A Robust Event Detection under Uncertainty in Video/Audio Surveillance Systems

Dissertation

Fadi Al Machot

Student number: 0961987

Klagenfurt, 2013

Alpen-Adria-Universität Klagenfurt

Faculty of Technical Sciences

Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende wissenschaftliche Arbeit selbstständig angefertigt und die mit ihr unmittelbar verbunden Tätigkeiten selbst erbracht habe. Ich erkläreare weiters, dass ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle ausgedruckten, ungedruckten oder dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte sind gemäß den Regeln für wissenschaftliche Arbeiten zitiert und durch Fußnoten bzw. durch andere genaue Quellenangaben gekennzeichnet. Die während des Arbeitsvorganges gewährte Unterstützung einschließlich signifikanter Betreuungshinweise ist vollständig angegeben. Die wissenschaftliche Arbeit ist noch keiner anderen Prüfungsbehörde vorgelegt worden. Diese Arbeit wurde in gedruckter und elektronischer Form abgegeben. Ich bestätige, dass der Inhalt der digitalen Version vollständig mit dem der gedruckten Version übereinstimmt. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

(Ort, Datum) (Unterschrift)

Abstract

This thesis is mainly concerned with the development of a comprehensive reasoning system for complex event detection under uncertainty. It discusses the consideration of uncertainty in the frame of complex event detection involving multiple video-sensors. Uncertainty is related to the state of having limited knowledge or where it is impossible to describe the existing state exactly or to predict the possible outcome. A series of approaches considering uncertainty in event detection are known, for example, confidence functions in a Boolean data type format, fuzzy modeling approach and Dempster-Shafer approach. The latter uses belief and plausibility functions to describe the reliability features.

The presented work will focus on this by trying to give an answer to 8 major questions:

1. What are the major functional, design and performance requirements of event detection in video surveillance systems?

2. What are the major methodological approaches for the functional, design and performance requirements?

3. What are the major requirements of spatio-temporal event detection?

4. What is meant by uncertainty? What are the different forms of its occurrence? How does the state-of-the-art cope with different dimensions of uncertainty in surveillance systems?

5. What are the proposed solutions regarding spatio-temporal event detection?

6. How can imperfect sensed context-information be handled?

7. What are the requirements of emotion detection in the frame of human surveillance? What are the different forms of uncertainty related to emotion detection from human speech streams? Are there limitations of the related state-of-the-art?

8. What is the proposed solution regarding emotion detection from human speech streams?

The thesis addresses diverse state-of-the-art approaches for the major requirements of surveillance systems, spatio-temporal reasoning and context modeling. It shows the limitations of the state-of-the-art approaches and compares them with the proposed solutions. The work consists of two case studies, which expose a complex event detection system based on Answer Set Programming under uncertainty and Semantic Web. Furthermore, it shows the power of using Answer Set Programming for complex event detection compared with the run time of Semantic Web.

It addresses diverse approaches of handling uncertainty in surveillance systems. It presents an approach which combines Hidden Markov Model (HMM) and Answer Set Programming (ASP) for complex event detection. The concept still ensures high performance even when it is implemented in embedded platforms with limited hardware resources. A comprehensive description of the overall architecture of the proposed system is presented. It shows that the exposed approach increases the detection rate to 95%.

Event detection on embedded platforms requires a model-free and a computational inexpensive approach in order to have an easy and small solution, which allows an integration to FPGA-based (Field Programmable Gate Array) smart camera without the need of a bigger FPGA.

Therefore, the thesis presents a solution based on a foreground-background-segmentation using Gaussian mixture models to first detect people and then analyze their main and ideal orientation using movements. This allows one to decide whether a person is staying still or lying on the floor. The system of our case study has a low latency and a detection rate of 88%. Another key of this algorithm is the use of Gaussian mixture models for image segmentation which is not sensitive to the light and small movements in the background of a scene and considers shadow detection that has an influence on the overall event detection process.

Furthermore, the work presents an approach of emotion detection from human speech streams based on a Bayesian Quadratic Discriminant classifier. It discusses the origins of uncertainty of emotion detection systems and the limitation of the proposed systems. Hence, a case study and a related concept is presented with an overall performance of 88%.

Contents

Chapter 1

Introduction

Public security has been becoming more important in the last 20 years. Surveillance systems potentially offer a good solution to the present-day security and safety challenges in public areas. All over the world, governments are under pressure to solve security and safety problems. Consequently, monitoring costs have greatly increased.
Furthermore, the huge amount of visual information gathered on airports, highways and streets cannot be processed through human beings alone without any form of computer-based assistance. Surveillance systems can also help to detect abandoned objects, injured people that are lying on roads or in diverse facilities to detect and identify criminal actions in public places. Therefore, governments have had to equip the important urban areas with thousands of multiple types of sensors including video cameras and even microphones to detect and record the events of interest when they occur.

The use of technology for surveillance began in the 1970s with Closed-Circuit Television (CCTV) systems that were analog based. These systems were designed using cameras, multiplexers, time-lapse Video Camera Recorders (VCR) and monitors. Over time, CCTV the price of installations increased; although, the price of components was relatively cheap, the need for frequent manual operation was not cost effective. In the 1990s, companies started to install Digital Video Recorders (DVR). In a DVR, a digital storage media such as a computer hard drive is used for storing the video recordings. Using the DVR the quality of the saved records were much better than the CCTV. Therefore, the manual operations were reduced and the costs of surveillance were also decreased. By 2003, there were more computer-based DVRs on the market that could handle multi-camera inputs and provide additional functionality such as alarm handling, scheduled activation of cameras, activity detection and alarm notification. Furthermore, it is being increasingly accepted that multiple sensors networks perform perfectly comparing to the single type based surveillance system.

Every digital video surveillance system can be divided into three modules: video capture module, network interface module and central office module. The video capture module usually consists of a set of cameras and a video encoder device. This module captures the video and compresses the raw video data by a given video coding standard (MPEG2, MPEG4, H.263). The network interface module processes the video coded stream and delivers it to IP. The central office module monitors every video channel and controls the camera's actions.

The major challenge of event detection in surveillance systems is the vagueness or ambi-

guity that occurs due to the low quality of the low level features in a surveillance system. Furthermore, event detection can be either explicit or implicit. Explicit event detection requires the definition of different rules and training, whereas implicit event detection does not use any of these rules and creates the models automatically. These event detection methods make use of pattern recognition, support vector machines, hidden Markov models, Bayesian networks, Kalman filtering, principal component analysis and others [1] [2].

In this Chapter, a comprehensive explanation of the problem statement in the frame of video surveillance systems will be illustrated, a list of research questions and the research methodology will be addressed and the scientific and practical significance will be explained. Finally, a list of the related publications and the overall architecture of this thesis will be listed.

1.1 Motivation and general context

Computer vision is a field that includes methods for acquiring, processing, analyzing and understanding images. In general, computer vision is dealing with high-dimensional data which is captured from the real world, in order to produce numerical or symbolic information (low level features) about a scene.

Event detection is a field that depends on the extraction of low level features from a scene and combines them to apply an inference about a specific event.

Consequently, building and designing a video surveillance system needs many design, functional and performance requirements. This thesis will give a comprehensive difference between each requirement and will show the methodological approaches from the state-of-the-art.

Modern multimedia surveillance and monitoring systems use different types of sensors. This creates a challenge because different sensors provide the correlated data stream in different formats and at different rates. Also, the designer of a system can have different confidence levels from different sensors during the detection of different events.

However, the visual features alone are generally not always sufficient to understand a scene and to analyze it. Human's brains are able to guess and understand the scenes in daily life because of observing multiple features such as body action, voice information and the interpreted knowledge of understanding.

Therefore, the quality of the low level data should be high and the uncertainty about the tracking, detection and recognition of objects in a scene should also have a confident level of uncertainty, otherwise the reasoning system will not perform well. Thus, the overall performance of the system will be decreased. In this context, monitoring the emotion of people in multimedia sensor networks is an important factor to build a robust video understanding system. Thus, spatio-temporal reasoning under uncertainty is required for complex event recognition of object behavior where temporal entities play a major role for event recognition.

This thesis aims to detect events in video surveillance systems despite of the lack of knowledge, incompleteness and low quality of low level features. It proposes an approach to reduce the complexity of the processing time of different types of data. Especially if the system is built to perform well in an ecological environment where the power is low and the hardware resources are limited.

However, the visual features alone are generally not always sufficient to understand a scene and to analyze it. Thus, a technical interpretation of human's emotions is deeply needed in modern surveillance systems.
Therefore, the potential and the role of emotion detection from audio streams of human speech will be illustrated and a novel methodological approach will be proposed.

This thesis presents the approach of emotion detection from human speech streams, discusses the origins of uncertainty of such emotion detection systems and the limitation of the proposed systems. Hence, a case study and a related concept will be presented and the overall evaluation of the performance of emotion detection will be illustrated.

1.2 Short description of the research questions and objectives of the thesis

1. **What are the major functional, design and performance requirements of event detection in video surveillance systems?**
   The main purpose of this question is to describe all functional, design and performance requirements of a video surveillance application in order to design and develop the optimal system architecture with respect to the use case of the system.

   In surveillance systems the functionality means the capability of the surveillance system to provide useful functions to detect events that are occurring in real-time (short term), e.g. a person is shooting using a gun, and events that are occurring within a long period of time, e.g. the analysis of people's trajectories moving in a specific area (long term) with respect to the time. Furthermore, the system has to be able to record and document the events to allow the user to see and observe the area of monitoring.

   The design of a video surveillance system requires the right decision to choose the right type of sensors, the ideal video management system and the type of storage.

   Consequently, the performance of the surveillance system has to perform well with respect to the accomplishment of the surveillance system requirements measured against preset known standards of accuracy, completeness, cost and speed.

2. **What are the major methodological approaches for each of the requirements for the group in Q1? How far do they satisfactorily solve (or not) the requirements with respect to their limitations?**
   Regarding the functional requirements modern video surveillance systems are using network cameras that give the ability to create and maintain an effective and reliable IP surveillance system. They are cost effective solutions where users can build a high performance and a scalable wired or wireless IP video surveillance system. Moreover, the major function of a surveillance system is to support the system by spatio-temporal event detection to verify the previous discussed requirements in question 1.

   The performance requirements of surveillance systems are difficult to achieve because of the trade off between the different requirements. The main problem is that a high recognition rate could require a high power consumption because of the high

computation time. Therefore, the design of recognition concepts has to be as accurate as possible, consume less power and be cost effective to run it on an embedded platform.

3. **What are the requirements of a) spatio-temporal context modeling and related ontologies, b) spatio-temporal reasoning (short term), c) spatio-temporal (long term), d) real-time spatio-temporal reasoning and e) spatio-temporal reasoning under uncertainty?**
   Spatio-temporal reasoning is one of the most important challenges in visual event detection systems. Many events and video understanding requires the temporal entities to decide for a specific complex event. Different types of events need a temporal sequence to be recognized, especially in the frame of middle and long term event detection.

   The major requirements of a spatio-temporal context modeling are that the model must restrict the domain of application, provide a support for recording of provenance and processing of information. In addition to this, the model should include tools that permit the definition of new contextual categories and should allow reusability in other independent modeling tasks.

   Regarding Qb and Qc, the system has to detect events with respect to temporal constraints, needs high quality low level data, high performance sensor fusion and a consistent simple ontology.

   The major requirements of Qd for real-time reasoning is that the system should keep the row data moving "in-stream", without any requirement to store them to perform any operation or sequence of operations. Furthermore, the system should process on chip to reduce data transfer between different components and a consistent ontology should be used.

   Reasoning under uncertainty is a major challenge where the low level data should be accurate and complete. The reflection of reality is needed to perform well during the inference. In order to achieve this, degrees of confidence are needed to handle uncertainty in different levels.

4. **What are the limitations of the previous concepts in Q3?**
   There are different approaches for event detection and recognition and every approach has its advantages and disadvantages. In this thesis, the limitations of the state-of-the-art will be considered deeply and in every Chapter we will illustrate the limitations of every approach.

   There are different approaches for event detection based on static threshold methods and probabilistic methods. The statistical approaches are the simplest and most computationally straight-forward. The probabilistic methods for event occurrence and other related probabilities are computed and assessed rather than computing and testing statistics from a sample data set.

   Clearly machine intelligence approaches are widely applied, e.g. particle filtering, genetic algorithms, neural networks, intelligent agents and fuzzy based systems.

   The major limitations of the previous approaches are: uncertainty handling, power

consumption, computational time, the lack of the consideration of temporal constraints, running on embedded platforms and the detection rate.

5. **What is uncertainty? What are the different forms of its occurrence and eventuality in relation to different sensor types and functions? What are the different taxonomies of uncertainty? How does the state-of-the-art cope with different dimensions of uncertainty in surveillance systems?**
Uncertainty means the state of having limited knowledge where it is impossible to describe the existing state exactly or to predict the possible outcome.
The media streams in multimedia sensor networks are often correlated; the system designer has different confidence levels in the decisions obtained.

   Vagueness or ambiguity due to the low quality of low level features in a surveillance system are sometimes described as "second order uncertainty" where uncertainty is even about the definitions of uncertain states or outcomes. In video surveillance systems we consider two main types of uncertainty; uncertainty in inference processes and uncertainty in data of sensors perception caused by weather, fusion or noise coming from sensors.

6. **Novel solutions to the points a, b, c, d and e of Question 3?**
The novel solutions concentrate on building robust and adaptive surveillance systems which are easy to implement on embedded platforms and offer the real potential of robustly detecting a huge number of complex events in real time and long term. Most of the previous methods do not consider uncertainty clearly; correct quantification of the probability of materialized events serves as an important tool for decision making.

   In this thesis, a method is defined based on the combination of Hidden Markov Model (HMM) and Answer Set Programming (ASP) for efficient approximation of new event materialization in feature space. The algorithm enables a quick method to compute the probabilities of a set of events. The approach increases the detection rate to 95% because of the power of HMM and the optimization power of ASP. This work suggests a model-free algorithm for position detection and estimation of humans combined with Gaussian Mixture Models (GMMs) for image segmentation with a detection rate of 88%.

   Additionally, a robust system is proposed to detect the emotions from human speech streams of people using a low number of features and can detect their emotion with a high level of accuracy over (88%). The proposed emotion detection system can run on an embedded platform and detect emotions in real-time.

7. **What are the requirements of emotion detection in the frame of human surveillance? What are the different forms of uncertainty related to emotion detection? Are there limitations of the related state-of-the-art?**
The thesis considers the requirements of emotion recognition systems of human speech in the frame of Advanced Driver Assistance Systems.

   Driver fatigue, stress and drowsiness cause traffic accidents. Road crashes are more frequent than in other transportation modes (air, sea and railways). Safety can be improved by designing a system to detect the behavior of drivers based on their

voices. In this thesis, a great amount of stress is made in evaluating the emotion classification algorithms over the Berlin database to propose an algorithm that is scalable and non-sensitive to gender. It summarizes the major origins of uncertainty and proposes the minimum required features that should be extracted from the audio data to detect emotions of humans. The work shows that the Bayesian Quadratic Discriminant classifier performs well and can be implemented easily on embedded platforms.

8. **A demo example of an audio based emotion detection.**
   Automatic emotion recognition plays a major role in surveillance systems. When we analyze the audio signals or speech, most of the audio signals are more or less stable within a short period of time. People express emotions differently depending on the speaker, sex, race and even language. In order to the recognition to perform more robustly, the training process is required to contain more samples in the database to identify the emotion. Consequently, the emotion detection system requires robustness & reliability, low-cost and feasibility and efficient inference approach, low-power consumption and finally, it should not need a cooperation from the driver side.

   In this thesis, a comprehensive architecture of an emotion detection system from human speech streams is proposed. This system shows that the Bayesian Quadratic Discriminant classifier is an appropriate solution for emotion detection systems, where there is a real-time detection.

   The emotions (angry, happy, sad, normal and fear) are classified using Bayesian Quadratic Discriminant (BQD). The concept aims to show that a speech emotion recognition system will be useful to understand the state and emotions, for example "a driver to increase safety and control the car autonomously".

1.3 Overall research methodology

This thesis introduces the most common difficulties and challenges in event detection problems. It describes the most frequently used event detection methods and provides different examples and case studies for event detection in video/audio surveillance systems. The major task is to explore the relationship between event detection, modeling and simulation.

In the frame of this thesis, we provide comprehensive illustration of event detection approaches by the presentation of the advantages and disadvantages of the related state-of-the-art research. Consequently, the thesis proposes different algorithms for event detection based on probabilistic, stochastic, model free and logic based concepts for complex event detection.

There are well defined methodological approaches for the functional, design and performance requirements of surveillance and monitoring systems. The industrial solutions should always be cost effective, easy to maintain and perform well.

Therefore, a feature extraction module has been developed which is implemented in $C++$ and OpenCV. OpenCV is an Open source Computer Vision library from Intel. The system has been tested in two scenarios: the first one is on the highway and the

second one is in a parking place. 24 test cases have been tested for the recognition of cars, dogs and humans.

Regarding spatio-temporal reasoning and context modeling, several requirements have been taken into account. The context information models have to be able to handle the information of context sources with respect to its large amount and different input resources.

Context information entities/facts may depend on other context information entities: for example, a change in the environment may impact the values of other properties and yield to inconsistencies that are not desired in the model. Moreover, the management of context histories is difficult if the number of updates is very high.

Another methodological approach considered is reasoning which uses context information to evaluate whether there is a change in the environment of the situation or to detect a specific behavior of the object observed in the scene. Reasoning techniques can also be adopted to derive higher level context information. Therefore, it is important that the context modeling techniques are able to support both consistency verification, and reasoning about complex situations.

For context modeling and event detection the SRSnet project was an optimal test environment. The SRSnet project focuses on the design of a smart resource-aware multi-sensor network capable of autonomously detecting and localizing various events such as screams, animal noise, tracks of persons and more complex human behaviors. The project's research areas include: collaborative audio and video analysis, complex event detection and network reconfiguration.

Regarding the detection of human falls in elderly houses, an example scenario is performed using a test environment which is a 6 * 4 meters room and a network camera which has been installed in the middle of the seal.

Uncertainty and its origins are considered based on the major methods for the management of different uncertainty taxonomies, e.g. ignorance, incompleteness, inaccuracy and inconsistency. We addressed different methodological approaches to handle uncertainty, e.g. Bayes theorem, certainty factors, Dempster-Shafer theory and fuzzy theory.

The thesis proposes a concept of handling uncertainty based on the combination between hidden Markov model and Answer Set Programming. A simulation tool is built during this work which allows one to test the proposed approaches for uncertainty management faster than in real systems.

The simulation tool helps to analyze the incoming data immediately and reports the results obtained. Therefore, the evaluation phase of the proposed approach is based on random trajectories of people using the developed tool to create history data.

Our simulation tool is developed in *C#*; it generates data, trains and evaluates the overall concept. The data sets from the history are divided into two parts, a training data set and a test data set. We evaluated the proposed HMM using different samples with different history data.

Event detection based on audio data (human speech streams) is also considered to detect and recognize human emotions for event detection in Advanced Driver Assistance Systems (ADAS).

The test environment of the emotion recognition system is the Berlin emotional data base. The Berlin emotional speech database is developed by the Technical University, Institute for Speech and Communication, Department of Communication Science, Berlin.

It has become one of the most popular databases used by researchers on speech emotion recognition, thus facilitating performance comparisons with other studies. 5 actors and 5 actresses have contributed speech samples for this database and it mainly has 10 German speakers.

1.4 Significance and contributions of the thesis

This section addresses a comprehensive summary of the major innovative contributions of the thesis which gives a scientific significance and practical significance to this work.

1.4.1 Comprehensive summary of the major innovative contributions of the thesis

In the frame of this thesis different research questions have been considered and intensive research has been done to find the optimal answer to every question. The thesis covers the major fundamental and advanced research topics in the area of video surveillance systems. It starts with the functional, design and performance requirements and its methodological solutions.

It addresses the major functional requirements of surveillance systems to provide useful functions to detect events that are occurring in real-time, short term and long term. Furthermore, the system has to be able to record and document the events to allow the user to see and observe the area of monitoring.

Choosing an optimal design for a video surveillance system requires the use of a mix of different camera types. For instance, an organization may use infrared fixed cameras around a perimeter with Pan Tilt Zoom (PTZ) cameras for indoors. Outdoors they may have a fixed megapixel camera covering the warehouse and a number of fixed IP cameras covering the entrance and hallways.

Hybrid Network Video Recorders (NVR) and Digital Video Recorders (DVR) support IP cameras and are directly connected to analog cameras. This provides simplicity and reliability.

Most existing state-of-the-art methods for event/object recognition are model based systems that are computational and expensive to run on tiny embedded platforms. Another challenge is that the detection of objects in ultra-fast computation time is also needed, e.g. in ADAS the driver has no time to think if a dangerous situation occurs.

In this thesis, reasoning about context information in the domain is supported by two types of reasoning mechanisms: rule-based reasoning and probabilistic/stochastic reasoning.

There are different approaches regarding uncertainty in video surveillance systems. The most famous concepts are using Monte Carlo simulations, Bayesian networks, Bayes theorem, certainty factors, Depster-Shafer theory, fuzzy theory and hidden Markov models.

The thesis addresses the methodological approaches and its limitations for handling uncertainty provided by different examples and supported by a specific case study.

Different taxonomies of uncertainty have been explained:

- **Ignorance:** This means that there is an object in the environment of the surveillance system which is not known.

- **Incompleteness:** This is in contrast to ignorance.
- **Inaccuracy:** This deals with the potential measurement errors that may occur.
- **Inconsistency:** This means that there are conflicting hypotheses about an object data.

Furthermore, a novel approach is defined and based on Answer Set Programming ASP where a weight should be calculated before every feature and then the rule with highest probability will be chosen using the optimization power of ASP. The approach enables a quick method to compute the probabilities of a set of events. The approach increases the detection rate to 95% because of the power of HMM and the optimization power of ASP.

Event detection on embedded platforms requires a model-free and a computational inexpensive approach in order to have an easy and simple solution, which allows an integration to FPGA-based smart camera without the need of a bigger FPGA.

Therefore, the thesis presents a solution based on a foreground-background segmentation using Gaussian mixture models to first detect people and then analyze their main and ideal orientation using moments. This allows one to decide whether a person is staying still or lying on the floor. The system has a low latency and a detection rate of 88% in our case study.

Another key of this algorithm is the use of Gaussian mixture models for image segmentation which is not sensitive to the light and small movements in the background of the scene and considers shadow detection that has an influence on the overall event detection process.

In the frame of Advanced Driver Assistance Systems (ADAS), safety can be improved by designing a system to detect the behavior of drivers based on their voices. Driver fatigue, stress and drowsiness cause traffic accidents. Road crashes are more frequent than in other transportation modes (air, sea and railways).

In this thesis, a comprehensive solution based on Bayesian Quadratic Discriminant (BQD) classifier is developed. The system supports ADAS to detect the mood of the driver based on the fact that aggressive behavior on road leads to traffic accidents. Therefore, difficulty in emotion recognition in people's speech streams is due to the lack of an affect-related semantic and syntactic knowledge base.

This work proposes a system for emotion recognition consisting of two main steps: a features extraction step and a classification step. The features extraction step uses the energy, pitch and the Mel-frequency Cepstral Coefficients (MFCC).

The Berlin data base is used to evaluate the performance of the system which is one of the most popular databases for emotion recognition.

1.4.2 Scientific significance of the thesis

The detailed illustration of the methodological approaches of knowledge representation, context modeling and reasoning techniques gives this thesis a valuable reference for researchers in the area of video/audio surveillance systems. It forms a detailed survey about the architecture requirements based on the modern state-of-the-art approaches.

Regarding spatio-temporal complex event detection has been proven that the use of Answer Set Programming combined by context models as a knowledge base can significantly reduce the computational time needed to detect complex events on embedded platforms.

It opens a perfect research direction to combine Answer Set Programming (ASP) with other context modeling tools where ASP can be an optimal solution for computer-aided verification, configuration, constraint satisfaction, diagnosis, information integration, planning and scheduling, security analysis, Semantic Web, wire-routing, zoology, linguistics and many more.

This work employs ASP power to reason the context. The power of ASP can be summarized as a descriptive and expressive tool to describe real-life events and scenarios, e.g. the strength of logic programming with ordered disjunction and guess & check programs of ASP.

The need of temporal reasoning about context information can be solved using ASP where the time is usually represented as a variable that values are defined by an extensional predicate with a finite domain. Dealing with finite temporal intervals can be used to reason complex events in our case studies.

The management of uncertainty in surveillance systems needs arithmetic operations that are usually not well presented in logic reasoning tools. ASP offers the standard arithmetic functions and the absolute function. Furthermore, other arithmetic can be implemented and reused depending on the use case of the desired reasoning process.

Consequently, the extensions and the research in ASP has to be considered, e.g. the combination between ASP and fuzzy theory FASP. This combination offers the best of both worlds: from the answer set semantics it uses the power of its declarative non-monotonic reasoning capabilities while, on the other hand, the concepts from fuzzy logic allow to avoid the limitations of classical logic. As fuzzy logic gives a great flexibility regarding the choice for the interpretation of the concepts of negation, conjunction, disjunction and implication, the FASP can be applied in different areas of application.

The novelty of this work is that it proposes a robust approach based on the combination between Hidden Markov Model (HMM) and Answer Set Programming (ASP) where a weight should be calculated for all related extracted features and then the event with the highest probability will be selected using the optimization power of ASP.

In relation to the previous advantages, the optimization possibilities of ASP, e.g. the maximization and minimization, can be applied to choose the optimal sensor data despite of the different taxonomies of uncertainty in surveillance systems.

Furthermore, the cardinality and the constraints in ASP can be used in the body of ASP rules to give the developer the possibility to optimize the desired answer sets.

Another key of this work is that it suggests a model-free algorithm for position detection and estimation of humans. This would be combined with Gaussian mixture models for image segmentation which is not sensitive to the light, small movements in the background of a scene and considers shadow detection that has an influence on the overall event detection process.

Finally, in the frame of event detection in audio based surveillance systems, the thesis proposes reliable features that can be used to detect emotions from human speech streams and suggests a classifier to decide between 5 different types of emotions (happy, sad, angry, normal and fear). Extensive research has been done in this area and a high detection rate is obtained compared to the related state-of-the-art.

1.4.3 Practical significance of the thesis

Event detection and recognition is an effective approach to reduce the costs of monitoring all over the world. The world population has experienced continuous growth in the last 100 years.

Video surveillance systems have an important role in our daily life nowadays. They reduce risk, increase the safety of the society and decrease the costs of monitoring. The proposed reasoning concept in this thesis has proven that using the developed reasoning concept in surveillance video systems can be applied to mitigate risk.

The proposed approaches can detect high-risk events quickly and can react quickly, whereas low-risk events may take weeks to be realized.
However, the cost of storing surveillance data remains expensive. The longer the data is kept the more storage is needed and in turn, the higher the cost. The proposed event detection reasoning concept helps to store sufficient required videos and delete others that are not important. It also considers the reduction of power consumption and limited hardware resources. In the United Kingdom there are over 1.85 million surveillance cameras[1]. This means that the practical use of the concept has a wide market to be applied in order to save the expenses of governments.

The concept of the combination between context models, ASP and uncertainty consideration, could have an application in public health surveillance and biological informatics. For example, in predicting missed genome sequences and predicting the impact of combining different chemical contents in human cells.

The concept can also be used to model the interaction of biological networks even though ASP is a great tool box for the modeling of biological network semantics and allows one to model specific networks with little effort.

The approaches discussed in this thesis can be applied to geographical systems for earthquake and zunami occurrences, threat detection and management of homeland security. Usually, they can also be used in systems that are using different kinds of sensors to observe, predict and detect any scenario defined by users for many use cases.

1.5 List of publications in the frame of this thesis

Publications in Book Chapters

- Kyamakya K., Chedjou J.C., Al Machot F., Fasih A.: Enabling a Driver-Specific "Real-Time Road Safety" Assessment through an "Extended Floating Car Data" and Visualization System. *In: Unger H., Kyamakya K., Kacprzyk J. : Autonomous Systems: Developments and Trends. Springer Verlag GmbH*, pp. 277-294, 2011.

- Rass S., Al Machot F., Kyamakya K.: Fine-Grained Diagnostics of Ontologies with Assura. *In: Jao C. (Hrsg.): Efficient Decision Support Systems: Practice and Challenges - From current to Future, Intech*, 2011.

Publications (Journals)

- Al Machot F., Kyamakya K.: Real Time Complex Event Detection Based on Answer Set Programing. *In: ISAST Transactions on Computers and Intelligent Systems, University of Jyväskylä*, pp. 1-5, 2011.

[1] http://www.securitynewsdesk.com/2011/03/01/how-many-cctv-cameras-in-the-uk/

- Schwarzlmüller C., Al Machot F., Fasih A., Kyamakya K.: A Novel Support Vector Machine Classification Approach Involving CNN for Raindrop Detection. *In: ISAST Transactions on Computers and Intelligent Systems, University of Jyväskylä*, pp. 52-65, 2010.

- Fasih A., Schwarzlmüller C., Kyamakya K., Al Machot F.: Video Enhancement for ADAS Systems based on FPGA and CNN Platform. *In: International Journal of Signal and Image Processing, HyperSciences Publisher*, 2010.

Publications in Conferences

- Al Machot F., Haj Mosa A., Dabbour K., Fasih A., Schwarzlmüller C.: A Novel Real-Time Emotion Detection System from Audio Streams Based on Bayesian Quadratic Discriminate Classifier for ADAS. *In: Kyamakya K., Halang W.A., Unger W., Mathis W., Kaltenbacher M., Huemer M., Horn M.: Proceedings of the Joint INDS'11 & ISTET'11. Aachen: Shaker Verlag GmbH*, pp. 47-51, 2011.

- Al Machot F., Kyamakya K., Dieber B., Rinner B.: Real Time Complex Event Detection for Resource-Limited Multimedia Sensor Networks. *In: Rinner B., Foresti G.F.: Proceedings of the 8th International Conference Advanced Video and Signal-Based Surveillance (AVSS)*, pp. 468 - 473, 2011.

- Al Machot F., Tasso C., Dieber B., Kyamakya K., Piciarelli C., Micheloni C., Londero S., Valotto M., Omero P., Rinner B.: Smart Resource-aware Multimedia Sensor Network for Automatic Detection of Complex Events. *In: Rinner B., Foresti G.F.: Proceedings of the 8th IEEE International Conference on Advanced Video and Signal-Based Surveillancei (AVSS)*, pp. 402 - 407, 2011.

- Hartmann R., Al Machot F., Mahr Ph., Bobda Ch.: Camera-Based System for Tracking and Position Estimation of Humans. *In: Arslan T.: Conference on Design and Architectures for Signal and Image Processing (DASIP)* , Edinburgh University Press, October 2010, pp. 281-286, 2010.

- Al Machot F., Haj Mosa A., Fasih A., Schwarzlmüller C., Ali M., Kyamakya K.: A Novel Real-Time Emotion Detection System for Advanced Driver Assistance Systems. *In: Unger H., Kyamakya K., Kacprzyk J.: Autonomous Systems: Developments and Trends. Berlin, Heidelberg*, New York: Springer Verlag GmbH, pp. 267-276, 2011.

1.6 Organization of the thesis

In the frame of this thesis different approaches have been developed and the overall architecture of surveillance systems will be explained in Chapter 2.

Chapter 2 considers the overall architecture of surveillance systems and the major functional, design and performance requirements of surveillance systems. Then, it covers the methodological approaches to build, design and deploy surveillance system with high performance.

Clearly, each user requires a specific target to monitor and observe. Hence, Chapter 2

tries to answer the major critical questions with respect to the recent modern technologies in the frame of sensor networks and surveillance systems.

Chapter 3 focuses on the field of spatio-temporal modeling approaches based on knowledge representation and its related tools. Chapter 3 discusses the spatio-temporal reasoning requirements, the related methodological approaches of the state-of-the-art and its limitations.

Chapter 4 addresses the field of uncertainty with its definition in the frame of video surveillance systems, the origins of uncertainty and its taxonomies. In consequence of this, a detailed illustration of related works and its limitation in the field of event detection is given.

