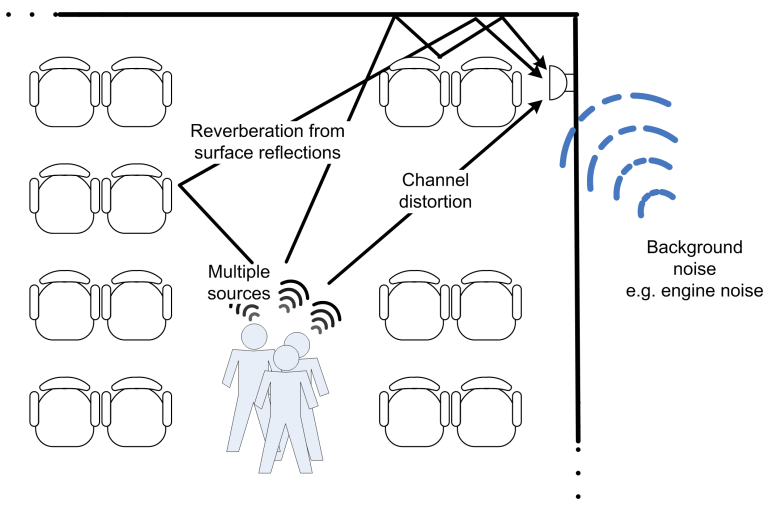


Figure 5.3: Sources of distortion.

environment specific factors, such as sources of distortion (see Figure 5.3), in the results. The energy decrease method discussed next, proved to be the more suitable method.

5.2.3.4 Level Difference Estimation

Sound source localization by interaural level differences depends on the diffusion of the signal energy as a sound signal travels through space. As a result, microphones will receive the same signal with a different volume depending on the distance to the sound source. The instantaneous sound pressure (i.e. volume) received by the ith microphone can be modeled as:

$$y_i(t) = s(t)/d_i. \tag{5.9}$$

where $y_i(t)$ is the sound pressure as received by microphone i and $s(t)$ is the source signal, d_i is the distance from the source to the ith microphone. These equations ignore the time shift in the signal, caused by the traveling distance (it is assumed that these differences are too small to have a significant influence).

A convenient and more usable method to work with the signals received by the different microphones, is by doing calculations on the *effective* sound pressure instead of the instantaneous sound pressure. To calculate the *effective* sound pressure (p_i) for microphone i while having a sound source at a fixed location during time interval $[0, T]$, the following equation is used:

$$\begin{aligned} p_i &= \sqrt{\frac{1}{T}\int_0^T y_i^2(t)dt} \\ &= \frac{1}{d_i}\sqrt{\frac{1}{T}\int_0^T [s^2(t)]dt} \end{aligned} \tag{5.10}$$

From Equation 5.10 it follows that the effective sound pressure is inversely proportional to the distance from the microphone to the sound source: $p_i d_i = p_j d_j$. The ratio of the effective sound pressure between microphone i and microphone j is p_i/p_j (denoted as r_p) and can be written in terms of the distance from a microphone to the sound source:

$$r_p = d_j/d_i \tag{5.11}$$

When the effective sound pressures for microphone i and microphone j are known, r_p can be calculated. The results of sound source localization with the above method are discussed in Section 11.1.2.

5.3 Object detection

In order to perform automated aggression detection we need a system that can recognize objects. However, only certain objects (and/or the sound they produce) are good indicators for aggression. People are the most important objects in aggression. In addition to detecting the human object, it is also important to recognize the role of a person, e.g. aggressor, police or conductor, and recognize his behavior (i.e. the actions and interactions). Furthermore, it is necessary to track this information over time.

Object detection algorithms are very numerous and vary in many different aspects depending on the goal of the detection algorithm and the working environment. In this section we will focus on those algorithms that detect objects relevant to aggression in the train compartment. Object detection implies classification (determining the type) of a detected entity into an object class. Thus object detection algorithms are essentially classification algorithms, which means that they try to give each object a class it belongs to (e.g. faces), ignoring objects of unknown class (e.g. birds).

First we present the video related algorithms followed by the audio related ones.

5.3.1 Human detection

There are many characteristic features by which humans can be detected. In a side project on behavior detection using video [92], we have analyzed and experimented with many of these features such as skin color, human body, torso, face detection, and speech detection. In the video domain, face detection is the most promising one for the train compartment as the other human body part based detection method suffered more from the effects of (self) occlusion. The skin color based detection method proved to be too reliant on stable lighting conditions. On the other hand, detecting the presence of humans by means of speech detection (see Section 5.2.3.1) performs well, but does not give an accurate estimation of the position of the human.

5.3.1.1 Face detection

One of the hottest areas in Artificial Intelligence research is face detection and recognition. Face detection is defined as being a computer technology that determines the location and size of human faces in arbitrary (digital) images. It detects facial features and ignores everything else, such as buildings, trees and bodies. Unlike face recognition, face detection does not inform about identification or describe anything about facial expressions or emotions seen on a face. Currently the most widely used detector is the Viola and Jones [93] algorithm. A rotation invariant version of this algorithm was created by Wu et al. [94].

Viola and Jones. In 2004 Paul Viola and Michael Jones introduced a variant of the AdaBoost machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. The algorithm is based on the rapid evaluation of three types of simple rectangular features chosen for their computational efficiency. By using smart image representation (integral image) features can be evaluated in constant time, much faster than more complex alternatives such as eigenfaces, 2-D Gabor wavelets and 2-D discrete cosine transform. Even though features may be evaluated rapidly, it does not adequately compensate for their number, however (for example, given that the base resolution of the detector is 24x24, the exhaustive set of rectangle features is over 180,000. Thus, the object detection framework employs a variant of the learning algorithm AdaBoost to both select the best features and to train classifiers.

Rotation invariant face detection. The problem with Viola and Jones is that the face detector is trained on faces with a fixed orientation (e.g. frontal faces). Because of the multi-perspective nature of real live images, the algorithm needs to be adapted to yield good results. The face detector by Wu et al. [94] also uses AdaBoost but characterizes itself in its ability to detect faces in multiple orientations. It introduces a width-first-search tree structure that achieves high performances in both speed and accuracy for multi-view face detections.

However, contrary to the Viola and Jones detector (which is available through OpenCV[1]), an implementation of this face detector is not publicly available.

5.3.2 Unusual object detection

For unusual object detection we draw on the research area of Content Based Image Retrieval (CBIR) systems which aim to capture the semantics of scenes and objects present in images. Whereas detection algorithms such as the face detection algorithms discussed above focus on detecting a specific and well defined class of objects, unusual object detection emphasizes the ability to retrieve lesser defined objects or unknown blobs over frames. This way, the unknown object can be associated and tracked over frames.

Popular classifiers employed in CBIR include Support Vector Machines. These are termed as linear classifiers or hyperplane classifiers, which work under the assumption that the data we are classifying is linearly separable. Others try to compute a similarity measure (L1 or L2 norm) between these high dimensional features to return the relevant images. But in high dimensions, data becomes very sparse and distance measures become increasingly meaningless. Therefore we see a degradation in the quality of the results of CBIR systems.

To overcome the shortcomings mentioned above, Ali and Shah [95] propose an integrated framework of Kernel PCA and AdaBoost. The essential idea is to employ Kernel PCA as a non-linear feature extractor by mapping input space to a higher dimensional feature space, through a non-linear map where the data is linearly separable. The justification of converting data to higher dimensional space is often alluded to as the Cover's theorem. This theorem formalizes the intuition that the number of separations increases with the dimensionality as we can have more views of the class and non-class data. Once in the feature space, which is of high dimension, AdaBoost is used to uncover the patterns by selecting only the relevant (discriminative) dimensions. This selectivity not only reduces the dimensionality but also speeds up online classification and retrieval. Classifiers can be trained using small training sets which is an added advantage.

5.3.3 Recognition of environmental sounds

Recently, many efforts have been made to identify the sounds which fill our everyday acoustic environment. Goldhor [71] made recordings of environmental sounds under background noise conditions in domestic and business environments. Background sound levels were typically 25 to 30 dB below signal levels. For sounds generated by what we judged to be "consistent" sound sources (such as a smoke alarm,

[1]Open Source Computer Vision (OpenCV) is a library of programming functions for real time computer vision, available at http://sourceforge.net/projects/opencvlibrary/

but unlike a barking dog) each token (sample) was recorded with the microphone at a different distance or orientation.

The recognition system constructed for the study, effectively classifies isolated tokens from several dozen disparate sound sources. Two dimensional cepstral coefficients proved to be effective classification features. Only the cepstral coefficients representing "low frequency" spectral and temporal variations were required in order to obtain accurate classification. Mel cepstral coefficients appear to be almost as effective as linear frequency cepstral coefficients. The most difficult problems associated with constructing an environmental sound recognition system may prove to be dealing with reverberation, separating multiple simultaneous sound sources, and characterizing sequentially-structured sounds such as footsteps.

Atrey et al. [72] present a hierarchical approach for audio based event detection. The proposed approach first classifies a given audio frame into vocal and nonvocal events, and then performs further classification into normal and excited events. Events are modeled using a Gaussian Mixture Model optimized for four different audio features zero-crossing rate (ZCR), Linear Predictor Coefficient (LPC), Linear Predictive Cepstral Coefficient (LPCC) and Log Frequency Cepstral Coefficient (LFCC).

5.4 Tracking

Video tracking is the process of locating moving objects in time. The purpose of tracking (as opposed to motion detection or motion segmentation) is to ide ntify the same object in different (consecutive) frames. For instance a person who walks in front of the camera in frame one must be identified as the same person in frame two etc. Over time, the tracking algorithm provides the trajectory of this person's walk in front of the camera. There are several methods for tracking in video frames: color value based methods such as those discussed in motion segmentation can be used for blob tracking, while other methods that depend on domain information such as contour based tracking or feature based tracking can be used for high level tracking algorithms.

Veenman et al. [96] define the tracking problem as a sequence of n time instances. At each time instance t_k we have m_k measurements x_{ik} with $1 \leq i \leq m_k$ and $1 \leq k \leq n$. At t_1, m points ($m \leq m_1$) are identified among the m_1 measurements. The task is to track these m measurements over the whole sequence, or to return a set of trajectories that represent the motion of the m points from $t_1, t_2, ..., t_n$. A trajectory is a fully tracked path of a single point: $X_{i_1 1}, X_{i_2 2}, ..., X_{i_n n}$, with $l \leq i_k \leq m_k$.

Video tracking systems usually employ a motion model which describes how the image of the target might change for different possible motions of the object to track. Using a motion model, tracking becomes a continual process of prediction and update of the motion model and trajectories as shown in Figure 5.4.

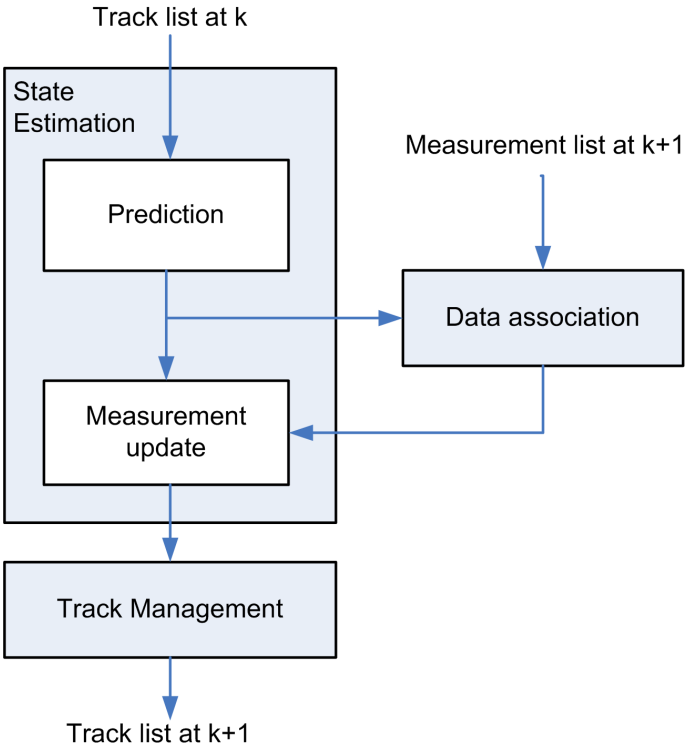


Figure 5.4: Basic elements of a tracking system and a block diagram of a recursive the tracking process (source: Abdeljaoued [80]).

Prediction The motion model is used to make a forward projection of the previous state estimate to obtain the predicted estimate for the current frame.

Measurement update A new measurement is incorporated into the predicted state estimate (list of trajectories) to obtain an improved estimate and motion model.

5.4.1 Target representation

Object tracking is an application-dependent task: the movement possibilities and constraints between a person in a train compartment and a flying airplane are quite different. As a result, there is not one general tracking methodology for solving all tracking problems. Different algorithms and procedures are needed and used for different applications. Target (motion) modeling and representation forms the first step in the design of a tracking system.

In the top down approach, the (motion) characteristics of high-level entities in the application domain (e.g. humans, objects) are defined in advance. Ideally, high-level primitives such as actions and events should be extracted from the video. This allows a semantic description of the video and thus intuitive tracking. Some researchers (e.g. [80]) argue that despite advancements in video segmentation, it is not yet feasible to reliably extract semantic primitives, especially in real-life scenarios. Therefore,

we rely on low-level features for target representation such as color, textures and motion and apply a bottom-up process for target tracking.

The choice of low-level features limits us to the low-level target tracking algorithms e.g.:

Blob tracking. Segmentation of object interior (for example blob detection, block-based correlation or optical flow).

Kernel-based tracking. An iterative localization procedure based on the maximization of a similarity measure (Bhattacharyya coefficient).

Contour tracking. Detection of object boundary (e.g. active contours or Condensation algorithm).

Visual feature matching. Tracking by means of one or a set of specific features that characterizes the target (e.g. facial feature tracking).

5.4.2 Data association

One of the difficulties in video tracking is to associate a new measurement to a separate track, especially when the objects are moving fast relative to the frame rate and the amount of tracks is numerous. This problem is known as the data association problem. There are a number of reasons why data association is difficult. In the tracking process, predictions are first made as to the expected locations of the current set of objects of interest. When these predictions are then matched to the actual measurements, problems may arise because of several different reasons. Predictions may not be supported by the measurements because objects have ceased to exist, objects have been occluded, new objects have appeared, or the measurements are a result of noisy sensors. In addition, more than one measurement may match a prediction or a single measurement may match more than one feature. Resolving these ambiguities is the essence of the data association problem. A number of statistical data association techniques have been developed.

5.4.2.1 Nearest-neighbor approach

An obvious approach is to use a metric to quantify the likelihood and that assigns measurements to their closest geometric features. The smaller the distance between a measurement and its predicted value, the more likely it is to have originated from it. If measurement x_{ik} at time k is normally distributed about its predicted value $\hat{x}_{ik}(k|k-1)$), then a common distance measure is the Mahalanobis distance defined by:

$$d^2(x_{ik},\hat{x}_{ik}) = (x_{ik}-\hat{x}_{ik})^T S(k)^{-1}(x_{ik}-\hat{x}_{ik}) \tag{5.12}$$

Where $S(k)$ is the covariance matrix. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant. This nearest-neighbor approach usually performs badly since there is always a finite chance that the closest measurement is incorrect.

5.4.2.2 Track-splitting filter

The nearest-neighbor methods make assignment decisions based solely on the current image frame. More accurate decisions can be made by postponing decisions and examining the information from several frames. In track-splitting, a track can have several branches, each branch denoting a different assignment of a measurement to a target. At the same time the same measurement may be assigned to more than one target. As a result, algorithms all have exponential complexity and heuristics must be used to constrain the search space. In addition, since the number of targets can change with time as objects enter and/or leave the camera's field of view, an algorithm is needed that is capable of initiating and terminating targets.

5.4.2.3 Probabilistic data association filter

The probabilistic data association filter (PDAF) and its extension to multiple targets, joint PDAF (JPDAF), solve the same task of measurement identification in a simpler way. In the JPDAF, hypotheses are built for the measurements and targets only for the current scan. In this way the number of hypotheses is additionally reduced but the chance of combinatorial explosion in dense target and clutter scenarios still remains. The joint-probabilistic data-association filter is a suboptimal approximation that has the advantage of fixed finite memory and computations.

5.4.2.4 Multiple hypotheses

In the Multiple hypotheses approach all measurements received at a scan are assigned to initialized targets, new targets or false alarms. A number of hypotheses are generated. Every one supposes a possible assignment scheme between measurements, received in all scans, and the targets - confirmed, new ones or false. Pruning and gating techniques are used to retain the most likely hypotheses and in this way to reduce their number.

5.4.3 Prediction

The choice of the target motion model represents one of the main issues to be addressed in the design of a multi-target tracking system. The most widely used state estimator is the Kalman filter. It has optimal estimation performance measured via

mean-squared error. However, this performance degrades rapidly in the presence of maneuvering targets. The more versatile Interacting Multiple Model filter (IMM) [8] has been proposed to cope with maneuvering targets. It uses different motion models in order to take maneuvers into account.

5.4.3.1 Kalman filter

Kalman filter is a classical optimal estimating algorithm for dynamical linear systems with Gaussian measurements and system noise. The modification of Kalman filter - Extended Kalman filter is developed for non-linear systems. The EKF gives particularly poor performance on highly non-linear functions because only the mean is propagated through the non-linearity. The unscented Kalman filter (UKF) uses a deterministic sampling technique to pick a minimal set of sample points (called sigma points) around the mean.

5.4.3.2 Particle filters

Particle filters, also known as Sequential Monte Carlo methods (SMC), are sophisticated model estimation techniques based on simulation. Particle filters generate a set of samples that approximate the filtering distribution to some degree of accuracy. With sufficient samples, they approach the optimal estimate. Sampling Importance Resampling (SIR), the original particle filtering algorithm, is a very commonly used particle filtering algorithm, which approximates the filtering distribution by a weighted set of P particles.

5.4.3.3 Interacting Multiple Models estimator

A very powerful approach for maneuvering targets tracking is known as the Interacting Multiple Models (IMM) estimator. Other well known estimators, such as Generalized Pseudo-Bayesian (GPB) estimators of different orders, Fixed structure IMM, Variable Structure IMM, and Probabilistic Data Association IMM, are variants of IMM. The most important feature of IMM is that it uses several models in parallel for modeling the estimated system.

5.5 Behavior recognition

After successfully tracking the moving objects in an image sequence, it is necessary to analyze and understand the object behaviors and to determine whether these behaviors are normal or abnormal. Object behavior understanding involves the analysis and recognition of motion patterns, and the production of high-level descriptions of actions and interactions.

Understanding of behaviors may simply be thought of as matching an unknown test sequence with a group of labeled reference sequences representing typical behaviors. A fundamental problem of behavior understanding is to represent these reference behavior sequences (e.g. from training samples) and to devise both training and matching methods for coping with small variations of the feature data within each class of motion patterns.

5.5.1 Behavior representation

Generally, there are two main categories of behavior representation methods: statistical models and formalized reasoning.

Statistical models. A representative statistical model is the Bayesian network model. This model interprets certain events and behaviors by analysis of time sequences and statistical modeling. These methods rest on lower-level recognition based on motion concepts, and do not yet involve high-level concepts, such as events and scenarios, and the relationships between these concepts. These concepts need high-level reasoning based on a large amount of prior knowledge.

Formalized reasoning. Formalized reasoning uses symbol systems to represent behavior patterns, and reasoning methods such as predicate logic to recognize and classify events.

5.5.2 Activity recognition

When considering research aimed at understanding action this thesis focuses on activities or behaviors where the entity to be recognized may be considered as a stochastically predictable sequence of states. These high-level events are composed of primitives (e.g. enter-compartment, sit-on-seat, exit-compartment) linked in a spatio-temporal structure that satisfies particular constraints.

Among previous studies on activity recognition, Ayers and Shah [97] detect actions of people in a room. Bobick and Davis [98] use temporal templates to recognize human activities. Aggarwal and Cai [99] review human activity recognition through application of the Hidden Markov Model and Davis and Shah [100] use finite state machines to model different phases of actions. For a discussion of recent work, see [101].

5.5.2.1 Temporal template matching

In the template matching approach movement data is matched against a predefined set of prototype activity templates. This method to classify human actions is based on the property of different actions to have different motion history patterns. Any

movement cue can therefore be used for template matching. These motion components are then accumulated to form a movement feature vector for a time period called a trajectory. The trajectory is compared to the predefined templates using a matching algorithm.

Template matching approaches differ in:

Motion feature used e.g. optical flow, MEI, MHI. The chosen feature determines the range of actions that can be detected but also how susceptible the method will be to noise.

How to deal with missing data. The easiest method is linear interpolation, but more advanced methods usually take into account properties such as the smoothness of the trajectory or maneuver characteristics of the target.

Matching algorithm. Quantifies the similarity between a trajectory and an activity template and chooses the best match e.g. nearest neighbor matching.

The motion features used by the temporal matching model of Bobick and Davis [98] are (1) a binary motion-energy image (MEI) which represents where motion has occurred in an image sequence, and (2) a motion-history image (MHI) which is a scalar-valued image where intensity is a function of recency of motion. Taken together, the MEI and MHI form a two-component temporal template: a vector-valued image where each pixel represents the motion at that pixel location.

The matching algorithm is rotation (in the image plane), scale, and translation invariant. Given a set of MEIs and MHIs for each view/action combination from training samples, statistical descriptions of the images are computed using Hu moments [102]. For each view of each action a statistical model of the moments (mean and covariance matrix) is generated for both the MEI and MHI. To recognize an input action, the Mahalanobis distance is calculated between the moment description of the input and each of the known actions.

5.5.2.2 Finite-state machines and HMM

A Finite-State Machine (FSM) is a model of behavior composed of a finite number of states, transitions between those states, and actions. The most important feature of a FSM is its state-transition function. The states are used to decide which reference sequence matches with the test sequence. State machine representations of behaviors have been employed in higher level descriptions.

Hidden Markov Models (HMM) in their essence are probabilistic finite state machines [103]. It allows a more sophisticated analysis of data with spatio-temporal variability. The use of HMMs consists of two stages: training and classification. In the training stage, the number of states of a HMM must be specified, and the corresponding state transition and output probabilities are optimized in order that the

generated symbols can correspond to the observed image features of the examples within a specific movement class. In the matching stage, the probability with which a particular HMM generates the test symbol sequence corresponding to the observed image features is computed.

5.5.3 Affect recognition

At the signal level, audio data already contains a wide array of information that can be used in the aggression detection process. Many events emit characteristic sound patterns when they occur (e.g. breaking glass, sound of the brakes when the train stops). Discrete audio event detection algorithms are discussed in [88]. On the other hand, emotion in the speech signal can be good indicators for affective aggression. The features used for emotion analysis of the audio data is based on recent work of Grimm and Kroschel [55]. The acoustic features extracted from the speech signal can be divided into the groups of 9 pitch related features, 5 speaking rate related features, 6 volume related features, and 26 spectral features:

Pitch related features. The pitch is the fundamental frequency (f_0) of an audio waveform. As the dynamics of the pitch over successive frames reveals more then the pitch of one isolated frame, pitch related features commonly encountered in literature are: mean value (f_0 *mean*), standard deviation (f_0 *std*), median (f_0 *median*), minimum (f_0 *min*), and maximum (f_0 *max*), 25% and 75% quantiles (f_0 *quartlow*, f_0 *quartup*), difference between f_0 maximum and minimum (f_0 *range*), difference of quartiles (f_0 *quartrange*).

Speaking rate related features can indicate the amount of stress or even the context of a conversation. Some speaking rate related features are: ratio between the durations of unvoiced and voiced segments, average duration of voiced segments , standard deviation of duration of voiced segments, average duration of unvoiced segments, and standard deviation of duration of voiced segments.

Volume related features. The most widely easy-to-compute frame feature is volume. Volume is a reliable indicator for silence detection, which may help to segment an audio sequence. Volume related features include volume mean, standard deviation, maximum, 25% and 75% quantiles, and difference of quartiles.

Spectral features. One of the most popular set of features used to parameterize the speech is the Mel-Frequency Cepstrum Coefficients (MFCC). Based on the human auditive system model of critical bands, the Mel-frequency scale is linearly spaced at low frequencies (below 1000 Hz) and logarithmically at high frequencies (above 1000 Hz). MFCC features are mean value and standard deviation of 13 Mel-Frequency Cepstral Coefficients.

All features can be extracted directly from the speech signal without the need of an automatic speech recognition (ASR) unit. The pitch related features are based on an

estimation of the pitch contour for voiced segments of the utterance. This is obtained by autocorrelation. The volume-based features are derived from the envelope of the time-domain signal.

We base the emotion recognition on the research of Grimm and Kroschel [55], in which 499 sentences of emotional speech of guests (both male and female) in a German TV talk-show were analyzed and provided the correlation values between the audio features and the emotion components Valence, Activation, and Dominance of the speakers. For each of the entities Valence, Activation, and Dominance, the scale of the classifier output ranges from 1 to 5 with 3 being the neutral value. The orientation of the scales is as follows: For Valence, 1 corresponds to a very positive, and 5 to a very negative emotion. For Activation, 1 corresponds to a very excited, and 5 to a very calm speaker. For Dominance, 1 corresponds to a very weak, and 5 to very strong speaker.

5.6 Discussion and conclusions

This chapter has discussed all the stages in the workflow of an automated surveillance system and has presented an overview of possible algorithms that can accomplish the tasks in each stage.

Tracking in the train compartment serves the purpose of recognizing activity of the people in the compartment. A commonly used tracking algorithm is the Kalman filter. However, given its characteristics, the train compartment does not work well with linear motion models such as the Kalman filter. The motions of people in the compartment tend to be too versatile for this. In addition, a lot of occlusion is expected to occur. Therefore, another solution is needed.

The best option is to make use of the characteristics of the compartment to support the tracking algorithm. In general, walking motions along the length of the compartment occur along the corridor. Motion along the width of the compartment occur when people try to sit on a seat. Multiple cameras with overlapping views can be used to enhance the results by track to track association.