Chapter 5 considers 2 novel approaches for event detection: the first approach combines Answer Set Programming (ASP) with Hidden Markov Model (HMM) to manage uncertainty in the frame of complex event detection and the second approach illustrates an algorithm for model-free position detection and estimation of humans.

Chapter 6 consists of 2 case studies, case study 1 proposes a complex event detection system based on Semantic Web, the second one shows the power of using ASP for complex event detection in video/audio surveillance systems.

Chapter 7 presents the approach of emotion detection from human speech streams, discusses the origins of uncertainty of emotion detection systems and the limitation of the state-of-the-art systems. Hence, a case study and a related concept will be presented and the overall evaluation of the performance of emotion detection will be illustrated. Finally, at the end of this thesis, the conclusion and the future work is presented in Chapter 8.

Chapter 2

Architecture of Surveillance Systems

Surveillance systems play an important role in traffic incident detection, travel time measurement and traffic management. They offer a good potential for helping to solve the present-day security and safety challenges in public transportation infrastructures. All over the world, transportation operators, security people and the police are being put under pressure to solve these security and safety problems. Due to this, monitoring costs have greatly increased. Furthermore, the huge amount of visual information gathered in public areas can no longer be processed through human beings alone without any form of computer-based assistance. Because of the previously mentioned importance, such systems have essential requirements that researchers have to consider in order to build the desired system and achieve their specified functions and performance. Although, there are many forms of observation and monitoring, e.g. directional microphones, communications interception, listening devices, Closed-Circuit Televisions or GPS tracking, video surveillance is the most popular form of surveillance

In this Chapter, the overall architecture of video based surveillance systems and its applications will be considered.

Then, the major functional, design and performance requirements will be discussed which will help to build a video based surveillance system with a high performance.

2.1 Surveillance systems and an overview of their application forms and scenarios

Intelligent video surveillance systems deal with the real-time monitoring of static and moving objects within a specific environment. The primary motivation of such systems is to understand, detect, recognize and predict the actions and the interactions of the observed objects autonomously based on the information acquired by sensors. The main steps of processing in an intelligent video surveillance systems are: moving object detection and recognition, tracking, behavioral analysis and retrieval. These steps include the topics of machine vision, pattern analysis, artificial intelligence and data management [3].

There are three main technical evolutions of intelligent surveillance systems. The first generation started with analogue Closed-Circuit Television (CCTV) systems. They gave good performance in specific situations but they had the problem of using analogue techniques for image distribution and storage.

The second generation techniques automated visual surveillance by combining computer

vision technology with CCTV systems. This combination increased the surveillance efficiency of CCTV systems but they had the problem of robust detection and tracking algorithms required for behavior recognition.

The third generation presents the automated wide-area surveillance systems. They are more accurate than the previous generation due to the combination of different kinds of sensors. They have challenges in distribution of information (integration and communication), design methodology, moving platforms, multi-sensor platforms [3].

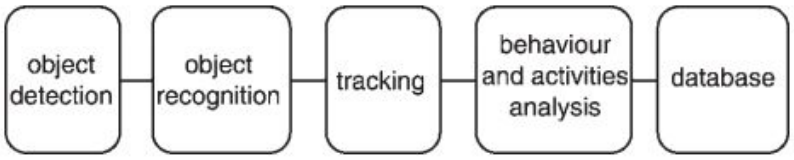

Figure 2.1: Traditional flow of processing in visual surveillance system. [3].

The typical flow of processing steps in video surveillance systems is illustrated in Figure 2.1. These steps constitute the low-level processing phase which is necessary for any video surveillance system.

Object detection: Usually, the main idea of object detection is the segmentation of images in foregrounds and backgrounds. The major two approaches are "temporal difference" and "background subtraction". The first approach consists of the subtraction of two consecutive frames followed by thresholding. The second approach is based on the subtraction of a background followed by a labeling process. Generally, morphological operations are used to reduce the noise and to correct the segmented shapes. The segmentation of images separates the image in two parts, the foreground and the background. The foreground of the image represents the objects to be detected in the scene. After that, different processes can be chosen, starting with the representation and description of the regions shape and ending with processing and analyzing the regions of interest. The results of the previous processes can be used in the field of boundary matching or mathematical models training. The final step is commonly performed to extract the low level features for event detection systems.

Object recognition: The object recognition and tracking step is normally a model-based technique. Different approaches can be used to classify the new detected objects. For example, Gaussian distribution [4], particle filters [5], hidden Markov models [6] and Support Vector Machine [7]. Tracking techniques can be split into two main approaches: 2-D models [8] and 3-D models [9].

Behavior analysis: The previous steps are important to extract features for event detection where the behavior of the observed object should be analyzed and understood. Furthermore, the analysis of the image and the understanding of the spatial/temporal content is also required to understand the behavior of the object. The overall architecture of event detection systems in surveillance systems is illustrated in Figure 2.2.

Suppose the system is detecting a vandalism in a bank, it is not possible to detect the event of vandalism against the Automatic Teller Machine (ATM) without knowing if the

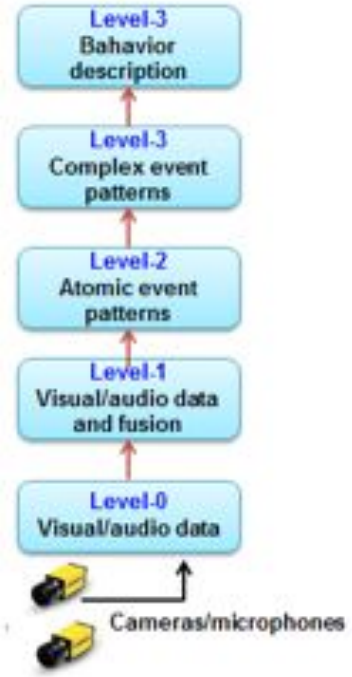

Figure 2.2: The overall architecture of video surveillance systems.

object is near by the (ATM) machine or not. Additionally, video streams consist of a sequence of frames (images). Thus the temporal issue should be considered and analyzed to understand which event occurs before another event. The overall architecture of event detection in surveillance systems has the following three layers:

1. **Object detection and tracking:** By extracting features using object recognition and object tracking algorithms; this involves image processing and pattern recognition.

2. **Primitive events detection:** By defining both behavior and rules that are related to objects simple events can be detected, like walking, running, shouting, etc.

3. **Complex event detection:** By building rules using rule engines acting on simple events a series of detected simple events can be joined together to form complex events.

Data base: The final stages in a surveillance system are storage and retrieval. The most used databases are data warehouses which is a database used for reporting and data analysis. It is a central repository which is created by integrating data from and multiple disparate sources (audio or video). The major disadvantage of a data warehouse is its expensive maintenance if it is underutilized.

Application areas: The major application areas of surveillance systems are applied in the following areas [3]:

- Transport applications, such as airports [10], railways [3], underground and highways [11].

- Public places, such as banks, supermarkets, homes, department stores[12] and parking areas [13].

- Remote surveillance of human activities, such as football matches and sport [14]. item Military applications [15].

2.2 The requirements of surveillance systems

Video surveillance systems have existed for 25 years, starting out as 100% analogue systems and gradually becoming digitized. The main purpose of this section is to describe all functional, design and performance requirements of a video surveillance application in order to design and develop the optimal system architecture with respect to the use case of the system. The focus is on the main concepts in the surveillance systems, e.g. real-time, dynamic reconfiguration and service composition.

2.2.1 The functional requirements

The video surveillance system has to provide different advanced functionalities, e.g. remote accessibility, spatio-temporal intelligent event detection, event management. It should be easy to integrate it and should offer a better scalability, flexibility and cost-effectiveness[1]:

- **Remote accessibility**: This means that the system has to offer the possibility to be configured and accessed remotely, enabling multiple authorized users to watch live and recorded videos at any time and from any location in the world.

- **Spatio-temporal intelligent event detection and event management:** The system has to offer the possibility to reduce the amount of uninteresting recordings by the detection of the interesting events autonomously and has to be able to sort and show specific statistics regarding the detected events and the distribution of those events within a specific period of time. Event management functionalities should simplify the use of the graphical user interface of the related software program, e.g. users can define the type of alarms or events and the sensitivity level of the system regarding event detection.

- **Easy integration possibilities and better scalability:** A video surveillance system should be able to grow with a user's needs. For example, any number of network products can be added to the system without significant or costly changes to the network infrastructure, e.g. adding a new network of audio sensor should be easy if audio based event detection is required. The system also must be able to be placed and networked from any location and the system must be as open or as closed as desired.

[1] Axis network video

- **Cost-effectiveness:** An IP surveillance system typically has a lower total cost of ownership than a traditional analog CCTV system. IP-based networks and wireless options are also much less expensive alternatives than traditional coaxial and fiber cabling for an analog CCTV system.

- **Network reconfiguration:** The automatic reconfiguration of the connected sensors to optimize the power consumption, switch on/off sensors in the region of interest and optimize data transfer and storage between the different nodes of the surveillance network.

2.2.2 The design requirements

Designing a video surveillance system requires decisions on the following major questions[2]:

1. What type of cameras should be used?

2. How to choose the ideal video management systems?

3. What type of storage should be used?

4. How should the saved videos be viewed?

The answer to the first question is that before one chooses the type of the camera, first the position of the camera must be specified. Surely, cameras must be deployed in critical areas where people or vehicles must pass to enter a certain area. After the determination of the observed area there are 4 cameras characteristics to decide on:

- **Fixed vs. PTZ:** A camera can be fixed to look only at a specific area or it can be movable through the use of panning, tilting and zooming. Most video surveillance systems use fixed cameras. The use of a PTZ camera is to cover wider fields of view.

- **Color vs. Infrared vs Thermal:** Today, in video surveillance systems the production of black and white image is only used when lighting is very low, e.g. night time. In those conditions, infrared or thermal cameras produce black and white images. Infrared cameras can produce clear image in the dark but are significantly more expensive than color cameras.

- **Standard Definition vs. Megapixel:** Now in 2012, megapixel is becoming the standard resolution used in new surveillance systems projects.

- **IP vs. Analog:** All surveillance cameras are digitized to view and record on computers, only IP cameras digitize the video inside the camera. Another important factor is that IP cameras support megapixel while analog cameras do not.

[2]Milestone White Paper Battening Down the Hatches: IP Video Surveillance and Access Control, A guide for security and IT leaders on the advantages of integrating video surveillance and access control

The second question focuses on how to choose the right video management software. The current video management software products all record compressed video streams from network cameras and encoders and intelligently route video to video monitors. They are also supported by camera and user administration interface. The modern systems display live video in graphical user interfaces, provide PTZ camera control and enable intelligent searching for recorded video.

The video management software systems in the market have wide variance in product features, usability and, of course, price. Product differentiators include scalability, network management, fault tolerance, operating system, browser-based software clients and the use of standard conventions and protocols. Clearly, the best product selection will depend on users system requirements:

- **Digital Video Recorder:** They are built computers which combine software, hardware and video storage all in one. By definition, they only accept analog camera feeds. Today, almost all DVRs support remote viewing over the Internet. They are simple to install but not flexible in the frame of hardware changes.

- **Hybrid Digital Video Recorder (DVR)):** They have all the features of standard DVR but they support IP and megapixel cameras. Most DVRs can be software upgraded to become Hybrid Digital Video Recorders (HDVR). Network Video Recorders (NVRs) are the same as DVRs but the difference is that a DVR only supports analog cameras but NVR only supports IP cameras. For using NVR with analog cameras an encoder should be provided.

- **Video Management Software (VMS):** It is a software application, like Word or Excel. It differs from DVRs or NVRs. It does not come with any hardware or storage. The user must load and make the PC/Server setting for the software. This provides potentially a lower cost and is much better than DVR/NVR appliances. Generally, VMS software is becoming the most commonly used recording approach in new surveillance systems [16].

The third question considers the storage of videos in the surveillance systems. Usually, the videos in video surveillance systems are stored for later retrieval and review. The average storage duration is around 30 days. However, a small percentage of organizations store video for a much shorter time or for a much longer time (some for a few years). It depends on the organization, company or users requirements. Furthermore, storage is getting cheaper and the amount of stored videos is getting higher. The different storage types are as follows:

- **Internal Storage:** It uses hard drives built inside of a Digital Video Recorder (DVR), a Network Video Recorder (NVR) or server. This method is still the most used form of storage. A Digital Video Recorder is an electronics device or application software which records video in a digital format to a disk drive, USB flash drive, SD memory card or networked mass storage device. Video on a DVR is encoded and processed at the DVR, while video on a NVR is encoded and processed on the camera, then streamed to the NVR for storage or remote viewing [17].

- **Directly Attached Storage:** This means that the hard drives are outside of the DVR, NVR or server but are 'directly' connected without using an IP network.

- **Networked Storage:** A device which is a server that is dedicated to nothing more than file sharing and storing videos from large numbers of cameras, e.g. Network-Attached Storage, NAS. They provide efficient, flexible and scalable storage for very large camera counts but generally at higher cost and complexity [18].

- **Onboard Camera Storage:** This allows the camera itself to record and save videos using, e.g. SD card, and rarely uses hard drives. Thereby, the surveillance system reduces the use of the network resources and it is independent. This is the least commonly used but likely the most interesting for future research.

Question 4 considers ways of viewing the recorded videos. Surveillance video is ultimately viewed by human beings. However, most surveillance video is never watched except for when it's needed for historical investigations. Some surveillance video is viewed live continuously or stored to be retrieved later. Especially, if the system has an intelligent event detection module, which stores only the relevant videos and does not consider all events round the hour.

- **Local Viewing:** This means there is a direct view from the DVR, NVR or servers. This way is ideal for monitoring small areas. It makes the video management system a local station, therefore it reduces costs.

- **Remote PC Viewing:** This is the most popular way of viewing videos in surveillance systems. In this approach, standard PCs are used to view live and recorded video. Usually, a web browser is used and users do not have to install or to worry about upgrading a client.

- **Mobile Viewing:** This allows users of a specific surveillance system to check immediately surveillance video using smart phones, e.g. iPhone, iPad and Android.

- **Video Wall Viewing:** This might be the best solution for large security operation centers that have hundreds or thousands of cameras. Video walls provide very large screens so that a group of people can watch.

Generally, the main focus during the design step is to choose the right sensors, the right network, the right coding and storage concepts and the optimal methodological approaches for automatic event detection. Therefore, the main goal of the design process is to build a video surveillance which verifies the following requirements:

- **Robustness:** The ability of a system to cope with errors and mistakes (internal or external factors) during the operation [19].

- **Reliability:** The ability of the system to perform its required functions under stated conditions for a specified period of time [19].

- **Multimodal:** The ability of the system using different types of sensors [20].

Finally, the overall design process depends on the desired application and the requirements of the user and the specified use case of the surveillance system.

2.2.3 The performance requirements

Hence, the most important requirement is the performance which means the accomplishment of the surveillance system requirements measured against preset known standards of accuracy, completeness, cost and speed. The major design and performance requirements are:

- **Real-Time:** Real-Time video surveillance systems must guarantee response within strict time constraints.
- **Detection Rate:** The automatic event detection system must be done in a high accuracy rate[20].
- **Low Resource Consumption:** A low measure of the resources, e.g. hardware and energy, is needed for the events detection or for the completion of a process or activity[21].

2.3 Methodological approaches for surveillance systems requirements

There are well defined methodological approaches for the functional, design deployment and performance requirements of surveillance and monitoring systems. The approaches described in the next subsections satisfy those requirements in the previous section and have been tested. The following approaches come from two different points of view: industrial and research. The industrial solutions should always be cost effective, easy to maintain and perform well. The scientific point of view considers the high performance of the automatic event detection system in the surveillance system, the power consumption issues and the good environmental solutions.

2.3.1 Existing approaches for functional requirements

IP video cameras can monitor the monitoring areas in real-time and alert to suspicious activities. They also can record events and produce valuable evidence. While some IP cameras are designed strictly for indoor placement, others are weatherized for outdoor use[3].

IP Network Cameras give the ability to create and maintain an effective and reliable IP surveillance system. They are cost effective solutions where users can build a high performance and a scalable wired or wireless IP video surveillance system. It helps users to monitor at any time, allows them to send live images and audio for remote monitoring, learning, troubleshooting, web broadcasting and any other activity requiring a remote presence.

Multiple users can control view and manage the system in real-time footage anytime using web browsers. Furthermore, they offer high resolution videos which can help to monitor the target area. Also, using IP surveillance systems makes installation and maintenance very easy. Usually, IP cameras must be configured for resolution, frame rate and server IP address to capture videos. In surveillance systems, it is possible that

[3]http://www.cisco.com

at least the frame rate and resolution could change at times throughout the day. Figure[4] 2.3 shows the standard functional requirements of surveillance systems.

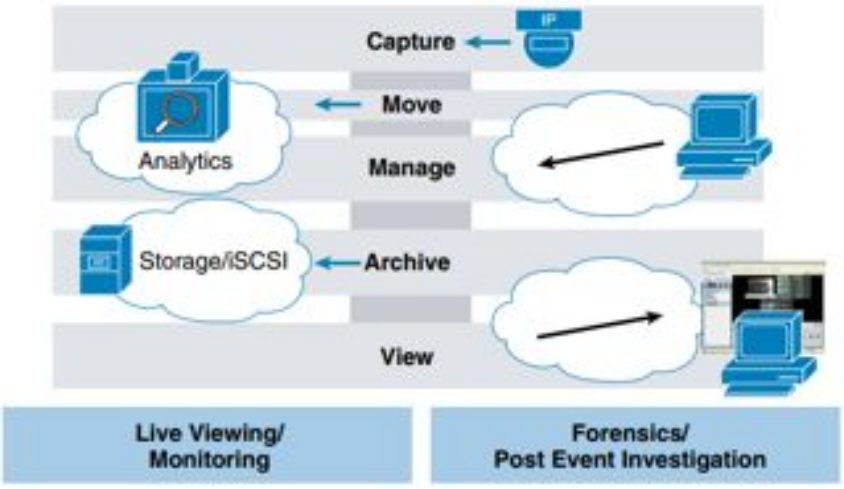

Figure 2.3: The standard functional requirements of surveillance systems (Cisco) system

Therefore, simply capturing data requires some control plane network traffic as well as keeping the clock of the camera in synch with a universal clock through protocols such as Network Time Protocol (NTP)[5]. NTP is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks[6].

A collective bandwidth is highly necessary when the deployment is made up of hundreds or thousands of cameras because of the possible packet loss during data transfer between different system nodes.

Media gateway devices, e.g. circuit switch or IP gateway, converts data from the format required for one type of network to the format required for another. Data input could be audio, video, or T.120 (real-time multi-point communications), which the media gateway would handle simultaneously. The media gateway controller is sometimes called a softswitch[7].

2.3.2 Existing approaches for design requirements

Most of modern surveillance systems are using a mix of different camera types. For instance, an organization may use infrared fixed cameras around a perimeter with a Pan Tilt Zoom (PTZ) overlooking the parking lot outdoors. Indoors, they may have a fixed megapixel camera covering the warehouse and a number of fixed IP cameras covering the entrance and hallways.

Hybrid NVRs and DVRs support IP cameras and are directly connected to analog cameras. This provides simplicity and flexibility. Customers can continue working with their existing analog cameras and slowly migrate to IP. Therefore, it decreases the costs

[4]http://www.cisco.com
[5]http://www.cisco.com
[6]http://tools.ietf.org/html/rfc5905
[7]http://searchunifiedcommunications.techtarget.com/definition/media-gateway

and provides many advantages as mentioned in the previous section. Hybrid DVR and NVR systems are the best choice. The lower cost, easier deployment and lack of client changes needed will make the hybrid DVR/NVR very attractive for applications needing recording of moderate camera counts at distributed facilities.

When selecting storage for an IP surveillance system there are four standard options: internal and Direct-Attached Storage (DAS), Network-Attached Storage (NAS), Storage Area Networks (SAN) and on-camera edge storage. All of these have a place in surveillance applications, with different manufacturers supporting different options.

The majority of surveillance projects still prefer using internal or direct attached storage where the hard drives are built inside of a DVR, NVR or server. This method is still the most used form of storage. However, networked storage is gaining in popularity.

A major problem is still whether video surveillance storage has redundancies, specifically how likely it is of a hard drive to crash. This problem is now becoming more and more common.

The solution is to use a data warehouse which is a database used for reporting and data analysis. It is a central repository for data which is created by integrating data from multiple disparate sources. The major disadvantage of this is that a data warehouse can be costly to maintain and this becomes a problem if the warehouse is underutilized. It seems that managers have unrealistic expectations about what they will get from having a data warehouse [22].

Virtualization techniques are important and should be used to segment the video endpoints and servers. The used PCs must have a sufficient CPU clock rate to decode the video feeds.

Camera feeds traverse the IP network from the camera source to the Media Server either as Motion JPEG (MJPEG) or MPEG-4. The Moving Picture Experts Group[8] (MPEG) is a working group of experts that was formed by International Organization for Standardization (IOS) and International Electrotechnical Commission (IEC) to set standards for audio and video compression and transmission.

MJPEG is typically transported via the Transmission Control Protocol (TCP). TCP provides guaranteed delivery of packages by requiring acknowledgment from the receiver. Packages that are not acknowledged are retransmitted. With MJPEG, each image stands alone, so the images that are displayed are of a good quality. MPEG-4 video is typically transmitted over the User Datagram Protocol (UDP), Real-Time Transport Protocol (RTP), or Real-Time Streaming Protocol (RTSP). UDP does not guarantee delivery and provides no facility for retransmission of lost packages. Table 2.1 shows the major differences between TCP protocol and UDP protocol.

Table 2.1: A comparison between TCP and UDP

TCP	UDP
Reliable	Unreliable
Connection oriented	Connectionless
Segment sequencing	No sequencing
Acknowledgment segments	No acknowledgment
Segment transmission and flow control through windowing	No windowing

[8] John Watkinson, The MPEG Handbook

2.3.3 Existing approaches for performance requirements

There are different approaches for event detection and recognition (actions and activities) whereat every approach has its advantages and disadvantages. The limitations of the state-of-the-art will be considered deeply in each of the Chapters 3, 4 and 5.

The researchers in [23] divided the main approaches into the following categories:

- Non-parametric approaches, e.g. Dimensionality reduction, Templates matching (2D and 3D) [24] [25] [26]).
- Volumetric: e.g. space time filtering, tensors and sub volume matching, [27], [28] and Support Vector Machine (SVM) [7].
- Parametric: e.g. Hidden Markov Models [29] [30].
- Graphical Models: e.g. Petri nets, propagation nets and dynamic Bayes nets.
- Syntactic: e.g. Context free grammars and attribute grammars [31]).
- Knowledge Based: e.g. logic rules and ontologies [32] [33].

Figure 3.1 shows an overview of action and activity recognition from the state-of-the-art. In the next Chapters the previous approaches will be discussed in details, the advantages and disadvantages of every approach will be considered and a comparison between the novel proposed solutions and the approaches listed in the state-of-the-art will be illustrated. Most existing state-of-the-art methods for event/object recognition are model based systems that are computationally expensive to run on tiny embedded platforms.

Another challenge is the detection of objects in ultra-fast computation time; this is also needed where the driver has no time to think in advanced driver assistant systems, for example. Reasoning about context based on the ontology supports the representation of both ontological and probabilistic knowledge; we could construct a context knowledge base for the application domain. Reasoning about context information in the domain is supported by three types of reasoning mechanism: ontological reasoning, rule-based reasoning and Bayesian reasoning.

The rule-based reasoning mechanism is the default reasoning mechanism supported by the context ontology. The ontological reasoner can be described as an instance of the rule-based reasoner. It works by propagating implication, predefined rules over the instance data. Probabilistic reasoning which uses as a standard Bayesian inference can be used to answer queries about the values of the properties of the instances.

Ontologies based on Semantic Web provide concise high-level definitions of activities but they do not necessarily suggest the right hardware to parse the ontologies for recognition tasks (Semantic Web) [34] [23]. Context Free Grammars expect perfect accuracy in the lower levels; they are not suited to deal with errors in low level tasks.

In complex scenarios involving several agents requiring temporal relations that are more complex than just sequencing, such as parallelism, overlap, synchrony [31]. Though Petri nets are an intuitive tool for expressing complex activities, they suffer from the disadvantage of having to describe manually the model structure [35] [23].

In Bayesian networks the evidence of one cause reduces the possibility of another cause given the evidence of their low prior probability which is especially difficult to model

in logical rule-based systems. Nevertheless, a fundamental limitation of using Bayesian network for knowledge representation is that it cannot represent the structural and relational information. Also, the applicability of a Bayesian network is largely limited to the situation which is encoded, in advance, using a set of fixed variables [36].

2.3.4 Existing approaches for deployment and operations requirements

Video surveillance deployment consists of cameras, video management software, servers, and storage. The IP network connects all these components into a converged network infrastructure. If the surveillance system is deployed in a small area the system components, video management software, server and storage components can be as simple as a single PC, an IP camera, and a simple Ethernet hub[9].

Very large deployments may need thousands of IP cameras, hundreds of servers, and a storage subsystem of terabytes capacity.

A collective bandwidth is highly required when the deployment is made up of hundreds or thousands of cameras because of the possible package loss during data transfer between different system nodes.

Managing the system also influences the network bandwidth requirements. For example, to schedule a backup of an archive, a sufficient bandwidth is required.

Quality of Service (QoS) is responsible for managing network congestion during periods where bandwidth is constrained. QoS manages the access to bandwidth by competing applications through prioritizing one application over another.

Security in the network is important to define the purposes:

- Where is equipment to be placed on the network?
- Who may access network equipment?
- How is access to this equipment controlled?
- How is data traveling over the network protected?

There are different approaches to surveillance systems deployments regarding the optimal deployments of the network sensors. In [37] they use a binary optimization scheme based on the branch and bound algorithm. They translate the camera constraints and the video processing requirements into spatial coverage.

Authors of [38] convert the resolution and field of view constraints on sensors into distances using an analytical process.

Authors of [39] propose an approach that relies on a spatial translation of constraints. Their approach is for fast exploration of potential solutions and hardware acceleration of inter-visibility computation.

[9] http://www.cisco.com

2.3.5 A global critical judgment of all various existing methodological approaches

The major questions of industrial companies are: What do they really want to accomplish with video surveillance? What are the costs related to video surveillance?

First, it is important to know the differences between analog vs. IP-based video. Analog is a standard traditional video where cameras are just recording devices. They can be networked, but it is limited by the technology. IP video incorporates robust capabilities that can support companies for a long time after the network deployment.

Experts estimate the current market is at about 80% analog and 20% IP[10]. Among those, there is an increased number of IP video surveillance deployments. Undoubtedly IP will replace analog.

In telecommunications, 4G is the fourth generation of mobile communications standards. A 4G system provides mobile ultra-broadband Internet access. The advent of 4G networks promise significantly more bandwidth and higher definition video. Therefore, it is an appropriate solution to video surveillance systems.

The question of cost is still mandatory. Videos are much more bandwidth intensive than data and therefore the most expensive to transmit. IP video systems are attractive because of their capability of video understanding and event detection. They have the advantage of enhanced image quality and the ability to remote via web-based applications.

Choosing the best deployment of the sensor in the field of monitoring should be optimized to provide the optimal coverage and lesser costs of deployments.

There are different optimization approaches that can be used [37] [38] [39].

PTZ stands for Pan, Tilt, and Zoom. A PTZ dome security camera differs from a fixed dome camera in that it can move left and right (pan) or up and down (tilt)[11].

PTZ dome security cameras have several advantages over traditional PTZ security cameras. They can move in all directions including 360 degree rotation and viewing straight down. Dome cameras utilize auto-flip to view objects directly under them. Auto-flip gives the camera the possibility to rotate automatically when something passes directly below it[12]. They provide the possibility of recording colored images during the day and black and white at night.

The disadvantage of wireless IP cameras is the security challenge, it is a difficult task to keep the network secure. The network needs experts in security to manage it and this can increase the overall cost of the system. The second disadvantage, mainly for large indoor areas, is a limited range of the wireless signal. In some cases the range of the wireless signal may not be sufficient to traverse through walls and the camera image will suffer[13].

Also, the data transfer has the problem of delay in large outdoor areas because of the bandwidth, encoding, decoding and transmitting between the nodes of the network.

[10] http://www.tyco.com
[11] http://www.securitycamera2000.com/help/questions/95/
[12] http://www.securitycamera2000.com/help/questions/95/
[13] http://EzineArticles.com/4907258

2.4 Summary

In Chapter 2, the major functional, design and performance requirements of surveillance systems have been comprehensively considered. We suggested the most relevant question for the functionality, design and performance of video surveillance systems.

Then, the answer of every point was illustrated in a detailed way. The overall methodological approaches from scientific and industrial point of view have been explained. The best choice regarding the requirements of the companies, associations or organizations depends on the use case and the sensitivity needed for the desired surveillance system.

The video surveillance system has to provide different functionalities, e.g. remote accessibility, spatio-temporal intelligent event detection, easy integration possibilities, cost-effectiveness and network sensors configuration. The design of modern video surveillance systems depends on the use and the requirement of users, however the most popular used nowadays are IP PTZ cameras in a sensor network.

Hybrid NVRs and DVRs video management systems support IP cameras and are directly connected to analog cameras. This provides simplicity and is cost effective for users who are still using the analog cameras. The performance requirements of video surveillance systems should consider the real-time, high detection rate and low resources consumption as important. There are different related approaches for performance requirements. Each approach has its advantages and disadvantages which will be addressed in the next Chapter. Clearly, the decision is always based on the requirements of users.

Chapter 3

Spatio-temporal context modeling and reasoning

Video surveillance systems provide many research challenges but the most interesting challenge is the one which considers human behavior. Human behavior combines both spatial and temporal resolutions in nature. This means that context becomes all important. Suppose a person is lying down on the floor for longer than 5 minutes in the kitchen. It could be a normal behavior if it was in the living room, but otherwise it is unusual.

In this example, the context can include spatial resolution on various scales (it is a kitchen and people do not lie on the floor of the kitchen to sleep). Another example is turning on the fire place in summer. Here, the context can include temporal resolution on different scales.

In the previous two examples, we explained the meaning of spatial and temporal resolution. It could also include information about how they reached their current situation: if the person went from standing to a lying position very suddenly there would be rather more cause for concern than if the person first knelt down and then lowered himself onto the floor. Representing all of these different temporal and spatial aspects together is a major challenge for video surveillance systems research.

In this Chapter, an overview of the meaning and the definition of knowledge representation and reasoning will be provided [40]. Ontologies are specifications of what exists or what we can say about the world. People were continuously trying to attempt to find ways to express their knowledge. Physics and mathematics have their own specific symbolic languages and many approaches to artificial intelligence with regard to finding the problem's optimal representation as most of the solution [41].

Therefore, ontologies and context models should be defined and play an important role in spatio-temporal event detection. Furthermore, the requirements of building a consistent ontology will be listed and explained.

The major methodological approaches for knowledge representation and reasoning based on logic programming will be illustrated. Sequentially, an overall view about the important methodologies from the state-of-the-art will be explained, these are dealing with spatio-temporal reasoning. Finally, the limitations of the proposed approaches will be considered and discussed in details.

3.1 Knowledge representation

Before the explanation of knowledge representation the meaning of knowledge has to be considered first. This question has been discussed by philosophers since the ancient Greeks. To understand the meaning of knowledge, it is important to look at how we talk about it informally.

First, observe that when we say something like, "Martin knows that his child comes home every day at 13:00 from school." This suggests that among other things, knowledge is a relation between Martin (as the person who knows this fact) and a preposition, which is the idea expressed by a simple declarative sentence, like, "His child comes home every day at 13:00 from school."

A similar story can be told by a sentence like, "Martin hopes that his child will come today at 12:00." The same proposition is involved but the relationship is different. Verbs like "knows", "hopes", "regrets", "fears" and "doubts" all denote propositional attitudes, relationships between agents and propositions [42].

A related notion that we are concerned with is the concept of *belief*. The sentence "Martin believes that x" is clearly related to "Martin knows that x". Therefore, in the first sentence we are not sure about the level of confidence which we have to give to Martin regarding x. In the second sentence, we can say that Martin is sure about what he knows regarding x.