When the train in is transit, not much motion is expected along the corridor. In this case, nearest neighbor matching is sufficient for data association. When many people are expected to enter or exit the train, multiple hypothesis association or a joint probabilistic data association filter may be more appropriate.

Behavior recognition analyzes behaviors and interactions between objects to decide the activity that is taking place (e.g. whether people are getting on or getting off). Behavior understanding is complex, as the same behavior may have several different meanings depending upon the scene and task context in which it is performed. Several possible solutions to this problem will be explored in Chapter 7 to Chapter 9.

Chapter 6

Computational framework and architecture

This chapter discusses a general Multi-sensor Data Fusion framework. It contains a motivation for the need of a separate framework, requirements, design choices and used techniques. It also contains a discussion about the communication infrastructure between components, the capabilities of the communication infrastructure and its main strengths and weaknesses.

6.1 Motivation and design goals

As discussed earlier, aggression detection is such a complex task that there exists no single algorithm that can accomplish the task on its own. Many algorithms will have to combine strengths to yield the final result. As a result, there is a need for an overall framework to facilitate this. Very important functions such as communication, resolution needs, resource management and exception handling rely on a good overall framework.

In recent years, application-specific frameworks based on emerging technologies such as peer-to-peer environments, message based middleware and service-oriented approaches have been developed. Most use XML (or related technologies) to support data manipulation and to facilitate component interaction. XMIDDLE, for example, is a mobile computing middleware using XML and a Document Type Definition (DTD) or Schema to enhance data with typed structure and a Document Object Model (DOM) to support data manipulation. It further uses XPath syntax to address the data stored in a hierarchical tree-structured representation. Other examples are Jini, Vinci, HP Web Services Platform and e-Speak.

The implementation of a framework based on the JDL model discussed in Section 3.3.2. The framework can function as a testbed for different fusion algorithms as separate

algorithms can be easily removed or replaced. The architecture of the system will show many similarities with distributed systems. The basic idea is to abstract out the process of doing the fusion (feature extraction, classification, etc.) from the fusion methods.

6.2 Requirements

From the discussion above, the goal of the framework can be summarized:

The data fusion software framework should provide mechanisms to build active software modules and enforces on these modules a set of rules that govern the interaction among the modules. The data fusion framework further deals with distribution of information to the right place at the right time (regardless of the actual software module).

In addition, the framework should support loose coupling of modules and the modules should be configurable from the framework. The modular components fulfill different tasks in the surveillance process and can be characterized as data acquisition units, processing units, communication components and directory service components.

Below is a list of additional requirements that are relevant to the framework.

Interest specification. There must be a mechanism that allows a module to indicate in what information (from other modules) it is interested and when. The application must be able to query the framework for information or to subscribe (with a set of conditions) to certain information.

Separation of concerns. Data that is used by a module might come directly from sensors (after interpretation) or there might be several layers of abstraction. The use of these multiple layers and the use of sensors should be transparent. Therefore, all components need to have a common external interface such that a framework can treat them in a similar fashion.

Transparent distributed communications. The fact that communication is distributed should be transparent to modules. This relieves the designer of having to build a communication framework. Furthermore the framework should support different communication mechanisms (e.g. RMI, Corba).

Storage. History must be stored such that the application, but also the context architecture itself, can use it to establish trends and predict future values.

Resource discovery. There should be a resource mechanism that is responsible for finding any applicable components (keeping a list of available components) and for providing the application with ways to access them. New services and

resources should be detected and their capabilities should be available to other services.

Data descriptions. In general JDL terminology will be used to describe data and modules. Other issues that need to be defined are the values that data can have, the services that the framework delivers etc. A common terminology makes it easier to develop modules, since it can be used to define the interface between modules, and the framework.

6.3 Framework overview

To support distributed architectures, the framework employs a communication protocol based on XML message passing. Architectures built on this framework have to be modular. The modules are classified into five types, each performing processing jobs according to the five levels of processing defined in the JDL model. The classification of a certain fusion algorithm must be considered as a tag describing the kind of processing done in the algorithm and by no means restrict the priorities of the algorithm. The framework is organized around the directory modules, the central point of communication and distribution of data between modules. The distribution mechanism is based on the Observer pattern.

6.3.1 Data distribution

Figure 6.1 shows the basic idea of the data acquisition and distribution system. The figure shows how the data acquisition aspect is separated from the data processing aspect of the system. The data acquisition part gets sensor information and notifies audio processing, video processing and/or combined data processing components if the data contains audio signals, video signals or both respectively. Multiple data processing components are allowed to register to the data acquisition component at run time.

The distribution of data from acquisition to processing follows the Observer design pattern. The Observer pattern [104], also known as Dependents or Publisher-subscriber pattern, defines a one-to-many dependency between objects so that when one object changes state all its dependents are notified and updated automatically. The Observer pattern can be used in situations when an object should be able to notify other objects without making assumptions about who these objects are.

In the data fusion framework only part of the functionality contained in the Observer pattern is used: only the object at the one side in the one-to-many relationship (the subject) is able to change state. The type of data distributed is not limited to audio and video (as in the data acquisition discussion above), also intermediate results from processing components may be distributed to other processing components by this mechanism.

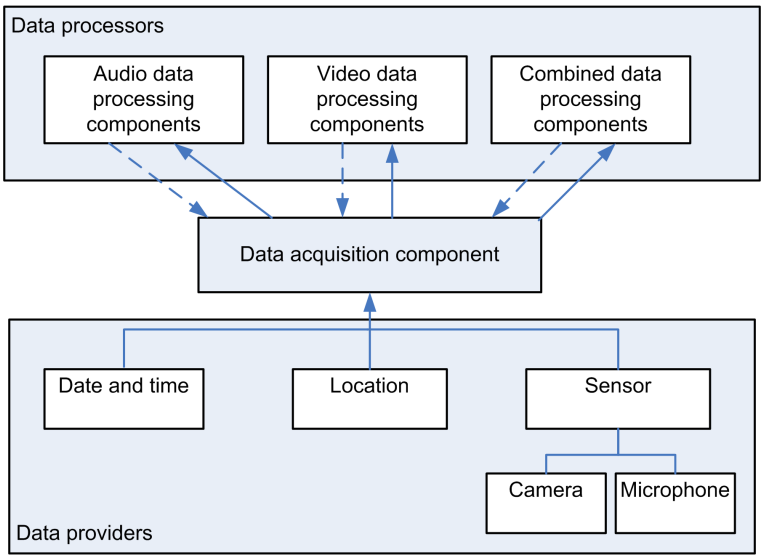


Figure 6.1: Data acquisition and processing system using the Observer pattern.

6.3.1.1 Structure

The key objects in the Observer pattern are subject and observer (see Figure 6.2). A subject may have any number of dependent observers. All observers are notified whenever the subject undergoes a change in state. In response, each observer will query the subject to synchronize its state with the subject's state.

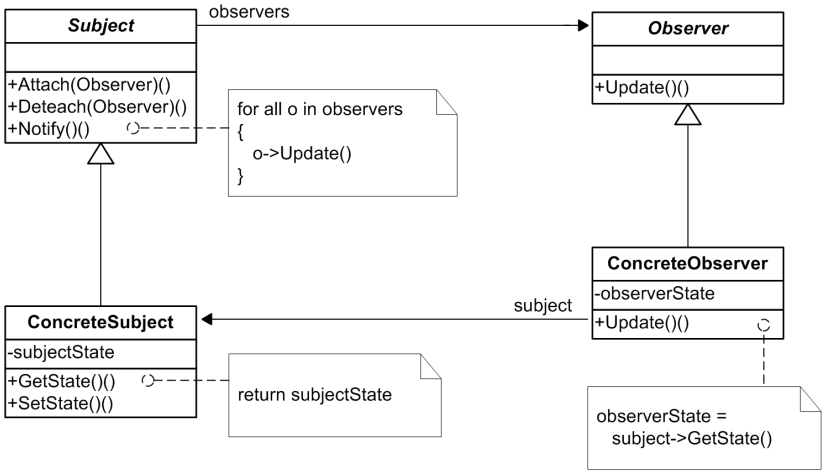


Figure 6.2: Structure of the Observer pattern.

Subject. The subject is the publisher of notifications. It sends out these notifications without having to know who its observers are. Any number of observers can subscribe to receive notifications.

ConcreteSubject. Stores the state of interest to ConcreteObserver objects and sends a notification to its observers when its state changes.

Observer. Defines an updating interface for objects that should be notified of changes in a subject.

ConcreteObserver. Maintains a reference to a ConcreteSubject object. Stores a state that should stay consistent with the subject's. Implements the Observer updating interface to keep its state consistent with the subject's.

6.3.1.2 Collaborations

Figure 6.3 shows the sequence diagram of the interactions in the Observer pattern. It illustrates the collaborations between a subject and two observers. In the diagram, Notify() is not always called by the subject. It can be called by an observer or by another kind of object entirely.

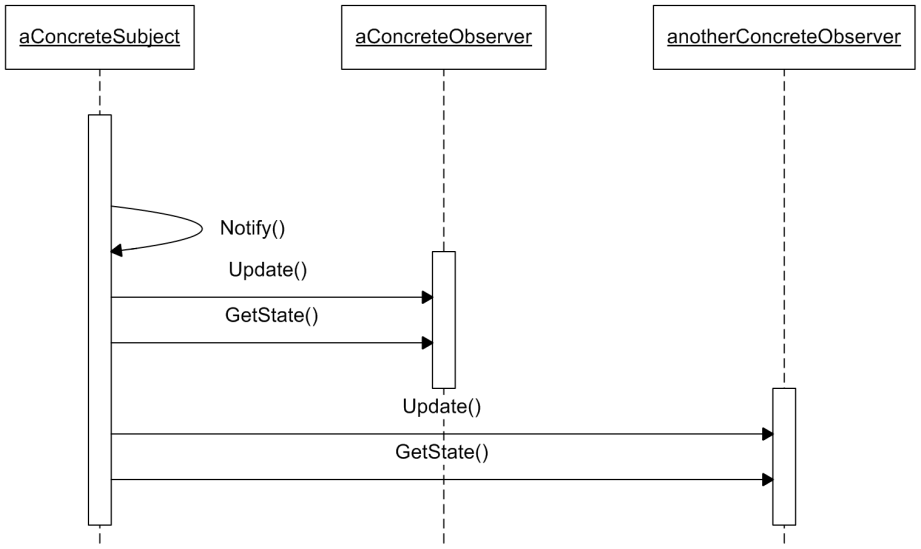


Figure 6.3: Sequence diagram of the Observer pattern.

ConcreteSubject notifies its observers whenever a change occurs that could make its observers' states inconsistent with its own. After being informed of a change in the concrete subject, a ConcreteObserver object may query the subject for information. ConcreteObserver uses this information to reconcile its state with that of the subject.

6.3.2 Communication between components

Three types of messages can be distinguished in the communication between components. (1) Service messages for service announcement, discovery and subscription, (2) data messages containing products of fusion nodes, that need to be transported to other fusion nodes for further processing and (3) feedback messages between nodes. The loose coupling among data fusion components is achieved by employing two architectural constraints in the communication between components.

Interface. A small set of simple interfaces in which only generic semantics are encoded is available to all data fusion modules. The interface keeps the complex logic of the fusion modules hidden, while providing a standardized way for the module to interact with the framework and other modules.

Messaging. Descriptive messages constrained by an extensible schema are delivered through the interfaces. No, or only minimal, system behavior is prescribed by messages. The schema limits the vocabulary and structure of messages while the extensibility of the schema allows new versions of services to be introduced without breaking existing services.

The interface has to abide to the following rules in order to meet the requirements described earlier (loose coupling, extensibility, etc):

- The messages must be descriptive, rather than instructive, because the service provider is responsible for solving the problem.
- The messages must be written in a format, structure, and vocabulary that is understood by all parties. Limiting the vocabulary and structure of messages is a necessity for any efficient communication. The more restricted a message is, the easier it is to understand the message, although it comes at the expense of reduced extensibility.
- Extensibility is vitally important. It is not difficult to understand why. The world is an ever-changing place and so is any environment in which a software system lives. Those changes demand corresponding changes in the software system, service consumers, providers, and the messages they exchange.

6.3.3 Service discovery

Because observers have no knowledge of each other's presence, we must create a mechanism that allows processing nodes to discover each other. A special directory node allows processing nodes to publish and discover available services. The

publishing method is based on the Universal Description, Discovery and Integration (UDDI) protocol[1].

UDDI is one of the major building blocks required for Web services, it creates an interoperable platform that enables applications to dynamically find and use services over a network. UDDI also allows operational registries to be maintained for different purposes in different contexts.

UDDI takes advantage of Worldwide Web Consortium (W3C) and Internet Engineering Task Force (IETF) standards such as Extensible Markup Language (XML), and HTTP and Domain Name System (DNS) protocols. Additionally, cross-platform programming features are addressed by adopting Simple Object Access Protocol (SOAP). The UDDI protocol is the building block that will enable fusion components to dynamically find and transact with one another.

6.3.4 Memory

In computer science, traditionally, memory has been used to store objects containing anything from plain text to data tables to complex applications. While all these objects together constitute the memory of the system, there is no obvious connection between the different objects i.e. the objects are stored independently. Biological systems on the other hand use as memory a system of neural connections that link neurons at the lowest levels of data. In addition, neural connections can have different strengths identifying the relative strength of each link.

Our recognition system uses a memory structure which is a hybrid of computer science and biological memory. We maintain the objects of computer science, which may contain low level features till high level objects, situations and scenarios but augment them by adding links between these objects that fully characterize each object within the memory structure. In this context, objects are called concepts and links between concepts are called associations.

The advantage of such a memory structure is that it enables cross-referencing of any concept contained in memory. In addition, this way of structuring the memory complies fully with the ontology and the ontological reasoning mechanism. A further convenience is that, from a technical perspective, XML and XML related technologies such as XML Schema, XQuery and OWL readily provide the capabilities to create and represent concepts and associations.

If we represent each XML element as a memory concept, then XML already provides a way to associate concepts with each other. With the addition of a logic or valued parameter that identifies the strength of a particular association all requirements to create the memory structure are essentially met.

[1] http://www.uddi.org

The memory component consists of this memory structure, supported by a search mechanism and an ontological grounding. The search mechanism is provided by XQuery and the ontological grounding is made of an XML Schema to govern the grammar and vocabulary used in the memory and an OWL ontology to guide the high level interpretation of the memory.

6.4 System architecture

At the highest level, the aggression detection system architecture consists of a number of detection units (one per train compartment) that are mutually connected and maintain a communication link to a central control point at the Utrecht train station. The connection with Utrecht is wireless, and any detection unit can send warning messages about upcoming or ongoing aggression in a compartment. Human operators at the control point can use the connection to request live data from the videos and microphones in the compartment to determine the actions to take.

6.4.1 Intercompartment architecture

Normally, detection units operate as standalone detectors, but in times of crisis neighboring units can exchange information to deal with the situation. This means that each detection unit is aware of its two neighbors. During an emergency (e.g. fire) a detection unit can inform (information push) its neighboring units about it. Alternatively, when trouble enters the compartment (e.g. hooligans) a unit can request information (information pull) of the group from the originating compartment.

6.4.2 Intracompartment architecture

The architecture of a single detection unit is based on loosely coupled data fusion components (see Figure 6.4). The architecture is organized around the Directory component, the central point for communication between component. The Directory component handles the distribution of all data. The communication protocol used is founded on XML. The distribution mechanism is based on the Observer pattern discussed earlier. The data processing algorithms are classified by their fusion level.

The architecture shows strong similarities with Service Oriented Architectures (SOA) used in web services [105]. SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents. A service is a unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by software agents on behalf of their owners. The SOA defines how the different modules communicate with each other.

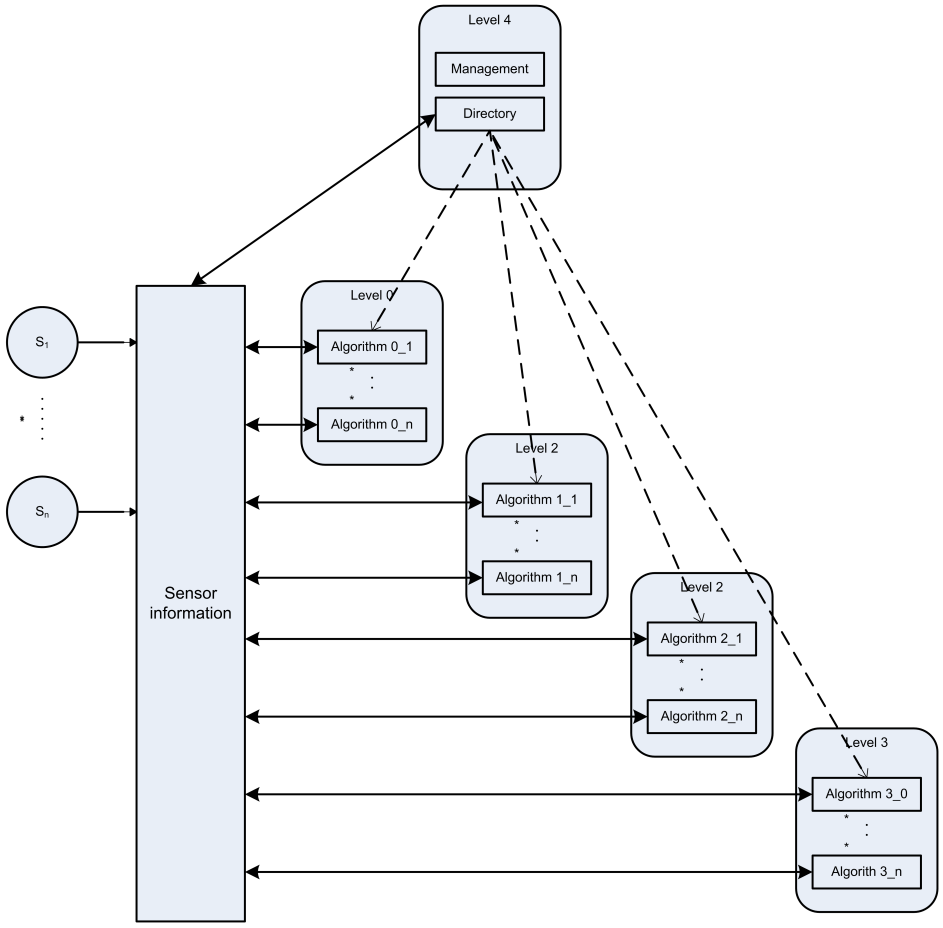


Figure 6.4: Overview of the intracompartment architecture. The architecture is based on loosely coupled data fusion components around a Directory component that handles the distribution of data among the data fusion components. The data fusion components are classified by their fusion level.

Because of the loose coupling the system architecture is not fixed. So, instead of presenting the components in the architecture. We will describe the classes of components. Each component is called a node. There are three types of nodes: sensor nodes, processing nodes and fission nodes.

6.4.3 Sensor nodes

The first class of components are nearest to the sensor inputs. The handling of sensor input is done by sensor nodes, these nodes differ from normal processing nodes in that they can only subscribe to a sensor (which can be streaming data from hardware

in real time or sensor data stored in the file system during testing). The problems associated with handling sensor data can be divided into the following:

- Storage: sensor data, if stored raw, require a significant amount of memory. Considering the multimedia data that we are dealing with, the storage space becomes immense. This results in problems in memory management and search and retrieval. One possible solution to reduce memory usage is by storing features of the data instead of the raw data itself. This however might result in loss of detail.
- Reliability: each sensor node, in essence, is a service provider for processing nodes deeper in the data fusion tree. As such, it is important that sensor nodes maintain a high level of reliability or, in case the sensor fails, are provided with the means to notify observers that a failure has occurred. And similarly, when a sensor is online again, that the service is available again. The initial measure of reliability is reported when a sensor node registers itself on the system. This measure of reliability (and accuracy) is called the service level agreement (SLA).
- Processing: sensor nodes require a certain amount of processing power in order to read and preprocess incoming data from the sensors so as to provide the level of service stated in the SLA. In the framework, sensor nodes continuously monitor their performance and lower or increase their SLA if necessary.

The SLA is an important attribute of a sensor node (and for other nodes as well) as it serves as a means to communicate reliability and accuracy measures to other nodes and acts as measure for the sensor node to assess its own performance. Thus a SLA requires a unified language for the sensor node to clearly and unambiguously state its capabilities and a protocol for other nodes to interpret the capabilities stated in the SLA.

6.4.4 Processing nodes

Processing nodes are data fusion components where the actual data fusion takes place. It receives sensor information from sensor nodes (or other processing nodes) and provides its results to all nodes that have subscribed to it. This results in a structure consisting of a number of processing nodes that finally produce the output. Each processing node does a processing step on data (be it low level features or high level object or situations) and provides output to other data fusion nodes deeper in the fusion structure.

A processing node consists of three parts:

- observer: takes care of the input data exchange with nodes earlier in the fusion structure.

- processing: where the actual processing takes place.
- subject: takes care of the output data exchange with nodes later in the fusion structure.

6.4.5 Fission nodes

In our solution, the human operator is still the final reviewer and decision maker. The time required for the operator to review information can lead to costly delays and errors. As a result, it is important for the system to present the detection results in a clear and consistent manner. It is the task of the fission component to collect data from all components and summarize it for the human operator. In the proposed solution, the human operator will be presented with a panel containing colored bars indicating the (maximum) danger for each type of aggression in all the train compartments (Figure 6.5). In essence, the panel summarizes the aggression status of the railway system. The operator can set a threshold for each bar. If the danger surpasses this threshold, an sound alarm will be issued by the fission component. At the alarm sound, the human operator can inspect the alarm message further by examining which compartment the alarm is coming from.

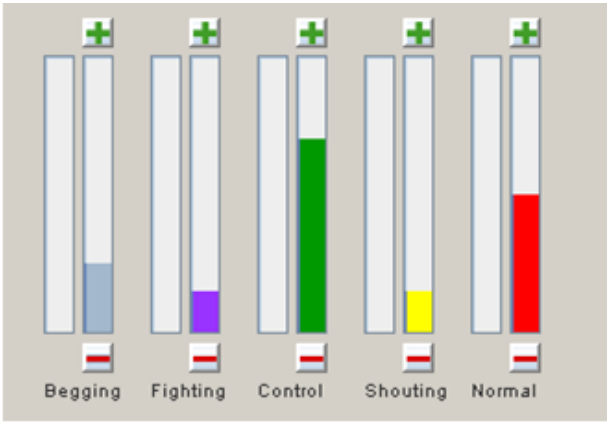


Figure 6.5: Proposed screenshot of the fission component, for each kind of aggression the human operator can set a threshold for a sound alarm.

6.5 Discussion

The Subject and Observer architecture described above enables an abstract coupling between fusion nodes. All a subject knows is that it has a list of observers, each conforming to the simple interface of the abstract Observer class. The subject does not know the concrete class of any observer. Thus the coupling between subjects and observers is abstract and minimal.

Because Subject and Observer are not tightly coupled, they can belong to different layers of abstraction in a system. A lower-level subject can communicate and inform a higher-level observer, thereby keeping the system's JDL layering model intact. If Subject and Observer are lumped together, then the resulting object must either span two layers (and violate the layering), or it must be forced to live in one layer or the other (which might compromise the layering abstraction).

In addition, the communication model supports broadcast communication. Unlike an ordinary request, the notification that a subject sends need not specify its receiver. The notification is broadcasted automatically to all interested objects that are subscribed to it. This gives the freedom to add and remove observers at any time.

Because observers have no knowledge of each other's presence, there exists a special directory node to allow processing nodes to publish and discover available services. The publishing method is based on the standard XML based UDDI specification.

A consequence of this architecture is that Observers can be blind to the ultimate cost of changing the subject. A seemingly simple operation on the subject may cause a cascade of updates to observers and their dependent objects. Moreover, dependency criteria that are not well-defined or maintained usually lead to spurious updates, which can be hard to track down. This problem is aggravated by the fact that the simple update protocol provides no details on what changed in the subject. Without additional protocol to help observers discover what changed, they may be forced to work hard to deduce the changes.

Chapter 7

A rule-based approach to reasoning about aggression

The previous chapters of this thesis have presented a number of techniques to detect different parameters or objects in the train compartment. In addition we have created an ontology that specifies the meaning and relationship of relevant concepts to aggression (and to each other). The final step is to build a system that combines the results of the detection algorithms with the relationships to reach a decision about the aggression situation.

7.1 Model

In the rule-based approach a set of rules describes which conclusion to draw given a certain input. Rules are usually extracted from interviews with domain experts in a so-called knowledge acquisition phase. In general, a rule consists of a condition and consequence as shown in Listing 7.1. The condition is a logical statement and the consequence specifies the hypothesis if the condition is true. The example below illustrates a rule in the language CLIPS.

```
(defrule script-example
   (perceive (feature f1))   # Condition 1
   (perceive (feature fn))   # Condition N
=>
   (assert (hypothesis H))   # Consequence
)
```

Listing 7.1: This rule proposes a hypothesis based on perceived features

7.1.1 Theory

The idea for the rule-based approach is inspired by the schema theory [62]. In this theory, schemas are cognitive structures that link declarative and procedural knowledge together in patterns that facilitate comprehension of behavior within a context. The declarative part is comprised of object classes together with associated features and arranged in hierarchies in space and/or time [106]. Procedural knowledge for the understanding and enacting of behavioral patterns and routines is encoded in scripts. A classic example of a script is Schank and Abelson's [107] restaurant script that includes a structure of elements for entering a restaurant, ordering, eating etc.

Following the schema theory, an aggression script contains information about the concepts and events that have occurred when the aggressive scenario was encountered. Scripts are stimulated if the events in the script are really being observed in the environment. This is illustrated in Figure 7.1: in the figure, scripts are represented as poles. When a concept is observed, it ends up on a pole if the concept supports the script. This can be any number of poles, depending only on how many scripts contain the particular concept. Each concept adds a certain weight on the pole depending on the salience of the observation to the script. When a script reaches a threshold the script is triggered and the result is that the person recognizes the scenario.