Now, representation means a relationship between two domains where the first is meant to "stand for" by using the second domain. Usually, the first domain is a symbol because it is assumed to be easier to deal with symbols. The second domain is the meaning of that symbol. Children do not always represent their knowledge well, for example if they say "chocolate". Based on that word parents can directly understand that the child wants to eat a chocolate. It is not necessary for children to formulate a well formed sentence to express what they want. *"Knowledge representation then, is this: it is the field of study concerned with using formal symbols to represent a collection of propositions believed by some putative agent. As we will see, however, we do not want to insist that these symbols must represent the propositions believed by the agent. There may very well be an infinite number of propositions believed, only a finite number of which are ever represented [42]"*.

3.1.1 Why knowledge representation?

Knowledge representation is useful to describe the behavior of complex systems. Imagine, we have to design an intelligent system which can play chess autonomously. Surely, in the first step we have to represent the chessboard which is 8 by 8 black and white fields. Then we have to represent the rules of chess, e.g. how rooks, pawns, bishops, knights, queen and king can move. Consequently, we have to define how they can be killed? (out of the board), what is the goal of the game? (killing the king), what are our beliefs? (the location of the current figures). Finally, we have to define a winning strategy.

All of the previous steps could not be done without knowledge representation. We can symbolize the complex system and then we can use those symbols to define the beliefs, desires, goals, intentions and hopes.

Therefore, we can realize that systems based on knowledge representation have the fol-

lowing features:

- Developers can add new rules and easily make them depend on previous rules.
- Developers can extend the existing system description by adding new beliefs (facts).
- Developers can debug faulty system descriptions by the conductors between the beliefs (facts) and the rules.
- Developers can concisely explain and justify the behavior of the system.

Furthermore, the hallmark of a knowledge-based system is that by design it has the ability to know facts about its world and adjust its behavior correspondingly [42].

3.1.2 Ontologies in relation with context models

Ontologies are widely accepted instruments for the modeling of context information in video based surveillance systems.

"An ontology term is borrowed from philosophy, where an ontology is a systematic account of existence. For artificial intelligence systems, what "exists" is that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. Thus, in the context of Artificial Intelligence (AI), we can describe the ontology of a program by defining a set of representational terms [43]. "

An ontology of a video based surveillance system combines the names of entities (spatio-temporal) in the scene, e.g. classes, relations, functions, or other objects. For example, suppose we are trying describe a vehicle using a simple taxonomy:

- Ground vehicle
- Motorcar
- Four or more wheel car
- Car
- Truck
- Motorbike
- Train
- Ship
- Aircraft

Now, for building an ontology of vehicle we have to define the classes, relations and functions as follows:

- **Vehicle world**
- **Type**
- Ground vehicle
- Ship
- Aircraft
- **Function**
- To carry persons
- To carry freights
- **Attribute**
- Power
- Size
- **Component**
- Engine
- Body

Traditionally, for building a context model at the beginning of the application and its functionality has to be defined and then the important context ontologies have to be developed [44] [45]. In video based surveillance systems the context model means all the information that may influence the way a scene is perceived. The "state" of an environment is defined as a conjunction of predicates. The environment must be modeled to get the information observed in the environment; the position, orientation and types of objects. As well as position, information and the state of other objects must be observed. The first step in building a context model is to specify the desired system behavior. The developer then lists a set of possible scenarios, where each scenario is a relationship between entities and relations to be observed.

3.1.3 Overview of existing context models tools

Development of context-aware applications is inherently complex. These applications adapt to changing context information: physical context, computational context, and user context/tasks. Context information is gathered from a variety of sources that differ in the quality of information they produce and that are often failure prone. Traditionally, for building a context model at the beginning the application and its functionality has to defined and then the important context ontologies has to be developed [44] [45]. In video based surveillance systems the context model means all the information that may influence the way a scene is perceived. There are different context modeling approaches [46]:

- **Ontology Based Models of Context Information:** The formalism of choice in ontology based models of context information is typically OWL-DL [47] [34] [23]. Ontology based context modeling is widely used in various application domains and it is supported by a number of reasoning services [48].

- **Key-Value Models** This approach uses the most basic data structure capable of providing a basis for context modeling. They employ key words and KeyValue Pairs to represent data to define and implement contextual information. They are considered as the weakest context modeling approach because they do not provide a descriptive or an expressive spatio-temporal reasoning [49]. Key-Value Pairs combine links, nodes and other hypermedia objects and describe parameters of the context the hypermedia objects belong to. Objects described by key-value pairs are typically used to detect in which context the objects are visible.

- **Markup Scheme Models:** Markup languages are characterized by a hierarchical data structure using a combination of tags with attributes and content. The advantages of Markup Scheme Models are that the content of the context is generally defined recursively by other (markup) tags in a nested structure [46]. E.g., XML and RDF [50]. The Resource Description Framework,(RDF) is a W3C[1] technology. RDF describes resources with properties and property values. A property is a resource that has a name, for example a person or an animal. A property value is the value of the property, for example "Alexander" or "cat".

- **Graphical Models** The Graphical Models are usually used to model the context generally, e.g. Unified Modeling Language (UML) [46] [51]and Temporal Entity relationship diagrams [52]. The UML is a non-temporal conceptual modeling language. CML is based on Object-Role Modeling (ORM). It provides a graphical notation designed to provide the software engineer a comprehensive way for analysing and understanding the context requirements of a context-aware application [53]. The formality of ORM and the Context Modeling Language (CML) extensions makes it possible to support a straightforward mapping from a CML-based context model to a runtime context management system that can be populated with context facts and queried by context-aware applications [48].
  CML supports the evaluation of simple assertions as well as SQL like queries. It offers the ability to support querying over uncertain information. CML has several weaknesses. It has a "flat" information model in that all context types are uniformly represented as atomic facts. It emphasises only the development of context models for particular applications or application domains.

- **Object Oriented Models:** There is a corollary between the Graphical Modeling (GM) approach and the Object Oriented Modeling approach (UML) in that they are both predicated on the principle of Object-Orientation (OO) [46]. UML is a modeling language used to express and design documents, software and systems. Independent of implementation languages UML can be used from general initial design to very specific detailed design across the entire software development life

[1] http://www.w3.org/

cycle. It is a multi-diagrammatic language where each diagram is a view into a model. The diagram is presented from the aspect of a particular stakeholder and provides a partial representation of the system.

- **Logic-Based Models:** Logic addresses different scenarios in which an expression or facts may be derived. In logic-based context models a context is defined using facts (context properties) with expressions and rules to describe and define relationships and constraints [54] [55] [46]. In constraint programming, for example, mapping the high-level specification of a problem into constraints that will lend themselves well to processing also requires certain mathematical background and expertise in constraint modeling and solving. In prolog, to be an effective Prolog programmer one needs to understand how to use terms as data structures which is quite difficult.

- **Hybrid Approaches:** It combines different modeling techniques for different purposes often along different levels of interpretation/semantics [56]. For example, a hybrid model that combines the respective advantages of CML and ontology-based approaches.

3.1.4 General requirements for ontology based context models

The major context model requirements for video based surveillance systems are [57][58]:

- **Applicability:** *Does the model restrict the domain of application in any way?*
  The model is useful for all applications that need an abstract description of the video observation area and therefore restricts the domain to this task. It is not intended to be used in completely foreign domains, like a context model that works outdoors does not have to work for intelligent houses.

- **Traceability:** *To what extent does the model provide support for recording of provenance and processing of information?*
  Every object in the context has its reliability, as well quality measurement should be fed in the model. Mapping of quantitative data gathered from surveillance sensors to qualitative abstract values has to be done outside the model, because the input data of sensors differ with various sensing systems and as a consequence of this it is also true for the applicable processing algorithms. The context-model has to be abstract from such details. Since the source of the abstract object is recorded, the mapping can be made available if needed, with reasoning.

- **History and Logging:** *In what ways does the model address the issue of data logging and history records?*
  The long and short term history of objects in a scene,for example the actions of a person in the park, could provide valuable information for prediction of object's behavior.

- **Quality:** *Is the quality of information an issue when it is directly integrated into the model?*
  The model should include an object's source, the source's reliability, a quality measure provided by the source and a time span indicating the object's validity within the knowledge base. In other words, in video surveillance systems the measurement

of recognition accuracy, location accuracy and temporal data has to be involved in the model.

- **Satisfiability:** Does the model check the satisfiability of information context instances?
  The context model has to assign finite domain values; the allowed range interval has to be specified. It is important to be aware in the case of receiving strange object types into the context model.

- **Inference:** *Does the model include tools that permit the definition of new contextual categories and facts on the basis of low-order context?*
  The model should include tools that permit the definition of new contextual categories and facts on the basis of low-order context.

- **Reusability and Standardization:** *To what extent does the ontology allow reusability in other independent modeling tasks?*
  The model should allow reusability in other independent modeling tasks. It means if we have a model describing people, gender, age and most importantly a person's data. It should be possible to reuse such description to build a new model about university where students and professors are also people and have the same attributes as any other person.

- **Flexibility and Extensibility:** *How much effort and changes are needed to extend the ontology model?* New definitions can be added to the context ontology without necessary changes in the existing dependencies.

- **Granularity:** *What is the level of detail for the defined concepts?*
  The context-model consists of abstract objects that together represent a high-level description for complex event recognition in the domain. Refinement to finer descriptions, which are needed for the operative level, is easy (compare to criterion Flexibility and Extensibility).

- **Consistency:** *Are there explicit or implicit contradictions in the model?*
  No contradictions should be found in the ontological content.

- **Completeness:** *Does the ontology cover all relevant concepts, properties? Can the entities and their interactions be modeled?*
  A series of experiments should be done to test the consistency of the designed ontology. Different scenarios have to be modeled with the context-model.

- **Redundancy:** *Are there two or more concepts or instances defined with the same formal definition but using different names?*
  The model should not contain a lot of defined instances that have the same properties. The context-model could contain redundant properties but they should be selected only when it is necessary. For example, the property SensorID in the video features could be the same as property on the audio features class.

- **Readability:** *Does the ontology contain intuitive labels to denominate the ontological entities?*

In video surveillance systems, the labels of classes and properties must be chosen from the domain vocabulary with respect to their understandability by human ontology designers.

- **Scalability:** *Does the model scale well with respect to cognition, engineering and reasoning?*
  Cognitive scalability rates the understanding of the model by humans while engineering scalability assesses the available tool support with respect to the size of the ontology [58]. In video surveillance system networks, the number of entities and the instances generated by the videos are very high. Therefore, the choice of the right modeling tool of the context model is important to respect the real-time requirements issues.

3.1.5 Ontology Web Language (OWL)

Ontology Web Language (OWL) is based on different logical models which simplify the description of the concepts. Therefore, complex concepts can be built up in definitions out of simpler concepts. All rules are expressed in terms of OWL concepts (classes, properties, individuals). This means that rules can be used to extract new knowledge from existing OWL ontologies [59] [60].

3.1.6 Semantic Web Rule Language (SWRL)

In keeping with many other rule languages, SWRL rules are written as antecedent-consequent pairs. In SWRL terminology, the antecedent is referred to the rule body and the consequent is referred to the head. The head and body consist of a conjunction of one or more atoms. SWRL rules reason OWL individuals, primarily in terms of OWL classes and properties [61].

3.1.7 Judgment criteria of context modeling approaches

Context models are using the following 6 criteria derived from the survey of approaches to context modeling (see Table 3.1) [46] [62]:

1. **Distributed Composition (dc):** The possibility of the implementation in dynamic distributed systems for pervasive computing goals. Pervasive computing is the idea that almost any device around us can be imbedded with chips to connect the device to an infinite network of other devices. The goal of pervasive computing is to combine current network technologies with wireless computing, voice recognition and artificial intelligence to create an environment where the connectivity of devices is embedded in such a way that the connectivity is unobtrusive and always available[2]. Therefore, the context model should be able to be integrated with other contexts of the environment.

2. **Partial Validation (pv):** The possibility to validate contextual knowledge on a structural level as well as on an instance level against a context model as a result of

[2]www.webopedia.com/TERM/P/

distributed composition. The context model has to be capable to check the consistency of the ontology and find contradictories during the design process. Especially, consistency of ontologies is important when autonomous system agents are using ontologies in their reasoning. Reasoning with inconsistent ontologies can lead to erroneous conclusions.

3. **Richness and Quality of Performance (qua):** This means that context modeling approaches must inherently support quality and richness indication with respect to different uncertainty dimensions.

4. **Incompleteness and Ambiguity (inc):** This means that context models must incorporate the capability to handle uncertainty by interpolation of incomplete data on an instance level. Uncertainty should be handled in the context model. In real life scenarios, sensors do not provide correct sensed data or they send incomplete information. Reasoning without uncertainty handling in rule based systems can lead to wrong conclusions.

5. **Level of Formality (for):** This means the possibility of a precise representation of facts and rules of the domain. Edmund Husserl introduced a descriptive ontology (that it is an ontology) that concerns the collection of information about a list of objects that can be dependent or independent items (real or ideal). A formalized ontology attempts to construct a formal codification for the results descriptively acquired at the preceding levels[3]. The context model has to be built with respect to descriptivity and formality.

6. **Applicability to Existing Environments (app):** It is important that a context model is adaptable to enable use in existing domains, systems and infrastructure, such as ad-hoc networks and Web Services. The network is ad hoc when it does not rely on a preexisting infrastructure. An ad hoc network typically refers to any group of networks where all devices have equal status on a network and are capable to associate with any other ad hoc network devices in link range. A web service is a software function provided at a network address over the web or the cloud. It is a service that is "always on" as in the concept of utility computing.

Table 3.1: The criteria derived from the survey of approaches to context modeling

Approach	dc	pv	qua	inc	for	app
Key-Value-Pairs Mod.	-	-	-	-	-	+
Markup Scheme Mod.	+	++	-	-	+	++
Graphical Mod.	-	-	+	-	+	+
Object oriented Mod.	++	+	+	+	+	+
Logic-Based Mod.	++	-	-	-	++	-
Ontology-Based Mod.	++	++	+	+	++	+

Table 3.1 shows that key-value-pairs are the weakest approach but they can be integrated with existing domains. Ontology based context models are the best for contextual

[3]www.ontology.co

information despite of their disadvantages regarding real-time constraints. Object oriented models are also good but they are not optimal in the level of uncertainty management. Mark-up schemes have advantages in the level of formality and consistency checking but they suffer from the lack of uncertainty handling. Graphical models do not offer formality and expressivity. Overall the best choice for pervasive computing requirements is the object oriented models, logic based models and ontology based models. Markup schemes are the optimal to enable the use of the context model in existing domains, systems and infrastructure, such as ad-hoc networks and Web Services.

3.1.8 Description of the limitations while considering the fixed criteria

In this section the limitations of context modeling approaches will be listed, see Table 3.2, page 52:

- **Ontology Based Models of Context Information:** The main problem with this approach is that reasoning in OWL-DL is already an expensive computation. [47] [34] [23].
- **Key-Value Models:** They are considered as the weakest context modeling approach because they do not provide a descriptive and an expressive spatio-temporal reasoning [49].
- **Markup Scheme Models:** Markup languages are characterized by a hierarchical data structure using a combination of tags with attributes and content. The advantages of Markup Scheme Models is that the content of the context is generally defined recursively by other (markup) tags in a nested structure [46]. E.g., XML and RDF [50].
- **Graphical Models:** This has a "flat" information model, in that all context types are uniformly represented as atomic facts. It also emphasises only the development of context models for particular applications or application domains [63] [53].
- **Logic-Based Models:** In constraint programming for example mapping the high-level specification of a problem into constraints that will lend themselves well to processing requires certain mathematical background and expertise in constraint modeling and solving. In prolog, to be an effective Prolog programmer one needs to understand how to use terms as data structures which is quite difficult. [54] [55] [46].
- **Hybrid Approaches:** Despite solving some challenges but hybrid approaches still share the limitations of the combined paradigms[56].

3.2 Reasoning

In general, reasoning is the formal processing of the symbols representing a collection of believed propositions to conclude representations of new ones (*beliefs* or facts).

3.2.1 What is reasoning and why reason

Reason is the capacity for consciously making sense of things, for establishing and verifying facts, and changing or justifying practices, institutions and beliefs based on new or existing information [64]. Given a precondition or premise, a conclusion or logical consequence and a rule or material conditional that implies the conclusion given the precondition, one can explain that logical reasoning has three main types:

- Deductive reasoning determines whether the truth of a conclusion can be specified for that rule, based solely on the truth of the premises. Example, "When it rains, grass outside gets wet".
- Inductive reasoning means that a conclusion can be taken after numerous examples are described in terms of such a rule. Example, "The grass got wet numerous times when it rained." Therefore, the grass always gets wet when it rains.
- Abductive reasoning selects a set of preconditions based on a true conclusion and a rule. Then it tries to select some possible facts that, if true also, can support the conclusion. Example, "When it rains, the grass gets wet."The grass is outside and nothing outside is dry." Therefore, maybe it rained.

Let us go back to the example, "Martin knows that his child comes home every day at 13:00 from school" and "Martin knows that his child comes directly home after finishing school". Now imagine that the child of Martin does not come home today at 13:00 pm. Now, the father starts to worry and wonders what has happened to his child. The way of thinking that the father adopts to try to know what is going on now with his child is "reasoning". First, the father could call the school to ask if the child is still there or not. Here, from the previous knowledge of the father, we see that by the use of his knowledge (beliefs) the father has started to worry. By calling the school he is trying to extract new knowledge to find a reason why his child does not come home on time. This is what we call *logical inference*. Reasoning is important for several reasons. First, reasoning is necessary to be able to make decisions based on the factual nature of a situation and not just an emotional response. For example, there are many illegal drugs that let people feel better or more powerful but in reality they have very harmful effects. For example, we might represent the following two facts explicitly:

- Person p is lying on the ground near the fireplace and the fire place is on.
- It is winter.

Now, by a logical reasoning we are able to say this is normal behavior and it is possible that a person who feels cold will lie nearby the fireplace in winter. But suppose now we remove the fact "It is winter" and we add a new fact "It is summer", so a logical reasoning will say this behavior is abnormal.

3.2.2 Rule engines

In inductive machine learning and data mining from large databases, it is important to know that the background knowledge can be used as good guidance for extracting information from the data. To achieve this goal a rule engine is needed. Rule-based

systems are successfully applied across a lot of domains. Interest in ontologies has become stronger to develop a common rule base that could be computed by different rule engines. Several rule languages have been developed such as RuleML, SWRL, Metalog and ISO Prolog among others. The Semantic Web Rule Language (SWRL) is intended to be the rule language of the Semantic Web. Rules can be given manually through the combination of facts using propositional logic or rules can be defined automatically when they are complex and we do not know the relationship between the extracted features, especially when they are numerical data. One of the most important rule generators is the Rough Set theory. It has many applications in information retrieval, data mining, expert systems and decision support. Almost all databases contain imperfections, such as missed values, noise or errors. The Rough Set theory is a good solution for dealing with these types of problems [65] [66]. The rule engine at a high level consists of three components: ontology, rules and data.

As previously mentioned in Chapter 3, the ontology is the representation model which is used for a specific environment. The rules do the reasoning and facilitate thinking. One of the well known rule engines are Jess[4], Jena[5] and pellet[6].

The term rule engine is quite ambiguous in that it can be any system that uses rules, in any form, which can be applied to data to produce outcomes.

A rule-based reasoning mechanism is the default reasoning mechanism supported by the context ontology. The ontological reasoner can be described as an instance of the rule-based reasoner. It works by propagating implication, which is a set of predefined rules over the instance data. Probabilistic reasoning that uses as a standard Bayesian inference can be used to answer queries about the values of the properties of the instances.

3.2.3 The requirements for spatio-temporal reasoning

Event detection combined with context modeling is wide spread. Researchers describe the automatic generation of event models based on qualitative reasoning and give statistical analysis of video input. The use of an existing tracking program which generates labeled contours for the objects in each frame and the view from a fixed camera is partitioned into semantically relevant regions based on the paths followed by moving objects. The objects that are moving along the same path at different speeds can be distinguished due to path indexing with temporal information. Via the usage of statistical methods event models can be created. They describe the behavior of pairs of objects.
Consequently, there are different fronts of requirements which should be considered and specified related to the goal of the surveillance system. For example should the designed system detect events in real-time?, e.g. the detection of vandalism in metro stations. Should the system detect events in a middle/long term point of view?, e.g. the detection of abnormal behavior because of the unusual change of people trajectories in the national park. Finally, what is the level of confidence of event detection of a surveillance system? The answer to the previous questions will be discussed in the next sections.

[4]http://herzberg.ca.sandia.gov/
[5]http://jena.apache.org/documentation/inference/index.html
[6]http://clarkparsia.com/pellet

Spatio-temporal reasoning (Short Term/Long Term)

As was pointed out before, interpreting human behavior in context involves reasoning about space and time. For example, preparing a meal at noon in the kitchen is usually perfectly normal behavior, but if the same activity occurs at 3 in the morning in the living room, it is behavior that needs some special attention. The requirements of spatio/temporal reasoning, short term/long term systems [67]:

- **Temporal inference observation:** The long term event should be detected within a specific time window and the reasoning process has to be performed with respect to the time constraints.
- **High quality low level data**: The raw data should be accurate and complete.
- **High performance sensor fusion:** High quality of the combined raw data from multiple sensors.

Spatio-temporal reasoning (real-time)

In addition to the previous requirements, the major requirements of real-time reasoning are that the system should keep the row data moving "in-stream" without any requirement to store it, to perform any operation or sequence of operations. Another requirement, is that the system should process on chip to reduce data transfer between different components. The requirements of real-time reasoning [68]:

- **Keep the row data moving:** It helps to process messages "in-stream" without any requirement to store them to perform any operation or sequence of operations.
- **Process on chip:** Processing on chip reduces data transfer between different components.
- **A consistent simple ontology:** The inference should be performed based on a simple ontology.

Spatio-temporal reasoning (under uncertainty)

Many specialists in decision theory, statistics and other quantitative fields have defined uncertainty as a lack of certainty. In other words, it is a state of having limited knowledge where it is impossible to describe exactly the existing state, a future outcome, or more than one possible outcome. The major requirements of spatio-temporal reasoning under uncertainty are in addition to all previous requirements. Usually, the measurement of uncertainty is calculated where probabilities are assigned to each possible state or outcome. This also includes the application of a probability density function to continuous variables [69]. The major origin of uncertainty is the lack of confidence on sensor data and sensor fusion. The major requirements of spatio-temporal reasoning under uncertainty are in addition to all previous requirements.Following requirements are needed [70]:

- **High quality low level data:** The raw data should be as accurate as possible and complete.

- **Posteriori (reflection of reality):** A distribution of reality is needed to be used as a reference during the inference.
- **Degrees of confidence:** It must be possible to express uncertainty in the form of success rates of detection or recognition modules.

3.2.4 Overview of spatio-temporal reasoning approaches

The major approaches to spatio-temporal reasoning (approaches) [23]:

- **Volumetric:** The volumetric approaches do not extract features from video streams frame by frame but they consider the video as a 3d volume, e.g. space time filtering, tensors and sub volume matching [27], [28] and Support Vector Machine [7].
- **Parametric:** They choose a model based on the temporal dynamic of the motion; the events are detected based on training data, e.g. Hidden Markov Models [29] [30].
- **Graphical Models:** They have been used to model complex scenes because of the characteristics of the inherent structure and semantics of complex activities that require higher level representation and reasoning methods, e.g. Petri nets, propagation nets and dynamic Bayes nets [35] [23].
- **Syntactic:** They try to express the structure of a process using a set of production rules to describe the real world events, e.g. context free grammars and attribute grammars [31].
- **Knowledge Based:** First, researchers used order logic and description logic to model the complex scenes. Then they did an inference based on logical rules that are useful to express domain knowledge as input by a user or to present the results of high-level reasoning in an intuitive and human-readable format, e.g. logic rules and ontologies [32] [33].

Most existing state-of-the-art methods for event/object recognition are model based systems that are an expensive computation to run on tiny embedded platforms. Another challenge is the detection of objects in ultra-fast computation time what is also needed.

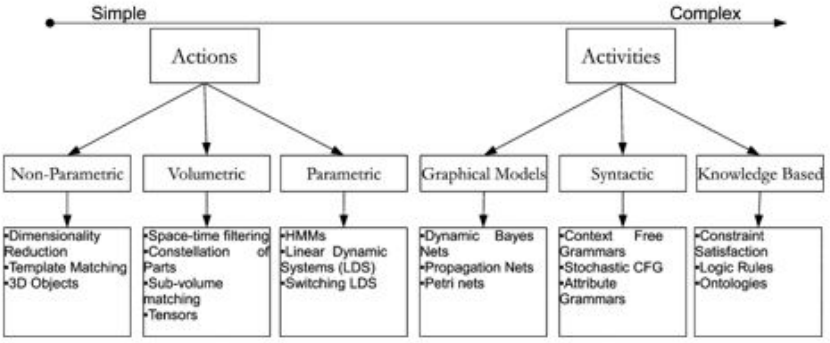

Figure 3.1: An overview of action and activity recognition from the state-of-the-art [23].

3.2.5 Judgment criteria and their justification for spatio-temporal reasoning approaches

Context reasoning approaches should respect the following 4 criteria derived from the survey of approaches to reasoning in context modeling, see Table 3.3, page 52:

- **Embedded platforms:** Is the approach suitable for embedded platforms?
  Surveillance systems have to be capable of running on embedded platforms, e.g. smart cameras. Using embedded platforms the cost of systems can be highly decreased.

  Surveillance systems that are using a central processing unit (central computer) and a repository to archive events need an infrastructure of network connectivity (LAN or WLAN).

  When the design of the surveillance system requires a data transfer and processing in a central station, the security of the system has to be managed well. Otherwise, for example, the privacy issues and evidence for police can cause a real problem if the surveillance system is not well protected. Using embedded platforms can help to reduce the effect of the previously mentioned problems.

- **Temporal reasoning:** Does the approach allow temporal reasoning?
  Temporal reasoning is a major requirement in surveillance systems because of the need to detect long-term events. Temporal reasoning can help to detect complex events. Complex events are recognized by the patterns between simple events.

  A simple event should be detected first, for example a running state or a walking state. Temporal reasoning can be used to detect the sequence of simple events that occurs over time. Additionally, it can be used to detect abnormal behavior, for example in modern smart homes turning on the heater in winter can be abnormal behavior.

- **Real-time:** Does the approach consider the real-time processing?
  Real-time video surveillance systems must guarantee response within strict time constraints. Fire and accidents should be detected as fast as possible otherwise the surveillance system has no sense in being deployed.

- **Uncertainty:** Does the approach allow uncertainty handling?
  Uncertainty means the state of having limited knowledge where it is impossible to describe exactly the existing state or to predict the possible outcome. Logical statements are usually precise about the world in many different forms. They are useful for capturing knowledge and applying it.

  Sometimes it is not possible to express a general statement with the totality of a logical universal. There are cases where it might be a fact or a belief is not sure. In surveillance systems, there are different types of uncertainty. For example, when there is an object in the environment of the surveillance system which is just not known.

  For the reasoning process of a surveillance system, this means that the content of the knowledge base may not have the required details that are necessary for the

decision process. Sensors in real scenarios can deliver incomplete information. It means that there is no hypothesis related to an object or attribute value at all e.g. the object type is known but the speed of the object is unknown.

Furthermore, while uncertainty is concerned with the measure of trust that is put into the data provided by a sensing system, inaccuracy deals with the potential measurement errors that may occur.

Inconsistency can occur during the reasoning process of surveillance systems. This means that there are conflicting hypotheses about an object data, e.g. two sensors are giving different object types with a high belief.

Volumetric approaches, e.g. space time filtering, tensors, sub volume matching and Support Vector Machines, need time for training in the case of Support Vector Machine (SVM) but they can run on embedded platforms. When the support vectors of SVM are generated during the training phase, those support vectors can be used to reason context and enable event detection on embedded platforms. See Table 3.3. Matching 3d volumes is not suitable for embedded platforms because of the need to save a huge number of features of samples to match and the complexity of matching computational approaches.

Using volumetric approaches can recognize events in real-time, but not faster than syntactic approaches. They do not manage uncertainty well and depend on training and matching approaches without a consideration of uncertainty. SVM training needs false positive samples that can reduce the confidence level of the model.

Parametric, e.g. Hidden Markov Models (HMMs), are used because they have proven to be effective in a number of domains, especially in prediction and recognition.

One of the most important advantages of HMMs is that they can easily be extended to deal with complex domains. To detect Markovian assumption means that the emission and the transition probabilities depend only on the current state, which does not map well to many real world scenarios in the frame of complex event detection[7].

Furthermore, HMM supports the detection of several Markovian events because each HMM uses only positive data,. They scale well and can be combined into larger HMMs.

HMMs only use positive data to train. In other words, HMM training involves maximizing the observed probabilities, for examples belonging to a class. However, it does not minimize the probability of observation of instances from other classes. HMM can run on embedded platforms and reason temporal events but it is not optimal for real-time constraints.

Graphical Models share the same criteria as parametric approaches but the prior can effect the probability of another cause, which can influence the whole inference process.

Syntactic Context free grammars and attribute grammars are optimal to run on embedded platforms and in real-time. They suffer because of the lack of temporal reasoning support and the lack of uncertainty management possibilities.

Knowledge based approaches, e.g. logic rules and ontologies, are not optimal for real-time constraints. They depend on the complexity of the ontology, the consistency of the ontology and they do not suggest the right hardware to parse the ontologies for recognition tasks. Therefore, they are not optimal for embedded platforms but perform well regarding uncertainty handling and temporal reasoning.

[7]http://www.cse.unsw.edu.au/ waleed/phd/html/node36.html

3.2.6 Description of the limitations while considering spatio-temporal reasoning

In this section, we consider the limitations of the previous concepts for spatio-temporal reasoning. See Table 3.4, page 52 and Table 3.5, page 53:

- **Ontologies:** Ontologies based on Semantic Web provide concise high-level definitions of activities but they do not necessarily suggest the right hardware to parse the ontologies for recognition tasks (Semantic Web) [34] [23]. Context reasoning is generally feasible for non-time-critical applications. For time-critical applications, such as security and navigating systems, we need to control the scale of context dataset and the complexity of rule set. A tentative solution is to perform static and complex reasoning tasks, e.g. description logic reasoning for checking inconsistency, in an off-line manner. From a system deployment point of view, we need to decouple context processing and context usage in order to achieve satisfactory performance. In this way, context reasoning is independently performed by resource-rich devices, such as a residential gateway; ubiquitous services hosted by this client can acquire high-level context from a centralized service instead of performing excessive computation themselves [71].

- **Petri Nets:** Petri nets are an intuitive tool for expressing complex activities. They suffer from the disadvantage of having to describe manually the model structure [35] [23].

- **Bayesian Networks:** The evidence of one cause reduces the possibility of another cause given the evidence of their low prior probability which is especially difficult to model in logical rule-based systems. Nevertheless, a fundamental limitation of using Bayesian network for knowledge representation is that it cannot represent the structural and relational information. Also, the applicability of a Bayesian network is largely limited to the situation that is encoded in advance using a set of fixed variables [36].

- **Support Vector Machine:** It does not directly model the global geometry of local parts instead considering them as a bag of features [72] [23].

- **Hidden Markov Model:** It does not detect human behavior perfectly because human behavior is not Markovian behavior [73].

- **Context Free Grammars:** Because deterministic grammars expect perfect accuracy in the lower levels, they are not suited to deal with errors in low level tasks [31].

- **Chronicle Recognition System (CRS):** The language includes predicates for persistence and event absence [55]. However, the CRS language does not allow mathematical operators in the constraints of the temporal variables. Consequently, CRS cannot be directly used for activity recognition in video surveillance applications. Logic programming approaches do not explicitly address the problem of uncertainty in the observation input stream.