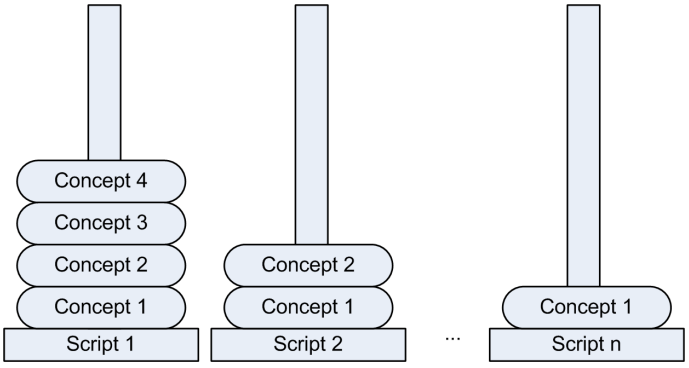


Figure 7.1: Scripts can be seen as stacks upon which concepts can be put.

A key idea in the theory is that some events can be more salient for a script than others (in a car accident, a damaged car is more salient than a police officer although both are things that one would expect to see) and that events need not necessarily occur in a strict sequence. Furthermore, the existence of a threshold ensures that not all events in the script need to be observed before the situation can be recognized.

7.1.2 Competing scripts

The aggression detection approach modeled after the scripts in the schema theory is called the competing scripts approach. The competing scripts approach is essentially a rule-based reasoning scheme in which (aggressive) scenarios are modeled as

scripts. Scripts are represented as a sequence of events. In the beginning, when a small number of events are observed, many competing scripts may be in the reasoning system. But by further observation and the detection of events over time, we arrive at a final hypothesis. This process is illustrated in Figure 7.2.

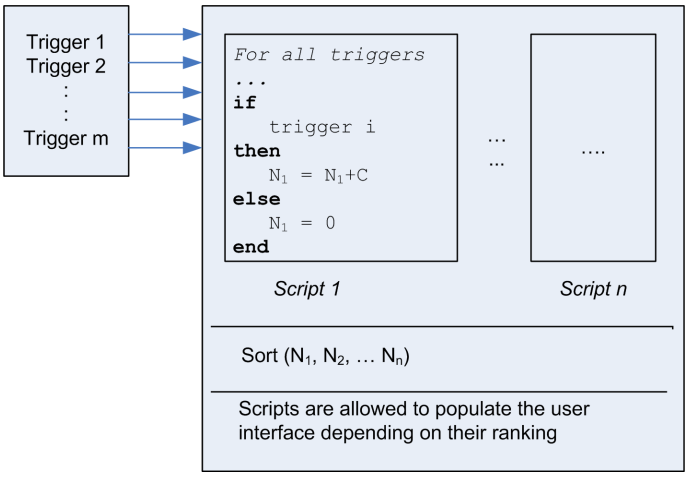


Figure 7.2: Overview of the rule-based approach. As new data enters the system in the form of triggers, competing scripts get re-evaluated until a final hypothesis is decided.

A model of the environment enables us to perform more sophisticated inferencing. A model must contain a static model (issues are granularity, which objects to represent, which relationships to represent, etc.), it must be scalable and it must be able to cope with contradictory facts. If multiple sensors are combined to obtain information, this means that the model must be able to deal with contradictory input.

7.2 Designing the rule base

To construct the rule base of the rule-based system, the expert knowledge obtained from interviews with human experts and results from analyzing aggression scripts, scenarios and surveillance videos, have been formalized. The first stage in creating the rule base is to create a list of possible aggressive scenarios that can occur in the train compartment. We have defined more than 40 scenarios (we refer to Chapter 10 for more information on how the scenarios were created and validated). Table 7.1 shows the list of hypothesis groups. To preserve space, similar scenarios (e.g. same scenario with different outcomes) have been grouped into one hypothesis. Others are less ethical scenarios (e.g. gun fire, vandalism, graffiti) have been omitted from the list because they cannot be performed in experiments.

Next a list of high-level concepts, that have been observed in the scenarios, are identified. This list is a subset of the observable concepts in the aggression ontology in Chapter 4. The list includes concepts such as “standing up” or “screaming”. For each of these high level concepts, there are many (combinations) low level feature

Table 7.1: List of hypothesis groups, similar scenarios have been grouped into one.

Hypothesis code	Description
H_1	Verbal and physical harassment (aggressor vs. victim)
H_2	Lost wallet and ensuing confusion
H_3	Medical emergency and call for help
H_4	Entering or exiting full or empty train
H_5	Confrontation with fighting verbal aggression (two aggressors)
H_6	Ticket checking
H_7	Drunk, hooligan and noisy crowds
H_8	Barricading/blocking entrance and exits

detection algorithms associated with it (as discussed in the perception model in Section 4.3.3). Cross referencing the concepts with the detectable features, results in the list of possible features (or observations) that show support for the concepts e.g. for "screaming" the observations would be "high volume", "sound in speech frequency" and "location of sound corresponds to the location of a face". This way, the features are related (via the high level concepts) to the scenarios/hypotheses. The strength or salience of the relationship is specified in an influence matrix (discussed below).

7.2.1 Influence matrix

The influence matrix consists of two parts: the main influence matrix and the link matrix. In the first part, the matrix indicates for each feature how much the feature contributes to the likelihood of a hypothesis. This contribution is expressed in a 5 point saliency scale: each feature affects the hypothesis in either a very positive (++), positive (+), negative (-), very negative (- -) or neutral way. In this valuation system, very positive means that the feature was extracted from that scenario. A positive value indicates that the feature was not present in the media but could be expected with reasonable probability. A negative value indicates that the feature is not expected to occur in the scenario and a very negative value indicates that the scenario is close to "mutually exclusive" with the feature (highly unlikely to occur). A neutral rating means that the feature and hypothesis are independent. A list of features and their description is shown in Table 7.2 while Table 7.3 shows the influence matrix for the first column of features.

The link matrix specifies the correlation between pairs of features (either negative or positive). As observations are rarely independent of each other, the observation of one feature can give expectations about the possible observation of the correlated features. This information can be used in various ways. Features can be causally related: the effect event is observed but not the cause event (or the other way around). Since both events are strongly correlated in the influence matrix, the cause of certain

Table 7.2: List of aggression related observable cues extracted from recordings and interviews with experts.

nr	Description	nr.	Description
f_1	Physical contact	f_{33}	Disgusted face
f_2	Staring	f_{34}	Grabbing other person
f_3	Directed attention (all passengers)	f_{35}	Person falls
f_4	Leaning over	f_{36}	People alerted
f_5	Uncomfortable body motion	f_{37}	Concerned/desperate sound
f_6	Relaxed body language	f_{38}	Shaking head
f_7	Pushing other person away	f_{39}	Mobile phone call
f_8	Uncomfortable face	f_{40}	Grabbing item from other person
f_9	Pointing finger	f_{41}	Panic body language
f_{10}	Assertive body language	f_{42}	Impatience
f_{11}	Innocence gestures	f_{43}	Throwing items
f_{12}	Excusing gestures	f_{44}	Talking
f_{13}	Aggressive pushes	f_{45}	Happy face
f_{14}	Wrestle/fight	f_{46}	Person walking fast
f_{15}	Person walking away	f_{47}	Person reaching for luggage
f_{16}	Person returning	f_{48}	Standing people
f_{17}	Person running	f_{49}	Shouting
f_{18}	Person looking around rapidly	f_{50}	Rapid arm movement
f_{19}	Person leaning over others	f_{51}	Person approaching
f_{20}	Person touching/feeling around	f_{52}	Conductor present
f_{21}	Passengers looking around	f_{53}	Small arm movements
f_{22}	Crouching	f_{54}	Giving item
f_{23}	Person searching in bags	f_{55}	Conductor inspecting item
f_{24}	Person pulling on item	f_{56}	Person out of balance
f_{25}	Angry face	f_{57}	Person holding knife/gun
f_{26}	Showing items	f_{58}	Person holding bottle/cigarette
f_{27}	Unhappy face	f_{59}	Person drinking
f_{28}	Unpleasant gestures	f_{60}	Dropping luggage
f_{29}	Indecision gestures	f_{61}	Feet on bench
f_{30}	Throwing item	f_{62}	Singing/laughing
f_{31}	“I give up” gestures	f_{63}	Crowded
f_{32}	Cursing		

events can be inferred or the effect of certain events can be anticipated (e.g. when there is fire we can expect smoke). On the other hand, simultaneous observation of two negatively correlated features (e.g. singing in an empty train) may indicate something unusual. Furthermore, the same event may be observed by sensors of different modalities e.g. “explosion in video”, “loud bang in sound”, the influence matrix can be used to relate concept from different modalities.

Correlation between features is used in the rule-based approach in the generation module to automatically fill in missing observations (see Section 7.3). The concept of correlated features is taken a step further in the emergent based approach in Chapter 9 where we use the correlation as a factor for the link strength of features. In this

Table 7.3: Influence matrix.

Feature	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8
f_1	++	++	++	++	++		+	++
f_2	++		+	-				
f_3	++	++	++		++	++	++	
f_4	++	+	++	-	+	+	+	
f_5	++							
f_6	++		-					
f_7	++		+	-				++
f_8	++	+	+			+	+	++
f_9	++	++	++		++	++	++	
f_{10}	++							
f_{11}	++		+	-		+		
f_{12}	++	+						
f_{13}	++		-	-				++
f_{14}	++		-	–		-		++
f_{15}	+	++			+	++	+	-
f_{16}		++						
f_{17}		++	+			+		-
f_{18}		++	++			+		
f_{19}	+	++	+	-				
f_{20}		++						
f_{21}		++	+					
f_{22}		++			–			-
f_{23}		++		++				
f_{24}		++						
f_{25}	+	++	-	-	+			++
f_{26}	-	++						
f_{27}	+	++	+		+			+
f_{28}	+	++						
f_{29}	+	++			+	+		
f_{30}		++					+	
f_{31}	+	++			+			
f_{32}		++						

approach, related features are automatically clustered together based on their link strength. We refer to Chapter 9 for an example of the influence correlation matrix.

7.2.2 Specifying scripts

A script is made of two parts: (1) a list of events and (2) a scenario specification. The events list contains a subset of all the possible events in all the scripts. The scenario specification can be viewed as a knowledge base containing a priori information about the scenario. It is essentially a set of rules. To make the analogy with linguistics, the events would represent the vocabulary i.e. the words that can be used to construct a sentence. While the scenario would specify how the words can be

rearranged while still retaining the same meaning. In addition, the scenario would specify how the sentence is to be interpreted.

A script can be specified by writing the sequence of events and their consequences (e.g. information for the user interface module to generate the appropriate messages specifying the type of aggression in the compartment) as rules. An expert system's inference engine controls the application of these rules. CLIPS[1] was used as the expert system shell as it provides the capabilities needed to develop the inference engine that is used for the script selection task.

7.3 Implementation overview

As mentioned earlier, the CLIPS expert system shell [108] was used to provide the functionality needed for inference for the script selection task. The rule-based detection model described above is integrated into the aggression detection framework described in Chapter 6. To be more precise, the inference model is implemented as a fusion node shown in Figure 7.3. The generation module is implemented as a separate node, it examines the output of the inference node and suggests missing observations as input to the inference as described in Section 7.2.1.

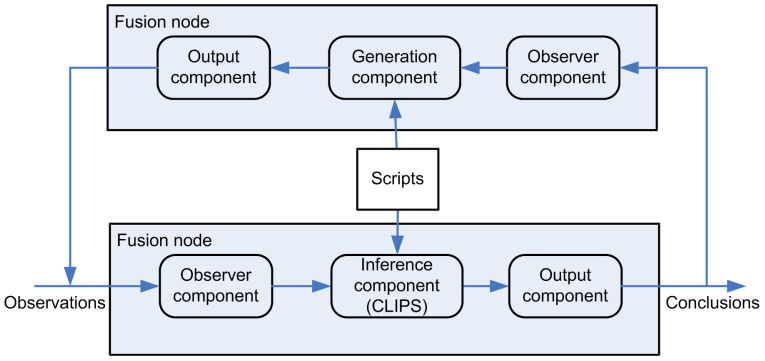


Figure 7.3: Rule-based aggression detection fits as fusion nodes in the software framework.

Observer component. The observer component is responsible for subscribing to information providers. The provider can be the event/object detection algorithms described in Chapter 5

Inference component. Given incoming observations from the observer component, the rule-based inference component determines the score for each competing script and selects the most probable script. The inference component is a wrapper around the CLIPS library.

Generation component. The flexibility of scripts is that they can account for observations that have been missed by the system (e.g. occlusion). As long as the

[1] http://clipsrules.sourceforge.net/

```
do (at each time step)
        for (each hypothesis)
                decay some of this hypothesis' support
        end for
        if (observed feature)
                for (each hypothesis)
                        update this hypothesis' support
                end for
        end if
end do
```

Listing 7.2: Program flow for the rule-based model.

observations that have been done provide enough evidence for a certain scenario, a missing observation can be automatically filled in.

Output component. The output module collects the analysis results from the inference engine and notifies the framework about it. Other modules such as a user interface described in Section 6.4.5 can subscribe to this information to present it to the user (e.g. degree and type of aggression).

7.3.1 Program flow

In the real-time 'run' of the rule-based system, a steady stream of features enters the system. This results in the calculation of all plausibilities for all hypotheses to be assessed using the influence matrix. As a feature in the stream is processed (i.e. is currently observed) the support of all hypotheses it contributes to is incremented with the respective contribution values. The program flow for this model specifies how the system should update its hypothesis support during a 'run' of the incoming feature stream. This flow forms the basis of the rule-based system and takes the form shown in Listing 7.2.

7.3.2 Hypothesis evolution in time

As the train compartment is observed over time a steady stream of observed features enters the system. Obviously, the most recent observations should have precedence over older ones as a wrong conclusion would be drawn if old observations of peace and calm intervened with newer observations of fighting and violence. In order to limit the influence of old observations a saliency decay function is implemented as follows.

Whenever a feature enters the system, a timestamp (t_s) is attached to it (if the feature is already in the system, t_s of the old observation is updated). When the support for a hypothesis is calculated, the saliency value of each feature is adjusted with a value depending on how long ago the feature was last observed:

$$s_{t+1} = s_t - \delta(t - t_s), \tag{7.1}$$

where s_t is the salience at time t and δ is a fixed decay factor. If the saliency reaches zero, the feature is removed from the system. The update function is called on each time step and gradually decreases the likelihood of the hypothesis if no new features are observed. Currently, the decay factor is fixed for all features. This could be easily modified to separate factors for each observable feature.

7.4 Evaluation

This section presents the results obtained after testing the developed rule-based system. An example of a single test run will be dissected, coupled with a reflection on its results. The internal workings of the rule-based system are not that complicated. Therefore, extensive testing of the software was quite straightforward. The application was designed in such a fashion as to output all its processing steps, allowing clear-cut evaluation and validation of its decisions. In practice validating the application entails running the system, while a test user is providing the required input (either through the console or through the reporting tool discussed in Section 11.5.1), and manually checking whether or not the console output matches the desired output. Because of the algorithm's simplicity, a human is very well capable of performing this final verification step. In a similar fashion, any unexpected and undesired results were easily mapped onto flaws in the underlying programming code, which were subsequently fixed. Having discussed the testing and validation process from a general point of view, it is now time to examine a specific practical example of a single run of the expert system.

7.4.1 Evaluation process

The CLIPS program commences by providing the user with a brief introduction, followed by a listing of all the features in its database. Next, the user is allowed to input features into the system. The features indicated in this manner will consequently be activated. As a confirmation, the application lists the feature descriptions when they are set as active. The system will ignore any invalid feature. After a new feature is received, the hypotheses are processed sequentially. During this process, each considered hypothesis is output with all its properties, including its positive and negative features and which one of these are currently activated. Finally, its total score in the present circumstances is displayed. Since all its features are output, including their active or inactive status, the scoring process of a scenario is totally transparent to the user. As a final step, the application provides a list of the scores of the scenarios' hypotheses and determines the dominant hypothesis. The result of this step is again communicated to the user.

```
Activated feature 1  (Agitated voice)
Activated feature 20 (Person invading personal space)
Activated feature 9  (Path is crowded)
Activated feature 5  (Loud talking)
Activated feature 7  (Normal voice [multiple])
Activated feature 16 (People pushing/shoving each other)
Activated feature 25 (Person moving towards other person)
Activated feature 28 (Shouting)
Activated feature 19 (Complaining gesturing)
Activated feature 17 (Person calling on cell phone)
Activated feature 29 (Shouting [multiple])

CHECKING FOR HIGH-LEVEL FEATURE ACTIVATIONS
High-level feature 'People are complaining/annoyed' 10) was activated.
All of its lower-level features were previously activated:
19 Person gesturing in a complaining fashion
7 Normal voice [multiple]

High-level feature 'People are in a quarrel' (12) was activated.
All of its lower-level features were previously activated:
25 Person moving towards other person
16 People pushing/shoving each other

SELECTING SET OF HYPOTHESES
Regarding hypothesis 'H5 - Noisy' with the features:
29 Shouting [multiple] (active)
All of the hypothesis salient features are active.

Regarding hypothesis 'H4 - Mobile Phone' with the features:
5 Loud talking (active)
17 Person calling on cell phone (active)
All of the hypothesis salient features are active.

Regarding hypothesis 'H3 - Medical Emergency' with the features:
22 Person is unconscious (inactive)

Regarding hypothesis 'H2 - Barricade' with the features:
9 Path is crowded (active)
20 Person invading personal space of other person (active)
All of the hypothesis salient features are active.

Regarding hypothesis 'H1 - Harassment' with the features:
1 Agitated voice (active)
20 Person invading personal space of other person (active)
All of the hypothesis salient features are active.
```

Listing 7.3: Evaluating the observed features against the hypotheses

7.4.2 Results

A specific practical example of a single run of the expert system is displayed in listing 7.3 and 7.4 below. After new features enter the system they are activated and the analysis is started. The application starts searching through the activated features and extracts the ones that serve as salient features to certain scenarios. The activated salient features and their corresponding scenarios are made visible (Listing 7.3).

Next, the system finds what supporting features of the activated scenarios are also activated and assigns a score to every active scenario and displays the scenario with the highest score. The entire process of finding supporting features and assigning scores to scenarios is fully transparent to the user (Listing 7.4).

```
FINDING SUPPORTING FEATURES FOR HYPOTHESES
Regarding hypothesis 'Scenario 1 - Harassment'.
6 Normal voice (weight 1 ) (inactive)
12 People are in a quarrel (weight 10) (active)
26 Person pointing at other person (weight 1 ) (inactive)
27 Person shaking head (weight 1 ) (inactive)
28 Shouting (weight 3 ) (active)
Processing active weights...
Raw 10, normalized: 0.625
Raw 3 , normalized: 0.1875
The scenario has been awarded a normalized score of 0.8125

Regarding hypothesis 'Scenario 5 - Noisy'.
3 Conductor warns passengers (weight 2 ) (inactive)
4 Hand gestures (weight 1 ) (inactive)
7 Normal voice [multiple] (weight 3 ) (active)
11 People are frolicking about (weight 5 ) (inactive)
Processing active weights...
Raw 3 , normalized: 0.272727272727273
The scenario has been awarded a normalized score of 0.272727272727273

Regarding hypothesis 'Scenario 4 - Mobile Phone'.
10 People are complaining/annoyed (weight 9 ) (active)
20 Person invading personal space of other person (weight 1) (active)
Processing active weights...
Raw 9 , normalized: 0.9
Raw 1 , normalized: 0.1
The scenario has been awarded a normalized score of 1.0

Regarding hypothesis 'Scenario 2 - Barricade'.
1 Agitated voice (weight 6 ) (active)
2 Commanding voice (weight 3 ) (inactive)
7 Normal voice [multiple] (weight 1 ) (active)
16 People pushing/shoving each other (weight 10) (active)
28 Shouting (weight 2 ) (active)
Processing active weights...
Raw 6 , normalized: 0.272727272727273
Raw 1 , normalized: 0.0454545454545455
Raw 10, normalized: 0.454545454545455
Raw 2 , normalized: 0.0909090909090909
The scenario has been awarded a normalized score of 0.863636363636364

Scenario 2 - Barricade scored 0.863636363636364
Scenario 1 - Harassment scored 0.8125
Scenario 5 - Noisy scored 0.272727272727273
Scenario 4 - Mobile Phone scored 1.0

Scenario 4 - Mobile Phone scored the highest with 1.0 points.
Therefore it is the dominant hypothesis.
```

Listing 7.4: Finding un-observed supporting features to determine the most plausible hypothesis

7.5 Conclusion

In modeling rule-based systems, determining the structure, i.e., the number of rules in the system, requires the prior knowledge of the problem. This is usually determined by experts.

This chapter focuses on a data-driven approach since very few experts were available to determine the number of rules in the system. In addition, not all the concepts and conditions most experts reason with can be detected accurately and consistently at the current moment. The technology needs to mature to a state that the results achieved by current algorithms can cope with the challenging conditions in the train compartment. For that purpose, the input space is limited to the features that we are currently able to detect and this limitation determines the number of rules in the classification problem. However, as technology matures and more algorithms become feasible we foresee better results for this system. On the other hand, huge difficulties in extending and maintaining a consistent rule base in the combinatorial explosion that will ensue is a limiting factor. That is why we discuss alternative approaches in the following two chapters.

The broadness in the definition of aggression and emotion categories implies that they can overlap in their information spaces. One way to tackle this problem is by using fuzzy inference systems (FIS) based on fuzzy rules instead of the crisp rules proposed in this section. A FIS based on fuzzy rules is basically a rule-based system to cope with the uncertainties in the given problems. The fuzzy extension is very natural to the current rule-based system and thus not difficult to achieve. In the next chapter we discuss the Bayesian alternative that summarizes complex relations between entities in terms of uncertainty values.

A major drawback of the rule-based approach is the complexity of building and maintaining large rule-based systems. A rule base of three or four thousand rules may require a massive, highly-trained staff to maintain. In addition, even though the threshold in the scripts allows for some flexibility in the recognition of scenarios, the reasoning system is essentially unable to cope with unforeseen scenarios. In other words, the system would not be able to make creative responses as a human expert would in unusual circumstances. The system is not able to adapt to changing environments, unless the knowledge base is changed (which leads to the aforementioned maintenance problem). On the other hand the simplicity of the rules allows for the analysis of the inferred results and easy trace back to the root of errors simply by following the rules that were triggered.

Chapter 8

A Bayesian-based approach to reasoning about aggression

Bayesian networks greatly condense the information and relationships needed to create a decision framework. They can be applied in situations which are too complex to be automated with rule-based expert systems. While expert systems use key insights to turn data into information and decisions, they only allow yes or no (if-then) rules. This restricts their usefulness to only tackling simple problems. When there are more than just a few contributing factors for a decision, or if some factors involve gradations or uncertain information, the number of rules needed to fully define the decision making process can grow very quickly, resulting in a combinatorial explosion. Additional rules are needed to account for situations where data is missing, or where inputs come in the form of possibilities or positions in a spectrum, not facts.

8.1 Bayesian networks

Probabilistic methods, such as Bayesian networks, can be used where uncertainty to our knowledge is relevant to a given problem. Either because the knowledge is based on incomplete or imprecise measurements of the physical quantities or the measurements themselves might be incomplete due to limited resolution of the sensors, but might be also imprecise due to noise in the measurement process.

A Bayesian Network (BN) is a directed graph, together with an associated set of conditional probability tables (CPT) to model probabilistic relationships among a set of variables. BNs can be used in domains where uncertainty plays an important role. Nowadays, BNs appear in a wide range of diagnostic medical systems, fraud detection systems, missile detection systems, etc. Bayesian reasoning can be applied to cope with the slight variations that can occur in typical scenarios in the surveillance environment. BNs have a couple of properties that make them suitable to be used in

surveillance. First, they allow one to learn about relationships between variables. In addition it is possible to combine expert knowledge and data into a BN. Finally, BNs are able to model causal relationships between variables.

8.1.1 Bayesian network components

BNs have a qualitative and a quantitative component. The quantitative component of the BN is also called the probabilistic model and is expressed by the assignment of conditional probability distributions to the variables. The probabilistic model encodes stochastic relations between all observed and unobserved variables. Given the observed variables Y, probabilistic methods answer queries about a hypothesis X, by probabilistic inference, that is, by computing a conditional probability distribution $(X|Y)$. The main building block of BN theory is Bayes' Theorem. This theorem is stated as follows:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{8.1}$$

In this formula $P(X|Y)$ is the posterior probability of the hypothesis X, given the data Y, or the likelihood of the data, $P(X)$ is the prior probability of the hypothesis X, and $P(Y)$ is the prior probability of the data Y, or the evidence.

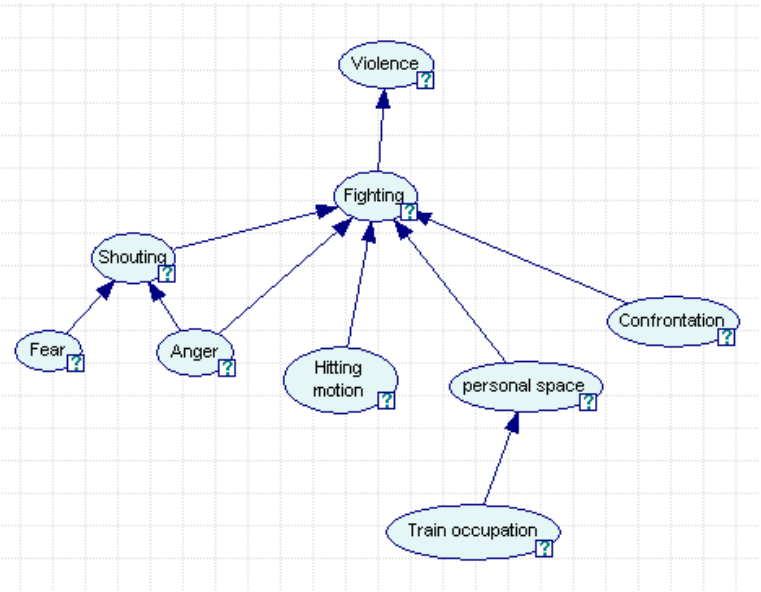


Figure 8.1: Example of a BN with nine nodes. All nodes are random variables that influence or are directly influenced by fighting.