- **Event Tree:** There is a perfect global synchronous clock that is unsuitable for non-centralized management and distributed systems of clock drift and loose coupling. Due to the lack of consideration of unpredictable delay, it cannot make breaking and mobile detection in a mobile database efficiently [74].
- **Diagram Detection Method:** This only provides the simple time model in which every event is regarded as a certain time point. Atomic events are based on definitions, while complex events are based on semantic [75].
- **Automata:** Automata approaches can neither detect parameter-events nor express event-disorder. Thus, it cannot meet requirements of distributed systems [76].

3.3 Answer Set Programming

The importance of ASP lies in the fact that it provides meaning to logic programs with default negation "¬". Many interesting applications exist in planning, reasoning about action, configuration, diagnosis, space shuttle control, spatial, temporal and probabilistic reasoning, constraint programming, etc.

The Technical University of Vienna (TU-Wien) hosts the research group "knowledge based systems", whose members are running a project on "Answer Set Programming for the Semantic Web". The goal of this project is research towards methods for providing advanced reasoning services in the context of the Semantic Web, using declarative knowledge representation and reasoning techniques.

A logic program in the language of *AnsProlog* (also known as *A-Prolog*) is a set of rules in the form:

$$a_0 \leftarrow a_1, \ldots, a_m, \neg a_{m+1}, \ldots, \neg a_n \tag{3.1}$$

where $0 \leqslant m \leqslant n$, each a_i is an atom of some propositional language and *not* represents *negation-as-failure*. A negation-as-failure literal (or *naf-literal*) has the form $\neg a$, where a is an atom. Given a rule of this form, the left and right hand sides are called the *head* and *body*, respectively. A rule may have either an empty head or an empty body, but not both. Rules with an empty head are called constraints, while those with an empty body are known as *facts*.

A *definite rule* is a rule which does not contain naf-literals and a *definite program* is solely composed of definite rules [77].

Let X be a set of ground atoms, i.e.. all atoms constructed with the predicate in Herband base of a logic program. The body in a rule of the form (3.1) is satisfied by X if $\{a_{m+1}, \ldots, a_n\} \bigcap X = \emptyset$ and $\{a_1, \ldots, a_m\} \subseteq X$. A rule with a non-empty head is satisfied by X if either its body is not satisfied by X, or $a_0 \in X$. A constraint is satisfied by X if its body is not satisfied by X.

Since logic programs unify declarative and procedural representations of knowledge; one way to reason is by using Horn clauses, backward reasoning and Selective Linear Definite clause (SLD) resolution. The *reduct* of a program is a possibility to generate answer sets. Given an arbitrary program, Π and a set of ground atoms, X, the reduct of Π w.r.t. X, Π^X, is the definite program obtained from the set of all ground instances of Π by:

1. deleting all the rules that have a naf-literal $\neg a$ in the body where $a \in X$, and

2. removing all naf-literals in the bodies of the remaining rules.

A set of ground atoms X is an answer set of a program Π, if it satisfies the following conditions:

1. If Π is a definite program, then X is a minimal set of atoms that satisfies all the rules in Π.

2. If Π is not a definite program, then X is the answer set of Π^X. (Recall that Π^X is a definite program and its answer set is defined by the first item [77].

The other advantage of ASP is that the order of program rules does not matter and the order of sub goals in a rule is not relevant. For example, if we have the infamous problem of "3-colorability", where we have a map and we want to check whether 3 colors (blue, yellow and red) are sufficient to color the map. The map is represented by a graph with facts about nodes and edges.

```
vertex(a), vertex(b), edge(a,b).
```

Every vertex must be colored with exactly one color:

```
color(V,r) :- vertex(V), not color(V,b), not color(V,y).
color(V,b) :- vertex(V), not color(V,r), not color(V,y).
color(V,y) :- vertex(V), not color(V,b), not color(V,r).
```

No adjacent vertices may be colored with the same color

```
:- vertex(V), vertex(U), edge(V,U), col(C), color(V,C), color(U,C).
```

Of course, we need to say what the colors are:

```
col(r).
col(b).
col(y).
```

After running this program we will get all possible coloring cases to color the whole map with three different colors. The other advantage of ASP is that the order of program rules does not (a) matter and the order of the sub goals in a rule does not a matter also.

3.3.1 Logic programming with ordered disjunction

Logic programming can be extended to allow us to represent new options for problems in the head of the rules. ASP gives us this ability by the way of ordered disjunctions. Using ASP under specific conditions reasoning from most preferred answer sets gives optimal problem solutions. Through Logical Programs with Ordered Disjunction (LPODs), such as normal logic programs, we are able to express incomplete and unfeasible knowledge through the use of default negation. This allows us to represent performances among intended properties of problem solutions that depend on the current context. It is possible to use the degree of satisfaction of a rule to define a preference relation on answer sets. [78] defines a rule as having *degree 1* under the following condition: when A is an answer set of P, then A satisfies all rules of P.
For example, let us plan a vacation: Normally you like to go to Mallorca but you prefer to go to Stockholm (denoted by the preference relation $\prec$). Unless it is hot, people usually prefer Stockholm to Mallorca. If it is hot, Mallorca is preferred to Stockholm. In summer

it is normally hot but there are exceptions. If it is winter, then Mallorca is no longer considered [78].

$$\begin{array}{rcll} Stockholm \prec Mallorca & \leftarrow & \neg hot & \text{(rule 1)} \\ Mallorca \prec Stockholm & \leftarrow & hot & \text{(rule 2)} \\ hot & \leftarrow & NOT \neg hot, summer & \text{(rule 3)} \\ \neg Mallorca & \leftarrow & rain & \text{(rule 4)} \end{array}$$

Without further information about the weather we obtain the single preferred answer set $A_1 = \{Stockholm\}$. There is no information suggesting that it might be *hot*, so rule 1 will determine preferences. A_1 satisfies all rules to degree 1. Now, if we add a new fact *summer*, then the new answer set is $\{summer, hot, Mallorca\}$. If we add the literal *hot*, then the new answer set is $\{summer, \neg hot, Stockholm\}$. Finally, if we add the facts *summer* and *rain*, the single answer set is $\{summer, rain, hot, \neg Mallorca, Stockholm\}$. We see that it is not possible to satisfy all rules to degree 1. As in real life, there are situations where the best options simply do not work out. Therefore, LPODs are well suited for representing problems where a certain choice has to be made. In general, using ASP we can optimize the solution we want to generate, we can improve the rules and define the constraints we are using to get the maximum optimization of the desired answer sets (solutions) [78].

3.3.2 Guess and check programs in ASP

Answer Set Programming (ASP) is widely used to express properties in NP, i.e.properties whose verification can be done in polynomial time, where answer sets of normal logic programs can be generated through solutions and polynomial time proofs of such properties. The solution of such problems can be carried out in two steps:

1. Generate a candidate solution through a logic program.
2. Check the solution by another logic program [79].

However, it is often not clear how to combine Π_{guess} and Π_{check} into a single program Π_{solve} that solves the overall problem. If we simply take the union $\Pi_{guess} \vee \Pi_{solve}$, it does not work and we have to rewrite the program.

Theoretical results prove that for problems with Σ_2^P complexity, it is required that Π_{check} is rewritten into a disjunctive logic program $\grave{\Pi}_{check}$ so that the answer sets of $\Pi_{solve} = \Pi_{guess} \vee \grave{\Pi}_{check}$ yield to the solutions of the problem, where $\grave{\Pi}_{check}$ emulates the inconsistency check for $\grave{\Pi}_{check}$ as a minimal model check, which is co-NP-complete for disjunctive programs. This becomes even more complicated by the fact that $\grave{\Pi}_{check}$ must not solely rely on the use of negation, since it is essentially determined by the Π_{guess} part. These difficulties can make rewriting Π_{check} to $\grave{\Pi}_{check}$ a formidable and challenging task [79].

As an example, if we are talking about planning the problem to find a sequence of actions, which takes the system from an initial state p_0 to a state p_n, in which the states are changing over time. Conformant planning looks for a plan L that works under all contingency cases that may be caused by incomplete information about the initial state and/or nondeterministic actions or effects which are Σ_2^P under certain restrictions [79].

Let's consider the problem of the "fire alarm"; an alarm is raised that there is a fire in a building that is supported through a fire alarm system. Possible actions (states) of the system turn off the electricity and then pump in water. Just turning off the electricity does not extinguish the fire, only additionally pumping in water guarantees that it is really extinguished. Using the following guess and check programs $fire_{guess}$ and $fire_{check}$ respectively, we can compute a plan for extinguishing the fire through two actions, $fire_{guess}$ and $fire_{check}$, the program $fire_{guess}$ guesses all candidate plans $P = p_1, p_2, ..., p_n$ using time points for action execution,

```
  fire_guess:
  % Timestamps:
time(0).
time(1).
  % Guess a plan:
  turn_off(T) v -pump(T) :- time(T).
pump(T) v -pump(T) :- time(T).
% Forbid concurrent actions:
:- pump(T), turn_off(T).
```

While $fire_{check}$ checks whether any such plan P is conformant for the goal $g = \neg extinguished(2)$ The final constraint eliminates a plan execution if it reaches the goal; thus, $fire_{check}$ has no answer set if the plan P is conformant.

```
  fire_check:
  % Initial state:
fired(0) v -fired(0).
  % Frame Axioms:
fired(T1) :- fired(T),
time(T),
not -fired(T1),
                T1 = T + 1.
  turned_off(T1) :- turn_off(T),
                    T1 = T + 1.
  % Effect of turning off:
  turned_off(T1) :- turn_off(T),
                    T1 = T + 1.
fired(T1) v -fired(T1) :- turn_off(T),
fired(T),
                              T1 = T + 1.
  % Effect of pumping:
  -fired(T1) :- pump(T),
                turn_off(T),
                T1 = T + 1.
  % Check goal in stage 2 (constraint):
:- not fired(2).
```

The program $fire_{guess}$ generates the answer set:

$$S = time(0), time(1), turn_{off}(0), pump(1)$$

which corresponds to the (single) conformant plan $\{P = turn_{off}, pump\}$ for goal not $fired(2)$. Using the method $fire_{guess}$ and $fire_{check}$ can be integrated automatically into a single program $fire_{solve} = fire_{guess} \vee \grave{f}ire_{check}$ It has a single answer set, corresponding to the single conformant plan $P = \{turn_{off}, pump\}$ as desired.

With these examples in mind, we now turn to the problem of diagnosing, such ontologies. What should have become evident by now is that spotting an error in a large-scale program is a challenging task. We deliver a solution that is flexible and can be implemented with widely standard components. In particular, our proposal does not require substantial

changes to an existing diagnostic engine, so it can be seen as an "add-on" or refinement of a debugging system. In ASP, time is usually represented as a variable in which its values are defined by an extensional predicate with a finite domain. Dealing with finite temporal intervals can be used to reason complex events in surveillance systems.

3.3.3 Strengths and limitations of ASP in comparison to traditional approaches

In this section, we address the advantages of ASP and we compare it with other existing paradigms, e.g. SAT, prolog and constraint programming. Answer Set Programming offers the following useful features [80]:

Classical Negation: ASP offers the classical negation that can be implemented via integrity constraints in which its effect is to eliminate any answer set candidate containing complementary atoms.

Built-In Arithmetic Functions: ASP supports a number of arithmetic functions that are evaluated during grounding. The following symbols are used for these functions: addition, subtraction, multiplication, integer division, modul function, exponentiation, absolute value, bit-wise AND, bit-wise OR, bit-wise exclusive OR, and bit-wise complement.

Built-In Comparison Predicates: ASP supports a number of built-in predicates permit term comparisons within the bodies of rules, e.g. equal, not equal, less than, less than or equal, greater than, greater than or equal.

Assignments: The built-in predicates := and = can be used respectively in the body of a rule to unify a term on it's right-hand side to a (non-ground) term or variable on its left-hand side. (respectively.)

Intervals: ASP supports integer intervals in the form $i..j$, where i and j are integers.

Conditions: Conditions allow for instantiating variables to collections of terms within a single rule. This is particularly useful for encoding conjunctions or disjunctions over arbitrarily many ground atoms as well as for the compact representation of aggregates.

Aggregates: An aggregate is an operation on a multi-set of weighted literals that evaluates to some value. In combination with comparisons, we can extract a truth value from an aggregate's evaluation; thus, obtaining an aggregate atom.

Pooling: ASP allows pooling alternative terms to be used as argument within an atom; thus, specifying rules more compactly.

Optimization: Optimization statements extend the basic question of whether a set of atoms is an answer set or whether it is an optimal answer set. Optimization in ASP is indicated via maximization and minimization.

Constraints: Constraints play an important role in Answer Set Programming because adding a constraint to a logic program P affects the collection of stable models of P in a very simple way; it eliminates the stable models that violate the constraint.

Modularization: It is a way of structuring and easing the program development process. Modular ASP programs consist of modules that are combined through suitable interfaces. This way, parts of a program can be developed and verified independently and they can be more easily reused.

ASP and Prolog: In prolog, to be an effective Prolog programmer one needs to understand how to terms as data structures, which is quite difficult. In SP, the order of rules is not important. However, the order of rules and subgoals in rule bodies in a Prolog program matter. Changing these may cause a working program to become useless. These features give a programmer control over the execution of search and give Prolog a programming language, a formalism in which one can implement algorithms. In this sense, Prolog misses true declaratively. ASP is "more declarative:" It is intuitive, requires less background in logic and its semantics are robust to changes in the order of literals in rules and rules in programs [81].

ASP and Constraint Programming: Mapping the high-level specification of a problem into constraints that will lend themselves well to processing, also requires certain mathematical background and expertise in constraint modeling and solving.

On the other hand, the language of ASP and its extensions were developed with knowledge representation applications in mind and their constructs were designed to capture patterns of natural language statements, definitions, and default negation. The language is simple and intuitive to use.

The major disadvantage of ASP is the computation time which is required in some complex NP hard problems. However, in the frame of reasoning in surveillance systems it reasons in real-time. Clearly, reasoning about events is less complex compared to standard heuristic search problems [81].

3.4 Summary

The development of video surveillance systems has different challenges to researchers. The classification of complex human behavior in a spatio-temporal context is a major challenge where the combination between time, spatial and uncertainty factors has to be considered. However, the major focus nowadays in this research area is context awareness where data concerning other activities/events taking place at approximately the same time or in approximately the same place; requires actions/scenarios to be considered as a group rather than in isolation. The Chapter has presented the main general approaches in how to create spatio-temporal event detection based on knowledge representation and reasoning. The Chapter has illustrated comprehensively the meaning of the terms, spatio-temporal and knowledge and reasoning, through providing several examples. Following this, the answers to the important questions "Why knowledge representation?" and "Why reasoning?" have been discussed. Knowledge representation is used to symbolize the description of a complex system. Reasoning where it is used describes the rules that already contain the solution to the problem. Spatio-temporal reasoning is an appropriate solution to the problem of classifying complex human behavior. This problem has been fundamental to the field of artificial intelligence since its beginning and despite massive efforts for more than sixty years has not been solved in general. Furthermore, the Chapter covered the major requirements of spatio-temporal reasoning under different cases (short/long-term, real-time and under uncertainty). Those requirements are specified based on the experience and the researchers related works from the state-of-the-art. Finally, a detailed explanation of the current proposed approaches has been discussed.

Generally, sub-symbolic approaches have their shortfalls, one of which is the lack of combination meaning with sequences of events. It might be simple enough to recognize someone in a vandalism sequence, but this does not mean that the system knows much about vandalism. Therefore, it would not be able to conclude that adults who are under the effect of alcohol or drugs behave the same as playing children.

Table 3.2: The limitations of context meddling approaches

Resource	Methods	Description
[49]	Key-Value Models	They have a weak support of temporal constraints
[50]	Markup Scheme Models	They have a weak support of temporal constraints
[47] [34] [23]	Ontology Based Models	The main problem with this approach is that reasoning in OWL-DL is already an expensive computation.
[63] [53]	Graphical Models	It has a "flat" information model, in that all context types are uniformly represented as atomic facts. It also emphasises only the development of context models for particular applications or application domains.
[54] [55]	Logic-Based Models	In constraint programming, for example mapping the high-level specification of a problem into constraints that will lend themselves well to processing also requires certain mathematical background, and expertise in constraint modeling and solving. In prolog, to be an effective Prolog programmer one needs to understand how to use terms as data structures which is quite difficult.
[56]	Hybrid approaches	Despite solving some challenges but hybrid approaches still share the limitations of the combined paradigms.

Table 3.3: The criteria derived from the survey of approaches to context reasoning under uncertainty

Approach	Real-time	Embedded plat.	Temporal reas.	Uncertainty
Parametric	+	+	++	++
Graphical Models	+	+	++	+
Syntactic	++	+	+	-
Knowledge based	+	+	+	+
Volumetric	+	++	+	+

Table 3.4: The limitations the previous concepts for spatio-temporal reasoning

Resource	Methods	Description
[34] [23]	Ontologies, e.g. Semantic Web	Ontologies based on Semantic Web provide concise high-level definitions of activities but they do not necessarily suggest the right hardware to parse the ontologies for recognition tasks (Semantic Web).
[35] [23]	Petri Nets	Petri nets are an intuitive tool for expressing complex activities; they suffer from the disadvantage of having to describe manually the model structure.
[36] [23]	Bayesian Networks	The evidence of one cause reduces the possibility of another cause given the evidence of their low prior probability, which is especially difficult to model in logical rule-based systems. Nevertheless, a fundamental limitation of using a Bayesian network for knowledge representation is that it cannot represent the structural and relational information. Also, the applicability of a Bayesian network is largely limited to the situation that is encoded in advance, using a set of fixed variables.
[73]	Hidden Markov Model	It does not detect human behavior perfectly because human behavior is not a markovian behavior.
[55]	Chronicle Recognition System (CRS)	The language includes predicates for persistence and event absence. However, the CRS language does not allow mathematical operators in the constraints of the temporal variables. Consequently, CRS cannot be directly used for activity recognition in video surveillance applications. Logic programming approaches do not explicitly address the problem of uncertainty in the observation input stream.
[72] [23]	Support Vector Machine	It does not model the global geometry of local parts directly, instead considering them as a bag of features.

Table 3.5: The limitations the previous concepts for spatio-temporal reasoning

Resource	Methods	Description
[31]	Context Free Grammars	Because deterministic grammars expect perfect accuracy in the lower levels, they are not suited to deal with errors in low level tasks [31]. Context Free Grammars expect perfect accuracy in the lower levels; they are not suited to deal with errors in low level tasks.
[74]	Event Tree	There is a perfect global synchronous clock which is unsuitable for no-centralized management and distributed systems of clock drift and loose coupling. Due to the lack of consideration of unpredictable delay, it cannot make breaking and mobile detection in a mobile database efficiently.
[75]	Diagram Detection Method	It only provides the simple time model, in which every event is regarded as a certain time point. Atomic events are based on definitions, while complex events are based on semantic.
[72] [23]	Automata	It does not model the global geometry of local parts directly, instead considering them as a bag of features.

Chapter 4

Complex event detection under uncertainty

Uncertainty means the state of having limited knowledge where it is impossible to describe the existing state or to predict the possible outcome exactly. Logical statements are usually precise about the world in many different forms. They are useful for capturing knowledge and applying it. Sometimes it is not possible to express a general statement with the totality of a logical universal. There are cases where it might be that a fact or a belief is not certain. For example, all cows are black and white. This is not always true as some cows are totally black and others are totally white. These cases show that in many situations it may be difficult to gauge something precisely or categorically.
Furthermore, to the intrinsic imperfection of the previous statements the way that we generate conclusions from data may also be imprecise [42].

In this Chapter, the design of an ontology design under uncertainty will be explored. The key for ontology research is to determine the mapping of representational information quality into the ambiguity and fuzziness fields. We will move from sound mathematical approaches in information theory to translation uncertainty in ontology [82].

Generally, there are different types of uncertainty. The first type is uncertainty in prior knowledge, e.g. some causes of an event are unknown and are not represented in the knowledge base of the video surveillance system. Another type is uncertainty in the model, e.g. models could be affected by noise and the noise is possibly represented in the model. Therefore, the model has a margin of error where the decision is not always true. Finally, uncertainty in perception, e.g. sensors do not return exact or complete information about the world, a system never knows its position exactly. Now, the major question is how to deal with uncertainty? The answer consists of two main approaches, the implicit approach and the explicit approach. In the implicit approach, we can deal with uncertainty by building procedures that are robust to uncertainty. The explicit approach deals with uncertainty by building a model of the world that describes uncertainty about its state, dynamics and observations. Then it reasons about the effect of actions given the model.

Usually, it is possible to reason uncertainty using three types of uncertainty: default reasoning, worst-case reasoning and probabilistic reasoning, Figure 4.1 shows the origins of uncertainty in surveillance systems. When using default reasoning we assume that the world is fairly normal. Abnormalities are rare. Consequently, an agent assumes normality

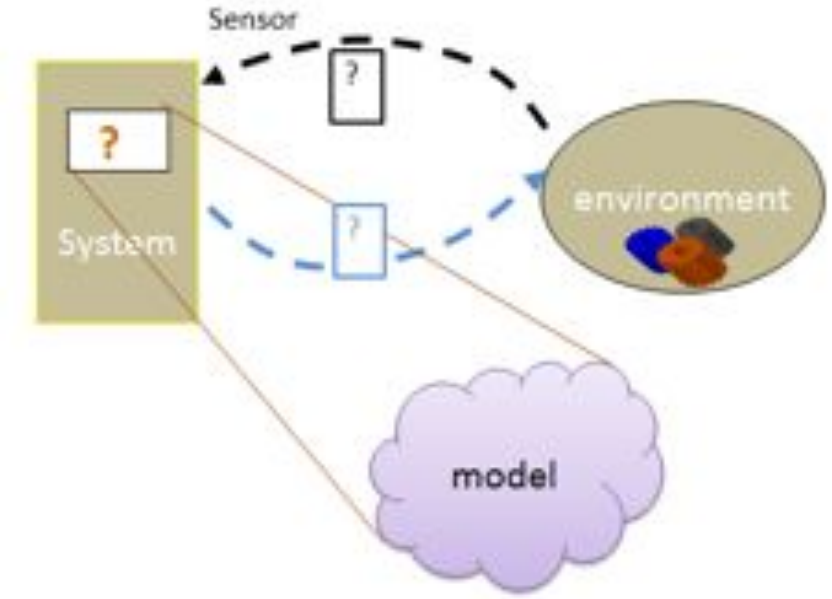

Figure 4.1: The origins of uncertainty in surveillance systems.

until there is evidence of the contrary. For example, if an agent sees a bird x, it assumes that x can fly, unless it has evidence that x is a penguin, an ostrich, a dead bird or a bird with broken wings.

Worst-case reasoning is the exact opposite of default reasoning. The world is ruled by Murphy's Law which means that uncertainty is defined by sets, e.g. the set possible outcomes of an action or the set of possible positions of an object. The surveillance system assumes the worst case and chooses the actions that maximize an utility function in this case.

In probabilistic reasoning, we assume that the world is not divided between "normal" and "abnormal", nor it is adversarial. Possible situations have various likelihoods (probabilities). The agent has probabilistic beliefs, pieces of knowledge with associated probabilities "strengths" and chooses its actions to maximize the expected value of some utility function.

For event detection in video based surveillance systems human behavior combines both spatial and temporal resolutions in nature. This means that context becomes all important.
Therefore, the design of an ontology, which we have discussed in the previous Chapter, has to satisfy the following properties [84]:

1. The model has to be able to capture long-range dependencies among observations at different spatial and temporal resolutions;

2. The model has to be probabilistic and should be learnable from the given training samples;

3. The model has to be able to detect events in real-time inference when the desired events occur.

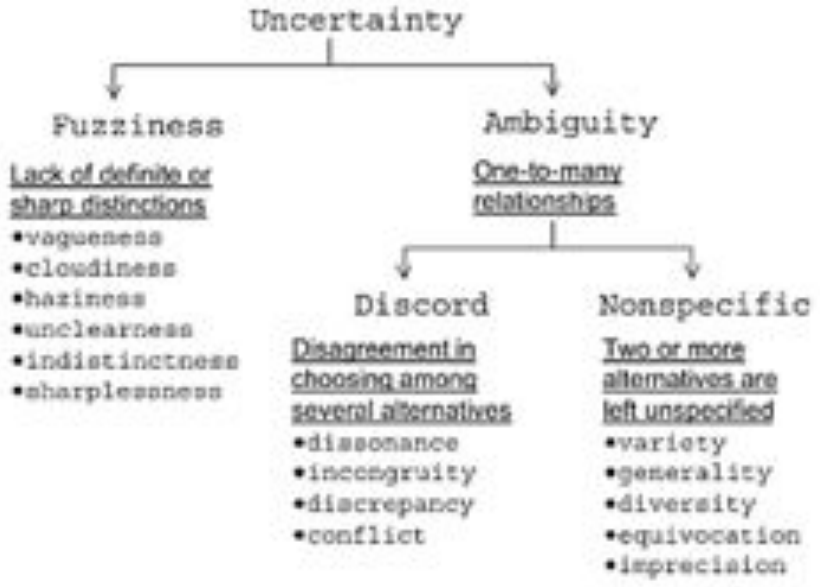

Figure 4.2: The major types of uncertainty [83].

4.1 Taxonomy of events: atomic, simple and complex events

Autonomous event detection in audio and video surveillance systems has been an important point of research in the last 15 years. Thus, different types of events have been detected and therefore different approaches have been used. Usually, researchers had started to detect simple or atomic events and then through the combination of the simple events they were able to detect complex events. An atomic event is a simple event that can be represented by an action of an object (.animal, person, vehicle, etc.) In a video system or a special sound in an audio system, such events can be:

- Object passing by a specified area.
- Object stopping in a specified area.
- Object sound in a specified frequency range.

Compared to an atomic event a composite or complex event can be defined as combination of atomic events. This can be:

- Object passing by a specified area and stopping in the next specified area.
- Object stopping in a specified area and a sound in a specified frequency range is recorded simultaneously.

For anomaly detection no former knowledge about the event that should be detected is necessary. The event models, which are probabilistic, are built autonomously and unsupervised. Thus, the system is able to detect the most frequent patterns which take place

in the scene. For example, the detection based on probability is done by considering the events with the probability of occurrence less than a predefined threshold. An appropriate example could be the clustering of trajectories of objects moving in a certain area; the clusters represent the normal state and trajectories outside the anomaly event [1].

Furthermore, the term incident is used in literature for detection of different events. Incident detection is mainly used in traffic scenarios. An incident can be defined as [2]:

"any non-recurring event that causes a reduction of roadway capacity or an abnormal increase in demand. Such events include traffic accidents, disabled vehicles, spilled cargo, highway maintenance and reconstruction projects and special non-emergency events (e.g., ball games, concerts, or any other event that significantly affects roadway operations)."

In other words, an incident can be an atomic event such as a traffic accident or a complex event, which could be, for example, a scene where first a traffic accident has been detected and following this there is spilled cargo and fire around the objects of interest.

Sequentially, the definitions of atomic events, anomalies and incidents have no sharp between each other. Therefore, anomalies and incidents can also be called simple events if they are combined complex events.

Simple events and complex events can be detected by inference (reasoning) process. Here, inference means the process of deriving logical conclusions from premises known or assumed to be true [1].

4.1.1 Taxonomies of uncertainty

A well-accepted typology of uncertainty is the one proposed by Klir and Yuan [85], Figure 4.2 shows the different types of uncertainty. Information can be uncertain due to many different reasons. It can be inaccurately measured, it can change over time, its source can be unreliable or unconfident, it can have ambiguous meanings etc. Parsons, the author of [86] found different taxonomies of uncertain information and found out some common terms:

- Ignorance: Ignorance means that there is an object in the environment of the surveillance system that is just not known. For the reasoning process of a surveillance system, this means that the content of the knowledge base may not have the required details which are necessary for the decision process.

- Incompleteness: Ignorance is in contrast to incompleteness. Incomplete information means that there is no hypothesis related to an object or attribute value at all, e.g. the object type is known but the speed of the object is unknown.

- Inaccuracy: While uncertainty is concerned with the measure of trust that is put into the data provided by a sensing system, inaccuracy deals with the potential measurement errors that may occur.

- Inconsistency: Inconsistency means that there are conflicting hypotheses about an object data, e.g. two sensors are giving different object types with a high belief.

[1] http://www.thefreedictionary.com/inference

4.1.2 Origins of uncertainty in knowledge based systems

Vagueness or ambiguity is sometimes described as "second order uncertainty," where uncertainty is even about the definitions of uncertain states or outcomes. Here, the difference is that this uncertainty is about the human definitions and concepts, not an objective fact of nature. It has been argued that ambiguity, however, is always avoidable while uncertainty (of the "first order" kind) is not necessarily avoidable. Uncertainty may purely be the consequence of a lack of knowledge of obtainable facts. You may be uncertain about whether a new rocket design will work, but this uncertainty can be removed with further analysis and experimentation. However, at a subatomic level uncertainty may be a fundamental and unavoidable property of the universe [87]. Most tasks requiring intelligent behavior have some degree of uncertainty associated with them.

The type of uncertainty that can occur in knowledge-based systems may be caused by problems with the data. For example:

- Data might be missing or unavailable.
- Data might be present but unreliable or ambiguous due to measurement errors.
- The representation of the data may be imprecise or inconsistent.
- Data may just be a user's best guess.
- Data may be based on defaults and the defaults may have exceptions.

The uncertainty may also be caused by the represented knowledge since it might,

- represent best guesses of the experts that are based on plausible or statistical associations they have observed.
- not be appropriate in all situations, e.g. may have indeterminate applicability.

Given these numerous sources of errors most knowledge-based systems require the incorporation of some form of uncertainty management. When implementing some uncertainty scheme we must be concerned with three issues:

- How to represent uncertain data?
- How to combine two or more pieces of uncertain data?
- How to draw inference using uncertain data?

4.2 Methodological approaches of reasoning under uncertainty

We will introduce three ways of handling uncertainty: the explanation of Bayes' theorem, Dempster-Shafer theory and certainty factor. This section is written with the help of the documents offered by the department of computer science at The University of Illinois, Chicago[2]:

[2]http://www.cs.uic.edu/ liub/teach/cs511-spring-06/cs511-uncertainty.doc

4.2.1 Bayes' Theorem

Conditional probability is defined as:

$$P(H|E) = \frac{P(H \cap E)}{P(E)} \quad for \quad P(E) \neq 0.$$

Furthermore, we have

$$P(E|H) = \frac{P(E \cap H)}{P(H)} \quad for \quad P(H) \neq 0.$$

In real life situations, the probability $P(H|E)$ cannot always be calculated. Bayes Theorem provides a rule for computing the conditional probability $P(H|E)$ from the probabilities $P(E), P(H)$ and $P(E|H)$.

From conditional probability:

$$P(E|H)P(H) = P(H|E)P(E) = P(H \cap E)$$

Thus,

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}.$$

Rule-based systems express knowledge in an IF-THEN format:

IF X is true, THEN Y can be concluded with probability p.

If we observe that X is true, then we can conclude that Y exists with the specified probability. For example, IF the sky is cloudy, THEN it will rain (0.75).

However, what if we reason abductively and observe Y, i.e. it will rain, while knowing nothing about X, i.e. the sky is cloudy? What can we conclude about this? Bayes' theorem describes how we can derive a probability for X. Within the rule given above, Y denotes some piece of evidence (typically referred to E) and X denotes some hypothesis (H) given:

$$P(H|E) = \frac{P(H|E)}{P(E)}. \quad or \quad P(H|E) = \frac{P(H|E)}{P(E|H)P(H) + P(E|H')P(H')}$$

Consider whether Rob has a cold (the hypothesis) given that he is sneezing (the evidence). The probability of his sneezing is the sum of the conditional probability that he sneezes when he has a cold and the conditional probability that he sneezes when he does not have a cold. In other words, the probability that he sneezes regardless of whether he has a cold or not.

4.2.2 Certainty Factors

Certainty factor is another method of dealing with uncertainty. One of the difficulties with Bayesian method is that there are too many probabilities required. Most of them could be unknown. The problem gets worse when there are many pieces of evidence. Besides the problem of amassing all the conditional probabilities for the Bayesian method, is another major problem that appeared with surveillance systems; the relationship of belief and disbelief. At first sight, this may appear trivial since obviously disbelief is simply the opposite of belief. In fact, the theory of probability states that $P(H) + P(H') = 1$ and so $P(H) = 1 - P(H')$. For the case of a posterior hypothesis that relies on evidence, C:

$$P(H|C) = 1 - P(H'|C) \tag{4.1}$$

The researchers developed a MYCIN model based on certainty factors [88]. This is a heuristic model of uncertain knowledge. In MYCIN two probabilistic functions are used to model the degree of belief and the degree of disbelief in a hypothesis. The function to measure the degree of belief is MB and the function to measure the degree of disbelief is MD.
MYCIN represents factual information as Object-Attribute-Value (OAV) triplets. MYCIN also associates with each fact a Certainty Factor (CF) which represents a degree of belief in the fact.

- -1 means the fact is false.
- 0 means no information is known about the fact.
- 1 means the fact is known to be true.

MYCIN combines two identical OAV triplets into a single OAV triplet with a combined uncertainty, computed as:

$$Uncertainty = (CF1 + CF2) - (CF1 * CF2)$$

For a logical rule the calculation of uncertainty is described as follows:

$$\begin{aligned} &CF(P1 \ \ or \ \ P2) = max(CF(P1), CF(P2)) \\ &CF(P1 \ \ and \ \ P2) = min(CF(P1), CF(P2)) \\ &CF(not \ \ p) = -CF(P) \end{aligned}$$

The single method handles only the cases where both certainty factors are positive. By additional methods the other cases of certainty can be handled.

$$\begin{aligned} &Uncertainty = (CF1 + CF2) - (CF1 * CF2) \\ &if \ \ CF1 \geq 0 \ \ and \ \ CF2 \geq 0 \\ &New \ \ Uncertainty = \frac{CF1 + CF2}{1 - min(CF1, CF2)} \\ &if \ \ -1 < CF1 * CF2 < 0 \end{aligned}$$

4.2.3 Dempster-Shafer Theory

Here, we discuss another method for handling uncertainty. It is called Dempster-Shafer theory. This appeared during the 1960s and 1970s through the efforts of Arthur Dempster and one of his students, Glenn Shafer. This theory was designed as a mathematical theory for evidence.
The development of the theory has been motivated by the observation that probability theory is not able to distinguish between uncertainty and ignorance owing to incomplete information [89]. Given a set of possible elements, called environment,

$$\theta = \{\theta_1, \theta_2, ..., \theta_n\}$$

These are mutually exclusive and exhaustive. The environment is the set of objects that are of interest to us. Each subset of θ can be interpreted as a possible answer to a question. Since the elements are mutually exclusive and the environment is exhaustive, there can only be one correct answer subset to a question. Of course, not all possible questions may be meaningful. The subsets of the environment are all possible valid answers in this universe of discourse. The term 'discern' means that it is possible to distinguish the one correct answer from all the other possible answers to a question. The power set of the environment (with 2^N subsets for a set of size N) has as its elements all the answers to the possible questions of the frame of discernment. In Bayesian theory, the posterior probability changes as evidence is acquired. The same as in Dempster-Shafer theory the belief in evidence may vary. It is customary in Dempster-Shafer theory to think about the degree of belief in evidence as analogous to the mass of a physical object. The mass of evidence supports a belief.

The reason for the analogy using an object of mass is to consider belief as a quantity that can move around, be split up and combined. A fundamental difference between Dempster-Shafer theory and probability theory is the treatment of ignorance. Probability theory must distribute an equal amount of probability even in ignorance.

For example, if you have no prior knowledge, you must assume the probability P of each atom :

$$p = \frac{1}{N}$$

where N is the number of possibilities, e.g. the formula $P(H) + P(H') = 1$ must be enforced. The Dempster-Shafer theory does not force belief to be assigned to ignorance or refutation of a hypothesis.

The mass is assigned only to those subsets of the environment to which you wish to assign belief. Any belief that is not assigned to a specific subset is considered no belief or nonbelief and just associated with environment θ. Belief that refutes a hypothesis is disbelief, which is not nonbelief.
A mass has considerably more freedom than probabilities, as shown in Table 4.1.

The overall belief in a proposition (A) is determined by the sum of all evidence supporting the proposition:

$$Bel(A) = \sum_{B \subseteq A} m(B) \quad Where \quad B \quad is \quad the \quad evidence$$

The belief $Bel(A)$ gives a measure of the extent to which a proposition is definitely supported.

However, the remaining evidence does not have necessarily disprove the proposition. A second measure, the plausibility $Pl(A)$ of a proposition is determined by that which indicates the extent to which the given evidence fails to refute a proposition:

$$Pl(A) = 1 - Bel(\overline{A})$$

Table 4.1: A mass has considerably more freedom than probabilities

Dempster-Shafer theory	Probability theory
$m(\theta)$ does not have to be 1	$\sum_i P_i = 1$
If $X \subseteq Y$, it is not necessary that $m(X) \leq m(Y)$	$P(X) \leq P(Y)$
No required relationship between $m(X)$ and $m(X')$	$P(X) + P(X') = 1$

Dempster-Shafer theory provides a function for computing from two pieces of evidence and their associated masses describing the combined influence of these pieces of evidence. This function is known as Dempster's rule of combination. Let $m1$ and $m2$ be mass assignments on θ, the frame of discernment. The combined mass is computed using the formula (special form of Dempster's rule of combination).

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X).m2(Y)}{1 - \sum_{X \cap Y = \phi} m1(X).m2(Y)}$$

Thus, $0 \leq Bel \leq Pls \leq 1$. Table 4.2 below shows some common evidential interval.

Table 4.2: Some common evidential interval

Evidential Interval	Meaning
$[1, 1]$	Completely true
$[0, 0]$	Completely false
$[0, 1]$	Completely ignorant
$[Bel, 1]$ where $0 < Bel < 1$ here	Tends to support
$[0, Pls]$ where $0 < Pls < 1$ here	Tends to refute
$[Bel, Pls]$ where $0 < Bel \leq PIs < 1$ here	Tends to both support and refute

4.2.4 Fuzzy Theory

Fuzzy logic [90] handles the problem of representing vagueness of concepts. The concept of fuzzy logic can be explained with an example. Let's talk about the speed of people. In this case the set S (the universe of discourse) is the set of different speeds. There are values in S that are not high speed and there are values that are in the borders between middle and high speed. To each speed in the universe of discourse, we have to assign a degree of membership in the fuzzy subset *high speed*. The easiest way to do this

is with a membership function based on the membership degree to S from the interval $[0, 1]$. The membership function S of a fuzzy set is formally defined as: $\mu_H : S \rightarrow \ [0, 1]$. The numerical scale for membership from the interval $[0, 1]$ allows the representation of gradation of membership. The membership functions are usually context-dependent and can be freely chosen as desired. With respect to the previous example, the speed's values that are members of the "high speed" set when applied to an indoor environment may not be members of that set in another indoor or even outdoor environment.

4.2.5 Hidden Markov Models

In HMM, the transition matrix consists of transition probabilities between the hidden states. In the training phase the transition matrix should be obtained between the hidden states as well as the confusion matrix between the observation and hidden states. Hidden Markov model is a dynamic statistical model consisting of [91]:

- Hidden set of states: $S = s_1, s_2, s_3, ..., s_n$
- Observed set of states: $O = o_1, o_2, o_3, ..., o_n$

HMM has a hidden sequence which generates an observed sequence.

The goal is to predict the next hidden states depending on the current hidden states and the next observed states. The following probabilities have to be specified in HMM:

- State Transition Matrix:

$$A = a_{ij} : a_{ij} = P(s_{j,t}|s_{i,t-1}) \tag{4.2}$$

- Confusion Matrix:

$$B = b_{ij} : b_{ij} = P(o_{j,t}|s_{i,t}) \tag{4.3}$$

A training phase and a test phase are required in HMM. The training phase usually works with the Baum-Welch algorithm to estimate the parameters (π (prior), A (transition matrix), B (confusion matrix)) for the HMM. This method is based on the maximum likelihood criterion [91].

The Baum-Welch algorithm is a particular case of a generalized expectation-maximization algorithm. It can compute maximum likelihood which estimates and posterior mode estimates the parameters (transition and emission probabilities) of an HMM, when given only emissions as training data.

4.3 Judgement criteria and limitations of event detection under uncertainty

Uncertainty in logic programming can be classified in different dimensions, e.g. reasoning about the truth, falsity and incompleteness of knowledge.

A stable model of the program P is defined as any set of atoms S such that S is a minimal model of the program P^S [92].

- The expression of uncertainty is possible based on stable model semantics as well as on disjunctions in the head of programs.

- Uncertainty in probabilistic logic programming can be expressed based on models that represent the semantic.When there is a program consisting of a set of logical clauses, then the probability distributions represent each model that satisfies the clauses in the program.

- Stochastic logic programs help to develop a grammar for a natural language. In statistical concepts it is possible to use stochastic grammar where production rules have associated weight parameters that contribute to a probability distribution. Using those weight parameters, the weight parameter p is learned from a given training set of example sentences.

In recent literature scientists differentiate between various types of uncertainty, e.g. subjective uncertainty, objective uncertainty, epistemic uncertainty, and ontological uncertainty [92].
Uncertainty in logic programming can be classified in different dimensions, e.g. reasoning about the truth, falsity and incompleteness of knowledge.

A stable model of the program P is defined as any set of atoms S such that S is a minimal model of the program P^S [92].

In [93], they use Fuzzy Event Detection (FED) where the FED is less sensitive to uncertainty sources. Their fuzzy model can be applied in distributed detection for a clustered network, where event notifications are aggregated in cluster heads.

Other researchers proposed the previous requirements (functionality and performance) using different methods. They increased the performance of their systems by considering the low computation time algorithms and ontologies that are needed to run the system on an embedded platform (SOC) [94].
The problem they faced was related to the task management, tracking and data processing. Despite using ontologies that provided concise high-level definitions of activities, ontologies tools did not necessarily suggest the right "hardware" to parse the ontologies for recognition tasks[23] [34].
For the achievement of high detection rate some researchers used Monte Carlo simulations [84] that have an expensive computational time. Others used Bayesian networks where the prior knowledge is very important but any lack in the prior knowledge can affect the whole inference and reasoning process [95]. Also, hidden Markov Model does not detect human behavior perfectly because the human behavior is not a Markovian behavior [73] [96] [97].

Furthermore, to solve the robustness and reliability regarding the design requirements, they use Stochastic Context Free Grammars (SCFGs) [98]. While SCFGs are more robust than Context Free Grammars (CFGs) to errors and missed detections in the input stream, they share many of the temporal relation modeling limitations of CFGs.

Context reasoning approaches should respect the following 5 criteria which has been derived from the survey of 'Approaches to Reasoning Under Uncertainty in Context Modeling' (see Table 3.3):

Table 4.3: The limitations of event detection under uncertainty approaches

Resource	Methods	Description
[93]	Fuzzy Event Detection	The FED is less sensitive to uncertainty sources. Their fuzzy model can be applied in distributed detection for clustered network, where event notifications are aggregated in cluster heads.
[94] [23] [34]	Ontologies	The problem is related to the task management, tracking and data processing. Despite using ontologies that provide concise high-level definitions of activities, ontologies tools do not necessarily suggest the right "hardware" to parse the ontologies for recognition tasks.
[95]	Bayesian Networks	The prior knowledge is very important but any lack in the prior knowledge can affect the whole inference and reasoning process.
[84]	Monte Carlo Simulations	They have an expensive computational time.
[98]	Stochastic Context Free Grammars	While SCFGs are more robust than Context Free Grammars (CFGs) to errors and missed detections in the input stream, they share many of the temporal relation modeling limitations of CFGs.

4.4 Summary

In this Chapter, a comprehensive discussion of uncertainty and its occurrence in video surveillance systems has been discussed. The relationship between uncertainty and probability, different taxonomies of uncertainty and its origins has been explained, e.g. ignorance which means that there is an object in the environment of the surveillance system that is just not known; incompleteness which is in contrast to ignorance; inaccuracy which deals with the potential measurement errors that may occur; inconsistency which means that there are conflicting hypotheses about an object data. We addressed different methodological approaches to handle uncertainty. e.g. Bayes theorem, certainty factors, Depster-Shafer theory and fuzzy theory.

Chapter 5

Novel complex event detection approaches

This Chapter consists of 2 main approaches. The first one considers the use of uncertainty to detect complex events in surveillance systems based on Hidden Markov Model (HMM) and Answer Set Programming (ASP). It combines the HMM and logic programming (ASP) to design a complex event detection system, where the performance is highly increased because of the consideration of uncertainty. The concept of the proposed case study is using HMM to predict the location of the object that is moving in front of multiple cameras. The output of a HMM model will be used in the knowledge base of ASP. However, every attribute of low level features, e.g. object type, object location, object speed, etc., has a single quality assessment for the whole event recognition process. The quality of the low level features is presented as a success rate of recognition. Bad quality constraints would negatively influence the quality statement for the whole decision, even if the information about the remaining constraints is reliable. Consequently, instead of degrading the final decision based on a single quality value, a weight of importance is assigned to every attribute. Using the rules of ASP simple events (run, walk, stop, position and direction) will be detected. Finally, the combination of simple events will be used to detect complex events.
We show that the use of ASP can significantly reduce the effort needed to detect complex events, while obtaining the same level of quality in the detected events. ASP is expressive, convenient and supports formal declarative semantics. Thus, ASP can be used to detect a large number of simple and complex events within a reasonable time frame that allows real-time operation with respect to limited hardware resources.

The system can use dlvhex[1]. dlvhex is the name of a prototype application for computing the models of so-called HEX-programs, which are developed in TU-Wien. The goal of dlvhex is to extend ASP towards an interface of Description Logics, which are the theoretical foundations of ontology languages like Ontology Web Language (OWL). For example the OWL context model which is designed in sections 6.2 can be combined with dlvhex.

The second approach is a computer vision based algorithm to detect people and analyze their position in space, especially recognizing people who are lying on the floor. Event detection on embedded platforms requires a model-free and a computationally inexpensive

[1]http://www.kr.tuwien.ac.at/research/systems/dlvhex/

approach in order to have an easy and small solution, which allows the integration of an FPGA-based smart camera without the need of a bigger Field Programmable Gate Array (FPGA). The solution is based on a foreground-background-segmentation using Gaussian Mixture Models (GMMs) to first detect people and then analyze their main and ideal orientation using moments. This allows one to decide whether a person is staying still or lying on the floor. The system has a low latency and a detection rate of 88%.

5.1 Complex event detection under uncertainty based on HMM and ASP

In this section, we consider a scenario to detect the flow of moving people. The task is to observe the flow of people in 3 different areas. Each area is observed by 2 cameras. After the observation of a specific flow in one of those areas the Hidden Markov Model (HMM) has to predict the flow of people in the destination area (destination camera). The observation states represent the flow of people in $area_1$ (Camera1 and Camera2), $area_2$ (Camera3 and Camera4), $area_3$(Camera5 and Camera6) see Figure 5.1.
Table 5.1 shows the combination of people flow classes in the observation states. Table 5.2 shows the people flow classes in the hidden state.
We observe three different flows: high, low and middle. The prediction of the flow of people in front of the destination camera helps to detect specific events, e.g. a crowd, a group of people that are walking in different directions or an abnormal crowd of people.

Table 5.1: The combination of people flow classes in the observation states

Observation state	Area1	Area2	Area3
O1	Low	Low	High
O2	Low	Low	Middle
O3	Low	Middle	Middle
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
O9	High	High	High

Table 5.2: The people flow classes in the hidden state

Hidden state	Destination Camera
S1	Low
S2	Middle
S3	High

Figure (5.1) shows that the HMM model is used to provide the knowledge base of Answer Set Programming (ASP) with a reliable level of features regarding the people flow that is moving in front of different cameras.

Using the rules of ASP simple events (run, walk, stop, position and direction) will be detected. The combination of simple events will be used to detect complex events.

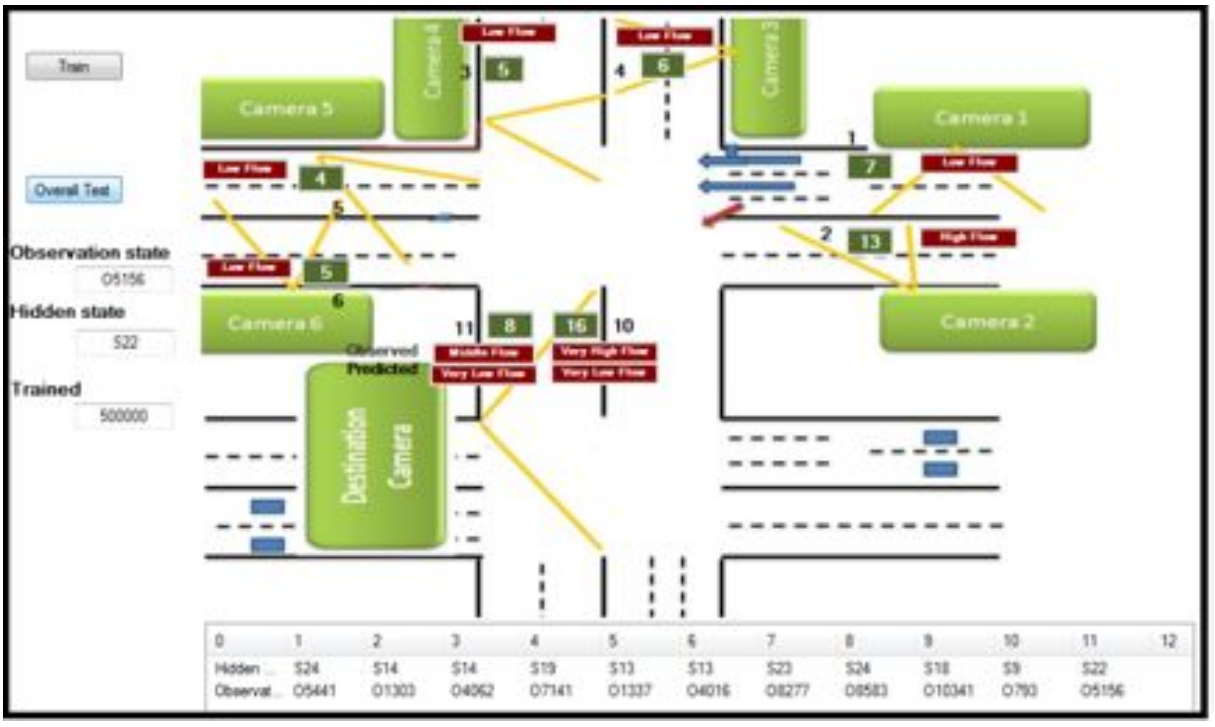

Figure 5.1: Description of the hidden and observation states by the simulation tool

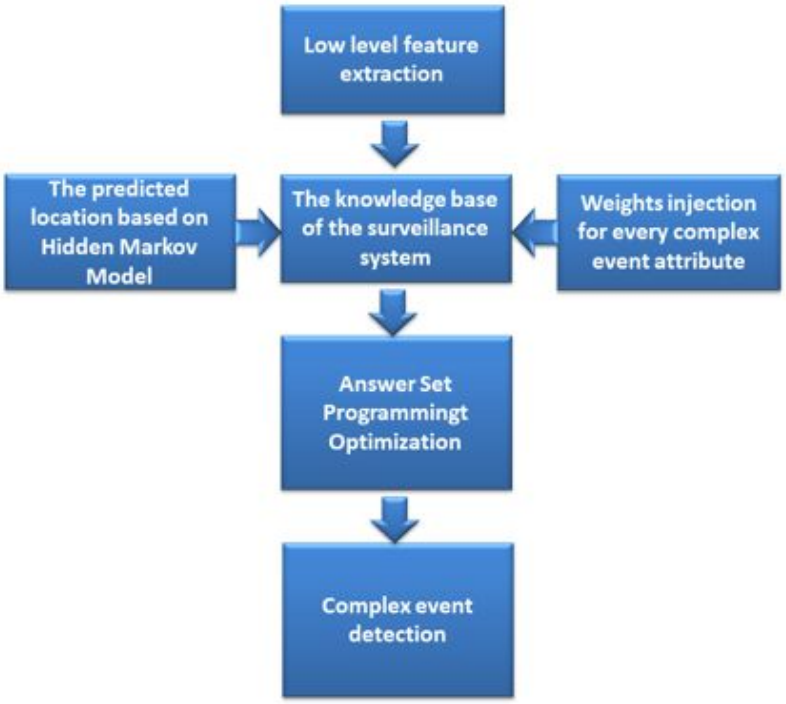

Figure 5.2: The overall architecture of the proposed complex event detection system under uncertainty

In our simulation scenario, we already have the observation and the hidden states. We just need the transition matrix and the confusion matrix. The goal is to compute the transition and the confusion probabilities.

For example, suppose we have the history data (10000 records) see Table 5.5. The first row represents the time line when we observe the state and the second row represents the hidden states sequence.

To compute the transition probabilities between the hidden states, we use the equation 1 and 2 and Table 5.5:

$$p(S_{1,t}|S_{1,t-1}) = \frac{\text{No. of recording S1 after S1}}{\text{No. of recording S1}} = \frac{2}{6} = 0.33 \tag{5.1}$$

$$p(S_{2,t}|S_{1,t-1}) = \frac{\text{No. of recording S2 after S1}}{\text{No. of recording S1}} = \frac{4}{6} = 0.66 \tag{5.2}$$

$$p(S_{2,t}|S_{2,t-1}) = \frac{\text{No. of recording S2 after S2}}{\text{No. of recording S2}} = \frac{1}{4} = 0.25 \tag{5.3}$$

$$p(S_{1,t}|S_{2,t-1}) = \frac{\text{No. of recording S1 after S2}}{\text{No. of recording S2}} = \frac{3}{4} = 0.75 \tag{5.4}$$

We use these results to build the transition matrix as in Table 5.7:

Table 5.3: The transition matrix A of the proposed example

Hidden/Hidden	S1	S2
S1	0.33	0.66
S2	0.75	0.25

$$p(o_{1,t}|S_{1,t}) = \frac{\text{No. of recording O1 and S1}}{\text{No. of recording S1}} = \frac{3}{6} = 0.5 \tag{5.5}$$

$$p(o_{2,t}|S_{1,t}) = \frac{\text{No. of recording O2 and S1}}{\text{No. of recording S1}} = \frac{3}{6} = 0.5 \tag{5.6}$$

$$p(o_{2,t}|S_{2,t}) = \frac{\text{No. of recording O1 and S2}}{\text{No. of recording S2}} = \frac{3}{4} = 0.75 \tag{5.7}$$

$$p(o_{2,t}|S_{2,t}) = \frac{\text{No. of recording O1 and S2}}{\text{No. of recording S2}} = \frac{1}{4} = 0.25 \tag{5.8}$$

In 5.4 we see the confusion matrix according to the previous results.

Table 5.4: The confusion matrix B of the proposed example

Hidden/Observation	O1	O2
S1	0.5	0.5
S2	0.75	0.25

After training the HMM, it can be used to predict the future states of the destination state after observing the current states in front of camera1 and camera2.

Table 5.5: History data of the hidden Markov model example

Time	1	2	3	4	5	6	7	8	9	10	11	12
Hidden	S1	S1	S2	S1	S2	S2	S1	S1	S2	S1	S2	
Observation	O1	O2	O2	O2	O1	O1	O1	O1	O1	O2	O1	O2

Thus, we have to compute the probabilities of all possible hidden states occurring at any given time ($t > 11$) with the transition influence of the previous hidden state $s_{h,t-1}$ and the observation state $o_{k,t}$. Such an influential relation can be represented by the function $a_t(s_{h,t-1}, o_{k,t})$ defined as follows [99]:

$$
\begin{aligned}
&a_t(s_{h,t-1}, o_{k,t}) \\
&= B[(:,[k])]^T .* A([h],:) \\
&= [(P(o_k|s_1)P(s_{1,t}|s_{h,t-1}), P(o_k|s_2)P(s_{2,t}|s_{h,t-1}), ..P(o_k|s_n)P(s_{n,t}|s_{h,t-1})]
\end{aligned}
$$

Where $A([h],:)$ is the hth rows of the state-transition matrix A and $B(:,[k])$ is the kth columns of the confusion matrix B. The symbol of the operator .* is an array multiplication, and thus, A .* B means the element-by-element vector multiplication of A and B. Using the HMM example in Figure (1) the goal is to predict the hidden state at time ($t = 12$).
In Table 5.5, we see that at time $t = 12$, the observation state is $O2$ and at time $t = 11$, we had the hidden state $S2$, using the previous equation, the probabilities of all possible hidden states occurring at time $t = 12$ is:

$$
\begin{aligned}
&a_t(s_{2,t=11}, o_{2,t=12}) = B[(:,[2])]^T .* A([2],:) \\
&= [P(s_{1,t=12}|s_{2,t=11}, o_{2,t=12}), P(s_{2,t=12}|s_{2,t=11}, o_{2,t=12})] \\
&= [B(2,1) * A(2,2), B(1,2) * A(2,2)] \\
&= [0.375, 0.0625]
\end{aligned}
$$

Finally, the probability of occurring $S1$ at time $t = 12$ is higher (0.375). Therefore, we take it as the predicted value of the tracked person status in front of the destination camera at time $t = 12$.

5.1.1 The knowledge base of the proposed case study based on ASP

In inductive machine learning and data mining from very large data bases, it is important to know that the background knowledge can be used as good guidance for extracting information from the data. To achieve this goal, we need a rule engine or a reasoning tool. Rule-based systems are successfully applied across a lot of domains. Using Answer Set Programming (ASP) we are able to define the rules of a system to detect defined simple and complex events [80]. The knowledge base of the proposed case study consists of the following entities:

1. **Object Entity:** The object entity has the following properties: hasId, hasType, hasZone, hasSpeed, hasDirection and a qualityRate for every attribute.

2. **Simple Event:** This is the simplest form of events, e.g. run, walk, shoot, etc.

3. **Complex Event:** A complex event is the combination of the simple events, e.g. a group of persons are running, a group of persons are fighting, a group of persons are in a forbidden area.

4. **Temporal Entity:** In ASP, time is usually represented as a variable in which the values are defined by an extensional predicate with a finite domain. Dealing with finite temporal intervals can be used to reason complex events in our case study.

5. **Flow Prediction Entity:** It is defined by the calculation of the flow of people using the proposed hidden Markov model (HMM).

5.1.2 Uncertainty in the knowledge base of ASP

Usually, the uncertainty handling approaches that are addressed in section 4.3, can be integrated with Answer Set Programming (ASP) to manage uncertainty in surveillance systems. In this section, we will illustrate an example of how to do this. In the state-of-the-art section, different approaches are explained for handling imprecise information within the knowledge base of a surveillance system. Usually, in surveillance systems the quality measure is assigned to the objects and their attribute values depending on the success rates and reliability of the sensing systems.

Based on this information, the reasoning process must consolidate the available quality information for several objects respective attributes to achieve an overall quality assessment as a measure of trust for the final decision. In the knowledge base of ASP different attributes should be combined for a quality assessment of the whole constraint.

Reasoning a single quality assessment for the whole event recognition process would mean that one or a few bad quality constraints negatively influence the quality statement for the whole decision, even if the information about the remaining constraints is reliable. Thus, instead of degrading the final decision, a single quality value is assigned to every attribute.

The simplest idea would be to use the minimum of all quality values for assessing the quality value of the whole constraint. A simple example will demonstrate this. Assume that the following quality values Q are given for the attributes:

- The speed of the first object $S1$, $Q = 0.96$
- The type of the first object $T1$, $Q = 0.97$
- The speed of the front object $S2$, $Q = 0.95$
- The type of the front object $T2$, $Q = 0.98$
- The predicted location of the first object $PL1$, $Q = 0.94$
- The predicted location of the front object $PL2$, $Q = 0.93$

The consideration of the minimum of the given quality measures for reasoning would give an overall quality of 93% to the given constraint. The result will always be just as good as the weakest quality measurement. If all available information is imprecise to a certain

degree, the final reasoning result will usually have a higher degree of imprecision than each of the single values alone. This is because it has been determined from a combination of imprecise values.

Therefore, a more realistic approach is to combine the available quality measures by multiplying them, similar to probabilities. Consequently, the result of the overall quality would be 0.76%. If one of the quality measurements is for example $T2 = 80$ (the type of the front object), the overall quality would be 0.62%. This shows the simple approach that the overall quality value depends on the weakest attributes information.

Thus, adding weights to the attributes that could influence the whole event recognition type could yield to the desired recognition without the consideration of the low single quality measurements. Suppose that the desired event recognition is "objects are running in different directions." In this case, the type of the objects is now less important than the speed and the predicted location. Now, if we add the following weights:

- The speed of the first object $S1$, $Q = 0.97$, $W = \frac{0.5}{6}$
- The type of the first object $T1$, $Q = 0.95$, $W = \frac{0.5}{6}$
- The speed of the front object $S2$, $Q = 0.99$, $W = \frac{1}{6}$
- The type to front object $T2$, $Q = 0.98$, $W = \frac{1}{6}$
- The predicted location of the first object $PL1$, $Q = 0.95$, $W = \frac{1.5}{6}$
- The predicted location of the front object $PL2$, $Q = 0.98$, $W = \frac{1.5}{6}$

The result would be:

$$Q = 0.97 * \frac{0.5}{6} + 0.95 * \frac{0.5}{6} + 0.99 * \frac{1}{6} + 0.98 * \frac{1}{6} + 0.95 * \frac{1.5}{6} + 0.98 * \frac{1.5}{6} = 0.97\%$$

If one of the quality measurements of the previous example is $PL1 = 60$, then

$$Q = 0.97 * \frac{0.5}{6} + 0.95 * \frac{0.5}{6} + 0.99 * \frac{1}{6} + 0.98 * \frac{1}{6} + 0.60 * \frac{1.5}{6} + 0.98 * \frac{1.5}{6} = 0.88\%$$

This approach yields the most natural result for the overall constraint. The imprecision of all attribute values is taken into account according to their importance for the final decision.

5.1.3 The integration of the knowledge base for ASP and HMM

In this section, we will illustrate an example of the integration between the hidden Markov model (HMM) prediction module and the knowledge base of Answer Set Programming (ASP). Suppose, that the HMM module provides a prediction of the next location of a specific object with a success rate of 96%. In the knowledge base of ASP, every fact v has a structure of $v(1, 96, 50, 97, 25, 95, 50)$ values:

- The first value is the low level feature ID.
- The second parameter is the success rate of the predicted object location.

- The third parameter is the weight of the importance of the related attribute.
- The fourth parameter is the success rate of the object type.
- The fifth parameter is the weight of the importance of the object type attribute.
- The sixth parameter is the success rate of the predicted flow of people using HMM.
- The seventh parameter is the weight of the importance of the flow of people.

Clearly, the specification of the weights depends on the type of the complex event detection. In the previous example, we had the event "objects are running in different directions." In this case, the type of the objects is now less important than the speed. Therefore, the weight of the object type can be less than the weight of other important attributes.

```
1. v(1,96,50,97,25,95,50).
2. v(2,98,25,94,75,93,75).
3. v(3,99,25,92,75,91,75).
4. uncertValue(ID,Q1,W1,Q2,W2,Q3,W3,FV):-
5. v(ID,Q1,W1,Q2,W2,Q3,W3),
6. FV =Q1*W1+Q2*W2+Q3*W3.
7. uncertValues(FV):-uncertValue(_,_,_,_,_,_,_,FV).
8. res(MaxVal):-MaxVal=#max[uncertValues(FV) = FV].
```

The previous code is an example of choosing the features with the highest probability based on ASP. The first three lines are the extracted low level features. Line 4, 5 and 6 calculate the measurement of accuracy of every feature extracted in the low level processing step. Line 7 assigns the values into their vector of accuracy rates FV. Line 8 chooses the maximum highest success rate of all received features in a specific time window (2 minutes in our case) and it assigns it to the variable MaxVal.
The previous example shows that choosing the best features to detect a complex event has the advantage of high speed calculation time on an embedded platform. We show that the use of ASP can significantly reduce the effort needed to detect complex events, while obtaining the same level of quality in the detected events. ASP is expressive, convenient and supports formal declarative semantics. Thus, ASP can be used to detect a large number of simple and complex events within a reasonable time frame that allows real-time operation with respect to limited hardware resources.