The qualitative part of the BN is represented by its structure (e.g. Figure 8.1), this is a directed, acyclic graph where the nodes represent the set of random variables (which can be discrete or continuous) and the directed arcs connect pairs of nodes. The arcs represent causal or influential relationships between variables. If an arc points from

X_1 to X_2 then X_1 is a parent of X_2 and X_2 is a child of X_1. A parent directly influences its children. The key feature of BNs is that they enable us to model and reason about uncertainty. The dependence between uncertain variables is modeled by a conditional probability table (CPT) for each node. The CPT expresses the relation between the possible outcomes of the child node with the outcome of the parent node. The belief in the various outcomes is also stored in each node. This is expressed as a probability.

All these components together form the joint probability distribution of the BN. This structure of the BN makes it ideal for representing the semantics of the aggression model as a graph of aggression related concepts and their relations (see Figure 8.2).

8.1.2 Bayesian learning

Another important concept is probabilistic learning. Often a model is a parametric function $P(X|\theta)$ dependent on unknown parameters θ. Probabilistic learning attempts to estimate the parameters from a training data set x_D. There exist two main approaches to probabilistic learning: frequentist and Bayesian. The former finds a single optimal parameter configuration θ^* by maximizing a function that measures how well the parameters fit to the training data. The latter considers an augmented model $P(X;\theta;x_D)$ where parameters become additional random variables, which can be estimated by inference.

The problem with probabilistic learning is that there needs to be a large amount of training data to get good estimates for the parameters in the BN. For a network of the size and complexity of our aggression system, the training data gathered from the experiments in Chapter 10 is not enough. Our chosen solution is to populate the conditional probability tables of the BN with values estimated from interviews with security and railroad experts. We first created an influence graph of relevant nodes for each scenario. Then we asked the experts to judge the relationship between the nodes (essentially asking them to estimate the CPTs per scenario). Finally, we merged the results into a single BN. For the prior distributions, the statistics of previous occurrences of aggression acts, gathered by the NS, give probabilities as to the time, place and frequency of the classes of aggression.

8.1.3 Bayesian inference

Inference is a basic task in BNs and is used to compute the posterior probability distribution for a query variable Q, given the set of evidence (observed) variables E. There is a possible third set of (hidden) variables H that are neither query variables nor observed variables. The posterior probability can be calculated by:

$$p(Q|E) = \frac{p(Q,E)}{p(E)} = \alpha p(Q,E) = \alpha \sum_{H} p(Q,E,H), \tag{8.2}$$

where α is a normalizing constant equal to $1/P(E)$. Informally, the equation says that a query can be answered using a Bayesian Network by summing out the variables in H and dividing that by the probability of the evidence. Inference in BNs is a NP-hard, but approximate algorithms have been developed to handle large multiply connected networks. There is a variety of approximate algorithms such as Direct sampling, Rejection sampling, Likelihood weighting and Markov Chain Monte Carlo (MCMC) methods. A detailed description of the algorithmic mechanisms of Bayesian inference is given in [109].

8.2 Bayesian network of the proposed system

Selection of an appropriate network architecture is another place where prior knowledge plays a role. One approach is to use a very general architecture, with lots of hidden units grouped in levels. The Bayesian network starts with a set of input nodes which are continually updated with information from the detection algorithms. These variables are referred to as the observables. The evidence gathered from the observables are then used for inference of the variables that are not directly observable, known as hypothesis variables.

The hypothesis variables are split into three main levels. It would have been possible to use only one node to weigh all the data gathered in the information variables. However, this would have resulted in an extremely large conditional probability table where one would need to assign conditional probabilities to many unrelated variables. Instead, intermediate nodes are created that only consider relevant information, and pass on their conclusion to the next level of hypothesis variables. Let us examine a few nodes from the analyzed example shown in Figure 8.2.

The Bayesian network first tries to evaluate the emotional state of the people in the station in the node *Emotion*, where information about the expressions, voice stress level, and loudness of voices are used. The nodes calculates the probability of stress being present using its conditional probability table, and the probabilities passed on to it from its parent nodes (not shown in the figure). The node indicates that there is an 50% chance of anger being present, and passes this information to the next level of hypothesis nodes. In a similar fashion, the *Fighting* node, for example, uses information about whether people are emotionally stressed, shouting, whether there are hitting motions, and whether the intra-personal space is too small.

The last level of the hypothesis variables combines the probabilities into nodes representing the scenarios that are important to the NS such as vandalism, violence, and fire (see Section 2.3.2).

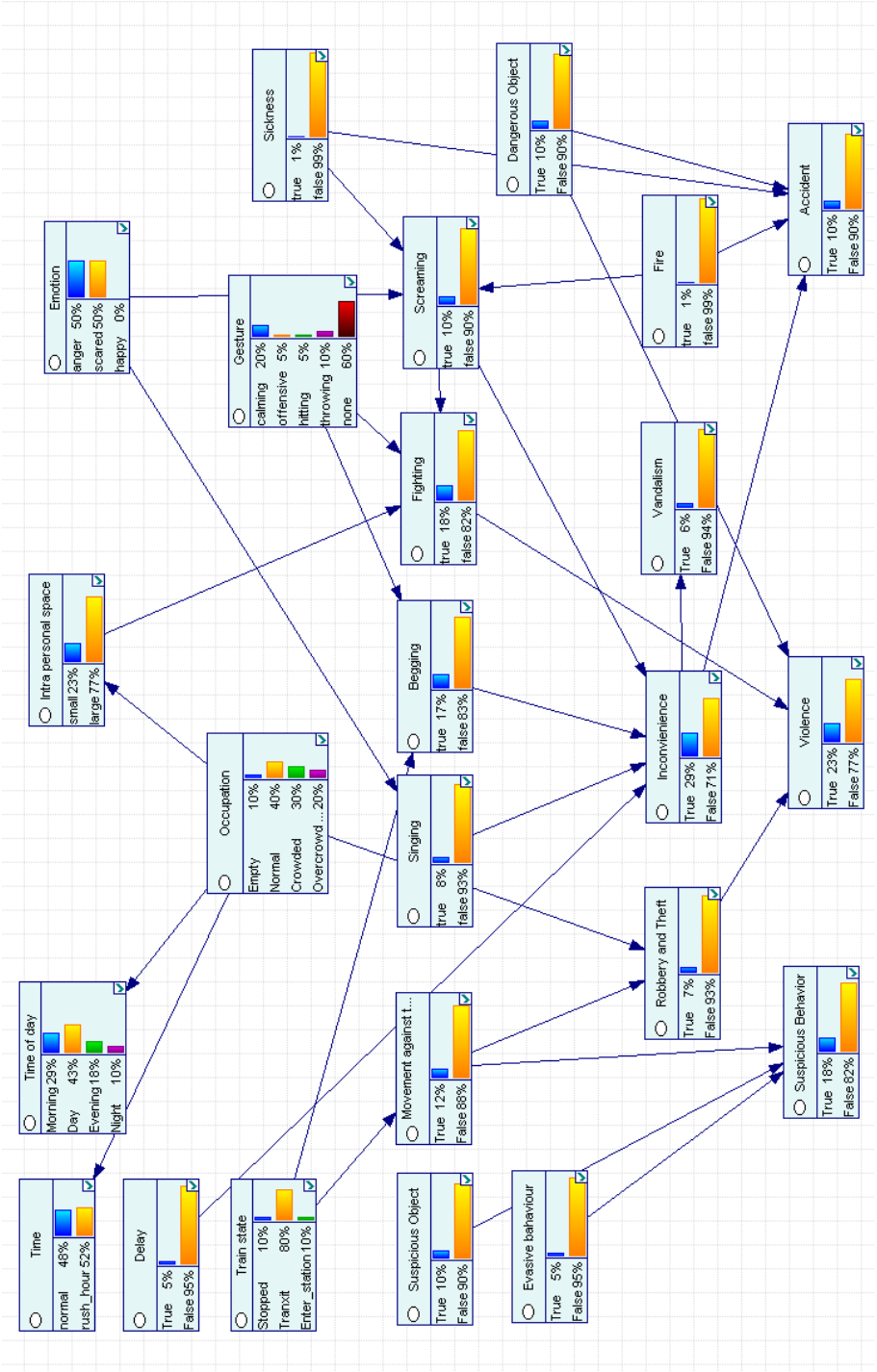


Figure 8.2: Bayesian network of the proposed system as a graph of concepts and their relations. The figure was made in GeNie.

8.2.1 Implementation

For the implementation of the Bayesian network and the Dynamic Bayesian extension (Section 8.2.3), we used the Gaia framework developed by Wiggers et al. [110] for context modeling in automatic speech recognition. The framework consists of a set of general purpose tools for inference with arbitrary DBNs. Although the library and tools of the framework were designed with language processing in mind, they can also be used for multi-modal fusion applications. The Gaia library is implemented in C++ and platform independent. Thus it could be easily incorporated as a fusion node in the framework of the aggression detection system. Figure 8.3 shows the components of the fusion node.

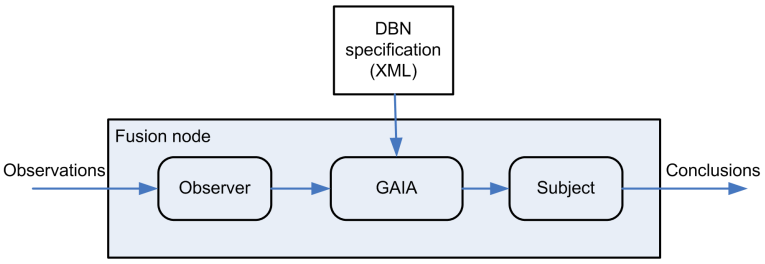


Figure 8.3: The Gaia library as a fusion node in the aggression detection framework.

Observer component. The observer component is responsible for subscribing to information providers. The provider can be the event/object detection algorithms described in Chapter 5.

Gaia component. Given incoming observations from the observer component, the Gaia component then calculates the probabilities of the aggression using the Gaia library.

Output component. The output module collects the analysis results from the inference engine and notifies the framework about it. Other modules such as a user interface described in Section 6.4.5 can subscribe to this information to present it to the user (e.g. degree and type of aggression).

8.2.2 Testing

The performance of the BN was tested using the annotations of the aggressive scenarios from the experiments in Chapter 10. Each scenario was assessed for all the relevant features and likelihood ratios. The posterior probabilities provided by the network were used to determine the ability of the system to recognize a scenario. Probability thresholds were defined for the output nodes so that the conclusion would be made if the probability exceeded that threshold.

8.2.3 Dynamic Bayesian extension

The BN representation shown in this section represents a snapshot of the situation in the train compartment. In order to follow the process as it unfolds over time, the Dynamic Bayesian networks [110] (DBNs) extension is needed. DBNs model processes that evolve over time by adding temporal dependencies between variables in a BN. A DBN can be defined by two BNs: a prior $P(X_1)$ and a transition model that defines how the variable at a particular time instance depends on the nodes at the previous time step.

$$p(X_t|X_{t-1}) = \prod_{n=1}^{N} P(X_t^i|P_a(X_t^i)) \tag{8.3}$$

where X_t^i is the ith variable in slice t. So the parents of a node in a DBN can either be in the current time slice or the previous time slice. Typically, first order Markov assumptions are made for the transition model, i.e. the nodes in a time slice only depend on the nodes in the previous time slice. This is illustrated in Figure 8.4. Each arc in the transition model denotes a dependency between variables in two "slices" of the network at consecutive times. The probability distribution is defined by the parameters of a Markov model: a matrix of transition probabilities and an initial state distribution.

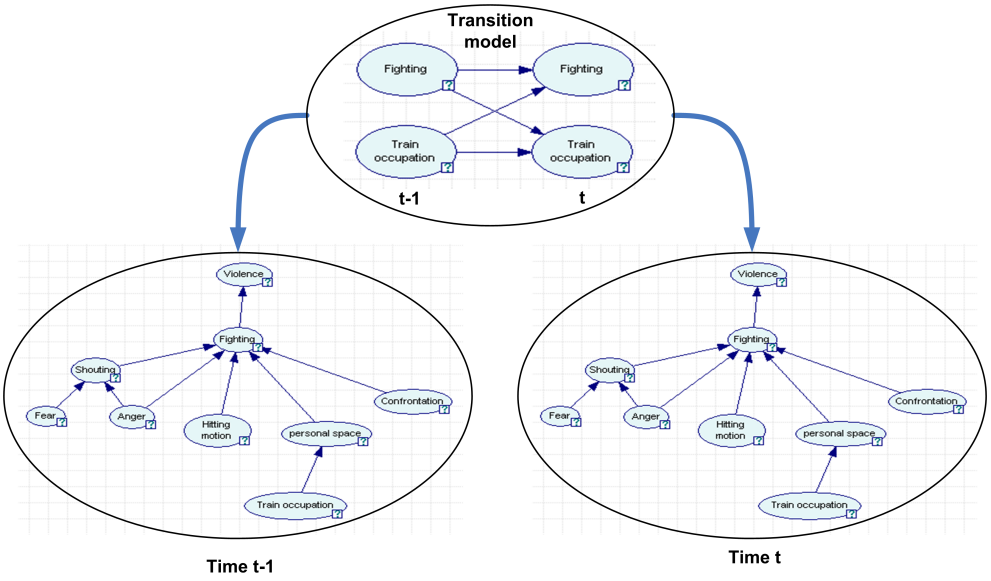


Figure 8.4: The DBN extension of the example BN in Figure 8.1 for two time slices. The structure and parameters repeat as the model is continues further over time.

8.3 Evaluation

In this section we evaluate the Bayesian approach using a case with real life footage. This particular footage was gathered during the experiments in the real train compartment. We analyzed and extracted the high-level cues from the footage using the annotation tool described in Section 10.2.2. These cues where entered into our BN. The results are evaluated here.

8.3.1 The footage

The specific scenario in our evaluation covers the begging situation which is categorized in the inconvenience category. The footage for this scenario is approximately 1:17 seconds long. We will go through several frames as an example of features we recognized, annotated and later used as evidence in our BN.

The footage starts (Figure 8.5) with a person (red square) walking in the hallway, stopping temporarily where there are people in the neighborhood on his way and stretching his hands and asking for something to people sitting (green squares).



Figure 8.5: Screen shot from the beggar scenario: a person walking slowly along the compartment and making begging gestures.

The footage continues as depicted in Figure 8.6 where people are complaining (red square) about his begging behavior (green square) and asking him to go work for his money.



Figure 8.6: Screen shot from the beggar scenario: people are complaining about the behavior.

In Figure 8.7 the person making the begging gestures invades the personal space of another person (red square), which results in a quarrel between the two persons

(green square). They are talking loud and even shouting. After another person (purple square) come between the two, the beggar moves on and keeps on making begging gestures along his way.



Figure 8.7: Screen shot form the beggar scenario: person is invading personal space.

8.3.2 Activating the features

We present, using the transcript from the previous section, which of the features we could identify and we need to use (activate) as evidence for our Bayesian network. Table 8.1 shows per frame presented in the previous section which of the features are identified.

Table 8.1: Selection of extracted high-level features from the beggar scenario.

Figure	Feature
Figure 8.5	Person is moving around making begging gestures.
Figure 8.6	People are complaining.
Figure 8.7	Person is invading space of another person. Person is talking loud. Person is shouting. People are in a quarrel.

8.3.3 Evaluation

After the extracted features are inserted as evidence in the BN, the situation nodes with the highest probabilities are Beggar and Fighting with respectively 0.95 and 0.57. The reason for the begging situation is obvious given the input feature nodes we annotated, but the fighting situation has such a high probability due to the fact that for a period of time there was a quarrel between the beggar and a passenger and they were shoving each other a little bit, along with invasion of personal space and loud talking. The structure of the network is such that some of the features annotated here have an influence on the fighting situation. Nevertheless the network did not infer the situation to be a high fighting situation, but inferred it to about a 50-50% chance of the situation being a fighting one.

Both the beggar and fighting nodes have a high probability. However, even though the beggar probability was higher than the fighting probability, that did not reflect

very well into the scenario nodes (output). There we see that violence was more likely than inconvenience, meaning that fighting had a bigger influence than the beggar situation. There could have been a couple of reasons that this happened. First, the CPTs were not that well adjusted for the two scenario nodes, but we checked here and the CPTs reflect that if a dangerous situation occurs, that it should be awarded a priority over the other scenarios. This could be a nice feature, as if such a combination of more scenarios or situations happens in a train environment, we want to know the ones where the probability of the aggression level is high and to have a human operator re-analyze the situation and act accordingly. Another reason could be that the number of nodes that influence the Inconvenience node is higher than that of the Violence node, making Inconvenience more difficult to infer.

8.4 Conclusion

In this chapter we outlined in brief the general theory of the BN network and presented the design of our particular network. In addition, we presented an evaluation case, which consisted of video and audio footage that we analyzed and annotated. In turn we inserted the evidence into our designed BN, which inferred the output probabilities of both the intermediate and output nodes. Although we had to fill in all conditional probability tables by hand (based on instructions from human experts), which was a tedious job, we can be quite happy with the results the network is outputting.

If there was a large amount of empirical data available, we could use that to train the conditional probability tables using one of the many known learning algorithms. In the network's current state, using the case presented, it can be shown that the job it was designed to do can be done in a satisfying way. However, the case also shows that the CPTs need to be fine tuned further, either by manual labor or by Bayesian inference. I the latter case, a huge amount extra data is needed.

Chapter 9

Aggression detection using an emergent based reasoning approach

The behavior understanding implementations in previous chapters are based on learned patterns of behaviors. In the rule-based case, when a detected behavior does not match a predefined pattern, it is an anomaly and the system will not be able to deal with it. In the Bayesian network case, the most similar scenario will have the highest probability. Since in both cases the reasoning system has not been designed to cope with unforeseen situations, the results may be unexpected. In this chapter an approach will be presented that constructs, by self-organizing and self-learning, scenarios for hitherto unknown patterns of behaviors.

9.1 Overview

Most currently known approaches for recognizing human activities use a top-down approach using predefined scenarios (e.g. [111, 6, 112]). In general, these approaches match incoming data with the predefined scenarios. The one that best matches the incoming data is the most probable. Sometimes the probability must exceed a certain threshold before a conclusion is made. Some of these systems work rather well and are able to recognize a lot of different scenarios.

However, most of these systems do not take into account the interrelation between different events. Different events may add up to trigger a scenario, while they may not be directly related to each other. The emergent model presented in this chapter, is a bottom-up approach in which the possible links between detected features or events are used to draw conclusions about the situation. By taking into account these links, results may be better than when only looking at the events themselves. This

approach may also enable the system to detect scenarios it has not been designed to recognize.

In the emergent approach, each feature has a salience value, which indicates how likely it is that the feature is related to an aggressive scenario. Link-strengths are used to indicate how likely it is that two different features are related. Link strengths are stored in a link matrix. Incoming events are arranged to form the best suitable semantic network of features.

9.2 The emergent model

The emergent model described in this section can be compared to solving a jig-saw puzzle, where one tries to fit the pieces of a puzzle locally to form a bigger whole globally. The system begins its puzzle when a salient event is observed. Next it tries to form a semantic network from more incoming features. A salient event is an event with a high salience value. First the system tries to create a kernel, by combining one or more salient events into a salient network. Such a network is considered a kernel if its total salience reaches a certain threshold. For combining the features, the system looks up the probabilities of events occurring together in the link matrix. These links between events are called link-strengths, which are set to zero if the events have nothing to do with each other and to a value > 0 otherwise. Two features with a high link strength will group together. The stronger the link-strength, the stronger the grouping. If a kernel emerges from this grouping, it forms the hypothesis for the situation at hand. The system can have multiple kernels at any time, in which case they become competing hypothesis. Each event can be used in more than one network.

To back up a hypothesis, the system looks for more events that have a link with the events already contained in the kernel. It does that by looking into all events it has received. When it finds an event that can be linked to the kernel (by looking into the table of link-strengths), it adds it to the network and calculates the new fitness of the network. If the new fitness is greater than or equal to the old fitness, the new network is stored, otherwise the old one is kept. If all observed events are fitted to the network, the system waits for new incoming events. When a new event enters the system, the system tries to link it to the currently known networks. To make sure the system does not base its hypothesis on too old information, each event is subject to a time decay function, which makes its contribution to the network smaller as time progresses.

9.2.1 The link matrix

The emergent model uses two tables to look up values. First, the influence matrix is a table that contains salience values, which indicate how likely a feature is related to a scenario. The influence matrix has already been discussed in Section 7.2.1. Second, the link matrix specifies the correlation between pairs of features. If observations

are dependent of each other, the observation of one feature can give expectations about the possible observation of the correlated features. The link matrix is a two-dimensional triangular matrix containing all the link-strengths. A link-strength is a value that indicates how likely it is that two different features are related to each other. A link-strength equal to zero means the two features involved cannot or will never be related. Table 9.1 shows a small link matrix for several related features (the actual link matrix is much larger). Because many features are independent, the entire link matrix is usually very sparse.

Table 9.1: Link matrix, where f_1 = Conductor enters, f_2 = Bumping into person, f_3 = Crowded, f_4 = Running, f_5 = Shouting/singing, f_6 = Evading conductor, f_7 = Negative emotions, f_8 = Grabbing wallet.

Feature	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
f_1	1	0	0	0.30	0	0.10	0.05	0.90
f_2	0	1	0.70	0.20	0	0	0.80	0.20
f_3	0	0	1	0	0	0	0.01	0.10
f_4	0	0	0	1	0	0	0.02	0.02
f_5	0	0	0	0	1	0	0.60	0
f_6	0	0	0	0	0	1	0.02	0.01
f_7	0	0	0	0	0	0	1	0.01
f_8	0	0	0	0	0	0	0	1

Link-strengths can be obtained by analyzing the scenarios. First semantic networks of features representing each scenario are created. An example of such a network can be seen in Figure 9.1. From these networks, the probability of the links in it can be determined. Determining the link strengths can be done by looking at how probable it is that the two events co-occur.

9.3 Implementation

All features that are extracted from the sensor data are fed into the system as events. A blob is basically a set of pixels that belong together, a human can be defined as a blob that has a face. Furthermore, blobs have identifiers, so as to determine if different events belong to the same blob and to detect movement. Movement can be determined by locating the same blob on different frames from the camera feed and determining the distance between the locations. Moving fast basically means a blob is moving quicker than someone walking, thus in practice its speed is greater than some threshold (see Chapter 11 for more details on feature extraction, face and activity recognition etc.). The flowchart of the emergent system is displayed in Figure 9.2.

All events have a salience value, which denotes how "serious" the event is, i.e. how much the event indicates an aggressive situation in the train. The events with the higher salience values are likely to end up in the kernel of a network, with the less salient events used to back up the hypothesis formed by the kernel. Furthermore,

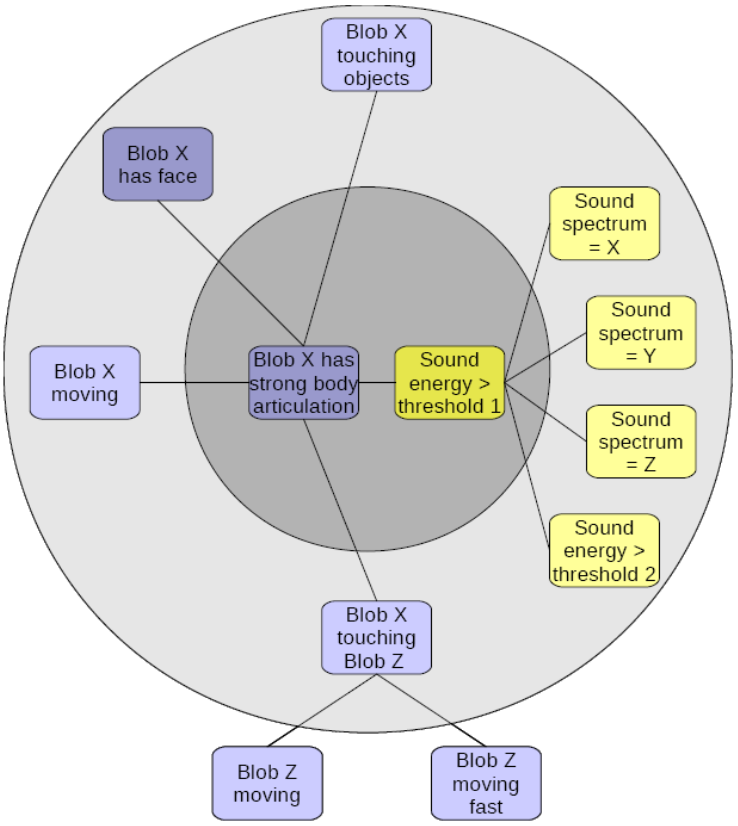


Figure 9.1: Semantic network for the Hooligans scenario.

each event is provided with a timestamp, for calculating its age. The age of an event is used to determine how significant it still is and to prevent the system from basing its hypothesis on information that is too old to be of any significance.