5.1.4 Simulation scenario and results obtained

In the evaluation phase, random trajectories of people are generated based on our designed simulation tool to create history data. Our simulation tool is developed in $C\#$; it generates data, trains and evaluates the overall concept. The data sets of the history are divided in two parts: a training data set and a test data set. We evaluate the proposed Hidden Markov Model (HMM) using different samples with different history data. In Table 5.6, we see the results of these tests where the columns show the overall success performance. This means there is a successful match between the observed states and

Table 5.6: The obtained results of different test scenarios of HMM module

Test	Number of Sample	Success Performance
1	10000	95%
2	20000	95.4%

the states at the current situation in the scene. The observed overall performance of the prediction has been on average 95%, especially after using 10000 training samples .

To evaluate the running time of the Answer Set Programming (ASP) reasoner atom-based embedded boards are used. A pITX-SP 1.6 plus board manufactured by Kontron[2]. It is equipped with a 1.6 GHz Atom Z530 and 2GB RAM.

iClingo[3] as a solver of ASP is used to detect the complex events [80]. It is an incremental ASP system implemented on top of clasp and Gringo. *iClingo* is written in *C* and can be run under *Windows* and *Linux*.

We measured the execution time of the ASP solver on our embedded platform. It shows that the knowledge base of ASP is far more suited for embedded operations because the overall execution time on average for more than 50 simple and complex events and 843 extracted features is 0.4s.

Furthermore, the system can parse over Ontology Web Language (OWL) using dlvhex[4]. Dlvhex is the name of a prototype application for computing the models of so-called HEX-programs and is developed in TU-Wien. The goal of dlvhex is to extend ASP towards an interface of Description Logics, which are the theoretical foundation of ontology languages like OWL. For example, the OWL context model which is designed in sections 6.2 can be combined with dlvhex.

5.2 The novelty of using ASP in video surveillance systems

Little research has been done in the frame of using Answer Set Programming (ASP) and in reasoning under uncertainty in surveillance systems. Reasoning support for the Semantic Web is currently mainly restricted to terminological reasoning in description logics. There is no broad consensus on what will constitute the logical layer of the Semantic Web.

The proposed approach can be used anywhere and anytime (the adjustment with new environments only has to be considered). The advantage of the proposed concept is that the modification of the knowledge base is easy. Sometimes, when introducing new knowledge to solve some specific problem, for example adding a new rule, we might introduce contradictions within the previous rules.

Human observers are not 100% reliable or consistent. They suffer from fatigue and the effect of emotional involvement. In surveillance systems such problems have to be avoided.

Spatio-temporal event detection in surveillance systems has different requirements that have to be covered. Spatio-temporal event detection in surveillance systems needs a

[2]http://www.kontron.com
[3]http://potassco.sourceforge.net
[4]http://www.kr.tuwien.ac.at/research/systems/dlvhex/

temporal reasoning whereby many other logic programming approaches suffer because of the lack of temporal constraints. In ASP, time is usually represented as a variable in which values are defined by an extensional predicate with a finite domain. Finite temporal intervals can be used to reason complex events in surveillance systems.

ASP supports a number of arithmetic functions that are evaluated during grounding. Therefore, all reasoning under uncertainty approaches can be implemented in ASP.

Conditions allow for instantiating variables for collections of terms within a single rule. This is particularly useful for encoding conjunctions or disjunctions over arbitrarily many ground atoms, as well as for the compact representation of aggregates.

An aggregate in ASP is an operation on a multi-set of weighted literals that evaluate to some value. In combination with comparisons, we can extract a truth value from an aggregate's evaluation; thus, obtaining an aggregate atom.

Optimization statements extend the basic question of whether a set of atoms is an answer set to an optimal answer set. Optimization in ASP is indicated via maximization and minimization. The use of this feature in ASP has the important task of reasoning under uncertainty in the field of video surveillance systems, e.g. the selection of the best low level features of hundreds to detect a specific event with respect to uncertainty.

Constraints play an important role in ASP because adding a constraint to a logic program P affects the collection of stable models of P in a very simple way. It eliminates the stable models that violate the constraint. This feature can be applied in video surveillance systems by the definition of the constraints in the environment, e.g. walking on a forbidden area or the recognition of abnormal behavior.

Ontologies based on Semantic Web provide concise high-level definitions of activities but they do not necessarily suggest the right hardware to parse the ontologies for recognition tasks (Semantic Web). It offers the best of both worlds because it has the expressive and descriptive power of Ontology Web Language (OWL) and the reasoning and arithmetic power of ASP.

The major advantages of using rule based systems are that each rule can be seen like a "unit of knowledge." Additionally, all the knowledge is expressed in the same format and more importantly the rules are in a natural format to express knowledge in a domain.

5.3 The novelty of combining ASP and HMM for reasoning under uncertainty

Dealing with uncertainty in surveillance systems needs arithmetic operations that are usually not well presented in logic reasoning tools. Answer Set Programming (ASP) offers the standard arithmetic functions and the absolute function. In addition to this, other arithmetic can be implemented and reused depending on the use case of the desired reasoning process under uncertainty.

A series of approaches considering uncertainty in event detection can be implemented based on ASP, for example: confidence functions in a Boolean data type format, the fuzzy modeling approach and the Dempster-Shafer approach. The latter uses belief and plausibility functions to describe the reliability features.

Consequently, the extensions of ASP have to be considered, e.g. the combination between ASP and fuzzy theory, Fuzzy Answer Set Programming (FASP). This combination offers the best of both worlds because of the answer set semantics, it uses the power of

its declarative non-monotonic reasoning capabilities. Meanwhile the concepts from fuzzy logic manage to avoid the limitations of classical logic. As fuzzy logic offers a great exibility.

The novelty of this work is that it proposes a robust approach based on the combination between Hidden Markov Model (HMM) and Answer Set Programming (ASP). A weight should be calculated for all related extracted features and then the event with the highest probability will be selected using the optimization power of ASP.

In relation to the previous advantages, the optimization possibilities of ASP, e.g. the maximization and minimization, can be applied to choose the optimal sensor data despite the different taxonomies of uncertainty in surveillance systems. Furthermore, the cardinality and the constraints in ASP can be used in the body of ASP rules to give the developer the possibility of optimizing the desired answer sets.

Usually, rule based systems suffer from a lack of trust (uncertainty). Therefore, this combination with HMM is required. The problem that must be faced is related to the task management, tracking and data processing. Despite the use of ontologies to provide concise high-level definitions of activities, ontology tools do not necessarily suggest the right "hardware" to parse the ontologies for recognition tasks. ASP offers the possibility to reason the descriptive and expressive power of Ontology Web Language (OWL).

In the state-of-the-art approaches, prior knowledge is important but any lack in prior knowledge can affect the whole inference and reasoning process. The combination between ASP and HMM can successfully reduce the effects of prior knowledge.

HMMs are used because they have proven to be effective in a number of domains, especially prediction and recognition. One of the most important advantages of HMMs is that they can easily be extended to deal with complex domains in order to detect several Markovian events. This is because each HMM uses only positive data scales well and can be combined into larger HMMs.

Markovian assumption means that the emission and the transition probabilities depend only on the current state. This does not map well many real world scenarios in the frame of complex event detection[5].

The basic theory of HMM is also very elegant and easy to understand. This makes it easier to analyze and develop implementations. Statisticians are comfortable with the theory behind hidden Markov models. HMMs offer a freedom to manipulate the training and verification processes. HMMs are still very powerful modeling tools and are far more powerful than many statistical methods.

One of the advantages of the HMM-based approach is that several knowledge sources can be combined into a single HMM. By representing all possible knowledge sources as HMMs, the recognition task of complex events becomes a search in an enormous HMM. In the case, that no knowledge is added, a recognition model can be simply created by putting all complex event models in parallel and adding an initial state and a final state. The initial state of the recognition model has a null transition to the initial state of each gesture model. A null transition is a transition that has a transition probability but does not emit any output symbol. Therefore, it does not consume any time [100].

The combination between OWL, HMM and ASP offers a powerful approach to combine the advantages of an OWL context model which are explained in sections 3.2.5 and 6.2 plus the advantages of ASP that are explained in section 3.3.3 and in section 5.2. There-

[5]http://www.cse.unsw.edu.au/ waleed/phd/html/node36.html

fore, it associates the advantages of rule based systems plus the advantages of stochastic approaches and can handle uncertainty using ASP during the reasoning process. Furthering this, it can detect different simple and complex events in real-time, run on embedded platforms and reason with respect to temporal constraints.

5.4 A model free event detection and position estimation of humans

In England, a third of the population that are over the age of 65 have a fall each year. In addition to this, half of these persons fall at least two times. Women are at a greater risk than men, with half of all women over the age of 85 having a fall in any given year [101].

It is important to find a technical solution to detect the falls of elderly people, which enables them to efficiently call for help. In an instance where a heart attack is the reason for a fall, each second is important to save a person's life. Consequently, the system must have a low latency, so that an alarm is sent as soon as possible.

A model-free and computationally inexpensive approach for the fall detection of humans is needed in order to have an easy and simple solution. It must allow an integration on FPGA-based smart cameras [102] without the need of a bigger Field Programmable Gate Array (FPGA). Therefore, a model free approach is used. The usage of one camera only leads to greater acceptance for in-house usage due to its minor need for technical devices, which we think is important in assisted living environments.

5.4.1 Related works on model-free event detection

The Institute of Robotics at the University of Braunschweig, Germany, is developing a system that will ensure a long, independent and secure life for elderly and handicapped people in their home environment. An initial supervision system is already operational and currently undergoing tests. The image processing system is able to detect people automatically by using a camera and to decide whether an ambulance should be informed or not. The institute has developed several model-free and model-based image processing algorithms that enable the tracking of the people and the detection of falls in a room [103]. The system has also been developed for active supervision approaches that allow the identification of individuals who fall during the night, i.e. in the dark. Therefore, infra-red emitters are attached in various positions on the ceiling. Whenever these lamps turn on and off, the person supervised casts a shadow in different directions. The shadow information is used to distinguish between people standing and people lying on the floor [104]. However, the system only works in covered areas provided with a huge amount of signals. This requirement is hard to obtain in most living environments of elderly people. Also, the segmentation approach is not adaptive to small background movements, like those that normally occur in-house.

A team at the University of Massachus etts uses a network of overlapping smart cameras in a decentralized procedure. They can compute inter-image homographies that allow the location of a fall to be detected in 2D world coordinates by calibrating the respective data. The aim of their work is to build a system which implements fall detection procedures. They succeeded by using a more sophisticated Support Vector Machine (SVM) classifier.

40 image sets were used to train the system and each image was utilized in order to build a classifier that decides whether an event is regarded as a fall or not. The system requires the manual calibration of one camera per a group of overlapping cameras. The localization error of the system is less than $50cm$. The system comprises of very low-power, low-resolution cameras and motes can be used to detect and localize falls [105]. However, as mentioned before we tried to build a model free system without the need of training data and manual calibration. Furthermore, the usage of several cameras per a room is an exclusion criterion, because this technical requirement cannot be guaranteed in most living environments.

5.4.2 Advantages and novelty of using model-free event detection

The aim of this work is to develop an algorithm that processes video streams to detect people and analyze their position in space, especially recognizing people who are lying on the floor. The problem of object detection and evaluation is well established in computer science. In general, model and training based approaches are used. Viola and Jones [106] presented AdaBoost in 2001, where a boosted cascade of classifiers based on haar-like features is used. Integral images were used to speed up the process. In [107] Ashbrook et al. used Pairwise Geometric Histograms (PGH) to detect objects even with occlusion or scene clutter present. Rotation invariant histograms are built out of geometric parameters of line segments.

The PGH calculated out of training images is later compared against the PGH from the scene image using the Bhattacharyya metric. Another way to solve the problem of the usage of optical flow [108]. The flow of walking people is directed uniquely to the whole body. If the person falls, then its optical flow is pointing into several directions. This fact could be used to detect falls as abnormal behavior of the optical flow. However, as well as model and training-based approaches optical flow requires a lot of training with respect to computational power.

A model-free and inexpensive computational approach is needed in order to have an easy and simple solution, which allows integration on embedded platforms. Therefore, a model free approach is used. The usage of one camera only leads to greater acceptance for in-house usage due to its minor need for technical devices, which we think is important in assisted living environments.

Event detection on embedded platforms requires a model-free and an inexpensive computational approach in order to have an easy and simple solution, which allows the integration of an FPGA-based smart camera without the need of a bigger FPGA. Therefore, the thesis presents a solution based on a foreground-background segmentation using Gaussian Mixture Models (GMM) to first detect people and then analyze their main and ideal orientation using moments. This allows one to decide whether a person is staying still or lying on the floor. The system has a low latency and a detection rate of 88% in our case study. Another key of this algorithm is the use of Gaussian mixture models for image segmentation that is not sensitive to the light and small movements in the background of a scene and considers shadow detection, which has an influence on the overall event detection process.

5.4.3 Detailed concept description of model-free event detection

To achieve the high performance of OpenCV library, it must run on high speed processors. If it does not, it will have a sub-optimal performance. As an example, the performance of OpenCV library under AMD processors is not equal to the performance under Intel Processors. However, in our case, all tests are run under Linux version 2.6.24-22-generic, Ubuntu 4.2.3-2ubuntu7, gcc version 4.2.3. The model name of the CPU is Intel(R), Core(TM)2 Duo CPU 2.00 GHz, cache size is 2048 KB. OpenCV version is 2.0 from September 30th 2009. The test environment is a 6 * 4 meters room where a network camera has been installed in the middle.

After finishing the programming of the final algorithm, the system was tested under several conditions. The test consists of nine positions within the room, for each position there were five iterations to test the system. This means that 45 tests were conducted in total. The test scenario included one person falling to the floor of our test room whilst being observed by the camera. Sometimes the person fell with motion and sometimes without motion. The test time was in the afternoon and it was used artificial room lighting.

Figure 5.3: The test of the system

5.4.4 The overall architecture of the system

To explain our approach to the problem, the system will be described in general. Humans will be monitored through a fish eye camera, which is situated in the middle of the ceiling. The special perspective of this kind of lens gives an easy parameter to detect whether people are staying or lying, because the main axis of standing people is pointing to the middle of the image. This property is the basic principle of the algorithm. When the main axis of the segmented person is differing from its expected direction (the middle of the image), we classify the person as lying. Falls are detected as moving from an upright position into a lying one within a small time slot. The steps are shown in Figure 5.4.

a) Capture the current image of the room.

b) Segment the moving object from the static background using adaptive GMM.

c) Calculate the ideal orientation ϕ as it should be for upright persons.

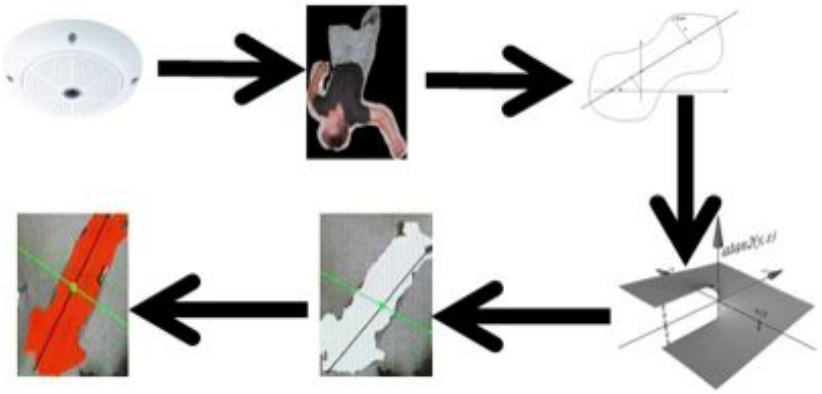

Figure 5.4: The working flow of the system

d) Calculate the main orientation θ of the person.

e) Use ϕ and θ to decide whether the person is upright or lying on the floor.

Now we will introduce the system in details by looking at each step individually.

Image capturing

The most important part of the image capturing process is the fish eye view of the camera. It has several advantages. Firstly, it can cover big rooms due to its wide angle view. It captures 180x160°. Secondly and more importantly is the algorithm. The main axis of upright standing people on an image is always pointing to the middle of the image, as can be seen in Figure 5.5. The above mentioned is only true if the camera lens is mounted in parallel to the room's floor, i.e. the image capturing direction is perpendicular to the floor. But in normal rectangular rooms this property is always true if we mount the camera to the ceiling of the room. A third advantage is that the camera automatically deskews the image after capturing for an easy processing.

Segmentation

The second part of the algorithm is the foreground-background segmentation to obtain the mask of the person. In order to locate moving foreground objects in videos, a segmentation technique called Gaussian Mixture Model (GMM) is used. GMM is an important tool in image data analysis. The segmentation of images means to divide the video frame into different types of classes or regions, background and foreground. Therefore, we can suppose that each pixel belongs to a Gaussian distribution with its own variance value and covariance matrix. These parameters of the model are learned by an Expectation-Maximization algorithm. Each mixture component consists of a Gaussian with a mean μ and covariance matrix $\sum$, i.e. in the case of a 2D color space:

$$p(\xi|j) = \frac{1}{2\pi|\sum j|^{0.5}} exp^{0.5(\xi-\mu_j)^T \sum_j^{-1}(\xi-\mu_j))} \tag{5.9}$$

So, by a Gaussian background mixture model each pixel is modeled as the sum of k weighted Gaussians. The weights show the frequency and the Gaussian is identified as

Figure 5.5: Example image view of the fish eye camera (without deskew [109])

part of the background model, updated adaptively with the learning rate α and the new observation.

The idea of Maximum Likelihood Estimation (MLE) is to make an assumption about a specific model with unknown parameters and then to define the probability of observing a given even, which is conditional on a specific set of parameters. This means after one has observed a set of outcomes in the real world. Then, it is possible to choose a set of parameters that are most likely to have produced the observed results. If we have a likelihood function, this could be a mathematical distribution $f(x_1, x_2, .., x_n; \Theta_1, \Theta_2, ..., \Theta_n,)$, where the data x_i's are normally distributed (Gaussian distribution) and independent. Thus, we can calculate the parameters Θ:

$$L(X|\Theta) = \prod_{i=1}^{n} f(x_i, \theta) \tag{5.10}$$

Next, the likelihood function should be maximized by calculating $\frac{\partial L}{\partial \theta} = 0$, but this would be difficult. Therefore, we use a simplified algorithm called EM (Expectation-Maximization).

EM maximizes the likelihood of fitting a mixture model to a set of training data. It is important for this algorithm that a prior selection of the model order and also the number of k components is incorporated into the model. It is important to be aware that mixture models do not work well in case of constant repetitive motion and high contrast between pixel values (edge regions), because the object appears in both the foreground and background [110].

Calculating the orientation

After segmentation of the image into foreground and background we have to determine the two main angles ϕ and θ. θ is the main orientation (main axis) of the detected object, see fig 5.6. ϕ is the expected orientation of a person standing upright with the same center of mass as the detected person, see Figure 5.7.

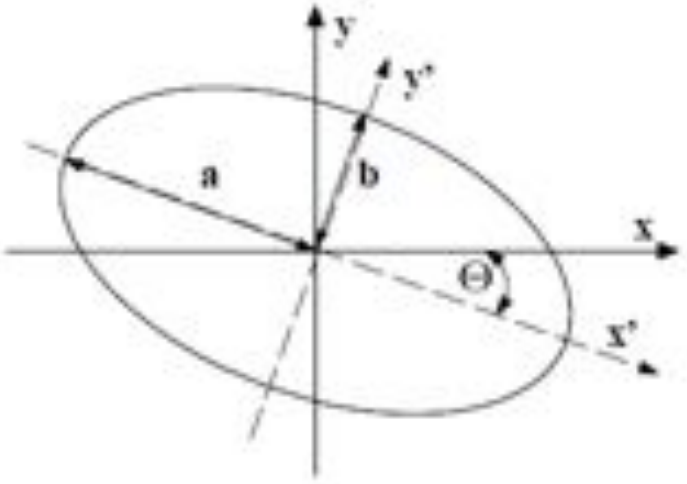

Figure 5.6: Main orientation θ of an abstract object (ellipse) within an image using a fish eye camera

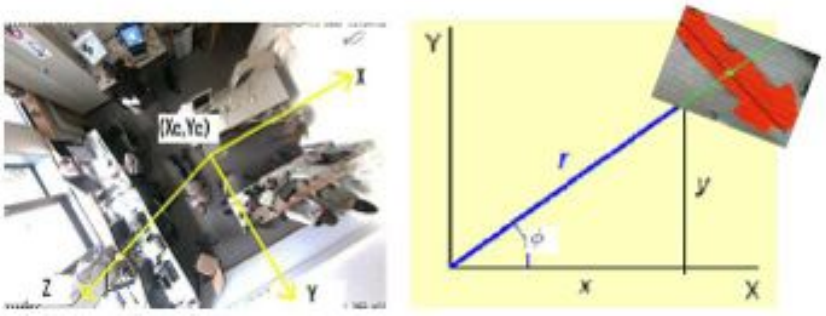

Figure 5.7: The calculation of the ideal orientationϕ

The two angles are calculated using moments. Moments are the processing of certain weighted averages of the values of each pixel of an image. They are usually chosen so that they reflect the desired characteristics of the image. They are useful to describe individual objects in a segmented image. The definition of moments as a gray value function $I_B(x, y)$ of an image is as follows, where p and q are the order of the moment [111]:

$$\mu_{p,q} := \int_Y \int_X x^p y^q I_B(x, y)\, dxdy$$

The integration is calculated over the area of an image I_B. When we have segmented

an object, we get a binary image mask M in which the pixels contain the value of either one or zero. Than we also can integrate over the objects area:

$$\mu_{p,q} := \int_Y \int_X x^p y^q I_B(x,y) M(x,y)\, dxdy$$

Zero order moments $\mu_{0,0}$ are the sum of the pixel values of an image. First order moments $\mu_{1,0}$ and $\mu_{0,1}$ describe the horizontal and vertical center of the mass of an object.

$$\mu_{0,0} := \sum_x \sum_y I_B(x,y)$$

$$\mu_{1,0} := \frac{\sum_x \sum_y x I_B(x,y)}{\mu_{0,0}}$$

$$\mu_{0,1} := \frac{\sum_x \sum_y y I_B(x,y)}{\mu_{0,0}}$$

They are later used to calculate the ideal orientation ϕ of a person, relative to its center of mass. Second order moments are the squared values of the row or column-counters multiplied by the value of the rows of the image. The normalized second order moments are associated with the orientation of the object and used for the main orientation θ:

$$\mu_{2,0} := \sum_x \sum_y x^2 I_B(x,y)$$

$$\mu_{0,2} := \sum_x \sum_y y^2 I_B(x,y)$$

$$\mu_{1,1} := \sum_x \sum_y xy I_B(x,y)$$

The orientation of the object is defined by the tilt angle between the positive x-axes and the axis around which the object can be rotated with minimal inertia [112]:

$$\theta := \frac{1}{2} \tan^{-1}\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right)$$

Fall decision

Now that the orientation of a person is calculated, we have all relevant data to make the final position estimation of a person and decide whether there was a fall or not. The main decision parameter is the deviation between the ideal orientation ϕ and the main orientation θ of the segmented object. A fall is detected in one frame if

$$|\phi - \theta| \leq \epsilon,$$

see also Figure 5.8. The specific threshold ϵ was determined empirically, i.g. the angle of a falling person has been observed several times. To ensure the robustness of the approach and to avoid false positive detections due to a one frame error, a frame counter is added. If we detect a person lying on the floor in one frame we increment a counter c. If we don't

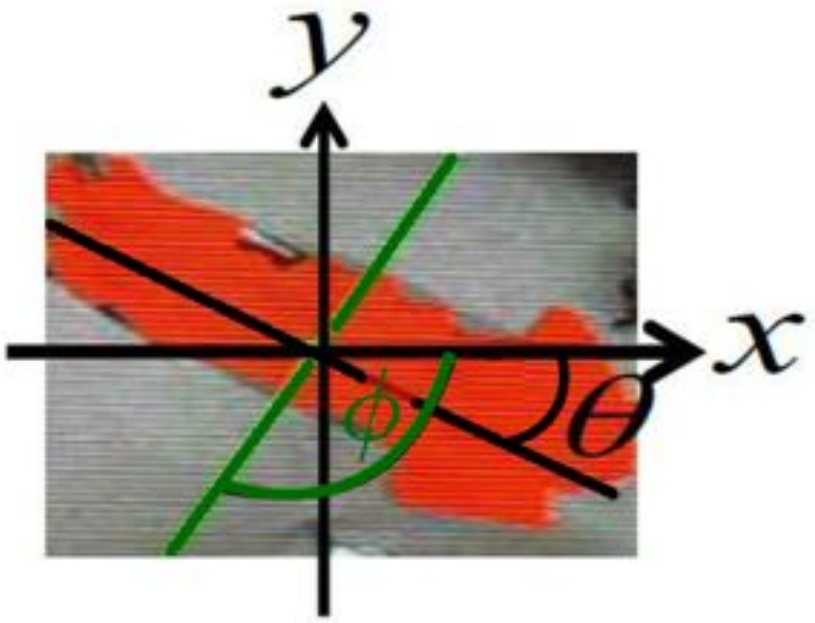

Figure 5.8: Example view of a fall with a large deviation between the main orientation θ (black) and the ideal orientation ϕ (green)

detect it, we reset the counter, $c = 0$. If c increases beyond a predefined threshold, then we alert a detected fall.

It needs to be mentioned that we perform the orientation estimation for each detected person separately. Therefore, the foreground mask is separated into connected components and the algorithm analyzes each component independently. This way also multiple orientations in a room can be evaluated and therefore multiple falls can be detected.

Shadow detection

One of the most frequent problems of computer vision systems deployed in environments suffused with light are the shadows. Especially if the background subtraction operator is affected by shadows, since shadows are detected as part of the element in motion. Despite using a Gaussian Mixture Model (GMM) to segment the objects, the shadows still posed a problem to the performance and so an algorithm was found to detect the shadows and delete them from the images.

Cucchiara et al. [113] proposes the usage of the three parameters of the HSV color system (Hue, Saturation, Value (brightness)) to detect shadows. The HSV color space corresponds closely to human perception of color in cases where the color information improves the discrimination between shadow and object. Intuitively, of course, when there is a shadow on a background texture, the hue value (color) stays almost the same. The saturation value decreases and the shadow causes a reduction in the brightness value, because shadows make textures look darker than normal. So for each foreground pixel its HSV values are compared to the HSV values of the pixel at the same position in the learned background image:

$$\alpha \leq \frac{V_i(x,y)}{V_B(x,y)} \leq \beta$$

$$S_i(x,y) - S_B(x,y) \leq \tau_s$$
$$|H_i(x,y) - H_B(x,y)| \leq \tau_h$$

Here i is the current frame and B is the background frame. If a foreground pixel fulfills all the conditions, it gets removed from the foreground mask.

After the removal of shadows the performance of the system has been highly increased. The results of the shadow detection are good. The quality of the overall algorithm is discussed in the previous section. The parameters of the shadow detection are 0.2 for α, 0.95 for β, 5 for the τ_s and the τ_h is 15.

5.4.5 Performance results obtained and related comments

The best parameters to measure the performance of the system are the measures of sensitivity and specificity. Sensitivity measures the proportion of actual positives, i.g. what is correctly identified, and specificity measures the proportion of negatives, i.g. what is incorrectly identified. Table 5.7 shows us the measurements: 86% of true positives are detected and 88% of the events (the person is not lying on the floor) are detected.

A theoretical, optimal prediction should achieve 100% sensitivity and 100% specificity. Specificity is the number of true negatives divided by the sum of true negatives and false positives. Sensitivity is the number of true positives divided by the sum of true positives and false negatives.

The system is a real-time system. Real-time systems are used when it is imperative that an event is reacted to, within a strict deadline. This type of reaction is required for our system. Therefore, the detection time is between 2-20 seconds. The Figures are examples from the test, see 5.3. In some tests we noticed that if the body is partly covered by a chair or a table, the detection still worked correctly.

	Detected		
	true	false	
positive	TP=40	FP=4	
negative	TN=43	FN=5	
specificity			$= TN/(FP+TN)$ $= 43/(4+43)$ $= 91\%$
sensitivity			$= TP/(TP+FN)$ $= 40/(40+5)$ $= 88\%$

Table 5.7: Specificity and sensitivity

However, Figure 5.9 illustrates the weak point of the system where it is difficult to achieve the threshold value. This is due to the ideal orientation that indicates the angle of an upright person being the same as the real orientation that indicates the angle of a fallen person.

Figure 5.9: The weak point of the system

5.5 Summary

Probability theory and logic programming are important for many complex events detection applications involving uncertainty. It requires a detailed analysis and understanding of the domain and also a great deal of data to alleviate the problem of uncertainty.

We demonstrated that the combination between stochastic methods (HMM) and logic programming (ASP) provides a powerful tool to detect complex events, especially the behavior of people in a crowded scene. The high performance of the HMM model for creating a prediction provides the ontology with facts with less than 5% uncertainty. Therefore, the overall performance of the system is increased. ASP can be used to detect a large number of simple and complex events within a reasonable time frame that allows real-time operation with respect to limited hardware resources.

The proposed algorithm used to estimate the position of people in space is also used to detect falls. The solution is based on a calculation of the deviation of the main and ideal orientation of segmented objects from a fish eye camera, where the main orientation of standing people is pointing to the middle of the image. Moments are used to calculate the orientation and the deviation between the two main and ideal orientations. In turn this is compared with a specific threshold to decide if the person has fallen.

The system has a low latency and a detection ratio of 88%. The high sensitivity and specificity renders this system fit for usage in assisted living environments. The usage of only one fish eye camera per room and a standard PC makes it easy to integrate into a normal apartment. Additionally, the system could be provided with infrared vision to detect falls during the night.

In the future, we want to integrate the algorithm into a smart camera [102], the most intensive computation task is the segmentation using GMMs. When realizing this part as hardware in the FPGA and the rest of the algorithm in software running on the PowerPC of the FPGA the algorithm would speed up a lot.

Chapter 6

Case studies related to complex event detection under uncertainty

This Chapter applies the proposed solutions in Chapter 5 in 2 case studies. The approaches are applied in the frame of a SRSnet project to check and judge their performance. The first case study is the design of a video surveillance system based on Semantic Web. This case study proposes a robust solution to representing context models for a video surveillance application, especially in order to recognize complex events which cannot be directly detected. The solution is based on building an ontology for representing prior knowledge related to video events. The designed ontology is composed of all high level semantic concepts in the context of the test environment.

The second case study presents a real-time complex event detection concept for resource-limited multimedia sensor networks. A comprehensive solution based on Answer Set Programming (ASP) is developed. We show that ASP is an appropriate solution to detect a large number of simple and complex events (video-audio understanding) on platforms with limited resources, e.g. power consumption, memory and processing power. Then, we underline different origins of uncertainty in video surveillance systems and explain the major approaches to handle uncertainty in different levels. Furthermore, we demonstrate the high performance of ASP compared to that of Semantic Web.

6.1 Scenario definition for case study 1 and case study 2

In our case study, we aim to construct a resource-aware multi-sensor network. The goal is to deploy a sensor network consisting of video and audio sensors that is capable of detecting complex events in an environment with limited infrastructure (especially without access to the power grid).

This specifically means that there is no access to the power grid and thus, the sensor nodes must be able to operate on battery and renewable energy as long as possible. As a multimedia sensor network, the SRSnet needs not only to record and transmit sensor information, but also perform on-board data processing. In this case, object detection, localization and tracking will be performed on audio and video data.

An integral part of this project is the detection of high-level events. We want to fuse low level events detected by audio and video processing to higher-level complex events.

Additionally, the network must react to events and to new task assignments. This requires a component for dynamic network reconfiguration that reconfigures sensor parameters and nodes according to events, task assignments and resource requirements. This component receives information on detected events and on the current orientation of Pan Tilt Zoom (PTZ) cameras as the input.