9.3.1 Fitness function

For each kernel, the system has to calculate the fitness of the network. This is done using a fitness function. Intuitively the fitness function should be based on the size of the network (i.e. bigger networks with many connections are fitter), and also on the link-strengths in the network. A possible function for the fitness (F) is:

$$F = \frac{1}{N} \sum_{i=0}^{N} \sum_{j=0}^{N} t(e_i) t(e_j) l(e_i, e_j), \tag{9.1}$$

where N is the number of events in the network, e_i is the i_{th} event of the network, $t(e_i)$ is the salience of e_i after the time decay function (see Section 9.3.2) and $l(e_i, e_j)$ is the link-strength between event i and event j. This function takes the sum of all links and divides them over the number of events in the network. This makes sure that a larger network will score higher than a small one. Two independent features

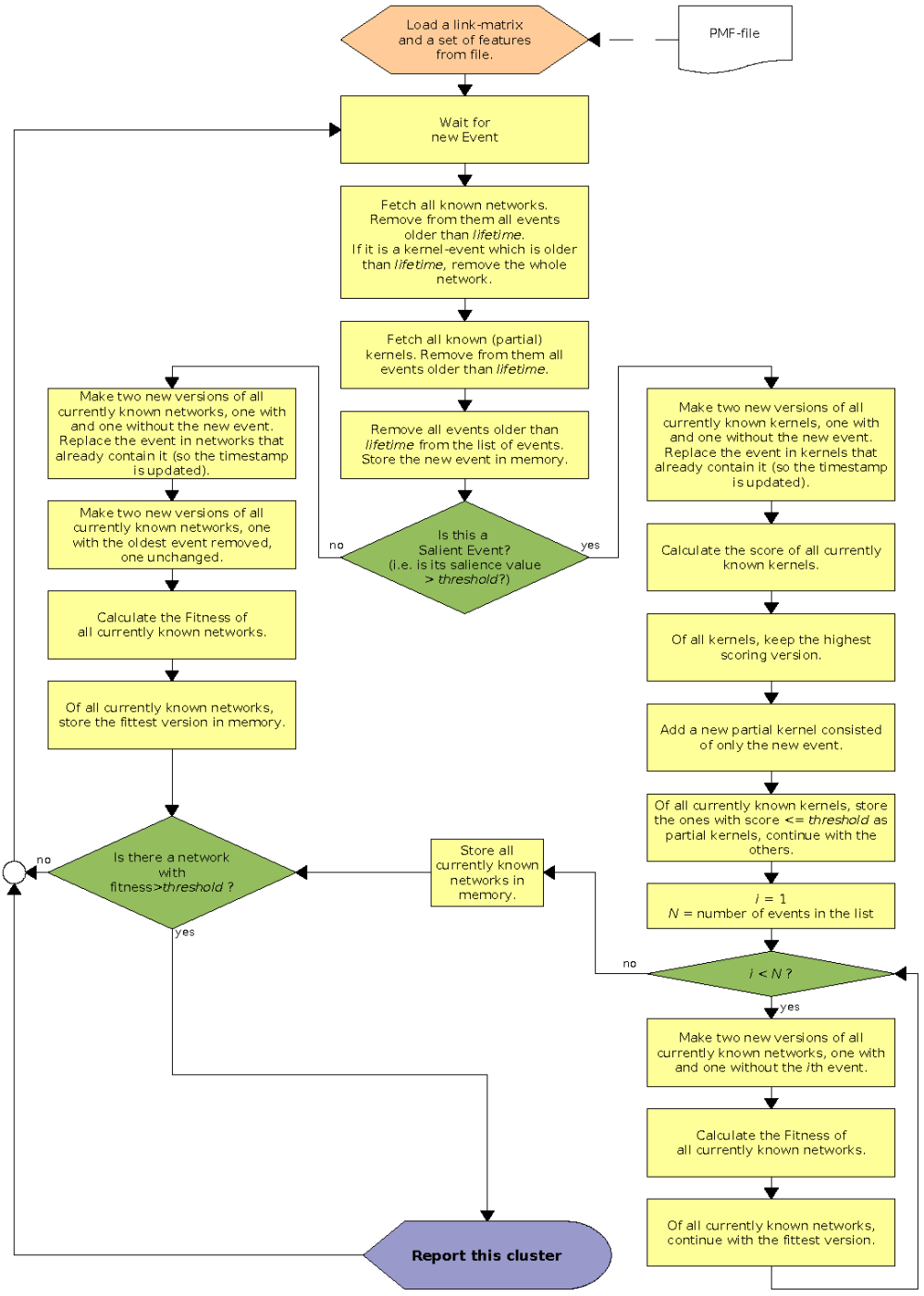


Figure 9.2: Program flow of the implementation of the emergent model.

(with link-strength = 0) are automatically left out of the equation as it has no effect on the fitness. Especially strongly interconnected networks (where events are linked to multiple other events in the network) score very high when using this fitness function.

It is also possible to take the average salience value of the events into account, by

multiplying Equation 9.1 with the average salience value. The threshold fitness for reporting the network should be adjusted to the fitness function used.

9.3.2 Time decay function

To make sure the system does not base its hypothesis on old information, events are subject to a time decay, limiting their contribution to the system as time progresses. Different time decay functions are possible:

$$t(e) = \begin{cases} 1 - \frac{a(e)}{T} & \text{if } a(e) < T \\ 0 & \text{otherwise,} \end{cases} \tag{9.2}$$

$$t(e) = \begin{cases} 1 - sin\left(\frac{\pi \cdot a(e)}{2 \cdot T}\right) & \text{if } a(e) < T \\ 0 & \text{otherwise,} \end{cases} \tag{9.3}$$

$$t(e) = \begin{cases} cos\left(\frac{\pi \cdot a(e)}{2 \cdot T}\right) & \text{if } a(e) < T \\ 0 & \text{otherwise,} \end{cases} \tag{9.4}$$

$$t(e) = \begin{cases} \frac{1}{2} \cdot cos\left(\frac{\pi \cdot a(e)}{2 \cdot T}\right) + \frac{1}{2} & \text{if } a(e) < T \\ 0 & \text{otherwise,} \end{cases} \tag{9.5}$$

where $a(e)$ is the age of the event, defined as the difference between the current time and the time of the most recent occurrence of the event (i.e. the event's timestamp), and T is a predefined maximum lifetime of the event. Equation 9.2 is a linear function, where $t(e)$ gets smaller when $a(e)$ approaches T. Equation 9.3 on the other hand makes the decay start out fast, but slower near the end of the maximum lifetime. Equation 9.4 works oppositely, in that the decay starts out slow but goes faster when nearing the maximum lifetime. The last function (Equation 9.5) is a hybrid version, where the decay starts out slow, goes progressively faster, but goes slower again when approaching the maximum lifetime of the event. For each of these functions, it holds that $t(e) = 0$ if $a(e) \geq T$, as the event is considered too old to be of any significance.

9.4 Results

The results of testing are relatively disappointing for this specific approach. Building a usable link matrix proved to be quite a difficult task, resulting in the system showing explosive growth. Using different fitness functions or time decay functions did not provide a solution either. The resulting networks of features are therefore not very useful for determining the problem at hand.

Every feature or event can be linked to every other if not directly, then most probably via another route. This problem arises because each feature can accept an infinite number of links. A possible solution to this problem is not to search through all possible network clusters, but rather in a one-dimensional sequence of events. Sequences of events are analogous to sequences of letters that form words. Scenarios in this model can be described in terms of probabilities of event transitions. The system can calculate bi- and trigrams for scenarios, based on the same logic for calculating them for words in speech recognition. This eliminates the possibility of explosive growth by limiting the amount of links an event can have (which is logically only two links in this approach).

The first tests were done on sequences of letters, combining incoming letters in order to try and create a word with the highest possible score, with the score being calculated using the bi- or trigrams. Starting with incoming letters (events), the system tries to add letters to either the front or the end of each of the available letters. Before adding letters, the system removes the ones already used in the word from the list, to avoid using letters more than once. After adding letters to the beginnings and endings of the words, the system stores the newly found words in the same list as the words that were already found in the last step. The system calculates the score for all currently known words and stores them with the word itself. The system takes the x highest scoring words, dropping the others, and tries to extend them again by either adding a letter to the front or the end. This process is repeated length-of-the-input times, to make sure that the maximum word length can be constructed.

9.5 Discussion

Self-organizing behavior can be found almost anywhere in nature, on any level or magnitude [113]. The most well-known examples that are commonly associated with this phenomenon are based on actuation or effect in groups, flocking or swarm behavior: schools of fish survive by together pretending to be a bigger entity, troops of ants bridge over relatively wide chasms by clinging on to each other, even mathematical fractal figures that consist of simple elements and rules and yet look complex.

This chapter is inspired by emergence which is closely related to self-organizing behavior. Emergence accounts for simple rules, for simple units, that individually say nothing at all about the combined behavior, to be responsible for complex behavior. In this case, every unit sends little bits of information to a central module where it gets combined with other bits of information until it becomes a meaningful high-level concept, especially considering the low information content provided by each separate unit.

The many ways in which events can be linked to a kernel, leads to a well-known problem: the curse of dimensionality. As more events are added to the system, and consequently the dimension of the data grows, it transpires that the algorithm needs more

resources to be able to model the increased input space. Unless the algorithm is improved, the consequences are that calculating the network fitness becomes slower and more fault tolerant. At the moment of writing, we are trying solve the problem by limiting the amount of links a events can have with different approaches.

Part III

Results

Chapter 10

Experiments

This chapter presents a detailed overview of the aggression detection experiments conducted in a real train compartment. In the experiments four microphones and four cameras, located at different positions in the train compartment, were used to capture data from semi-professional actors continuously and simultaneously. The actors were performing aggressive as well as non aggressive scenes in the compartment. In these experiments, data was only captured (in the final situation, sound and video based algorithms should extract features from the incoming data in order to detect aggression). This chapter discusses the details of the experimental setup and presents the resulting data products.

10.1 Data acquisition

A first step towards automated aggression detection is collection of a realistic dataset in a real train in order to evaluate aggression detection algorithms. Due to the scarcity of these recordings and the privacy issues involved, it is not possible to use real data. Against this backdrop, we hired semi-professional actors to perform aggressive scenarios in a real train. The actors are given scenario descriptions to perform, we used multiple microphones and cameras to record their actions. Three sets of experiments were conducted.

- The first set of experiments was inside a Benelux train that was made available by the Dutch railway company (NS/ProRail). The train was described in detail in Section 4.1. There was an experienced train conductor to play the part and give advise. The train was standing still while the recordings where made (although the air conditioning system in the train was active and producing background noise). The resulting dataset consists of recordings of aggressive as well as non-aggressive scenarios with multiple cameras and multiple microphones.

- The second experiment was done around the Delft Zuid train station and was focused on abandoned luggage and behavior of people around the train stations. In this experiment the data was captured using two fixed and one mobile camera. The audio was recorded using the microphones in the cameras.
- A final set of experiments was conducted to test and evaluate the detection system. The recorded data for these experiments was inside a life-size train compartment and train station built by the NS and used for demonstration purposes and festivities. The same setup of sensing hardware as in the first set of experiments was used here.

10.1.1 Requirements

Since it is not feasible (or legal) to collect real live data from train compartments, we collected data while actors acted out certain predefined scenarios in a real train compartment. Because we want the results to be realistic, there are some additional requirements to such a data collection process. Following is a list of requirements for the data, specifically related to the automated detection of aggression using data fusion.

- The data should contain aggressive as well as non-aggressive situations. This requirement is achieved by including aggressive as well as non aggressive scenarios. In addition certain scenarios have been split into aggressive as well as non aggressive outcomes.
- Realism, the situations have to be realistic and it is important that all the scenarios really do occur in reality. The realism of each scenario will be verified by professionals of the Dutch Railways.
- Flexibility, we view a recording of an aggressive scenario as just one instance of that scenario. As a result, it is undesirable to impose too many restrictions in the scenario description (since that would limit the number of instances we can get for a scenario). A scenario description should enable an actor to understand what situation is expected/requested and then allow him the freedom of a personal interpretation.
- Re-usability, in the future, we want to be able to hire other actors to perform the same scenarios again (resulting in another instance of the scenario). This is comparable with emotion recognition for facial expressions where data collection is done by having several people perform the basic emotions.
- Humans should be able to (easily) classify the aggression in each situation correctly. There is no point in trying to automatically detect aggressive situations, if even humans are not able to do it. However, if human are able to do so, it may be worth while to find out how they do it and what audio/visual cues they have used. This information can be used to annotate the data.

10.1.2 Experiment setup

The sensor setup used to capture the scenarios specified in the storyboards consists of four cameras and four microphones. The location of the sensors in the train compartment and their orientation is shown in Figure 10.1. The cameras are mounted on the roof of the compartment and face downward with an angle. In addition, the cameras are aligned in a straight line along the center axis of the train. The orientation of the cameras is such that they face the center of the compartment. Most scenarios were performed in the middle of the train, where the two cameras in the middle have the largest overlap.

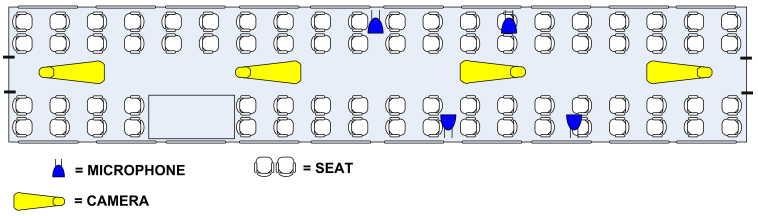


Figure 10.1: The locations of the sensors seen from a top view of the train compartment. All sensors are attached to the roof of the train.

The cameras also have zooming, panning and tilting capabilities, but these settings were fixed during the recordings. The locations of the four microphones are indicated in Figure 10.2. They are all mounted at the same height and face inward in the train compartment. All microphones are connected to a multichannel recording system to provide synchronized audio streams.



Figure 10.2: All cameras face downward with an angle and the center of the compartment has the largest camera coverage and overlap.

The scenarios performed by the actors are based on storyboards. For each scenario, a storyboard was made and verified. In designing storyboards about aggression in trains it is necessary that the stories are realistic, that is to say, based on real life experiences. To ensure this, the process of creating a storyboard is split up into two steps: storyboard design and storyboard verification.

10.1.2.1 Storyboard design

The goal of storyboard design is to develop realistic prototype stories. Therefore, it should contain a clear plot, with all the key scenes and key concepts. The process of designing a story can be split up into several steps.

- The choice of the main theme. At first it has to be decided what will be the dominant theme in the context. Is the story about hooligans, travelers with non-adapted behavior, pickpockets, or travelers without a ticket.
- Once the main theme is chosen the designer has to decide about the main aggressive acts or behavior. Options are hooligans demolishing a train, start fighting with other travelers or the conductor.
- The next step is to decide about a reasonable rising action. The behavior during the rising action should culminate in the aggressive act. So during the rising action there is a sequence of acts with increasing tension, threat of aggression.
- Next, the designer has to design one or more representative scenes for every step. A scene is specified by a sketch.
- Once the designer has a rough outline of the story, he looks if the story is complete or over complete. He will add or delete sketches.
- The final step is the refinement of the sketches and creation of the world model. A sketch should express the semantic content of a scene, the main actors should be presented in a prototype way. In Figure 10.3 sketches 2 and 3 for example, we see a hooligan showing non adaptive behavior and a conductor giving a reprimand. Furthermore, the designer should create a world model by specifying the objects and their relationships within the storyboard.

10.1.2.2 Scenarios and storyboards

Given the requirements for the data, a rigidly scripted approach, in which all actions are specified in advance, has serious limitations on the flexibility of the scenarios. Storyboards [114] do not have this limitation. In a storyboard the viewer is confronted with a sequence of sketches, each sketch represents a characteristic scene (see Figure 10.3). In addition, all the sketches, put in the correct sequence, tell a story. The general structure of our storyboards contain five phases:

Context. In the first frame of the storyboard a general context is presented. The situation, time and location is sketched.

Problem definition. In the next frames key player(s) are introduced and an indication of the main theme is given i.e. one or more people violating the rules of general accepted correct behavior.

Stimulus-response reaction. The story may continue in increasing violence or decreasing violence. In the first case, the aggressor shows behavior of overt aggression. This can result in more aggression by the aggressor or aggression against the aggressor. We observe a spiral of increasing violence. Otherwise, violence can be neutralized or decreased by counteracts e.g. passengers who inform the aggressor that he is violating the rules. As a result the aggressor adapts his behavior.

Climax. The stimulus-response behavior results in a climax e.g. an explosion of aggression, a peaceful arrangement, a successful repression of aggression, or some actors leave the scene.

Final result. The last frames show the final result (happy or unhappy end).

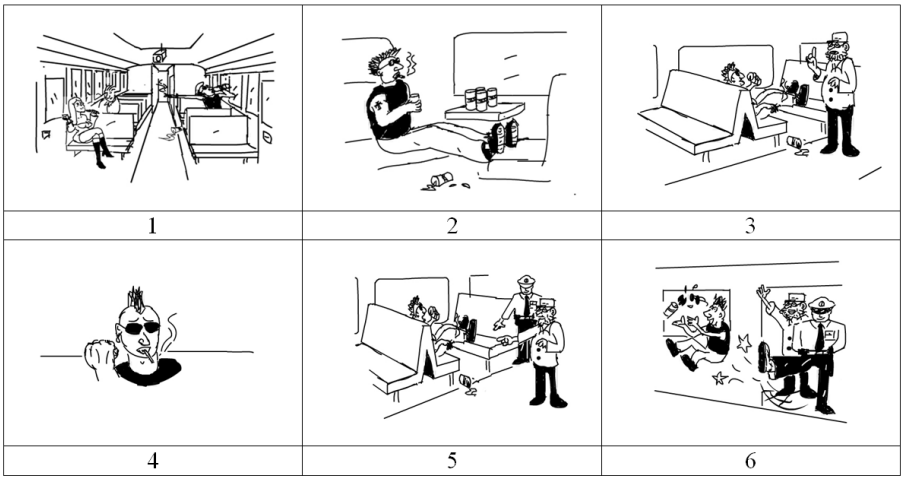


Figure 10.3: The five phases of our storyboard model: context (1), problem definition (2), stimulus-response reaction (3 and 4), climax (5) and final result (6).

10.1.2.3 Storyboard verification

According to the "Cognitive dissonance" theory developed by Festinger [115], the actors do not need fixed lines to memorize; the aggressive situation will emerge spontaneously as the actors will try to interpret the situation they have seen in the storyboard. One thing we have to make sure of is that the same storyboard will invoke the same interpretation in all actors as it did with the storyboard designer. In a verification step, performed beforehand, a test group of ten students was used to verify that

each storyboard invokes the same story in the mind of the viewer. The students were individually asked to write down the story that came up in their mind once they saw the storyboard. We only used the storyboards that induced the same (and correct) story in all the students. To fulfill the realism criteria, we asked a train conductor with 15 years of experience to verify the realism of the scenarios.

Appendix B shows the list of scenarios and scenario descriptions and motivations. Some scenarios have been split up into the different paths the situation could go into. In some cases there is no aggression (e.g. ticket checking); these scenarios serve as examples of normal situations. Other scenarios do have some form of aggression or aggressive behavior, but should not be classified as such because it falls under acceptable behavior (3a, 6a). Some cases of scenarios 10 are difficult to act out (graffiti) as it involves destruction of property. These have not been performed. Finally the normal scenarios such as entering and exiting the train have been recorded.

10.1.3 Data products

In the end, aggressive and normal scenarios are recorded in sequences which total up to about one and a half hours of audio and video data. The data contains the scenarios as described earlier as well as recordings of normal and spontaneous situations. All the data of the sensors is stored in separate streams (four audio streams and four video streams). The four video cameras captured colored video at about 13 frames per second, at a resolution of 640x256 pixels.

Each microphone captured sound generated by the actors performing the scenarios at a sample rate of 44100Hz with a 24 bit sample size. Each track is synchronized in hardware with sample accuracy. The audio data can be addressed in a single synchronized project consisting of the four streams of the four microphones, or as separate mono audio streams for each separate microphone (Figure 10.4).

10.2 Data preparation

The collection of data is the first part in a two step data acquisition process. The raw data contains the cues for the detection system to detect the factors (or combination of factors) that constitute the aggressive situations. Now that we have a collection of raw data in the form of recorded signals (audio or video data) and also the information about the physical circumstances surrounding the recording of each signal, such as time, place, equipment used, people involved, descriptions of the events depicted in the signals, and so on. Our next task is to represent this information in a flexible and general way. In other words, the data has to be (semantically) labeled or annotated.

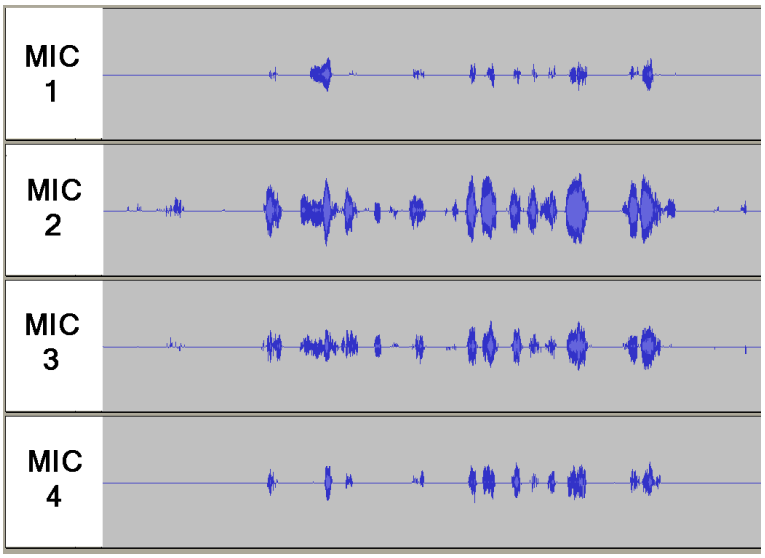


Figure 10.4: Four waveforms of a shouting scene recorded by the microphones. The waveforms are different in energy yet similar in form, indicating that sound source localization (see Section 5.2.3.3) may be feasible.

10.2.1 Data annotation

One of the goals of data acquisition is to enable re-usability of the results. This requires grounded data annotation with truth values about the situation in the train compartment. This means that we have to describe all the objects (including their characterizing features and relations with other objects) that play a role in the world of aggression in the train compartments. Therefore, an annotation language is needed to express this information.

We used the concepts defined in the aggression ontology developed earlier in Section 4.3.2 to provide the vocabulary for the annotation language. This has several benefits. First, the annotations can be directly related to the developed world model and second, the annotation language gives an indication of the completeness of the ontology. In other words, if the annotation language falls short to express the situation in the train, it indicates that the ontology needs to be extended. A graphical view of the relationships between elements in the annotation language is shown in Figure 10.5.

The schema definitions are included in Appendix A. Essentially, an annotation file starts with the introduction of the sensors, the introductions of the objects and then a list of events in which the introduced sensors and objects play a role.

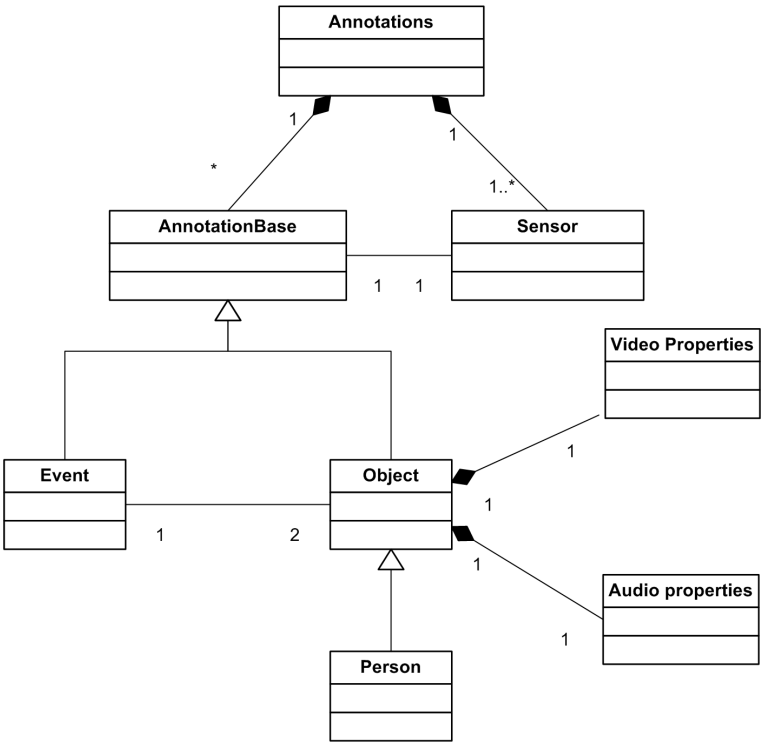


Figure 10.5: Graph of the relationships between elements in the schema.

10.2.2 Annotation tool

During the annotation of the data, a tool was developed to aid the annotation process. The annotation tool is conceived to be useful for creating publishable annotated data to accompany the raw video and audio data. This annotated data can be used as ground truths to evaluate algorithms. The user interface of the annotation tool is designed to guide the user through the annotation process giving feedback whenever possible and disabling options that are not logical. Many checks and help tips have been implemented in the system so that the user can work efficiently and with minimal errors. The user interface was designed to be simple, intuitive, and visually appealing. New icons of people, expressions, relations, actions and objects can be added to the interface by modifying a GUI file.

The user interface consists of a number of areas. These are numbered in Figure 10.6. During the annotation process, a video file can be loaded. After loading the file, the user interface shows the current frame (the area indicated by (1) in Figure 10.6) and five subsequent frames in a film strip below it (2). The arrows on the filmstrip allow the user to quickly navigate through frames. The icon selection area (3) allows the user to select and annotate objects, events and relationships. The location of an object can be indicated by drawing a rectangle in the corresponding area in the current frame. In this way the presence and location of faces, bags, etc. can be annotated.

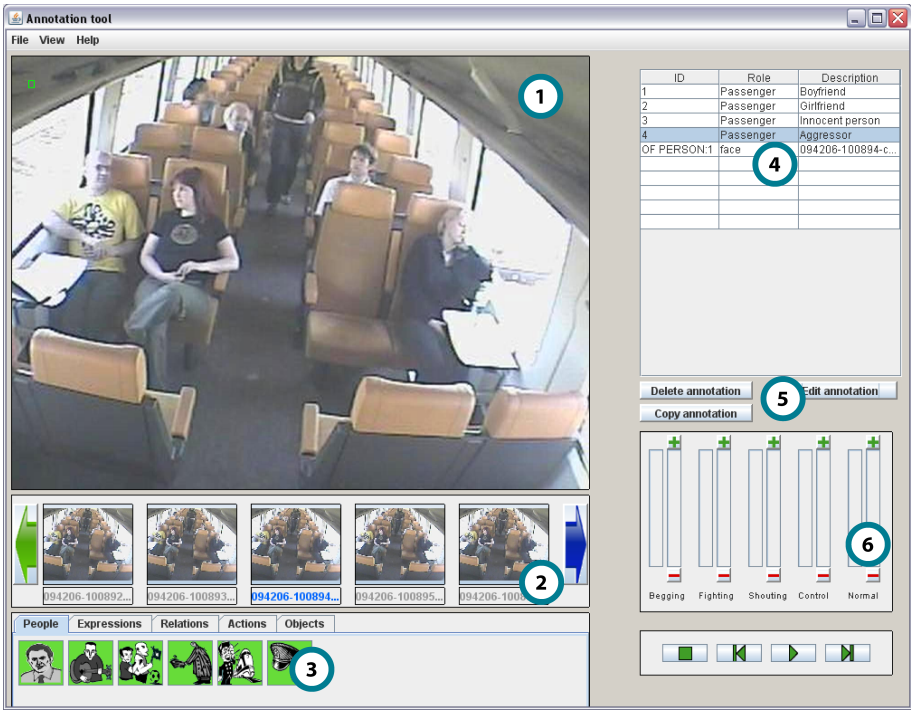


Figure 10.6: Screen shot of the annotation tool.