A resource-aware multi-sensor network can be deployed in environments, like national parks, to help protect sensitive environments. In order to provide an interface for users, we will employ a multimedia data warehouse to collect detected events and multimedia artifacts. Then, users can query the database for interesting events in time and space. The data warehouse is meant to be deployed outside of the sensor network itself, e.g. as a cloud service. To feed information into the data warehouse we will use web services which are accessed by the sensor network. This architecture enables us to save energy by only connecting to the data warehouse on demand. A persistent connection is not needed.

Figure 6.1 shows an overview of the project parts and the data flow between them.

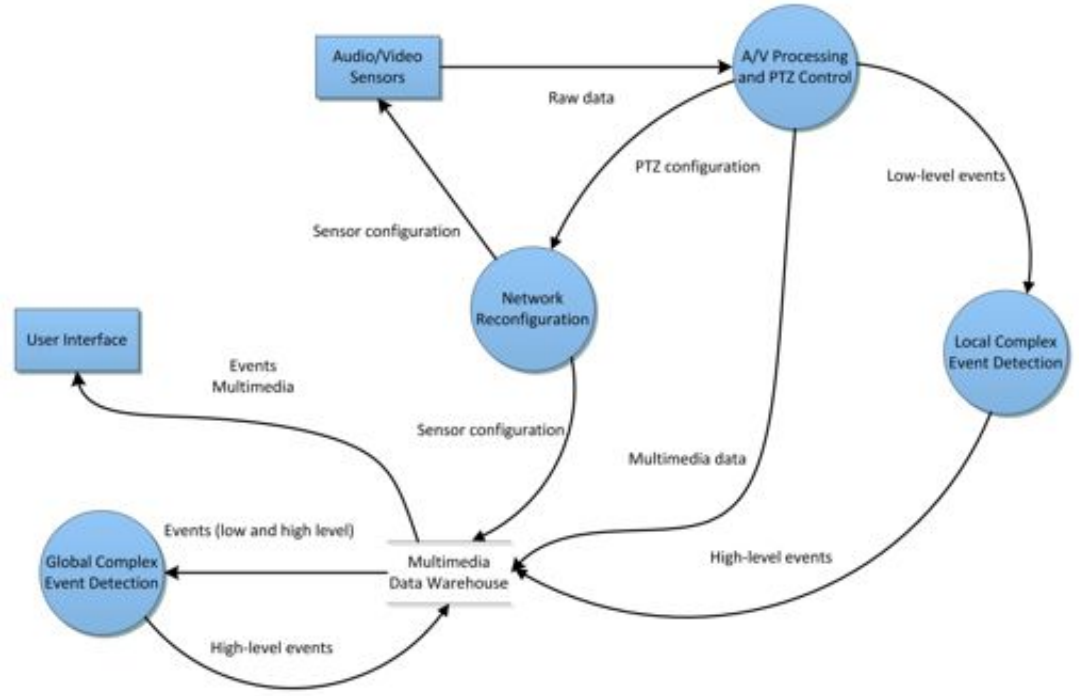

Figure 6.1: The components of SRSnet and the data flow between them. Bulbs indicate flow of data while lighting indicates operations or actions.

6.2 Case study 1: A comparison between Semantic Web and ASP for complex event detection in video-audio-based sensor networks

The Smart Resource-Aware Multi-Sensor Network project (SRSnet) is an Interreg IV research project funded by the European Community. The SRSnet project focuses on the design of a smart resource-aware multi-sensor network capable of autonomously detecting

and localizing various events, such as screams, animal noises, tracks of persons and more complex human behaviors. The project's research areas include the following modules:

- Collaborative audio and video analysis.
- Complex event detection.
- Network reconfiguration.

The SRSnet is demonstrated in a biologically sensitive environment, namely in the Hohe Tauern National Park. This national park was chosen as a testing environment as it offers realistic case studies. It also profits from the results of the project by having a better understanding of visitor and animal movements within the mostly restricted natural preserve area. The bridge between Chapter 5 and Chapter 6 is that the SRSnet project will be considered as the test environment to trial, evaluate and compare different reasoning approaches under different uncertainty handling approaches in the frame of surveillance systems.

6.2.1 The knowledge base designed for SRSnet

The ontology design of the audio/video surveillance system, two super classes "event" and "object" should be defined. From the super-class "event" three other sub-classes must be defined: simple event, spatial event and temporal event. Figure 6.2 shows the overall architecture of the proposed surveillance system based on Semantic Web.

These derived classes are not disjointed as an event can be created by inheriting from multiple classes. These specializations are defined as follows:

- **Object specializations:** For the Object concept. Types are observed from sensors, e.g. person, dog, group of persons, etc. Each object has the following properties:
  - objectId
  - hasObjectType
  - hasSpeed
  - hasDate
  - hasTime
  - hasDirection
  - hasCameraId
  - hasFrameId
  - hasX
  - hasY
  - hasUncertaintyObjectType
  - hasUncertaintyCordination
- **Sound specializations:** The sound types are observed from sensors, e.g. shot, scream, howl, etc. Each sound entity has the following properties:

Figure 6.2: The overall architecture of surveillance systems based on Semantic Web.

  - hasSoundType
  - hasSoundArrayID
  - hasSoundTime
  - hasSoundDate
  - hasSoundCorX
  - hasSoundCorY.
  - hasUncertaintySoundCordination.
  - hasUncertaintySoundType

- **Simple event:** This is the simplest form of events. A single entity is generally involved without interactions with secondary entities. Each event has these main properties: Duration, BegginIn, EndIn and the Event UID.

- **Complex event:** A complex event consists of a combination of different simple events.

- **Temporal entity:** This describes time entities for a specific event within a predefined period of time.

- **Event specializations:** Each detected event whether it is simple or complex has the following features:

Figure 6.3: The overall architecture of surveillance systems based on Semantic Web

- hasEventType
- hasObjectType
- hasSpeed
- hasDate
- hasTime
- hasCameraId
- hasFrameId
- hasSoundArrayID

The scene context defines all the information that happened during a scene. This is important during the detection of an event. Therefore, we have defined spatial context to describe the environment and object context to help with the detection of the relationships between the objects. Our spatial context was defined through the mapping between pixels and the real region of interest which is defined in the ontology with specific values. Figure 6.3 shows a snapshot of the designed ontology based on OWL.

6.2.2 Test and simulation environment

The development of the proposed system consists of the following steps. The first step is the extraction of the features from the scenes. Then, the modeling of the ontology defines the spatial and object context. The final step is to build the rules to detect events. Figure 6.4 shows the test environment in the park.

$$ObjectEntity(?x) \wedge Forbidden(?x, ?z) \rightarrow AlarmEvent(?z)$$

Figure 6.4: A snapshot of the test environment in the park

- **Features extraction:** For data generation and simulation a simple tool has been implemented using OpenCV[1] to extract and generate features for the evaluation of the proposed reasoning system. The goal of the features extraction is to extract the type, speed, direction and the coordinates of the moving objects. Our overall method consists of four main steps:

  - **First step:** is the segmentation, including the shadow detection.
  - **Second and third steps:** are the contours and their related features of the extracted contour.
  - **Fourth step:** is the tracking of the objects including the matching of these objects with our samples that are saved in the XML file.

  Upon receipt of video streams, the video frames are smoothed using a Gaussian kernel. Subsequently, the system starts to use Gaussian Mixture Models (GMMs) to segment moving objects. The segmentation succeeds by using multiple GMMs. This is a good way to learn the background and to model it by one or several Gaussian distributions and to determine its parameters. Next, the segmentation shadows have to be removed using the HSV color system and the contours are corrected using morphological operations. Contour moments are important in order to extract as much information about the segmented object as possible and to get the center, area, orientation and shape. The extracted features of the samples have

[1] http://opencv.org

been saved in an XML format with different classes of objects, such as humans, cars, dogs, etc. These will be compared with the contours. To avoid challenges with overlapped objects, we have stored several patterns of overlapped objects in several scenarios. Extracted features have been included and each case has been classified in its own class in the XML file.

- **Ontology design:** Protege as editor was used for the ontologies and Semantic Web Rule Language (SWRL) rules. The design of the ontology consists of the following steps. First, the specification of the desired events must be detected. Then, the context data (spatial context and object context) is designed. Next, the specification of the classes is needed to build the ontology. After the classes are defined, the properties must be given. Finally the consistent of the ontology must be tested using the reasoning tools that are provided in Protege. (see Figure 6.3).

- **Detection of complex events:** The ontology model permits the use of a reasoner that can check if the definitions of the statements in the ontology are consistent or not. It can also recognize which concepts are the best for which definitions. Therefore, the reasoner can help to maintain the hierarchy correctly. This is particularly helpful when dealing with multiple class hierarchies. The rules were built using the jess engines, which are helpful to show the resulting SWRL rules.

6.2.3 Results obtained and related comments

In this section, we will show some results of the proposed concept. To test the system in order to detect the complex events we used the Jess Rule Engine for evaluating SWRL rules. We have created a set of instances of some objects included in our ontology. The system has been tested in a parking place (see Figure 6.4).

Figure 6.5: The overall architecture of surveillance systems

- **First Example:** At the beginning, simple events can be defined and detected through the use of the extracted features. For example, to detect that a person is walking or running the rule is:

  $$Person(?x) \wedge hasSpeed(?x, ?y) \wedge sqwrl : greaterThat(?y, 5) \wedge has(?x, ?ID) \rightarrow run(x)$$

  This rule means if the object is a person and has an Id and a speed which is greater than 5 then this person is running.

Figure 6.6: The overall architecture of surveillance systems

- **Second Example:** In this example, we want to detect objects that are walking in a forbidden area. At the beginning we have to define the rule for forbidden areas:

  $$ObjectEntity(?x) \wedge hasZone(?x, ?Z) \wedge hasID(?x, ?id) \wedge sqwrl : equal(?Z, 0) \rightarrow Forbidden(?area, ?Z)$$

  Through the use of our spatial context where regions of interests have been defined, we gave the value zero for the forbidden areas which could be visited. The previous rule defines a forbidden area as the area that has the value zero. After the definition of forbidden areas, we can define the rule to detect objects which come into this forbidden zone (Figure 6.5 and 6.6):

Detection of different events or situations has become an important topic in audio and video surveillance systems within recent years. In particular the surveillance of public areas, such as airports or train stations, has been the focus of research and development. Event detection in combination with context modeling is widespread. Especially ontology-based on Semantic Web; this is an expressive technology that has the ability to describe certain aspects of the world. It supports developers to model the relationships between the sources that can be described. After the design of the ontology, the relevant information of the scene can be described in terms of scene-related entities (object, event and context.) Then, the complex events can be defined based on a list of simple events. It is important for the development of video understanding systems (outdoors systems) to use the paradigm of ground truth through geometric correction. This will be needed for the description of regions of interests in the image or to detect the coordination of moving objects nearby the important regions. In the next case study, we use Answer Sets Programming (ASP) to detect complex events which provides an appropriate environment to build complex rules.

6.3 Case study 2: Complex event detection based on ASP

This case study presents a real-time complex event detection concept for resource-limited multi-sensor networks. A comprehensive solution based on Answer Set Programming (ASP) is developed. The case study discusses different examples of handling uncertainty in surveillance systems.

6.3.1 The structure of the knowledge base founded on ASP

The structure of our knowledge data base consists of:

1. **Object Entity:** The object entity has the following properties: hasId, hasType, hasZone, hasSpeed, hasDirection and a qualityRate for every attribute.
2. **Simple Event:** This is the simplest form of events, e.g. run, walk, shot, etc.
3. **Complex Event:** A complex event which is the combination of the simple events, e.g. a group of persons are running, a group of persons are fighting, a group of persons are in a forbidden area.
4. **Temporal Entity:** In ASP, time is usually represented as a variable in which the values are defined by an extensional predicate with a finite domain. A finite temporal interval in ASP can be used to reason complex events in our case study.
5. **Direction Entity:** This is defined by the calculation of the orientation angle of the object. The direction has a value between 0 and 360 as a primary feature, then in the rules, we map these values to eight possible directions, North, Northeast, Northeast East, Northwest, Northwest West, etc. (see Figure 6.7)

Each object has the following properties:

- objectId
- hasObjectType
- hasSpeed
- hasDate
- hasTime
- hasDirection
- hasCameraId
- hasFrameId
- hasX
- hasY

- hasUncertaintyType
- hasUncertaintyCorType

Each sound entity has the following properties:

- hasSoundType
- hasSoundArrayID
- hasSoundTime
- hasSoundDate
- hasSoundCorX
- hasSoundCorY.
- hasUncertaintySoundcCor.
- hasUncertaintySoundType

Each detected event whether it is simple or complex has the following features:

- hasEventType
- hasObjectType
- hasSpeed
- hasDate
- hasTime
- hasCameraId
- hasFrameId
- hasSoundArrayID

The knowledge base is used as an input to the solver in order to generate the answer sets, which present the detected simple and complex events.

6.3.2 The ASP rules

The scene context plays a major role during the detection of an event. Therefore, we define the features of the existing objects in the environment. The objects have two different types of features: sound features and video features. These features are extracted from an audio/video subsystem. Features and coordinates are extracted using object recognition and object tracking algorithms.

Here, we illustrate an example of a simple event (a human is running),

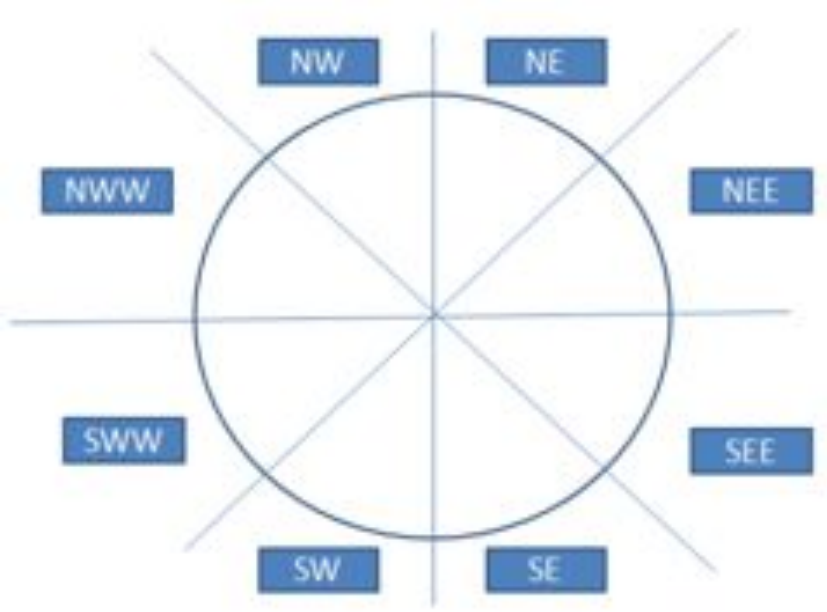

Figure 6.7: The observed directions

```
run(X,S,T,Z,D,C1,C2,FI,CI,ST,SAI,SZ,OT):-
S>5,
object(X),
hasSpeed(X,S),
hasTime(X,T),
hasZone(X,Z),
hasX(X,C1),
hasY(X,C2),
hasDate(X,D),
hasFrameId(X,FI),
hasCameraId(X,CI),
hasSoundType(X,ST),
hasSoundArrayID(X,SAI),
hasSoundZone(X,SZ),
hasObjectType(X,OT),
OT=human.
```

To detect the complex event of a group of people who are running, we check for at least two persons running whereat the distance between them is less than 3 meters. The condition of the distance is declared in the predicate $near(X_1, X_2)$, where X_1 and X_2 present the observed objects. The condition of running is declared in the predicate $run(X_1, OT)$, where X_1 is the observed object and OT is the type of the detected object. The last condition makes sure that the two observed objects are different.

```
groupPersonsRunning(X1):-
run(X1,OT1),run(X2,OT2),
near(X1,X2),
OT1=human,
OT2=human,
X1!= X2.
```

The detection of the complex event, (a group of people are running in different directions) is made when there are at least three persons near to each other and when they are moving in three different directions, e.g. $NorthwestWest, Northeast, Southeast$. The last three conditions make sure that the detected persons are not the same.

```
diffDirections9(X1,X2,X3):-
northWestWest(X1),
northEast(X2),
southEast(X3),
near(X1,X2),
near(X2,X3),
X1!=X2,
X2!=X3,
X1!=X3.
```

6.3.3 Methodological approaches used for handling uncertainty

This case study investigates the types of imprecise information that are likely to occur in the knowledge base of a surveillance system. The most important uncertainty cases are [114]: ignorance, incompleteness, inconsistency and inaccuracy, which are discussed within the context of reasoning process in surveillance system. The different types of imprecision, the different techniques for getting a quality assessment and handling the concerned information during the reasoning process will be illustrated. Finally, a method for integrating the resulting quality values of several low level features into Answer Set Programming (ASP) will be outlined.

As mentioned in Chapter 4 there are different types of uncertainty in surveillance systems:

- Ignorance: Ignorance means that there is an object in the environment of the surveillance system that is just not known. For the reasoning process of a surveillance system, e.g. the content of the knowledge base may not have the required details which are necessary for the decision process.

- Incompleteness: In contrast to ignorance, incomplete information means that there is no hypothesis related to an object or attribute value at all, e.g. the object type is known but the speed of the object is unknown.

- Inaccuracy: While uncertainty is concerned with the measure of trust that is put into the data provided by a sensing system, inaccuracy deals with the potential measurement errors that may occur.

- Inconsistency: Inconsistency means that there are conflicting hypotheses about an object data, e.g. two sensors are giving different object types with a high belief.

Bayesian approach

In modern video based surveillance systems, the detection, recognition and tracking functions are chip executed. Suppose that we have two different cameras in the cluster of a specific sensor network. An object is walking in front of two cameras; the system could recognize that there is a human in the environment. Every camera is by the success rate of object recognition; the success rate determines the probability measure for the sensed object. The two cameras are independent, camera C1: detection ratio is 98% camera C2: 95% These success rates are realistic in relation to the industrial recognition rate of smart cameras. If both cameras support the hypothesis $sucess1$ with a high probability, the overall quality measure improves. The new probability measure for the sensed object is

given by a contrary probability in the event that both cameras are wrong:

$$P(s1) = 1 - (\overline{C}1 * \overline{C}2) = 0.999 \approx 99\%$$

Now, imagine that Camera C_1 recognizes the object type T_1 with a probability of 98% and camera C_2 recognizes the "same object" with a probability of 80%. The two events are mutually exclusive, the object can either be the type recognized by C_1 or by C_2. Also, there is a chance that both cameras C_1 and C_2 are wrong and the object type is completely different. The probability for C_1 and C_2 is then given as:

$$P(success1) = 1 - (\overline{C}_1 * C_2 + \overline{C}_1 * \overline{C}_2) = 0.98 = 98\%$$
$$P(success1) = 1 - (C_1 * \overline{C}_2 + \overline{C}_1 * \overline{C}_2) = 0.95 = 95\%$$

The camera probability of C_1 is relative to camera C_2 being right. The major question is: Which camera should be trusted? If the success rate of the second system is much lower, then the choice could be to trust the camera with the higher success rate. If both success rates are high, the second hypothesis cannot be ignored. In this case, a safe way would be either ignorance or the choice of the "safer" hypothesis. When camera C_1 reports an object type as a *dog* and camera C_2 reports the same object type as a *cat*, the type has to be chosen in consideration of prior knowledge that a cats existence in the environment is less likely than a dogs. Therefore, the type of a dog can be considered. Conditional probability can be used within a surveillance system to combine the success rate of a camera with its certainty values (readability measurements), in order to improve the overall quality measure. The success rate $P(C_1)$ is a quality measure for the low level features extraction system integrated into the smart camera. If we have 2 sensing systems, they have both the same detection rate but different reliability values. Using Bayes theory, if we know $P(C1|C2)$ then $P(C2|C1)$ can easily be calculated. Therefore a system with higher conditional probability can be chosen.

The Bayesian methods have a number of advantages that indicate their suitability in uncertainty management. Most significant is their sound theoretical foundation in probability theory. Thus, they are currently the most mature of all of the uncertainty reasoning methods. Although, Bayesian methods are more developed than other uncertainty approaches, they are not flawless. They require a significant amount of probability data to build a knowledge base. Furthermore, human experts are normally uncertain and uncomfortable about the probabilities they are providing.

Certainty factors

The type of relationship between the hypothesis and evidence plays an important role in determining how the uncertainty will be handled. The reduction of these relationships to simple numbers could yield to the removal of relevant information, which might be needed for reasoning with a high success rate about the uncertainties. The main disadvantage of Bayesian method is that there are too many probabilities required. Most of them could be unknown. The problem gets worse when there are many pieces of evidence. Another major problem that appeared in surveillance systems was the relationship of belief and disbelief[2]. At first sight, this may appear trivial since obviously disbelief is simply the

[2]http://www.cs.uic.edu/ liub/teach/cs511-spring-06/cs511-uncertainty.doc

opposite of belief. In fact, the theory of probability states that $P(H) + P(H') = 1$ and so $P(H) = 1 - P(H')$. In the case of a posterior hypothesis that relies on evidence, C:

$$P(H|C) = 1 - P(H'|C) \tag{6.1}$$

Researchers have developed a MYCIN model based on certainty factors [88]. Certainty factors are a heuristic model of uncertain knowledge. In MYCIN two probabilistic functions are used to model the degree of belief and the degree of disbelief in a hypothesis; a function to measure the degree of belief MB and a function to measure the degree of disbelief MD.
MYCIN represents factual information as Object-Attribute-Value (OAV) triplets. MYCIN also associates with each fact a Certainty Factor (CF), which represents a degree of belief in the fact.

- -1 means the fact is false
- 0 means no information is known about the fact
- 1 means the fact is known to be true

MYCIN combines two identical OAV triplets into a single OAV triplet with a combined uncertainty, computed as:

$$Uncertainty = (CF1 + CF2) - (CF1 * CF2)$$

For a logical rule the calculation of uncertainty is described as follows:

$$CF(P1 or P2) = max(CF(p1), CF(p2))$$
$$CF(P1 and P2) = min(CF(p1), CF(p2))$$
$$CF(notp) = -CF(P)$$

For example, suppose we have the following rule, IF

1. The person is walking, and
2. the person has a gun, and
3. the person starts to shoot.

THEN, there is the suggestive evidence (0.7) that the identity of the person is a hunter. This can be written in terms of posterior probability:

$$P(H|C1 \cap C2 \cap C3) = 0.7$$

where the C_i corresponds to the three patterns of the antecedent.

$$P(H'|C1 \cap C2 \cap C3) = 0.3$$

For example, ten pieces of evidence might produce a $MB = 0.955$ and one disconfirming piece with $MD = 0.899$ could then give:

$$CF = 0.955 - 0.899 = 0.056$$

The definition of CF was changed in MYCIN in 1977 to be:

$$CF = \frac{MB - MD}{1 - Min(MB, MD)}$$

This softens the effects of a single piece of contradicting evidence that is combined with many confirming pieces of evidence. Under this definition with

$$MB = 0.955, MD = 0.899,$$
$$CF = \frac{0.999 - 0.799}{1 - min(0.999, 0.799)} = 0.1$$

For example, given a logical expression for combining evidence such as:

$$E = (C_1 \ \ AND \ \ C_2) \ \ or \ \ (C3 \ \ AND \ \ NOT \ \ C4)$$

the evidence E would be computed as:

$$E = max[min(C1, C2), min(C3, -C4)]$$

for values:

$$C1 = 0.8, \quad C2 = 0.6, \quad C4 = -0.6, \quad C5 = -0.2$$

the result is:

$$= max[min(0.8, 0.6), min(-0.6, -(-0.2)]$$
$$= max[0.6, -0.6]$$
$$= 0.6$$

Then, there is suggestive evidence (0.7) that the identity of the person is a hunter. Where the certainty factor of the hypothesis under certain evidence is:

$$CF(H, C) = CF(H, C1 \cap C2 \cap C3) = 0.7$$

and is also called the attenuation factor.

The attenuation factor is based on the assumption that all the evidence $C1, C2$ and $C3$ is known with certainty. That is,

$$CF(C1, c) = CF(C2, c) = CF(C3, c) = 1$$

What happens when all the evidence is not known with certainty?
In the case of MYCIN, the formula $CF(H, c) = CF(C, c)CF(H, C)$ must be used to determine the resulting CF value since

$$CF(H, C1 \cap C2 \cap C3) = 0.8 \ \textit{is no longer valid for uncertain evidence.}$$

For example, assuming:

$$CF(C1, e) = 0.6$$
$$CF(C2, e) = 0.7$$
$$CF(C3, e) = 0.3$$

Then,

$$\begin{aligned} &CF(C, c) = CF(C1 \cap C2 \cap C3, e) \\ &= min[CF(C1, e), CF(C2, e), CF(C3, e)] \\ &= min[0.6, 0.7, 0.3] \\ &= 0.3 \end{aligned}$$

The certainty factor of the conclusion is

$$\begin{aligned} &CF(H, c) = CF(C, c)CF(H, C) \\ &= 0.3 * 0.8 \\ &= 0.24 \end{aligned}$$

The CF formalism has been quite popular with expert system developers since its creation. This is due to its simple computational model that permits experts to estimate their confidence in a conclusion being drawn.

It permits the expression of belief and disbelief in each hypothesis, allowing the expression of the effect of multiple sources of evidence.

It allows the knowledge base to be captured in a rule representation, while allowing the quantification of uncertainty. Many systems, including MYCIN, have utilized this formalism and have displayed a high degree of competence in their application areas. Some studies have shown that changing the certainty factors or even turning off the CF reasoning portion of MYCIN does not seem to affect greatly the correct diagnoses. This revealed that the knowledge described within the rule contributes a lot more to the final derived results than the CF values.

Dempster-Shafer

Ignorance means that there are some things in the environment of the surveillance system that are just not known or cannot be sensed exactly. This means that the content of the knowledge base includes all required information, which are necessary for the reasoning process. For example, to stay on the topic of the sensing of object types, a sensing system may be quite sure that the current object type is an "animal", but it may not be able to tell if it is a *dog*, a *cat* or a *horse*. This form of ignorance is rather difficult to represent with Bayesian probability, which requires the hypotheses to be atomic. Also, within Bayesian reasoning the condition automatically holds that in the case of low support of a hypothesis A, support of the converse $\overline{A} = 1 - P(A)$ is automatically high, which is not necessarily true in reality [58].

If a sensing system reports an object type is a "dog" with a low probability, it will not necessarily report high probability for all other types of the object. In this case, the system knows nothing about the opposite of the hypothesis. The so-called theory of evidence, or Dempster-Shafer theory, provides a more general model for belief that overcomes the problems of the Bayesian approach.

Using the Dempster-Shafer theory, a basic belief mass m (or basic probability assignment) is assigned to a proposition. A proposition is either a single element (elementary proposition) or a set of elements from a given frame of discernment ω. It contains all valid propositions for a given real world interpretation. For example, $A = \{cat, dog\}$ would be a proposition, whereas $B = \{cat\}$ would be an elementary proposition. To assign the belief mass m to a set of propositions allows the representation and handling of ignorance. Additionally, it is possible to support a proposition with a very low belief. For example, we can support the proposition of $A = \{cat\}$ with 0.3. This does not mean that A is supported with 0.7. On the contrary, it is a non-commitment to the remaining propositions of the frame of discernment. We are sure that it is unlikely that an object type *cat* exists in the environment, but beside that it can be everything from the given frame of discernment. So, the remaining belief of 0.7 is assigned to ω. The overall belief in a proposition A is determined as the sum of all evidence supporting the proposition:

$$Bel(A) = \sum_{B \subseteq A} m(B) \quad where \quad B \quad is \quad the \quad evidence$$

The belief $Bel(A)$ gives a measure of the extent to which a proposition is definitely supported. However, the remaining evidence does not have necessarily disprove the proposition. A second measure, the plausibility $Pl(A)$ of a proposition is determined to indicate

the extent to which the given evidence fails to refute a proposition:

$$Pl(A) = 1 - Bel(\overline{A})$$

$Bel(A)$ and $Pl(A)$ give an interval for the credibility of an object, with $[1, 1]$ representing absolute certainty and $[0, 1]$ representing total ignorance. There is no definite support for a proposition, but the proposition is a completely plausible belief assignment from different independent sources $m1$ and $m2$ can now be combined using Dempster's rule of combination:

$$m1 \oplus m2(Z) = \frac{\sum\limits_{X \cap Y = Z} m1(X).m2(Y)}{1 - \sum\limits_{X \cap Y = \phi} m1(X).m2(Y)}$$

To continue the example with object types, sensing system A gives the following belief measures:

$$m_A = (\{cat, dog\}) = 0.98$$
$$m_A = (\{horse\}) = 0.02$$

This means, sensing system A is sure that the object type is either a *cat* or a *dog*. However, it does not completely rule out the possibility of the object type being a horse. Sensing System B gives the following belief measures:

$$m_B(\{dog, horse\}) = 0.7$$
$$m_B(\{\omega\}) = 0.3$$

System B is quite sure that the object type is a *dog* or *horse*. The system does not support any further belief information, so the remaining 0.3 are assigned to the frame of discernment. Combining the belief assignments $m1 \oplus m2$ gives revised belief assignments to the propositions:

$$m_{1,2} = (\{dog\}) = 0.67$$
$$m_{1,2} = (\{cat, dog\}) = 0.29$$
$$m_{1,2} = (\{horse\}) = 0.02$$

For every proposition, the belief and plausibility measure can now be calculated:

$$Bel(AB) = (\{dog\}) = 0.67$$
$$Pl(AB) = (\{dog\}) = 0.67 + 0.29 = 0.96$$
$$Bel(AB) = (\{cat, dog\}) = 0.67 + 0.29 = 0.96$$
$$Pl(AB) = (\{cat, dog\}) = 0.67 + 0.29 = 0.96$$
$$Bel(AB) = (\{horse\}) = 0.02$$
$$Pl(AB) = (\{horse\}) = 1 - 0.98 = 0.02$$

The credibility intervals for the given propositions are:

$$\{dog\} : [0.67, 0.96]$$
$$\{cat, dog\} : [0.96, 0.96]$$
$$\{horse\} : [0.02, 0.02]$$

If the type of the object is known, but some attribute values are missing, default values are usually used for reasoning. For a surveillance system, the default-value approach has to be applied with care. While it is suitable for some attributes, there are also values which cannot be substituted with a default when missing. Also, depending on the type of the missing information, distinguishes how to deal with its absence.

Using default values:

Belief measures should not be confused with a probability measure. While probabilities usually have some numerical base from which they are obtained, for example a number of test runs to obtain a success rate, a belief is a purely subjective measure, similar to a subjective probability. It is a reasonable assessment given by a knowledgeable agent, without any numerical or statistical basis [58].

- Default value can be used if available.
- Default can be calculated from history values: Z-score.
- Default can be defined if from prior knowledge.
- A plausible value can sometimes be obtained with mathematical methods like Dempster-Shafer theory.

A simple example can demonstrate this. Let us assume that the current speed of an object is $(8m/s)$, the initial distance to the front object is 10 meters and the speed of the front object is unknown. After one second, the distance has decreased to 2 meters. Thus, the speed of the front object can be derived as $6m/s$ [58].