All the annotations of the current frame are displayed in the table in area (4). Annotations can be edited, removed or copied to other frames using the controls in area (5). Finally there is a rule-based aggression detection system running to verify the annotations and show the intermediate results (6). The rule-based system is discussed in Chapter 7. The resulting annotation is an XML file that uses the same vocabulary and semantics of the ontology defined in Section 4.3.2. An example of an annotation file can be found at the end of Appendix A.

10.2.3 Camera alignment

Only with the knowledge of the camera orientation parameters can we map objects in 2-D image coordinates to 3-D world coordinates. Obviously, this mapping is important for multi-camera object tracking, where positions in each 2-D camera image has to be mapped to coordinates in a common 3-D space. The inverse of the mapping is also very useful. For instance, in aggression detection using audio data, the inverse mapping enables us to visualize the 3-D coordinates of an audio source onto the 2-D coordinate system of a camera image. The process of determining the camera parameters is also known as camera calibration. In the following section, the Direct Linear Transform (DLT) method is applied to calibrate one camera. A similar approach can be used to calibrate the three other cameras.

10.2.3.1 The DLT method

The imaging process produced by a projective camera can be interpreted as a sequence of three projective transformations: given a point $p = (x_w, y_w, z_w, 1)$ in homogeneous world coordinates and a point $q = (f \cdot x_i, f \cdot y_i, f)$ in image coordinates corresponding to the projection of p onto the image, then the mapping of p to q can be expressed as:

$$q = K \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot M \cdot p \tag{10.1}$$

Where K represents the intrinsic parameters of the camera and is given by:

$$K = \begin{bmatrix} \sigma_x & \sigma_\theta & u_0 \\ 0 & \sigma_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10.2}$$

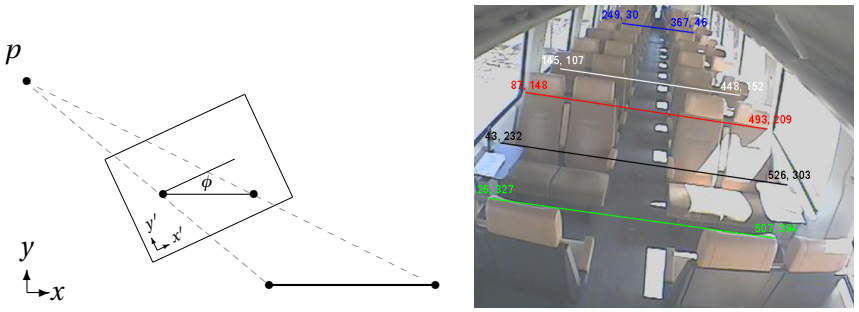
With (u_0, v_0) the coordinates of the principal point, and σ_x and σ_y the scale factors along the axes of the image (u and v resp.). The parameter σ_θ describes the skewness of the two image axes. Nowadays, the pixels are almost perfectly rectangular in most cameras and thus σ_θ is very close to zero.

M represents the extrinsic parameters of the camera and is given by:

$$M = \begin{bmatrix} \ddots & \vdots & \iddots & \vdots \\ \dots & R & \dots & T \\ \iddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10.3}$$

Where R is the rotation and T the translation which relates the world coordinate system to the camera coordinate system. The optical center of the images can be approximated at the true image center and the camera orientation relative to a xyz axis system may be specified by three Euler angles: ϕ, θ and ψ.

Any rotation of the camera about the z-axis (ϕ) of the camera coordinate system, causes all horizontal lines to be rotated the same amount in the resulting camera images (Figure 10.7a). ϕ is the angle between a line along the x axis of the world coordinate system and the projection of it into the image plane. As a result, ϕ can be estimated by calculating the rotation of some lines in the image known to be horizontal. Since all the seats in the train are ordered in straight rows, finding a few of these lines is very easy (see Figure 10.7b). ϕ is then approximated by averaging the angles of these lines with a real horizontal line.



(a) (b)

Figure 10.7: (a) A horizontal line in the x, y system projected onto the image surface (x', y' system). The angle formed by this projection and a horizontal line in the x', y' system determines the camera rotation parameter ϕ. (b) Lines known to lie parallel to the x-axis can be used for determination of ϕ.

To validate the result, the image can be rotated in the reverse angle. In the resulting figure (see Figure 10.8), all the chosen lines should be approximately horizontal and the train's corridor should run vertically to a single vanishing point.



Figure 10.8: Figure 10.7(b) after rotation with ϕ.

In a similar fashion θ and ψ can be estimated. For more details we refer to [116]. After determining all the parameters for the DLT model, we can fill in these values in the model and construct the projection matrix. Note that a conversion factor is needed to convert measuring units from cm into pixels (and vice versa). This conversion factor depends on the size of the CCD sensor of the camera and focus length. Since we do not know these settings a value was approximated by visual fine tuning.

10.3 Evaluation

This chapter described in brief the experiments that were conducted to collect usable data for testing and evaluating the aggression detection system. The choice to use storyboards to describe scenarios to the actors turned out to be a good one. As

opposed to rigidly scripted approaches, storyboards offer more flexibility. As a result, many different versions of the same scenario could be captured for certain scenarios.

The subsequent annotation of the captured data was greatly simplified by the implemented annotation tool. The rule-based expert system running in the background of the annotation tool allowed for real-time analysis of the annotation results on the aggression level. The ease of incorporating the expert system in the tool is a direct result of the use of the concepts, defined in the aggression ontology, as the vocabulary for the annotation language. The drawback of this close relation is that the tool is very specific for aggression in trains. Much modification is needed to make it suitable for a wider area.

Calibrating the cameras in the train turned out to be a straightforward process, given the abundance of reference and calibration points naturally available in the train compartment. In the experiments, the cameras where installed in a fixed location and had fixed settings. Given the linearity of the calibration method, we do not foresee problems in real-time calibration, even with variable settings (zooming, panning, tilting).

Chapter 11

Results

The aggression detection system has been designed to combine results from different algorithms: low level features are used to infer objects such as discrete events and humans. Higher up in the process, meaningful labels and attributes are given to the objects e.g. human objects receive attributes such as expression, position, movement. In addition environmental attributes such as compartment crowding, distance between people in the compartment, distance change etc. can be extracted. A tracking algorithm keeps track of object motion, while a behavior recognizer assigns typical behavior to detected motion patterns. Finally all this information and attributes are fused by an inference system into an assessment of the aggression level and potential in the compartment. This chapter presents the results of the separate processing algorithms followed by an evaluation of the reasoning models. The algorithmic results will be presented in the same order as they appear in the detection model in Chapter 5.

11.1 Environment modeling

11.1.1 Background subtraction

To allow for an understanding of the performance of background subtraction algorithms with respect to aggression detection in the train compartment, it is necessary to evaluate the algorithms with the actual videos taken in the train compartment. The results given in the literature are usually limited to lab conditions and are by no means representative for complex situations. The type of situation for which it is needed imposes many other problems. A background subtraction algorithm that works well in one scenario may not necessarily work as well in another. Three different algorithms where evaluated: running average, Gaussian mixture and the codebook method discussed in Section 5.2.1.

11.1.1.1 Test data

Research in background subtraction is an ongoing area with many different algorithms being developed. To make it easier to evaluate current and future algorithms, a test data set was created that contains the problem examples of train environments for current techniques. Table 11.1 shows a list of the problems.

Table 11.1: Problem examples for background subtraction.

Problem	Description
Illumination	Changes in illumination alter the color composition of the background. In color and intensity based algorithms, this change causes a large difference in the subtraction and therefore increases the number of false positives.
Relocation	If luggage is moved from one location to another, this causes change in two regions. The new position of the object and the former. Both regions will be picked up as foreground due to the change in color.
Shadows	Objects can cast shadow areas which are darker than the background color in that area that will then be wrongly classified as foreground pixels due to the illumination change in the shadow region.

In literature more problem examples are mentioned. A well known one is non-static backgrounds e.g. changing scenery visible through the train compartments windows causes fluctuations in pixel values resulting in false matches in these areas. Test data of this type was omitted because solving this problem requires high level knowledge (e.g. what is a window?). Another problem is camera movement. Unfortunately, our dataset only contains data from fixed cameras. In the future the background subtraction data set could be extended with data from moving (zoom, rotate, panning) cameras.

11.1.1.2 Results

Speed. The fastest amongst the methods evaluated is certainly the running average, where, for each pixel, the classification is just a thresholded difference and the background model update adapts just a few operations. The Mixture of Gaussians (MoG) is the slowest of the three methods with a complexity that increases with the number of Gaussian distributions used in the mixture.

Memory. The memory complexity per pixel results in similar rankings for each algorithm. The running average method requires one floating point value for each pixel. In the codebook method, five floating point numbers ($\bar{R}, \bar{G}, \bar{B}, \check{I}, \hat{I}$) and four integers ($f, \lambda, p, q$) are needed. The number of codewords depends on the variation of pixel values. The average number of codewords in each pixel is

around four. MoG requires five floating point numbers for each distribution: RGB means, a variance, and a weight.

Accuracy. The comparison of accuracy between the algorithms was limited to visual inspection of the algorithms' results. Where possible we used the advised values for parameters for the methods tested, as such, there is no guarantee that the used values are optimal for the dataset.



(a) (b) (c)

Figure 11.1: Background subtraction method using the codebook method (b) of the original image (a). Post-processing connects foreground pixels to create blobs and can recover errors (c).

11.1.1.3 Discussion

There is no perfect background subtraction algorithm that will work within all scenarios as there are so many uncontrollable variables (e.g. missing and ambiguous data). That is why many algorithms have to be tailored for the application. It is apparent that for aggression detection, fast processing is required. Thus, the choice of a background subtraction method is a trade-off between speed and accuracy. Background subtraction using the codebook method was chosen because it gives a good balance between both. In addition, the OpenCV (mentioned earlier in Section 5.3.1.1) implementation of background subtraction is coupled with post-processing filters to reduce the number of false positives (as shown in Figure 11.1(c)).

11.1.2 Sound localization

The sound captured by the four microphones during the experiments is used to test and evaluate the SSL algorithm described in Section 5.2.3.3. The test data consists of recordings of a person speaking a fixed sentence at predefined locations inside the train. We used the difference between the predicted location and the actual location of the sound source to finetune the algorithm parameters and estimate the accuracy of the method in the train compartment.

The final parameters used for the algorithm are shown in Table 11.2. For evaluation of the SSL we used the audio recordings of the aggressive scenarios[1].

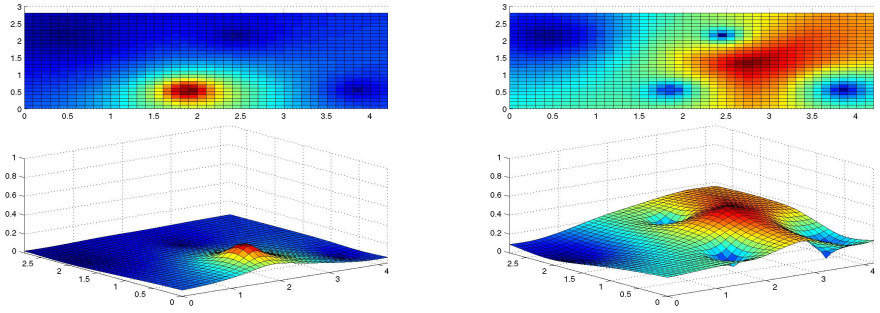
[1]The source code of the SSL software (Matlab, C++) and the test and evaluation data are available online at http://www.zheek.com/phd/software.php?option=ssl

Table 11.2: SSL algorithm parameters.

Parameter		Value
Method	:	ILD
Error measurement	:	Mean Square
Sound speed	:	343m/s
Filter	:	BANDPASS 300-8000Hz
Sampling frequency	:	44100Hz
SNR Threshold	:	1.000000
Gain compensation	:	YES
Sample length	:	200ms
Room height	:	3.00m
Room width	:	2.80m
Room length	:	4.20m
Grid density	:	0.10m

- A bandpass filter over human speech frequencies results in a huge increase in SSL accuracy as much of the engine noise is filtered out.
- When there is no sound produced, there is no reason to perform SSL. To filter out this data, signal to noise ratio (SNR) thresholding is applied on all the sound samples. The localization algorithm is only applied on the data when the signal ratio is above the threshold ratio. The SNR is obtained from known silent sound sections for each microphone separately.
- Since not all microphones are equally sensitive, the sound signal has to be normalized (gain compensation). Normalization is obtained by scaling with the energy in the silent section.
- The performance of the algorithms is influenced by the choice of the sample length. A higher sample length e.g. 500 ms means that the algorithms have more data to work on and will return more accurate results. On the other hand a high sample length may cause confusion with moving sources as shown in Figure 11.2. For the evaluated data, a good compromise between robustness and accuracy was found at 200ms.
- Grid density decides the accuracy of the SSL. By choosing a small grid density the algorithm will be less accurate but faster (and vise versa). A grid density between 10 cm turns out to work well for the chosen room size.

Because all microphones were positioned at the same height, no reliable height information of the sound source can be obtained. The result of this limitation is that the SSL returns a 2-D position of the sound source. This position is however a valuable asset fused with video information (if the sound source is near a human face). The accuracy of this position depends on the chosen grid density, the chosen sample length and the type of sound source (stationary or moving).



(a) (b)

Figure 11.2: Plot of the inverse of the mean square errors of SSL in a section of the train compartment. The four minima indicate the positions of the four microphones. The global maximum is the expected position of the sound source. A fixed source results is sharp spikes (a) while moving targets generate a larger spreading (b).

The evaluation was done in a section of the train compartment. To extend SSL to the entire compartment, about ten microphones are needed. In this case, it is advisable to perform SSL on the four microphones with the strongest signal (instead of all ten). This is because the sound signal quality declines rapidly with distance, so the weakest microphones may contain too much noise. This method is also more computationally efficient.

11.1.3 Motion estimation

To get an indication of the general movement of the detected blobs obtained from background subtraction, motion templates can be used. The silhouette of the blob is used to track its movement. By the movement of the blobs over several frames new silhouettes are captured and overlaid with the (new) current time stamp. Older motions gradually fade. These sequentially fading silhouettes record the history of previous movements and thus are referred to as the motion history image (MHI). Once the motion template has a collection of object silhouettes overlaid in time, we can derive an indication of overall motion by taking the gradient of the MHI image. We use the motion templates algorithm implemented in OpenCV (see Figure 11.3). Matching a motion template gives a (context independent) indication of the general direction of motion. This is not to be confused with the behavior templates discussed in Section 11.4.2.

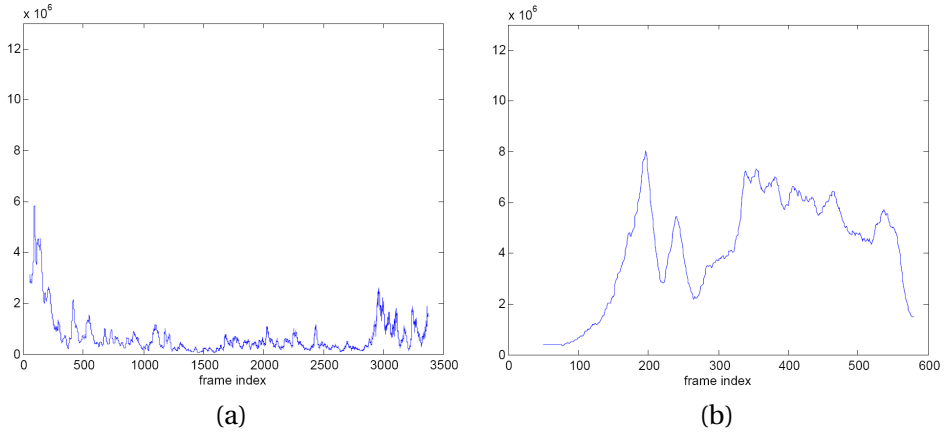
11.1.4 Energy signatures

As the MHI image assigns values to a pixel according to the amount of change (motion) observed in the last few frames, it can be used to create a energy signature of the image by summing these values. The energy signature shows the amount of change in the scene and thus gives an indication of the amount of action that is taking place



Figure 11.3: Motion estimation of a person walking through the train corridor using silhouettes of detected blobs overlayed in time.

in the scene. Figure 11.4 shows the energy distributions for two different scenarios. Optionally, the energy signatures can be calculated per masked area giving more details about the general distribution and location of the motion. Energy signatures provide a quick way to calculate the amount of motion in the train. To be useful for aggression detection however, this information needs to be combined with other attributes of the train compartment (e.g. train status) in order to infer if the amount of motion corresponds to the type of situation.



(a) (b)

Figure 11.4: The amount of motion in the compartment during certain actions results in different energy distributions. Graph (a) shows the energy during ticket checking, while (b) is the energy during the 'enter train' scenario.

Similar to the video case, the energy entropy of the audio signal can be used as a parameter to detect audio data containing burst of sounds (for violent scene detection). The energy entropy in audio is computed by dividing each audio frame into segments of K samples each. The signal energy is computed over each of these segments and normalized by the overall frame energy. Then, the energy entropy I, of the clip is defined as follows:

$$I = -\sum_{i=1}^{J} \sigma_i^2 log_2 \sigma_i^2, \tag{11.1}$$

where J is the total number of segments in the clip and σ_i^2 is the normalized energy of the i-th short segment in the clip.

11.2 Object detection

11.2.1 Human detection

The method we consider for the purpose of person detection is face detection. Given the characteristics of the compartment, other human characteristics are less effective (e.g. human body detection suffers too much from occlusion). When a face is detected in an image, obviously this means a person has been detected as well. In addition, the number of faces detected, indicates the number of people in the compartment. We used the frontal face detection method implemented in the OpenCV library, which is based on the method proposed by Viola and Jones [93].



Figure 11.5: Frontal face detection using Viola and Jones algorithm.

The Viola and Jones method is capable of accurately detecting faces for which the classifier is trained in reasonable time. It is however not very robust under noisy circumstances, and the frontal face classifier used is very susceptible to changes in

orientation. In larger frontal faces of sizes around 50x50 pixels, we achieved very good detection rates. A snapshot of a video frame with all faces detected successfully is shown in Figure 11.5. However, if the size of a face drops below this threshold, detection rates fall dramatically, to a point where they hardly contribute to detection at all. In practice, faces to a resolution of 16x16 pixels can be detected, however the reduced size of detectable features and the higher signal to noise ratios in these smaller areas result in a very high rate of false negatives and false positives. This is illustrated in Figure 11.6.

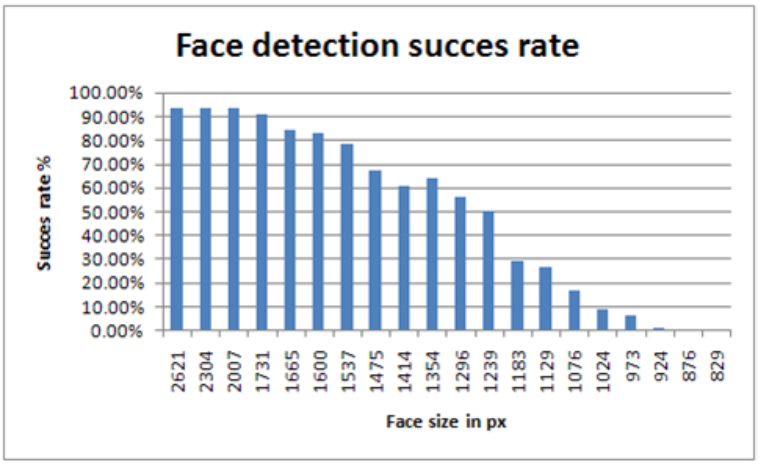


Figure 11.6: Face detection success rate for various face sizes.

Given the number of false positives, determining the number of people by counting the number of faces, is inaccurate to say the least. To get more accurate results, we first use a mask filter to remove all positives from areas where no faces are expected to be found. This excludes areas such as the windows and the ceiling, where false positives would commonly occur. Another solution is to install better (higher resolution) and more cameras.

Another attribute of the train compartment derivable from face detection is the distance between people in the train and the existence of clusters of passengers. Because face recognition gives a two dimensional x, y location of a face (and frontal faces can essentially only be captured by one camera), we have to estimate the third coordinate i.e. z. Currently, the z coordinate is roughly estimated using the size of the detected face (the bigger the face, the closer the person is to the camera).

11.2.2 Emotions

Emotion is the main indicator of affective aggression in the system. An active topic of research is the detection of human emotion from faces. However, the faces detected in the video data are usually too small to allow for reliable emotion detection. Instead we turn to the audio data for emotion detection.

Vamp[2] is an open source audio processing plugin system for plugins that extract descriptive information from audio data by means of audio feature extraction plugins. Implementations of extraction algorithms for the emotion related features described in Section 5.5.3 are available as Vamp plugins.

To obtain the emotion from audio features, the correlation values between the audio features and the emotion components Valence, Activation, and Dominance of the speakers described in [55] can be used. At the moment of writing, audio data from the harassment, hooligan and ticket checking scenarios are being used to test the emotion detection algorithms.

11.3 Tracking

We apply greedy nearest-neighbor matching to construct most probable tracks from the coordinates of detected faces. To account for false negatives, we search over a maximum of 10 frames increasing the search area by 5 pixels every frame without a face found. This produces satisfactory results as can be seen in Figure 11.7.

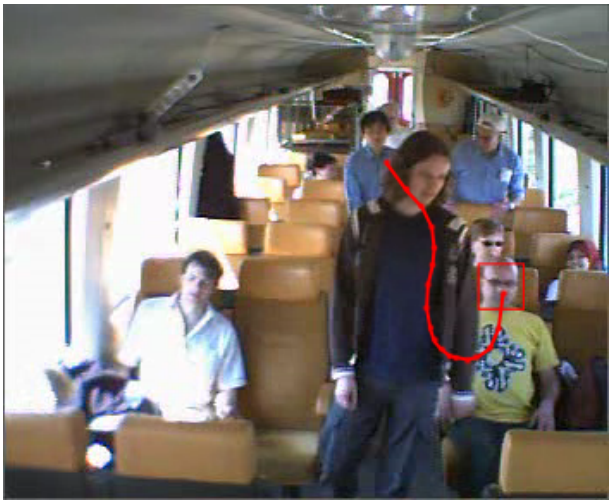


Figure 11.7: Track plot for fragment of scenario 8b, overlayed on last frame of the scenario. The track is manually enhanced to improve visibility.

To guard against false positives, we can apply a mask on the train compartment. Masking exploits the typical usage of certain areas in the train compartment. As every area has a very clear designation, this method is well suited. A mask designates an area and assigns values to expected behaviors for each location in that area. For example, the probability that a passenger will be walking is very low in the areas containing the seats and in the corridor we would expect passengers to be walking instead of sitting. We can define an area as a polygonal shape (see Figure 11.8 (b)).

In the border area between segments a softer division in the form of a probability for each pixel is made, with higher probabilities at the centers, and lower probabilities at

[2]http://www.vamp-plugins.org



(a) (b)

Figure 11.8: Empty train (a) and designation of areas for masking (b).

the edges of these segments. Remaining areas, or areas that are in a location where one would not expect a face to be detected, can be classified with an 'error' mask. For example, Figure 11.9 shows masks for the ceiling and window areas of the train compartment as these areas are unlikely to ever contain faces.



(a) (b) (c)

Figure 11.9: Scene masks for walking (a), sitting (b) and error/undefined behavior (c). The probability of the mask is visualized here as intensity values.

Other prediction methods have also been studied. Because of the confined space of the train compartment and the possible irregular movement of humans in it, model based methods, such as Kalman-filtering, have a hard time with the data.

11.4 Activity recognition

11.4.1 Unusual behavior

In Figure 11.8 we defined three areas to detect, those designated for walking, sitting and an undefined area in which we would expect only false positives or unlikely scenarios. We can use the error mask to quickly discard data we are not interested in, or which we simply are not able to classify accurately. For example when the train is in the station, the pixels in the corridor could have a mask value $M_{walk} = 0.6$ to indicate the high probability of observing walking in this areas, while the other actions are considered less possible, giving mask values $M_{stand} = 0.3, M_{sit} = 0$ and

$M_{error} = 0.1$. Alternatively, the mask can be used to simply detect people being in a certain (prohibited) location, regardless of their actions.

We also allocate areas for different scenarios, for example by using different scene masks during stops and when the train is moving. An example is given in Figure 11.10 where the scene is divided in segments according to seat numbers. It is most occupied when the train is moving, as we would expect most passengers to be seated. We can then alternatively use the walking masks (visualized in Figure 11.9(a)) to monitor the amount of movement during this phase, as we would expect it to be much less compared to when the train is stopped.

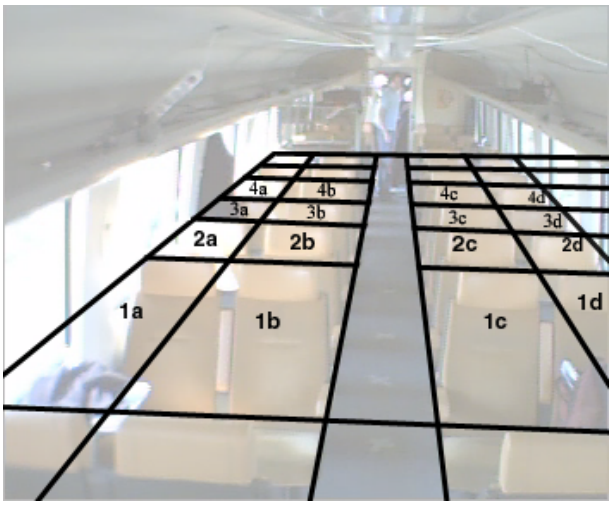


Figure 11.10: Scene segmented by seat numbers.

11.4.2 Template-based classifier

The template-based classifier compares the observed path of a person to typical paths for certain actions. A typical path or template is a specific observation of a complete action, that is, a path of observed locations of an actor over time, for which we know the action. The template consists of the coordinates and timestamps of named actions in the train (e.g. walking, running, sitting, begging). The location and timestamps are manually determined from experiment data. Thus templates are basically lists of locations-time couples of an actor during a particular action. For example, during the action 'walking' we observe the actor moving from one point in the frame to another, at a reasonably constant speed. For the action 'running' we would observe a similar path, only at a higher speed.

When we have observed a path using our surveillance system, the matching algorithm produces tracks of objects for the template matcher to identify. The template matcher chooses the best match with the template most closely resembling the track taken by the object.

The typical behavior template contains about 15 to 40 data points, at successive time intervals, but larger templates can be constructed to detect more complex behaviors. This set of coordinates is then used to compute the similarity with an observed path, expecting similar behaviors to have similar paths. We use the Mean Square Error (MSE) as a measure for similarity. We can then compute the MSE for all coordinates separately. To score an entire measurement against a template we compute the sum of these errors, and divide it by the number of points compared.

11.4.2.1 Discussion

The values for each pixel in the frame can be looked up when classifying a path. We expect a passenger walking through the middle of the train corridor to be occupying mostly points where the mask values will be similar to (M_{sit} =0, M_{walk} =1, M_{error} =0). Taking the average of these values over an entire observed path would return the highest average value for the 'walk' mask. This value can then be used to determine the action observed in a path.

This method can also be used to split a path in two. When a passenger walks through the middle of the train and then sits down, there will be a clear difference in the observed mask value somewhere along the path. Since our template matching method only uses general templates containing one action, we can use the mask values to identify the possibility of two or more actions in this track, and therefore split the track into pieces which are more likely to match a part of a template.

The tracks tend to be smaller in length than our templates, about 8-32 frames. This is because of the false negatives occurring in the detection data, creating gaps in the measurements. Therefore we try to match a large enough subset of behavior to a template. We used a minimum length of 10 frames containing three or more positive data points, but more data is desired for accurate results.

Even then a measurement may not be complete however, and therefore not cover the entire template. We might observe a passenger walking in the front of the train, while our template for that action covers a path from the front up to the back of the train. We must therefore try to match our measurements to the corresponding part of the template. To find the best correspondence we shift our measurement along the template, and use the lowest average error value because it is the best 'fit' for our measurement.

Finally, the activity recognition is based on tracks created from the results of face detection. Optionally, additional tracks can be based on blobs acquired from the background subtraction algorithm. Blob tracking is more robust in terms of detection rate, but face tracking is more accurate in the location of the object.

11.4.3 Abandoned luggage

Detecting abandoned luggage is an important function for the aggression detection system, amplified by the risks of terrorist attacks. As the original dataset inside the train compartment does not contain scenarios of abandoned luggage, an extra dataset was created of such scenarios at train stations. The method to detect abandoned luggage consists of two major components. First, there is blob tracker for tracking and detecting moving objects. In addition, there is a stationary object detector to mark and detect the abandoned objects.



Figure 11.11: A blob being tracked from frame to frame. The dotted bounding box illustrates the blob in the previous frame.

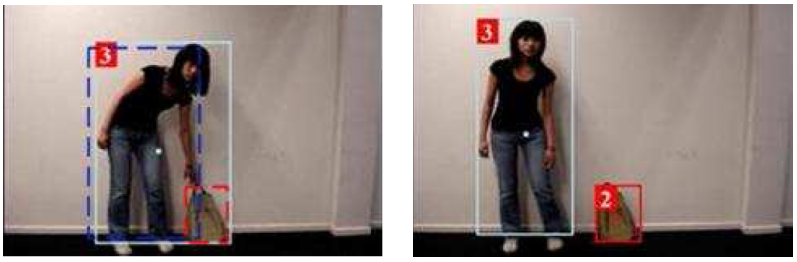


Figure 11.12: Blob 3 with child 2 split into separate blobs 3 and 2.

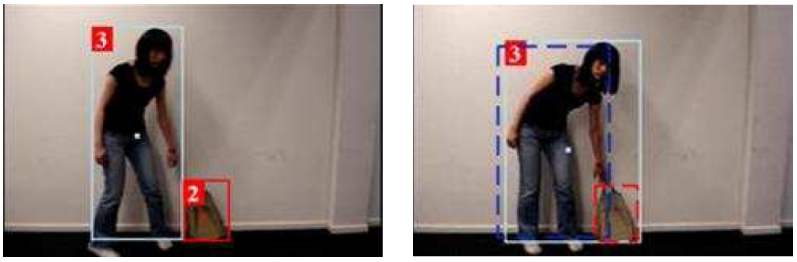


Figure 11.13: Blob 3 and 2 merged into root blob 3 with child blob 2.

The abandoned luggage detection method works as follows:

- Each detected object is assigned a unique ID number (Figure 11.11). This number will be staying with the object during the whole process, from splitting to merging.
- The detected objects are tracked and analyzed from frame to frame to detect object splits and identify the drop-off event.
- During a drop-off event, blobs are spitted (Figure 11.12). To identify the human from the bag, a decision tree is employed that takes into account the sizes and the motions of the detected child and parent objects.
- The distance of the bag to its owner is calculated at each frame. To calculate the distance, the center of mass of each object is used. If this distance surpasses a predefined threshold, the conclusion of "abandoned luggage" is made.
- If on the other hand the owner picks up his bag, a merging of blobs occurs (Figure 11.13).

11.5 Results of reasoning methods

In this thesis three different reasoning methods for aggression detection were described. In order to get evaluation results that are unbiased by errors in the detection components, the evaluation was conducted with ground truths. The ground truth is gathered after a process of manual enhancement of the results of the low level processing algorithms (see Figure 11.14). Because the low level processing results comply with the ontology vocabulary, the ground truths are also ontology compliant annotations.

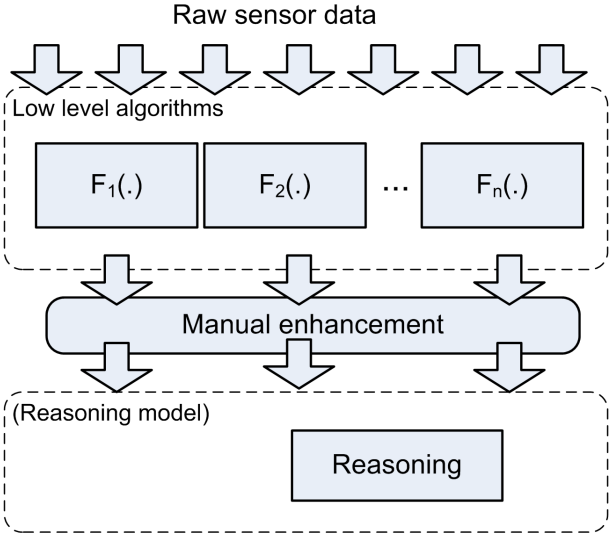


Figure 11.14: Evaluating reasoning methods with ground truth values.

11.5.1 Rule-based system

The rule-based aggression detection system performed and behaved as designed. Inserting the correct input in the rule base results in the expected conclusions. As a side project of the evaluation of the rule-based system, an aggression reporting tool was developed that allows bystanders (passengers in the compartment who are witness of aggression) with a smartphone to report their observations to a control center.



Figure 11.15: Screen shot of the icon reporting tool.

The system consists of three parts: a user interface to present icons to the reporter (Figure 11.15), an icon selection module that populates the user interface with icons from the competing hypotheses and a message generation module that fills in the missing icons once a conclusion has been made. The message generation module is also responsible for generating the appropriate message to send to the control center. In the control center, the rule-based reasoning system gathers the reports and draws conclusions based on them. The rule-based reasoning system has been tested with several people with smartphones sending reports. For quantitative results, we refer to [117].

11.5.2 Discussion

In this chapter we used the real data to test and evaluate the aggression detection system and its individual components. Since we used existing algorithms, the evaluation was based on a test set of selected “problem cases”. We purposively refrained from going into too much detail of the quantitative evaluation, but tried to give an overview of the chosen solution and the modifications that had to be made to the algorithms in order to better cope with the data in the test set.

The final aggression detection system consists of several discrete object and event recognizers (e.g. faces, movement and movement change, sound source, emotion). In addition, crowd attributes such as crowd size, inter-distance and total movement, are calculated. A tracking algorithm keeps track of human motion, while an activity recognizer tries to assign labels to these motion patterns. Finally, high-level inference engines give a judgment of the situation based on the available data.

An evaluation of the rule-based system was also presented in this chapter. The BN system and the emergent system have also been evaluated although not as thoroughly as the rule-based system. One can argue the point of such an evaluation. Indeed, in the case of the BN, the CPTs are filled in by experts, if the BN does not give correct answers, then the solution is a matter of modifying the CPT values until it does. For the emergent case, it is the strength of the links between concepts that can be finetuned. In both cases it is a lack of evaluation data that prevents us from deriving other conclusions about the performance of the reasoning methods. There is not enough data to conclude if the observed increased accuracy is a result of better inference or overtraining.

Chapter 12

Evaluation and conclusions

This thesis reports on research to build an automatic, multi-modal, aggression detection system while maintaining the privacy of train travelers. This has been accomplished by an integration and implementation of ideas and results from signal processing and AI. The aggression detection system presented in this thesis evolved from analyzing and observing aggression and the way human experts observe video images. Also current audio, visual and reasoning methods and algorithms were explored to try to design a system that can mimic the human recognition process.

The significance of our research lies in the novel application of existing technologies and development of new concepts to tackle the problem of data fusion for the surveillance of people in a train compartment. Compared to previous work, we adopt a more human centered approach to the detection problem by extracting knowledge and rules from security experts. As a result, different concepts and techniques have been used to solve existing technical problems, such as situation awareness, dealing with uncertainty and temporality. In addition we aim to solve the issues identified in previous projects such as difficulty with testing and comparing systems, by means of a software data fusion framework designed specifically for multi-sensor, multi-modal and multi-algorithm systems.

A description of the principles of the aggression detection system in the terminology of JDL fusion model can be of considerable advantage to a systematic use in future projects.

12.1 Summary of results

A system was designed to function as a warning system working alongside a human operator. The system indicates which compartments have a high probability of aggression. To calculate the probability of aggression, the system mimics human behavior and reasons with human knowledge. The achievements in the thesis are summarized below:

Aggression. Aggression is usually not directly identifiable from observable features. It can only be detected by considering features over time using, combining and reasoning about higher level concepts. In addition aggression always needs to be interpreted in the proper context. We categorized the different forms of aggression known in the research field and interviewed human experts to obtain the aggression-related cues and the reasoning process behind human aggression detection.

Aggression ontology. In order to achieve an aggression detection system with predictive abilities, prior knowledge of possible aggressive situations is used. A common vocabulary (and the underlying semantics of concepts in the vocabulary) was established using an XML language. The meaning and understanding of ontologies as well as the range of purposes that ontologies may serve was examined. We examined the best way ontologies can be built and the appropriate software to use for the use of the aggression ontology.

Aggression detection model. The detection model in this work imitates the human expert. Our aggression detection approach is essentially divided into two models: (1) the observation model which describes how low level features from observations are combined into high level concepts and (2) the reasoning model in which high level concepts are reasoned with in order to infer the presence of aggression. From earlier interviews with experts, it became clear that in the human aggression detection process, the human reasons in terms of high level concepts. Thus the reasoning model resembles human aggression detection.

Framework. A data fusion framework was built to facilitate the different fusion algorithms and allow them to communicate and use each others' results. The framework also makes it possible to easily replace or modify a component without altering the remainder of the application. As a result, it was possible to evaluate and choose the best technique, for each component of the system, from various competing techniques. The framework uses terminology from the JDL model for easy comparison and re-use with other similar systems.

Aggression detection system. On top of the framework the actual aggression detection system was built, based on a computational model consisting of low level components (that extract relevant features from audio and video data to combine them to high level concepts) and higher level components (that reason with higher level concepts to come to a conclusion). At the lower level an observation model was implemented to bridge the gap between the digital representation of sensor data to the high level semantic concepts encountered in human expert reasoning. The implementation consists of a number of off-the-shelf detection and classification algorithms (e.g. faces, movement and movement change, sound source, emotion). In addition, crowd attributes such as crowd size, inter-distance and total movement, are calculated. A tracking algorithm keeps track of human motion, while an activity recognizer tries to assign labels to these motion patterns.

Data collection. We used storyboards to specify scenarios of aggressive behavior. The scenarios were verified by a professional conductor on realism. In experiments in a train compartment, we used the storyboards to indicate to actors the scenarios we wanted them to perform. As the actors performed the scenarios, audio and visual data of their actions were captured with the cameras and microphones located in the train. We used the raw data to evaluate and compare separate feature extraction and classification algorithms and to test and evaluate the aggression detection system. For re-usability, the captured data was also annotated using an annotation language derived from the train aggression ontology. An annotation tool was developed to help in the annotation process.

High level inference. High level inference engines give a judgment of the situation based on the available data and prior knowledge. In this thesis we evaluated three inference approaches rule-based, Bayesian and emergent. The strengths and weaknesses of each approach for our particular problem are outlined in separate chapters.

12.2 System evaluation

Many proponents of automatic surveillance claim that it has been shown to be an effective crime deterrent. There are pitfalls in these studies that should also be taken into account when evaluating the automated aggression detection system in real life. The main issue to guide against is that the results may suffer from the Hawthorne Effect. The Hawthorne Effect refers to a problem that plagues any study of human behavior. The mere fact that the behavior is studied influences the behavior that is studied in the short term. In other words, the novelty of the system might, during the time that the system is evaluated, decrease the number of aggressive incidences. But over time, the novelty wears off and as people learn the weaknesses and blind spots of the system, aggression might revert to its previous levels. Another possibility is that the aggression decreases in the area under surveillance but is just displaced somewhere else without surveillance.

A different aspect in the evaluation of the system is the public acceptance. The British Columbia Civil Liberties Association[1] has compiled a list of criteria for video surveillance systems to be acceptable. This list has been adapted for the automated multimodal sensor case. In order to be acceptable, automatic surveillance of a public place must:

- Fulfill an important purpose, such as reducing the risk of physical harm or other illegal activities and not simply the control of nuisance behavior.

[1] http://www.bccla.org/positions/privacy/99videosurveillance.html

- Not simply drive a problem from one area into another area that does not have surveillance.
- Be less invasive of privacy than alternative means of addressing the problem. The privacy implications of this system has been explored in [118].
- Be advantageous to all, or at least to most of the people who are giving up their privacy.
- Provide the public with clear notification of its presence in the areas where surveillance occurs.
- Inform people of its rationale.
- Inform the public about who/what is monitoring the cameras, what use will be made of the data, how long it will be stored and other practical details.
- Not be used for other purposes.
- Be more efficient than alternatives.

12.3 Future work

Much improvement of the system can be achieved by adopting better low level detection algorithms, or algorithms that are better trained on the train environment. Because we have implemented or used third party software (when available) we will not focus on them here. Instead, we will discuss several directions for future work and research. We both present alternative approaches and extensions to our presented methods.

- Three sets of experiments were conducted to gather data to train the system. Despite these efforts there is a need to gather more data to get more examples of aggressive situations (e.g. recognition of environmental sounds could not be tested, because there was no suitable test in the dataset) and to construct a standardized training and test set for future systems.
- Construction of a sub-ontology describing the train compartment concepts in more detail (for example, windows, toilets, exit door, emergency exits etc.). Other details which are important for the lower-level features could be designed in a separate sub-ontology. Ontologies can easily be coupled together, using name spaces. Our designed aggression ontology can be used as a good basis for extension.
- In the Bayesian network approach we have focused on the BN representation which only represents a snapshot of the situation in the train compartment. In order to follow the process as it unfolds over time, the Dynamic Bayesian networks (DBNs) extension is proposed. The DBN models processes that evolve

over time. A DBN can be defined by two BNs: a prior $P(X_1)$ and a transition model that defines how the variable at a particular time instance depends on the nodes at the previous time step.

Appendix A

Annotation schema

In the W3C standard a schema requires one unique root element. The element 'Annotations' represents this required element. The 'Annotations' element consists of an unbounded number of annotations and sensors.

```
<xs:element name="Annotations">
   <xs:complextype>
      <xs:element name="Sensor"
                        type="SensorT"
                        maxOccurs="unbounded" />
   </xs:complextype>
   <xs:complextype>
      <xs:element name="Annotation"
                        type="Annotation"
                        maxOccurs="unbounded" />
   </xs:complextype>
</xs:element>
```

The detection of any event is due to some computations being done on the readings of a certain sensor. Each annotation should therefore include a description of the sensor and its settings at the moment of the event detection. A sensor itself is identified by its SensorID, in addition a sensor has a location and a position. In a concrete XML file, a sensor is referred to by its SensorID.

```
<xs:simpleType name="SensorT">
   <xs:attribute name="SensorID" type="xs:string" />
   <xs:element name="Location" type="xs:string" />
   <xs:element name="Position" type="xs:string" />
</xs:simpleType>
```

According to the schema language definition, the XML schema language supports object orientation, more specifically inheritance. Here we define a base element for an annotation called 'AnnotationBase'. It describes the most basic attributes of any annotation of events in multimedia: a reference to the sensor, a start time and a duration (for video this might be cam number, frame number and frame count). Furthermore, each annotation has a flag indicating if the annotation was provided by a user, or calculated automatically e.g. by interpolation, logical inference.

```
<xs:simpleType name=AnnotationBase">
   <xs:attribute name="sensor" type="xs:integer" />
   <xs:attribute name="starttime" type="xs:positiveinteger" />
   <xs:attribute name="duration" type="xs:positiveinteger" />
   <xs:attribute name="is_manual" type="xs:boolean" />
</xs:simpleType>
```

We have to distinguish between the different modalities in the description. Some events are non-informative without the audio, so audio is an important aspect to include in the inspection window. However, the demonstrator should be made (visually) aware of this, therefore, the event description should specify which modality the event applies to. We create two special elements to achieve this. Typically, the attributes 'is_visible' and 'makes_sound' indicate which modality is implied.

```
<xs:simpleType name="video_properties">
   <xs:attribute name="is_visible" type="xs:boolean" />
   <xs:attributes name="x_upleft" type="xs:positiveinteger" />
   <xs:attributes name="x_upright" type="xs:positiveinteger" />
   <xs:attributes name="y_upleft" type="xs:positiveinteger" />
   <xs:attributes name="y_upright" type="xs:positiveinteger" />
</xs:simpleType>

<xs:simpleType name="audio_properties">
   <xs:attribute name="makes_sound" type="xs:boolean" />
   <xs:attribute name="sound_source" type="3dlocation" />
</xs:simpleType>
```

Some events can only be described as an interaction between different objects, for example person A hits person B. To ease the description of events, events should be able to refer to objects (or maybe even other events). For example person A could be doing something else a few frames after hitting person B. It is very convenient to be able to refer to person A instead of having to redefine person A every time he does something. Therefore, we introduce an element called 'object' (which could be a person) with an id that can be used as a handle for referencing the object in other

annotations. As an 'object' also has to be detected by sensors, it inherits from 'AnnotationBase'. Because of the special role people play in aggression, a special 'person' element is extended from the 'object' element.

```
<xs:simpleType name="object" AnnotationType="object">
inherits all for AnnotationBase
   <xs:attribute name="id" type="xs:string" />
   <xs:attribute name="location" type="3dlocation" />
   <xs:element name="video_properties" type="video_properties" />
   <xs:element name="audio_properties" type="audio_properties" />
   <xs:element name="parts" type="object" />
</xs:simpleType>

<xs:simpleType name="person" AnnotationType="person">
   inherits from object (with parts => head, torso, hands, feet)
   <xs:attribute name="pose" type="xs:string" />
   <xs:attribute name="gesture" type="xs:string" />
   <xs:attribute name="activity" type="xs:string" />
</xs:simpleType>
```

The main element in this schema is the 'event' element. By properly interpreting the attributes of the 'event' element, one should be able to deduce exactly what the event is about. However, this might prove to be difficult in the real world in real time. If something goes wrong in the interpretation, it might be confusing to have no idea what is going on or why the system thinks the situation is aggressive. For this sake, we added a 'description' attribute that should tell the story. A quick glance at the description of the events should be enough to know what is going on.

```
<xs:simpleType name="event" AnnotationType="event">
inherits all for AnnotationBase
   <xs:attribute name="objectid" type="xs:integer" />
   <xs:attribute name="role" type="xs:string" />
   <xs:attribute name="subjectid" type="xs:integer" />
   <xs:attribute name="description" type="xs:string" />
</xs:schema>
```

To give an indication of the result, following is a sample annotation file.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>

<Annotations>

<!-- There is only one sensor: camera 4 -->
```

```
<Sensor name="camera 4" SensorID="cam4">
   <Location>"In the Train"</Location>
   <Position>"0, 1.5, 2.5"</Position>
</Sensor>

<!-- Introduction of our players: Boyfriend, Harasser and Girlfriend -->
<Annotation type="person"
            name="Boyfriend"
            id="1"
            sensor="cam4"
            starttime="0"
            duration="50000"
            is_manual="true">
   <video_properties is_visible="true"
                     x_upleft=""
                     y_upleft=""
                     width=""
                     height="" />
   <audio_properties makes_sound="false" sound_volume="0" />
</Annotation>

<Annotation type="person"
            name="harasser"
            id="2"
            sensor="cam4"
            starttime="0"
            duration="50000"
            is_manual="true">
   <video_properties is_visible="true"
                     x_upleft=""
                     y_upleft=""
                     width=""
                     height="" />
   <audio_properties makes_sound="false" sound_volume="0" />
</Annotation>

<Annotation type="person"
            name="Girlfriend"
            id="3"
            sensor="cam4"
            starttime="0"
            duration="50000"
            is_manual="true">
   <video_properties is_visible="true"
```

```
                    x_upleft=""
                    y_upleft=""
                    width=""
                    height="" />
   <audio_properties makes_sound="false" sound_volume="0" />
</Annotation>

<!-- Now come the events -->
<Annotation type="event"
            id="10"
            starttime="0"
            duration="50000"
            is_manual="true"
            objectid="1"
            role="stands"
            description="Person stands up">
   <video_properties is_visible="true"
                    x_upleft=""
                    y_upleft=""
                    width=""
                    height="" />
   <audio_properties makes_sound="false" sound_volume="0" />
</Annotation>

<Annotation type="event"
            id="11"
            starttime="0"
            duration="50000"
            is_manual="true"
            objectid="1"
            role="walks"
            subjectid="2"
            description="Person stands in front of other person">
   <video_properties is_visible="true"
                    x_upleft=""
                    y_upleft=""
                    width=""
                    height="" />
   <audio_properties makes_sound="false" sound_volume="0" />
</Annotation>
...

</Annotations>
```

Appendix B

Scenarios

B.0.0.1 Harassment

In the harassment scenario a man walks into the train and starts talking thrash to a girl sitting next to him. At a certain moment the girl leaves the train and our harasser focuses his attention on another girl. (A) Unfortunately for him, this girl is taken. The boyfriend of the girl starts a fight with the harasser. (B) The girl is alone, she walks away from the man.

B.0.0.2 No ticket

A train conductor is checking tickets in the train. (A) At a certain point she encounters a passenger that does not have a ticket. The conductor engages in conversation with the passenger. After a while it becomes clear that the passenger has no money and does not want to show identification. The conductor threatens to call the police, the passenger stands up and tries to persuade the conductor not to do that. The conductor remains calm and the situation is resolved in the end. (B) All passengers have their tickets. The ticket checking ends without incidents.

B.0.0.3 Asocial hooligan

A hooligan or group of hooligans enter the train with a beer bottles and other objects typically carried by soccer supporters. They start shouting, yelling and hitting the tables. They try to provoke other passengers into joining them. (A) The passengers do not respond. (B) Supporters from the other team are in the compartment. More yelling, shouting and fights start.

B.0.0.4 Drunkenness

The scenario starts as a man walks through the corridor of the train, the way of walking clearly indicates drunkenness. In this drunken state, the man starts to shout (at no one in particular). Other passengers feel uneasy. Finally, a conductor enters and calms him down. The conductor stays in the neighborhood to keep an eye on him.

B.0.0.5 Beggar

A beggar traverses asking for money. He is provoked by the passenger to do tricks for his pay. But as the tricks do not seem to produce the expected results for the beggar he starts attempts to physically get money. The scenario ends when the beggar realizes the futility of the attempt and moves away.

B.0.0.6 Medical emergency

As two passengers (a couple) walk through the corridor of the train, one of them suddenly falls down and passes out. The other passenger panics and starts making wild gestures and helpless sounds. Medical help is called (by mobile phone) and will be waiting at the nearest station. The panicked passengers is calmed down by other passengers. The scenario ends when the calmness returns.

B.0.0.7 Mobile phone harassment

People talking on their mobile phone in the train can cause great annoyance among other passengers. In this case a lady is unfortunate as her mobile phone goes off while she is in a train with a group of clowns. The lady tries to speak through the phone, but is over-shouted by the crowd. As she tries to speak louder, the clowns just shout harder and accompany their shouts with gestures. In the end, the lady walks away with a very annoyed expression.

B.0.0.8 Smoking

A passenger is smoking in a train compartment (which is not allowed). Another passenger asks if he can stop smoking. (A) The smoking passenger stops smoking or (B) continues smoking and starts acting aggressive as he is reminded or (C) is attacked for not complying to the request.

B.0.0.9 Shoes on seat

A teenager puts his shoe on the opposite seat. Because this is not proper behavior and also because his shoes are dirty, he is asked to stop. This scenario branches the same way as in the smoking scenario.

B.0.0.10 Graffiti and vandalism

In an otherwise empty compartment, (A) some passenger applying graffiti and (B) others are vandalizing the seats.

Summary

In many public places multiple sensing devices, such as cameras, are installed to help prevent unwanted situations such as aggression and violence. At the moment, the best solution to reach a safe environment requires human operators to monitor the camera images and take appropriate actions when necessary. In the wake of the terrorist attacks of September 11 2001, there has been a rapid growth in the volume of security cameras and other sensing devices for anti-terrorism and other security purposes. The increased application of these, often multi-modal, sensors has caused a digital data explosion that human operators have difficulty to keep up with. The need for a fully or partially automated system becomes all the more prevailing.

The main aim of this thesis is to report on our work to address the complex challenges that arise within the context of multi-modal automatic surveillance applications. In this thesis work, a multi-modal aggression detection system was built that fuses audio and video data from sensors located in a train compartment. Compared to previous work, we adopt a more human centered approach to the detection problem by extracting knowledge and rules from security experts. The aggression detection system is based on many hours of observing and studying professional operators at work as they analyze and respond on surveillance data.

Our aggression detection approach is essentially divided into two models: (1) the observation model which describes how low level features from observations are combined into high level concepts and (2) the reasoning model in which high level concepts are reasoned with in order to infer the presence of aggression.

In the observation model, feature extraction algorithms are used to transform audio and video signals into features, which are combined by classification algorithms into high level concepts. In the thesis, an analysis is made of the train compartment in particular, on the objects and situations that may be encountered in the train compartment. This analysis is formalized in a train aggression ontology. In addition an overview of relevant audio and video feature extraction and classification algorithms is given. Also the JDL model is introduced as a way to structure the wide range of available algorithms.

In the reasoning model knowledge of the human expert and high level reasoning is used to infer the presence of aggression. In essence this boils down to combining the results of the observation model to a description of the current scenario, and

comparing this to known scenarios. If the current scenario is similar to a known unwanted scenario or if the current scenario deviates too much from a known normal scenario, an alarm situation may be announced. There are a number of different approaches to accomplish the inference. In this thesis, three different inference methods are explored for their merits in aggression detection: expert system based reasoning, Bayesian reasoning and self organization/emergent reasoning.

To test and verify the results, several experiments were conducted in a real train. During the experiments, actors had to perform scenarios as described in storyboards. The storyboards where previously validated by security experts for their realism. As the actors performed the scenarios data was captured using multiple cameras and microphones. The acquired data was annotated using the vocabulary from the train aggression ontology and used as ground truths for the evaluation of the aggression detection system.

Samenvatting

Op veel publieke locaties worden steeds meer opnameapparatuur geplaatst om ongewenste situaties zoals agressie en geweldpleging te helpen voorkomen. Op dit moment is de beste oplossing om tot een veiligere omgeving te komen, het inzetten van mensen om de opnames te monitoren en waar nodig actie te ondernemen. Sinds de terroristische aanslagen van 11 september 2001, is de inzet van bewakingscamera's en andere opnameapparatuur tegen terroristische aanslagen en andere onveiligheden scherp toegenomen. De toename van deze, vaak multi-modale, sensoren heeft tegelijk geleid tot een digitale data-explosie die beveiligingsmensen moeilijk kunnen verwerken. Er is dus behoefte aan gehele of gedeeltelijke automatisering van het systeem.

Dit proefschrift is een rapportage van het onderzoek dat is verricht om de complexe uitdagingen die zich voordoen bij automatische surveillance-applicaties het hoofd te bieden. Tijdens dit onderzoek is een multi-modaal agressie detectie systeem gebouwd dat audio-en videodata van sensoren in een treincoupé combineert. In vergelijking met andere onderzoekers, staat bij ons de mens centraal. We gebruiken kennisregels, verkregen van beveiligingsdeskundigen om het detectie probleem aan te pakken. Het agressie detectie systeem is gebaseerd op vele uren observeren van beveiligingsdeskundigen terwijl zij aan het werk zijn.

Onze agressie detectie aanpak is in essentie opgedeeld in twee modellen: (1) een observatiemodel dat beschrijft hoe kenmerken van observaties op laag niveau kunnen worden gecombineerd tot concepten op hoog niveau en (2) een redeneermodel waarin met concepten wordt geredeneerd om de aanwezigheid van agressie af te leiden.

In het observatiemodel worden kenmerk extractie algoritmen gebruikt om de audio- en videosignalen om te zetten in kenmerken. Deze kenmerken worden in daarop volgende stappen door classificatie algoritmen gecombineerd tot hoog niveau concepten. In dit proefschrift is een analyse gemaakt van een trein coupe. In de analyse is in het bijzonder rekening gehouden met de objecten in the trein en de mogelijke situaties die kunnen voorkomen. De analyse is formeel vastgelegd in een trein agressie ontologie. Ook is er in dit proefschrift een overzicht gegeven van relevante kenmerk extractie en classificatie technieken. Het JDL model is gepresenteerd als een methode om dit brede scala aan technieken overzichtelijk te structureren.

In het redeneermodel wordt kennis van de beveiligingsdeskundige met behulp van automatische redeneertechnieken gebruikt om de aanwezigheid van agressie af te leiden. Dit proces bestaat in essentie uit het combineren van de resultaten van het observatiemodel in een beschrijving van het scenario. Dit scenario wordt vergeleken met de bekende scenario's: lijkt het scenario op een bekend agressief scenario of wijkt het scenario af van alle bestaande normale scenario's, dan wordt een alarmsituatie afgekondigd. Er zijn verschillende theorieën om te redeneren over scenario's. In dit proefschrift worden er drie uitgewerkt voor agressie detectie: expertsysteemgebaseerd redeneren, Bayesiaans redeneren en zelforganisatie- of emergent redeneren.

Om de resultaten te testen en te verifiëren, zijn er een aantal experimenten gedaan in een echte trein. Tijdens deze experimenten moesten acteurs scenario's, beschreven in storyboards, uitbeelden. De storyboards waren van tevoren beoordeeld op realisme door beveiligingsdeskundigen. Terwijl de acteurs de scenario's aan het uitbeelden waren werd de data opgenomen met meerdere camera's en microfoons. De opgenomen beelden zijn later geannoteerd met het vocabulaire van de trein-agressie ontologie en gebruikt als grondwaarde voor het evalueren van het agressie detectie systeem.

About the author

Zhenke Yang was born in Guangdong in China on October 14th 1977. In 1980 he moved to Suriname, where he spent most of his childhood in the capital Paramaribo. In 1995 Zhenke finished secondary school and went to study Technical Informatics at the Delft University of Technology in the Netherlands. In 2001 he obtained a Master degree in Technical Informatics, at the Knowledge Based Systems (KBS) group. The subject of his Master thesis was “Speech Interfacing in the Wireless Automobile Messaging Pilot”. The research was conducted at CMG in Rotterdam. After finishing his thesis, Zhenke continued working at CMG as a technical software consultant for two more years. In 2005 he started his PhD. research at the Man Machine Interaction (MMI) group at the Delft University of Technology. The research was conducted under supervision of Prof. Dr. Ir. L.J.M. Rothkrantz.

Bibliography

[1] H. Ferwerda, G. Verhagen, and E. de Bie, "Onderweg naar een veiliger openbaar vervoer 2004," Ministerie van Verkeer en Waterstaat, Adviesdienst Verkeer en Vervoer, June 2005.

[2] Z. Zhu, W. Li, and G. Wolberg, "Integrating LDV audio and IR video for remote multimodal surveillance," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2005)*, vol. 3, 2005, p. 10.

[3] D. Datcu, "Multi-modal recognition of emotions," Ph.D. dissertation, Delft University of Technology, 2009.

[4] A. G. Chitu, "Visual speech recognition," Ph.D. dissertation, Delft University of Technology, 2010.

[5] S. A. Velastin, B. L. Maria Alicia Vicencio-Silva, and L. Khoudour, "A distributed surveillance system for improving security in public transport networks," *Meas Control, Special Issue on Remote Surveillance Measurement and Control*, vol. 35, no. 8, September 2002, pp. 209–213.

[6] F. Cupillard, A. Avanzi, F. Brémond, and M. Thonnat, "Video understanding for metro surveillance," in *Proceedings of the IEEE International Conference on Networking, Sensing & Control*, vol. 1, Taipei, Taiwan, March 2004, pp. 186–191.

[7] D. Zillmann, *Hostility and Aggression.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1979.

[8] C. A. Anderson and L. R. Huesmann, "Human aggression, a social-cognitive view," in *The Sage Handbook of Social Psychology*, M. A. Hogg and J. Cooper, Eds. SAGE publications, Thousand Oaks, California, 2003, pp. 296–323.

[9] L. Berkowitz, *Aggression: Its Causes, Consequences, and Control.* Temple University Press, Philadelphia, 1993.

[10] A. H. Buss, *The Psychology of Aggression.* John Wiley & Sons, Inc., New York, 1961.

[11] A. Bandura, *Aggression: A Social Learning Analysis.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

[12] J. A. Joireman, J. Anderson, and A. Strathman, "The aggression paradox: Understanding links among aggression, sensation seeking, and the consideration of future consequences," *Journal of Personality and Social Psychology*, vol. 84, 2003, pp. 1287–1302.

[13] A. H. Buss and M. P. Perry, "The aggression questionnaire," *Journal of Personality and. Social Psychology*, vol. 63, 1992, pp. 452–459.

[14] T. Sakuma, T. Mukai, and S. Kuriyama, "Psychological model for animating crowded pedestrians," *Computer Animation and Virtual Worlds*, vol. 16, 2005, pp. 343–351.

[15] S. R. Musse and D. Thalmann, "Hierarchical model for real time simulation of virtual human crowds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 2, 2001, pp. 152–164.

[16] K. Yamashita and A. Umemura, "Lattice gas simulation of crowd behavior," in *International Symposium on Micromechatronics and Human Science*, 2003, pp. 343–348.

[17] E. L. Andrade and R. B. Fisher, "Simulation of crowd problems for computer vision," in *First International Workshop on Crowd Simulation (V-CROWDSŠ05)*, 2005.

[18] V. Gangaram Panday, "An ontology-based bayesian network approach to aggression analysis in a train environment," Master's thesis, Delft University of Technology, 2008. [Online]. Available: http://www.kbs.twi.tudelft.nl/Publications/MSc/2008-Panday-Msc.html

[19] S. C. Yudofsky, J. M. Silver, W. Jackson, J. Endicott, and D. Williams, "The overt aggression scale for the objective rating of verbal and physical aggression," *The American journal of psychiatry*, vol. 143, no. 1, January 1986, pp. 35–39.

[20] C. Choy, M. Chung, G. Harahap, M. Natadarma, and S. L. Wu, "Surveillance system using abandoned luggage detection," Man-Machine Interaction Group, Delft University of Technology, Tech. Rep., 2007. [Online]. Available: http://www.kbs.twi.tudelft.nl/Publications/BSc/2007-5st-BSc.html

[21] O.-W. Kwon, K. Chan, J. Hao, and T.-W. Lee, "Emotion recognition by speech signals," in *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, September 2003, pp. 125–128.

[22] S. Bou-Ghazale and J. Hansen, "A comparative study of traditional and newly proposed features for recognition of speech under stress," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, July 2000, pp. 429–442.

[23] M. Pantic and L. J. Rothkrantz, "Automatic analysis of facial expressions: The state of the art," *IEEE Transactions on Patterns Analysis and Machine Intelligence*, vol. 22, no. 12, December 2000, pp. 1424–1445.

[24] D. L. Hall and S. A. H. McMullen, *Mathematical Techniques in Multisensor Data Fusion*, 2nd ed. Artech House, inc., Norwood, Massachusetts, 2004.

[25] D. L. Hall and J. LLinas, "Multisensor data fusion," in *Handbook of Multisensor Data Fusion*, 2nd ed., D. L. Hall and J. LLinas, Eds. CRC Press LLC, Boca Raton, Florida, 2001, pp. 1–10.

[26] A. N. Steinberg, C. L. Bowman, and F. E. White, "Revisions to the JDL data fusion model," in *Proceedings of SPIE AeroSense (Sensor Fusion: Architectures, Algorithms and Applications III)*, 1999, pp. 430–441.

[27] J. Llinas and D. L. Hall, "An introduction to multi-sensor data fusion," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1999.

[28] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White, "Revisiting the JDL data fusion model II," in *Proceedings of the 7th International Conference on Information Fusion (Fusion2004)*, 2004, pp. 1218–1230.

[29] E. P. Blasch and S. Plano, "JDL level 5 fusion model 'user refinement' issues and applications in group tracking," in *Aerosense*, vol. 4729, 2002, pp. 270–279.

[30] W. Elmenreich, "A review on system architectures for sensor fusion applications," in *5th IFIP Workshop on Software Technologies for Future Embedded & Ubiquitous Systems*, 2007, pp. 547–559.

[31] C. Harris, A. Bailey, and T. Dodd, "Multi-sensor data fusion in defence and aerospace," *Aeronaut Journal*, vol. 102, no. 1015, 1998, pp. 229–244.

[32] J. Esteban, A. Starr, R. Willetts, P. Hannah, and P. Bryanston-Cross, "A review of data fusion models and architectures: Towards engineering guidelines," *Neural Comput & Applications*, vol. 14, 2005, pp. 273–281.

[33] G. M. Powell, "Opportunistic problem-solving in data fusion: The blackboard model," in *23rd Army Science Conference*, December 2002.

[34] H. P. Nii, "Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures," *The AI Magazine*, vol. VII(2), 1986, pp. 38–53.

[35] N. Carver and V. Lesser, "Blackboard systems for knowledge-based signal understanding," in *Symbolic and knowledge-based signal processing*. Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1992, pp. 205–250.

[36] P. Hou, X. Shi, and L. Lin, "Generic blackboard based architecture for data fusion," in *the 26th Annual Conference of the IEEE Industrial Electronics Society (IECON 2000)*, vol. 2, October 2000, pp. 864–869.

[37] I. D. Craig, "Formal techniques in the development of blackboard systems," *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, vol. 7, no. 2, 1993, pp. 197–219.

[38] B. V. Dasarathy, "Sensor fusion potential exploitation, innovative architectures and illustrative applications," *Proceedings of the IEEE*, vol. 85, no. 1, 1997, pp. 24–38.

[39] M. Bedworth and J. O'Brien, "The omnibus model: A new model of data fusion?" *IEEE Aerospace and Electronic Systems Magazine*, vol. 15, no. 4, April 2000, pp. 30–36.

[40] M. M. Kokar, M. D. Bedworth, and C. B. Frankel, "A reference model for data fusion systems," in *Proceedings of SPIE Sensor Fusion: Architectures, Algorithms, and Applications IV*, vol. 4051, 2000, pp. 191–202.

[41] Z. Yang, "Multi-modal datafusion for aggression detection in trains, a survey," Delft University of technology, Tech. Rep., 2006. [Online]. Available: http://mmi.tudelft.nl/index.php?option=com_pub&task=view&id=145644

[42] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 2nd ed. Academic Press, San Diego, California, 2003.

[43] E. K. Kelloway, V. Stinson, and C. MacLean, "Eyewitness testimony in occupational accident investigations: Towards a research agenda," *Law & Human Behavior*, vol. 28, no. 1, 2004, pp. 115–132.

[44] P. Ekman, "Facial expressions of emotion," *American Psychologist*, vol. 48, 1993, pp. 384–392.

[45] J. Cohn, A. Zlochower, J. Lien, and T. Kanade, "Feature- point tracking by optimal flow discriminates subtle differences in facial expression," in *3rd IEEE International conference on Automatic Face and Gesture Recognition*, 1998, pp. 396–401.

[46] M. Black and Y. Yacoob, "Recognizing facial expression in image sequences using local parameterised modes of image motion," *International Journal on Computer Vision*, vol. 25, no. 1, 1998, pp. 23–48.

[47] B. Boulay, F. Brémond, and M. Thonnat, "Applying 3D human model in a posture recognition system," *Pattern Recognition Letters*, vol. 27, 2006, pp. 1788–1796.

[48] J. Davis and M. Shah, "Toward 3-D gesture recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 13, no. 3, May 1999, pp. 381–393.

[49] A. Kapoor and R. W. Picard, "A real-time head nod and shake detector," in *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, November 2001.

[50] D. Datcu and L. J. M. Rothkrantz, "The recognition of emotions from speech using gentleboost classifier," in *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech)*, 2006.

[51] C. M. Lee and S. S. Narayanan, "Toward detecting emotions in spoken dialogs," *IEEE Transactions On Speech And Audio Processing*, vol. 13, no. 2, March 2005, pp. 293–303.

[52] V. Hozjan and Z. Kačič, "A rule-based emotion-dependent feature extraction method for emotion analysis from speech," *Journal of the Acoustical Society of America*, vol. 119, 2006, pp. 3109–3120.

[53] I. Shafran and M. Mohri, "A comparison of classifiers for detecting emotion from speech," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, 2005.

[54] C. Sobin and M. Alpert, "Emotion in speech, the acoustic attributes of fear, anger, sadness, and joy," *Journal of Psycholinguistic Research*, vol. 28, no. 4, 1999, pp. 347–365.

[55] M. Grimm and K. Kroschel, "Rule-based emotion classification using acoustic features," in *3rd International Conference on Telemedicine and Multimedia Communication*, October 2005.

[56] D. Ververidis and C. Kotropoulos, "Automatic speech classification to five emotional states based on gender information," in *Proceedings of 12th European Signal Processing Conference (EUSIPCO '04)*, September 2004, pp. 341–344.

[57] T. Polzin and A. Waibel, "Detecting emotions in speech," in *Proceedings of Cooperative Multimodal Communication (CMC)*, 1998.

[58] P. Kammermeier and M. Buss, "A human perception model for multi-modal feedback in telepresencesystems," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, 1999, pp. 1020–1025.

[59] J. J. Gibson, *The Senses Considered as Perceptual Systems*. Houghton Mifflin, Boston, 1966.

[60] B. G. A. Klein, *Sources of Power: How People Make Decisions*, 2nd ed. MIT Press, Cambridge, Massachusetts, 1998.

[61] A. Treisman, "Properties, parts and objects," in *Handbook of Perception and Human Performance*, K. R. Boff, L. Kaufman, and J. P. Thomas, Eds. John Wiley & Sons Inc, New York, 1986.

[62] J. M. Mandler, *Stories, Scripts and Scenes: Aspects of Schema Theory*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1984.

[63] A. Krijgsman, R. A. Vingerhoeds, and L. Boullart, *Application of Artificial Intelligence in Process Control*. Pergamon Press, Oxford, New York, 1992.

[64] D. Poole, A. Mackworth, and R. Goebel, *Computational Intelligence: A Logical Approach*. Oxford University Press, Oxford, 1998.

[65] G. Foresti, C. Micheloni, L. Snidaro, P. Remagnino, and T. Ellis, "Active video-based surveillance system: the low-level image and video processing techniques needed for implementation," *IEEE Signal Processing Magazine*, vol. 22, no. 2, 2005, pp. 25–37.

[66] O. Javed, Z. Rasheed, O. Alatas, and M. Shah, "Knight^M: A real-time surveillance system for multiple overlapping and non-overlapping cameras," in *Proceedings of the International Conference on Multimedia and Expo (ICME 2003)*, vol. 1, 2003, pp. 649–652.

[67] H. Buxton and S. Gong, "Visual surveillance in a dynamic and uncertain world," *Artificial Intelligence*, vol. 78, no. 1-2, 1995, pp. 431–459.

[68] R. Hosie, S. Venkatesh, and G. West, "Classifying and detecting group behaviour from visual surveillance data," in *Proceedings of the 14th International Conference on Pattern Recognition*, vol. 1, August 1998, pp. 602–604.

[69] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *the IEEE International Conference on Multimedia and Expo (ICME 2005)*, July 2005, pp. 1306– 1309.

[70] A. Härmä, M. F. McKinney, and J. Skowronek, "Automatic surveillance of the acoustic activity in our living environment," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2005)*, July 2005.

[71] R. S. Goldhor, "Recognition of environmental sounds," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP '93)*, vol. 1, April 1993, pp. 149–152.

[72] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICSSP06)*, May 2006, pp. 813–816.

[73] B. P. Lo, S. Thiemjarus, and G.-Z. Yang, "Adaptive bayesian networks for video processing," in *Proceedings of the 2003 International Conference on Image Processing, 2003 (ICIP 2003)*, vol. 1, 2003, pp. 889–892.

[74] M. J. Beal, H. Attias, and N. Jojic, "Audio-video sensor fusion with probabilistic graphical models," in *Proceedings of the 7th European Conference on Computer Vision*, May 2002, pp. 736–752.

[75] W. Zajdel and B. Kröse., "A sequential bayesian algorithm for surveillance with non-overlapping cameras," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 8, 2005, pp. 977–996.

[76] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, "Modeling individual and group actions in meetings with layered HMMs," *IEEE Transactions on Multimedia*, vol. 8, no. 3, June 2006, pp. 509–520.

[77] C. Wojek, K. Nickel, and R. Stiefelhagen, "Activity recognition and room-level tracking in an office environment," in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Heidelberg, Germany, September 2006, pp. 25–30.

[78] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, August 2000, pp. 831–843.

[79] C. W. Stauffer and E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, August 2000, pp. 747–757.

[80] Y. Abdeljaoued, "Feature point extraction and tracking for video summarization and manipulation," Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, 2001.

[81] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, 1997, pp. 780–785.

[82] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proceedings of the 6th European Conference on Computer Vision (ECCV '00)*, 2000.

[83] M. Piccardi and T. Jan, "Mean-shift background image modelling," in *Proceedings of the 2004 International Conference on Image Processing (ICIP)*, 2004, pp. 3399–3402.

[84] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, 2005, pp. 167–256.

[85] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, 1994, pp. 43–77.

[86] A. Hyvärinen, "Survey on independent component analysis," *Neural Computing Surveys*, vol. 2, 1999, pp. 94–128.

[87] B. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music.* John Wiley & Sons, New York, 2000.

[88] T. C. Andringa, "Continuity preserving signal processing," Ph.D. dissertation, Rijksuniversiteit Groningen, 2002.

[89] D. Erdogmus, K. E. Hild II, Y. N. Rao, and J. C. Principe, "Independent component analysis using Jaynes' maximum entropy principle," in *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation*, April 2003.

[90] J. H. DiBiase, "A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays," Ph.D. dissertation, Brown University, 2000.

[91] J. Voordouw, Z. Yang, L. Rothkrantz, and C. A. van der Mast, "A comparison of the ILD and TDOA sound source localization algorithms in a train environment," in *Proceedings of EUROMEDIA 2007*, 2007.

[92] Z. Yang, A. Keur, and L. Rothkrantz, "Behavior detection in dutch train compartments," in *Euromedia*, 2008, pp. 52–57.

[93] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, 2001, pp. 511–518.

[94] B. Wu, H. Ai, C. Huang, and S. Lao, "Fast rotation invariant multi-view face detection based on real adaboost," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 79–84.

[95] S. Ali and M. Shah, "An integrated approach for generic object detection using kernel PCA and boosting," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2005)*, July 2005.

[96] C. Veenman, E. Hendriks, and M. Reinders, "A fast and robust point tracking algorithm," in *Proceedings of the 1998 International Conference on Image Processing (ICIP 98)*, vol. 3, 1998, pp. 653–657.

[97] D. Ayers and M. Shah, "Monitoring human behavior from video taken in an office environment," *Image and Vision Computing*, vol. 19, no. 12, October 2001, pp. 833–846.

[98] A. Bobick and J. W. Davis, "Action recognition using temporal templates," in *Motion-Based Recognition*, M. Shah and R. Jain, Eds. Kluwer Academic Publishers, 1997, pp. 125–146.

[99] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," in *Proceedings of the IEEE Nonrigid and Articulated Motion Workshop*, June 1997, pp. 90–102.

[100] J. Davis and M. Shah, "Visual gesture recognition," in *IEE Proceedings Visualization and Image Signal Processing*, vol. 141, no. 2, Apr 1994, pp. 101–106.

[101] R. T. Collins, A. J. Lipton, and T. Kanade, "Special section on video surveillance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000, pp. 745–746.

[102] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, February 1962, pp. 179–187.

[103] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, August 2000, pp. 852–872.

[104] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1995.

[105] T. Erl, *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Prentice Hall, Upper Saddle River, New Jersey, 2005.

[106] C. Lindle, L. Nacke, and C. Sennersten, "Dissecting play - investigating the cognitive and emotional motivations and affects of computer gameplay," in *CGAMES*, 2008.

[107] R. Schank and R. Abelson, *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum, Hillsdale, New Jersey, 1977.

[108] J. C. Giarratano and G. D. Riley, *Expert Systems: Principles and Programming*, 3rd ed. PWS Publishing Company, 1998.

[109] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks Of Plausible Inference*. Morgan Kaufmann, San Mateo, Calif, 1988.

[110] P. Wiggers, "Modelling context in automatic speech recognition," Ph.D. dissertation, Delft University of Technology, 2008.

[111] S. Hongeng, F. Brémond, and R. Nevatia, "Representation and optimal recognition of human activities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 818–825.

[112] D. Datcu and L. Rothkrantz, "A multimodal workbench for automatic surveillance," in *Proceedings of ASCI 2004*, 2004.

[113] K. V. Laerhoven, "Embedded perception: Concept recognition by learning and combining sensory data," Ph.D. dissertation, Lancaster University, 2005.

[114] Z. Yang, X. Wang, and L. Rothkrantz, "The use of storyboards in audio-visual data collection," in *Proceedings of EUROMEDIA 2007*, 2007.

[115] L. Festinger, *A theory of cognitive dissonance*. Stanford University Press, Stanford, Calif, 1957.

[116] Z. Yang, "Train image calibration using the DLT model," Delft University of technology, Tech. Rep., 2006.

[117] Z. Yang and L. Rothkrantz, "Dynamic scripting in crisis environments," in *Proceedings HCI International 2007*. Springer-Verlag, Heidelberg, July 2007.

[118] Z. Yang, S. Fitrianie, D. Datcu, and L. Rothkrantz, *Advances in Artificial Intelligence for Privacy Protection and Security*, ch. 11: An Aggression Detection System for the Train Compartment. Word Scientific, Singapore, 2009, pp. 249–286.