Fuzzy Logic

Fuzzy logic is found to handle the concept of partial truth- truth values between "completely true" and "completely false". It is the logic underlying the modes of reasoning which are approximate rather than exact. Therefore, fuzzy logic [90] handles the problem of representing the ambiguity of concepts. Suppose we have different speeds of people s that are in an environment of a surveillance system. There are values in S which are not high speed and there are values which are in the ranges between middle and high speed. To each speed in the universe of discourse, we have to assign a degree of membership in the fuzzy subset $high \;\; speed$. The easiest way to do this is with a membership function based on the person's speed. Then, every member is assigned a membership degree to S from the interval $[0, 1]$. The membership function S of a fuzzy set is formally defined as:

$\mu_H : S \rightarrow \quad [0, 1]$. In our example, we can define the membership function as follows:

$$highspeed(x) = \{1, if speed(x) <= 10,$$
$$(20 - speed(x))/10, if 10 < speed(x) <= 20,$$
$$0, if speed(x) > 20\}$$

If the system is providing the following values of speeds in the observation area (see Table 6.1), then the degree of membership can be calculated:

Table 6.1: The speed's values provided by the surveillance system

Object ID	Speed	Degree of Membership
O1	5	1
O2	11	0.9
O3	15	0.5
O4	19	0.1
O5	25	0

So given this definition, we would say that the degree of truth of the statement that $O3$ has a high speed" is 0.5. Furthermore, if you have an event *abnormal* described as "if a person is running and shouting or shooting, then the system can detect an abnormal state". Consequently, using the fuzzy operators[3], it is possible to detect a specific degree of truth regarding the event is "abnormal":

- **Union:** The membership function of the union of two fuzzy sets C and D with membership functions, is respectively defined as the maximum of the two individual membership functions. This is called the maximum criterion (equivalent to "OR" operator).
- **Intersection:** The membership function of the intersection of two fuzzy sets C and D with membership functions, is respectively defined as the minimum of the two individual membership functions. This is called the minimum criterion, (equivalent to "AND" operator).
- **Complement:** The membership function of the complement of a fuzzy set A with membership function is defined as the negation of the specified membership function. This is called the negation criterion.

In the proposed example there is a degree of membership for every state exemplified as:

$$person(x) = 0.6$$
$$shooting(x) = 0.2$$
$$shouting(x) = 0.7$$

[3]www.doc.ic.ac.uk

The answer to the statement "if a person is running and shouting or shooting, then the system can detect an abnormal state" would be,

$$abnormal(x) = person(x)\ AND\ (shooting(x)\ \ OR\ \ shouting(x) = 0.7)$$
$$abnormal(x) = Min(0.6)\ \ OR\ \ (Max(0.2, 0.7))$$
$$abnormal(x) = 0.6$$

So given this definition, we would say that the degree of truth of the statement that $abnormal(x)$ has a high speed" is 0.6. There are different membership functions that can be used, e.g. Bell-shaped function, Gaussian function, triangular function and Trapezoidal function [115].

6.3.4 Simulation environment and parameter settings

For the feature extraction module, we developed a framework which was implemented in $C++$ and OpenCV. OpenCV is an open source computer vision library from Intel. The system has been tested in two scenarios: the first one is on the highway and the second one is in a parking place. It is running under Linux version 2.6.24-22-generic, Ubuntu 4.2.3-2ubuntu7, *gcc* version 4.2.3. The CPU used is an Intel(R), Core(TM) 2 Duo CPU 2.00 GHz, cache size is 2048 KB. The representative requirements for video processing in our scenario are: the frame size of the camera is 640*480 pixels and frame rate 30frames/sec. 24 test cases have been examined for the recognition of cars, dogs and humans.

6.3.5 Performance results obtained and related comments

To compare the run-time behavior of the previous Semantic Web model and the Answer Set Programming (ASP) approach, we performed several tests on an embedded platform that will also be used in our case study project. We use Atom-based embedded boards as example platforms. We tested all algorithms on pITX-SP 1.6 plus board manufactured by Kontron[4]. The board is shown in Figure 6.8. It is equipped with a 1.6 GHz Atom Z530 and 2GB RAM.

For Semantic Web, *Protege*[5] as editor was used for the ontologies and Semantic Web Rule Language (SWRL) rules. Jess[6], Pellet[7] and Jena[8] as rule engines are used for evaluating.

The Ontology Web Language (OWL) ontology and the knowledge base of ASP consist of 30 instances of video features and 18 instances of audio features. We defined 42 Semantic Web Rule Language (SWRL) and SPARQL rules and 48 ASP rules.

We use *iClingo*[9] as a solver of ASP [80]. It is an incremental ASP system implemented on top of clasp and Gringo. *iClingo* is written in *C* and can be run under *Windows* and *Linux*.

We measured the execution time of the Semantic Web implementation and the ASP solver on our embedded platform. Table 6.2 shows an overview of the execution time. It can be seen that the technology of ASP is far more suited for embedded operation than the Semantic Web solvers. In our project, this means that the complex event detection can be executed once or twice a second which enables the audio/video subsystem to collect sufficient data for detecting complex events. The detection of different events or situations has become an important topic in audio and video surveillance systems in the recent years. In this work, we have demonstrated the advantages and disadvantages of the most important technologies. We have also shown that the use of Answer Set Programming (ASP) can significantly reduce the effort needed to detect complex events while obtaining the same level of quality in the detected events. ASP is expressive, convenient and supports formal declarative semantics. We showed that ASP can be used

[4]http://www.kontron.com
[5]http://protege.stanford.edu/
[6]http://www.jessrules.com/
[7]http://clarkparsia.com/pellet/
[8]http://jena.sourceforge.net/
[9]http://potassco.sourceforge.net

Figure 6.8: The pITX-SP hardware platform used in our tests

	Semantic Web	ASP
Average [s]	111	0.40
Minimum [s]	108	0.39
Maximum [s]	114	0.46

Table 6.2: Execution time measurements

to detect a large number of simple and complex events within a reasonable time frame that allows real-time operation. We proved that ASP is an appropriate solution for complex event detection systems in multi sensory networks with limited resources. In our future work, we will use rule decision systems, which generate decision rules based on decision tables. By using Rough-set theory and genetic algorithms, we integrate the generated rules in ASP for detecting events where it is not possible to describe the related behavior.

6.4 Summary

In Chapter 5 different approaches are proposed for the detection of complex events in multimedia sensor networks, which can be either explicit or implicit. Explicit event detection requires the definition of different rules and training, whereas implicit event detection does not make use of any rules and creates the models automatically. Ontologies can be used for rule based systems to:

- describe the relations between entities of the environment, the states and the rules of the environment.
- help designing context models that represent, manipulate and access context information.

The context information should include the spatial and temporal information to detect short and long term complex events correctly. The first step in building a context model is to specify the desired system behavior. For an interactive environment, this corresponds to the environmental states defined in terms of the variables to be controlled by the environment and predicates that this should be maintained as true (rules). For each state, the designer lists a set of possible situations, whereby each situation is a configuration of entities and relations to be observed. Although a system state may correspond to many situations, each situation must uniquely belong to one state [116].

After building the context model, context reasoning extends context information implicitly by introducing deduced context derived from other types of context. It is a perfect solution to resolve context inconsistency and to make the inference over the defined rules to detect complex events [117]. Usually, complex knowledge must be learned especially with different procedural constructs and uncertainties. Such expert systems should be able to integrate information of learning and reasoning methods for building robust universal systems.

Chapter 7

Emotion recognition using human voice features

Driver fatigue, stress and drowsiness cause traffic accidents. Road accidents are more frequent than accidents in other transportation modes (air, sea and railways). Safety can be improved through the design of the vehicles and monitoring the behavior of the road users. Research in surveillance systems in the frame of modern driver assistance systems is increasing and the number of publications in the last 10 years has also been increased.

Driver monitoring plays a vital role in assessing, controlling and predicting the driver's behavior. The research concerning driver monitoring systems has been ongoing since the 1980s [118]. Firstly, the requirements of acoustic emotion detection systems will be explained. Then, uncertainty and its origins in emotion recognition systems will be illustrated.

In this Chapter, a comprehensive solution based on the Bayesian Quadratic Discriminant classifier (BQD) is developed. The developed system supports Advanced Driver Assistance Systems (ADAS) to detect the mood of the driver based on the fact that aggressive behavior on road leads to traffic accidents. We use only 12 features to classify between 5 different classes of emotions. We illustrate that the extracted emotion features are highly overlapped and how each emotion class is affecting the recognition ratio. Finally, we show that the BQD classifier is an appropriate solution for emotion detection systems, where a real-time detection is deeply needed with a low number of features.

7.1 Basic concepts related to emotion and its involvement in technical systems

The different types of emotion recognition and monitoring systems have been designed with the aim of increasing human-machine interaction. Usually, those systems are used for psychological analysis in clinics, robotic systems and ADAS.

7.1.1 What is emotion?

The word emotion includes a wide range of observable behaviors, expresses feelings, and changes in the body's state[1].

[1]http://library.thinkquest.org

Emotional states can be defined by a variety of changes in the chemical profile of the body due to changes in the condition of viscera and by changes in the contraction of various striated muscles of the face, throat, torso and limbs.

However, emotions are defined by alterations in the neural structures that cause these changes and also other significant changes in the state of several circuits in the brain itself. An emotion can be simplified as a specifically caused transient change in the condition of the body.

Psychology science has a long focus on negative emotions and their effects. Emotions are characterized by the quality of their subjective experience, which usually can be described as a counterpoint to cognition through various dimensions: direction (pleasant or unpleasant), quality (content of experience or attention or rejection), extent of activation and awareness. The intensity describes how much the person is excited. CE Izard (1981) identifies three levels of behavior to describe emotions and to define subjective experiences, the neurophysiological processes and observable behavior expressions. She assumes that emotions have a physical, a mental and behavioral controlling component.

Negative emotions, such as fear or anger, are helpful in some situations. Thereby, people have learnt to survive, recognize and avoid dangers. Stress also fulfills a useful function as an active life requires a certain level of stress. Even Darwin supported his theory of emotions through the observations of similarities in the emotional expressions of humans and animals. His conclusion is based on the observation of people's emotions from different parts of the world and stated that the emotion-specific expression is universally distributed.

Some psychologists have tried to subdivide emotions in categories. For example Wilhelm Wundt, the great nineteenth century psychologist, offered the view that emotions consist of three basic dimensions, each one a pair of opposite states: pleasantness-unpleasantness, tension-release and excitement-relaxation. However, this list has become more complex over time.

Plutchik suggests that there are eight basic emotions grouped in four pairs of opposites: joy-sadness, acceptance-disgust, anger-fear and surprise-anticipation[2]. Figure[3] 7.1 illustrates a very basic example of four primary emotions and their related states.

A baby knows the feelings of anxiety in the womb before birth, because it has learned to struggle, to move and suck its thumb. They do not know only their body. They know the voice of their mothers and fathers, their favorite song and favorite music and know the smell of their mother.

Unborn children can have experiences in the womb which make them later prone to anxiety. For example, if the mother is afraid of the father, a baby can feel this. Furthermore, children can hear the rapid heartbeat and the loud voice of the father. This experience is stored in the brain. After birth, the child falls into a torpor, when the father's voice has the same tone.

[2]http://library.thinkquest.org

[3]http://www.psychologyofmen.org/index.php?itemid=35

Figure 7.1: A very basic example of four primary emotions and their related states

7.1.2 How far is emotion detection important in a variety of technical systems?

The second class of approaches directly measures human physiological characteristics but in an intrusive way by involving measurement systems such as the Electroencephalogram (EEG) which monitors brain activities [119].
Figure[4] 7.2 shows an example of EEG. This is a procedure that records the brain's continuous electrical activity by means of electrodes attached to the scalp.

Figure 7.2: Electroencephalogram (EEG) - a procedure that records the brain's continuous electrical activity by means of electrodes attached to the scalp

Other authors used an Electrocardiogram (ECG) which measures heart rate variation, an Electrooculogram (EOG) which monitors eye movement and a skin potential level

[4]http://www.kernneuro.com/index-5.html

measurement technique. [120]. These approaches are accurate but they need electrodes that are attached directly to the human body.

And more recently, significant research has been focusing on developing non-intrusive techniques. Generally, these non-intrusive approaches involve machine vision as an alternative to a direct measurement of physiological characteristics and they do not need any cooperation from the human [121]. For emotion recognition applications, the machine should be as intelligent as a human brain to sense emotions. Humans can recognize and detect emotions by observing other people's action, speech and body language. The mental and physiological state is associated with a wide variety of feelings. When emotions change, muscle properties in the face, voice and body change. Consequently, by observing these muscle movements, human can analyze emotion.

Researchers have been studying several psychology aspects and computational aspects of emotion. However, they have still no clear idea about how to estimate emotion and how to differentiate emotions from each other. According to psychology science, emotions are classified into two models. The first one is discrete. This model categorizes emotions as entities with names and descriptions. The second model is a continuous model [122].

Technical based emotion means the conceptual knowledge from the psychology is translated into features that help a machine to recognize emotions using machine learning techniques. The most well known study on facial emotion recognition is the Facial Action Coding Systems (FACS). These systems identify the changes in a facial image by observing the facial muscles. For facial movement analysis, automatic classifiers for 30 facial actions (including the motions of blinking and yawning, as well as a number of other facial movements) from the FACS have been classified by using machine learning [123]. The speaker emotion recognition techniques are mainly classified into three categories:

- The long term averages of the acoustic features, like pitch or spectrum representations.
- Speaker dependent based on the utterance based features.
- Neural network based approach.

A body gesture based emotion recognition system tracks the body and hands of the subjects. It has been developed using different approaches. Furthermore, they observe temporal series of the selected motion cues over time, depending on the video frame rate. They apply several statistical moments to extract features that can be classified using machine intelligence techniques [124].

Multimodal emotion recognition is also used by different researchers using facial expression, body gesture and acoustic analysis.

7.1.3 Why consider emotion detection as a particular event detection?

Biometric surveillance is a technology which consists of several approaches that measure and analyze human physical behavioral characteristics for authentication, identification or screening purposes [125].

A facial thermograph is a technology which allows machines to identify certain emotions in people, such as fear or stress. Therefore, emotion recognition systems could be used to

identify if a suspect is nervous what might indicate that a suspect is hiding something, lying, or worried about something. A thermograph or a thermal camera can also be used to extract features from people's faces.

An important application of emotion recognition systems is the specification of a driver mode during driving to reduce the occurrence of accidents. Driver behavior is recognized as one of the main factors in the cause of accidents. The National Highway Traffic Safety Administration (NHTSA) estimates that in the USA approximately 100,000 crashes each year (resulting in more than 1,500 fatalities and 71,000 injuries) are caused primarily by driver drowsiness or fatigue[5]. Therefore, it is important to control, record and monitor a driver's status during driving.

In some applications, it may not be an important process for computers to recognize emotions, for example a surveillance system deployed in an airport. In some applications where computers take on a social role, such as an "instructor," "helper" or even "companion", it may enhance their functionality to be able to recognize users' emotions [126].

In interactive learning systems the recognition of the user's emotions helps computers to become a more effective tutor. Synthetic speech with emotions in computer "agents" could learn the user's preferences through the users' emotions.

Another application is to help the human to control their stress level. In clinical settings, recognizing a person's inability to express certain facial expressions may help experts to detect early psychological disorders of patients [126].

In video surveillance systems emotions are important to support the reasoning process, for example if there is a danger in the environment of the observation area. The detection of the danger can be highly accurate if we relate the emotions of people with other features that might be extracted from other sensor types.

7.2 The requirements of acoustic emotion detection systems

Emotion detection systems gather participants' emotions as well as proximity and patterns of driver speech by processing the outputs from the sensors. The sensors of emotion recognition systems should be low-cost, low-failure and should be connected in a feasible way. Thus, the emotion detection system can be used to understand the correlation and the impact of interactions and activities on the emotions and behavior of individuals.

The key objective of the following requirements is to develop and validate a robust and low-cost non-intrusive system capable of reliably measuring all parameters needed for recognizing the emotion of the driver during the driving process. Figure 7.3 shows an overview of the requirements of human speech emotion recognition systems.

- **Non-intrusiveness:** The measurement systems must be non-intrusive, which means there is no need of cooperation from the driver side.
- **Robustness and Reliability:** The audio surveillance system should be as reliable as possible to recognize the driver's state and robust to compensate sensor failures (because of the noise).

[5]www-nrd.nhtsa.dot.gov/Pubs/TSF2005.PDF

- **Low-cost and Feasibility:** The sensors selected for the emotion detection system should be low-cost and should be connected in a feasible way.
- **Efficient Inference Approach:** Efficient reasoning approaches are required to be exploited to extract high-level information from the available raw data of not always accurate sensors embedded in mobile phones.
- **Low-power Consumption:** An efficient system for this class of resource-constrained device, especially in terms of power consumption, needs to be devised.
- **Easily Programmable:** The emotion detection system should be easy to program and customize for different types of experiments regarding the change of requirements.

Figure 7.3: The overall requirements of human speech emotion recognition systems

7.3 Origin of uncertainty in human voice based emotion detection systems

Uncertainty can be handled in video surveillance systems and audio surveillance systems using similar approaches.

There are different types of uncertainty. The first type is uncertainty in prior knowledge, e.g. some causes of an event are unknown and are not represented in the knowledge base of the audio surveillance system. Another type is uncertainty in the model, e.g. models could be effected by noise and the noise is possibly represented in the model. Therefore,

the model has a margin of error where the decision is not always true. Finally, there is uncertainty in perception, e.g. sensors do not return exact or complete information about the world; a system never knows its position exactly.

Now, there is a major question to be asked: How does one deal with uncertainty in audio surveillance systems? The answer consists of two main approaches; the implicit approach and the explicit approach. In the implicit approach we can deal with uncertainty by building procedures that are robust to uncertainty. The explicit approach deals with uncertainty by building a model of the world to describe uncertainty about its state, dynamics and observations. Then, we reason the effect of actions given the model.

The difficulty in emotion recognition in people's audio streams is the lack of an affect-related semantic and syntactic knowledge base. There are different forms of uncertainty related to emotion detection [127]:

- It is quite difficult to define what emotion means [128].
- Long-term and short-term transitions of emotional states [129].
- How to determine the features that influence the recognition of emotion in speech [130]?
- Which classifiers must be used [131]?

It is still a major challenge to detect emotions through acoustic emotion detection systems because of the different resources of uncertainty [132] [133], e.g.

- Sickness
- Language
- Noise
- Ambiguity in emotional keywords

Usually, it is possible to reason uncertainty using three types of uncertainty. These are default reasoning, worst-case reasoning and probabilistic reasoning. By default reasoning we assume that the world is fairly normal. Abnormalities are rare. Therefore, an agent assumes normality, until there is an evidence of the contrary.

Worst-case reasoning is exactly the opposite of default reasoning. The world is ruled by Murphy's Law which means that uncertainty is defined by sets, e.g. the set possible outcomes of an action, the set of possible emotions in a continuous form of speaking. The surveillance system assumes the worst case and chooses the actions that maximizes a utility function in this case.

In probabilistic reasoning, we assume that the world is not divided between "normal" and "abnormal", nor is it adversarial. Possible situations have various likelihoods (probabilities). The agent has probabilistic beliefs, pieces of knowledge with associated probabilities or "strengths." Through this it chooses its actions to maximize the expected value of some utility function. The previous types are comprehensively explained in Chapter 4.

In the frame of facial emotion detection surveillance systems, face recognition is the first step in many human-computer interaction systems, e.g. expression recognition and

cognitive emotional state recognition. For example, in order to detect the emotions of the driver's face in ADAS, the driver's face must be detected first. In ADAS the emotion of the driver can be detected using facial or acoustic features or both. The origins of uncertainty for face detection are:

- The rotation of the face or frequent movements of the face up and down.
- The presence of beard, mustache, glasses etc.
- Occlusions because of long hair or hands.
- In-plane rotation.

Other origins can be considered in the size of images, lighting conditions, distortion, noise and the compression of images after capturing.

7.4 General limitations of the related state-of-the-art in human voice based emotion detection

In the field of ADAS, there are different types of driver monitoring systems in the first stages of this research. Researchers developed an approach to driver monitoring systems based on inferring both the driver's behavior and state from the observed or measured vehicle performance.

However, these approaches are strongly depended upon vehicle conditions, e.g. steering wheel movements, vehicle lateral position, lane change, speed variability, breaking, gear changing and reaction time [134], and road conditions, e.g. quality of lane markings, alternate lane markings during road repairs, as well as on environmental conditions, e.g. shadow, rain and night vision [135].

The main limitation is that they cannot help assessing the mood, the emotion and the stress state of the driver. These drawbacks have drawn the researcher's interest to monitoring the driver's behaviour directly. Thus, a second set of approaches has been created that directly measure the driver's physiological characteristics but in an intrusive way by involving measurement systems such as the Electroencephalogram (EEG), which monitors brain activities [136]; the Electrocardiogram (ECG), which measures heart rate variation; the Electrooculogram (EOG) which monitors eye movement; the skin potential level measurement techniques, etc.

In the field of emotion recognition systems, many researchers used Gaussian Mixture Model (GMM), Support Vector Machines (SVM), Multilayer Perceptron (MLP), and decision trees [137] [138] [139]. Furthermore, a base-level classifier may not perform well on all emotional states. For example, a GMM-based classifier may fail to correctly recognize a neutral emotion, while the MLP-based classifier shows its superiority on neutral emotion recognition. SVM can also fail because of the mix between the extracted features for different types of emotions.

Other researchers used hyper classifiers together [132] [137]. Three classifiers consisting of GMMs, SVMs and MLPs obtained results that do not show a high improvement in emotion detection in relation to other works in this field. Those approaches use over 30 features for emotion recognition in human speech [140] [141] [142].

In our approach we use only 12 features to classify between 5 types of emotion: afraid, normal, sad, angry and happy. Although for driver monitoring in ADAS the emotions normal, angry and happy could be the most important.

7.5 Specific limitations of the state-of-the-art of human voice based emotion detection while considering uncertainty

Regarding emotion recognition in emotion detection systems, there are different ways to handle uncertainty. Well known approaches are the probabilistic approaches for facial emotion recognition in video sequences. Typically, these probabilistic approaches have two main steps.

The first step is the selection of the representative features from the raw video that are extracted from references videos. The second is the use of the collected samples as centers for probabilistic mixture distributions for the tracking and recognition process. Probabilistic approaches allow a systematic handling of uncertainty.

The authors in [143] use a distance function d to measure the uncertainty in a recognition process. It assures at the same time that enough exemplars for a successful recognition under a variety of conditions are generated.

However, there are different origins of uncertainty in the frame of facial recognition. These typically arise due to the variations in the conditions of capturing the face images of a person as well as the variations in the personal information such as age, race, sex, expression or mood of the person at the time of capturing the face image. Authors in the fields of the fuzzy-geometric approach and symbolic data analysis for face recognition are considered for the modeling of uncertainty of information about facial features [144].

Another approach is the use of conditional regression forest. The regression forest learns the relations between facial image patches and the location of feature points from an entire set of faces.

In general, regression forests learn the probability over the parameter space given of a face image from the entire training set, where each tree is trained on a randomly subsampled training set to avoid over-fitting. The authors [145] are handling uncertainty by seeking to maximize the discriminative power of the tree. By maximizing the power of the tree, the class uncertainty for a split is minimized [145].

With reference to acoustic features, the speech of human is not a stationary signal. It has properties that change over time. Therefore, a single representation based on all the samples of a speech utterance is generally not robust to recognize emotions.

Instead, researchers define a Time Dependent Fourier Transform (TDFT) and Short-Time Fourier Transform (STFT) of speech that changes periodically as the speech properties change over time. The STFT is a fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time [146].

Authors in [147] built a Minimum Mean Square Error (MMSE) log-filterbank energy estimator for environment-robust automatic speech recognition. However, MMSE estimators of non-linear speech transformations are better as they combine MMSE with MFCC to reduce noise for robust speech recognition [148].

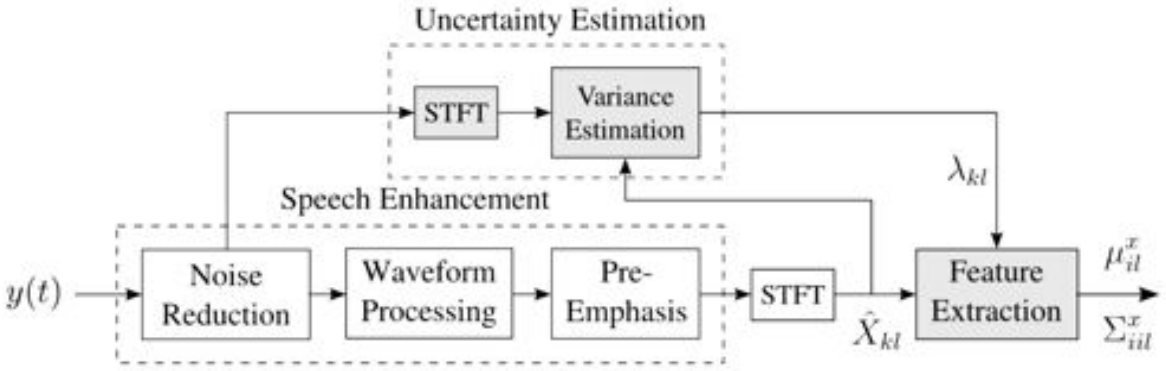

Figure 7.4: Uncertainty decoding for human speech noise reduction [149]

Furthermore, the combination between uncertainty propagation with observation uncertainty techniques can be applied to a realistic realization of robust distributed speech recognition to improve recognition robustness [149].
Figure 7.4 shows the concept of uncertainty decoding for human speech noise reduction. The disadvantage of their approach is that it increases the computation time.

The main disadvantages of the previous approaches are that they have a high computational time. The noise reduction approaches can effect the quality of features needed for emotion recognition. Even filtering out the ambient noise suffers from the same problem. Generally, acoustic based detection systems can perform well without noise. However, in the case of noise reduction it can affect the quality of low-level features.

The previous illustration considered the acoustic uncertainty in the level of features extraction. Uncertainty management concepts and their limitations are discussed in Chapter 4 in details.

7.6 Case Study: a real-time emotion detection system for advanced driver assistance systems

In this section, an overall architecture of the emotion detection system will be proposed. The Berlin emotional speech database is used to classify discrete emotions. This publicly available database is one of the most popular databases used for emotion recognition, thus, facilitating comparisons with other works. Ten actors (5m/5f) each uttered 10 everyday sentences (five short and five long, typically between 1.5 and 4 s) in German; sentences that can be interpreted in all of the seven emotions acted. Further, the raw database is evaluated by a subjective perception test with 20 listeners. In total, we extracted 12 features from each sample:

- The minimum, the maximum, the mean and the median of the energy.
- The minimum, the maximum, the mean and the median of the pitch of the signal.
- The minimum, the maximum, the mean and the median of the Mel-Frequency Cepstral Cofficient (MFCC) of the signal.

To extract the features, we used the statistical moments (minimum, maximum, mean and median) of 3 features (MFCC, Pitch and energy) and then for the classification,

Class	Class symbol
afraid	1
normal	2
angry	3
sad	4
happy	5

Table 7.1: The defined classes of the proposed emotion recognition system

we used a Bayesian Quadratic Discriminant (BQD) classifier. In this demo and this research question, the aim is to show that a speech emotion recognition system will be useful to understand the state and emotions of a driver to increase safety and control the car autonomously. Table 7.1 shows defined classes of the proposed emotion recognition system.

7.6.1 Overall systems requirements

The designed system considers drivers of vehicles. Usually, during recording their voices using audio sensors, like a microphone. The recorded data may be affected by noise due to the weather conditions or any other disturbances. For the reduction of the noise a filter operation is performed with a high pass filter.

Furthermore, during driving the engine causes a noise that has to be separated from the driver's voice. Also, the voice of the driver has to be separated from the voices of other persons during driving. Therefore, the system performance can be highly affected by noise and thus, the driver's voice has to be identified.

The voice mixture contains the co-passengers voice, the motor or vehicle noise, the environmental noise, the entertainment system voice along with the driver's voice. The only prior knowledge we have is the "driver voice". So we should find a method that is capable of using only this knowledge and separate the driver's voice from the mixture of voices.

In a driving situation, with this prior knowledge (driver voice), the GMM based voice separation technique and non-negative features based technique can be used to separate the driver's voice.

In the GMM model parameters (the covariance matrices and the prior weights, are estimated by using training samples from the each speaker or source.

The GMM is used to characterize the speaker's voice by a set of acoustic classes. In GMM a group of speaker can be chosen and using GMM the posterior probability can be calculated. The speaker with maximum probability is the identified speaker.

The use of Support Vector Machines (SVM) is one of the most popular techniques in speaker recognition. SVM requires training data for both the speaker and others. SVMs are used for recognition of both discrete and continuous emotions. While support vector classification finds the separation hyperplane that maximizes the margin between two classes, support vector regression determines the regression hyperplane that approximates most data points with precision. The SVM implementation is adopted with the Radial Basis Function (RBF) kernel employed.

The design parameters of a SVM are selected using training data via a grid search on a

base logarithmic scale. According to literature, a SVM performs better than the existing speaker recognition techniques [150].

Additionally, the designed system has to consider that the measurement system must be non-intrusive, which means there is no need of cooperation from the driver. Also, it has to be cost effective and capable of running on embedded platforms. Real-time response is very important because during driving, a simple delay in the ADAS can cause a horrible accident.

7.6.2 System engineering details

Systems engineering techniques are used to ease the design of complex systems. The proposed emotion recognition system has different system engineering requirements. During the development of the emotion recognition systems the following tools and methods have been used to better comprehend and manage complexity in systems:

- **System Architecture:** The conceptual model that defines the structure of the emotion recognition system is built for the specification of the behavior of the system and the desired output of the system.
- **Optimization:** The optimization is applied in the emotion detection system to find the best feature space that can increase the performance of the system and decrease the complexity.
- **System Analysis:** The main advantage of system analysis is the specification of challenges for every step of the design. Regarding the emotion recognition system the noise has the major challenge. During driving the engine causes a noise that has to be separated from the driver's voice. Furthermore, the voice of the driver has to be separated from the voices of other persons during driving. Therefore, the system performance can be highly affected by noise and thus, the driver's voice has to be identified.
- **Performance Analysis:** In signal detection theory, a Receiver Operating Characteristic (ROC), or simply ROC curve, is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting the fraction of true positives out of the positives vs. the fraction of false positives out of the negatives at various threshold settings[6]. The ROC curve is used to measure the overall performance of the recognition system.

The previous steps helped to design the emotion recognition system with respect to the different challenges in the frame of ADAS where the real-time response of the system is the major requirement.

7.6.3 System training concept and involvement of the Berlin Database of Emotional Speech (BDES)

Below is an explanation of the Berlin Database of Emotional Speech (BDES)
The Berlin emotional speech database is developed by the Technical University, Institute

[6]http://en.wikipedia.org

for Speech and Communication, Department of Communication Science, Berlin. It has become one of the most popular databases used by researchers on speech emotion recognition. Thus, facilitating performance comparisons with other studies. 5 actors and 5 actresses have contributed speech samples for this database. These mainly consist of 10 German utterances, 5 short utterances and 5 longer ones and recorded with 7 kinds of emotions: happiness, neutral, boredom, disgust, fear, sadness and anger. The sentences are chosen to be semantically neutral. Therefore, they can be readily interpreted in all of the seven emotions simulated. Speech is recorded with 16 bit precision and 48 kHz sampling rate (later down-sampled to 16 kHz) in an anechoic chamber[151] [152].

The training phase of the system consists of the following steps:

1. Download the emotion sentences from the official database of BDES.
2. Sort the sentences with respect to every emotion type.
3. Create folders for every emotion type.
4. Extract the features from every emotion type.
5. Save the features in a specific format.
6. Apply different classifiers and check the performance with respect to different feature spaces. The feature spaces have been defined based on statistical analysis and principal components analysis.

The prototype of the system is written in MATLAB[7]. MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks[8], MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran.

The signal processing toolbox of MATLAB is used for training and PRTools[9] is used for classification. PRTools is a Matlab Pattern Recognition Toolbox for representation and generalization.

Signal processing toolbox provides industry-standard algorithms for analog and Digital Signal Processing (DSP). It can be used to visualize signals in time and frequency domains, compute FFTs for spectral analysis, design FIR and IIR filters, and implement convolution, modulation, resampling, and other signal processing techniques. Algorithms in the toolbox can be used as a basis for developing custom algorithms for audio and speech processing, instrumentation and baseband wireless communications[10].

7.6.4 Feature extraction concepts

Features are extracted from the real-time data by performing time and frequency domains algorithms. These algorithms extract temporal and spectral features. These features are extracted based on the amplitude and spectrum analyzer of the audio data.

[7]http://www.mathworks.de/products/matlab/
[8]http://www.mathworks.de/
[9]http://prtools.org/
[10]http://www.mathworks.de/products/matlab/

After windowing, we performed the feature extraction methods for estimating the acoustic features that are mostly used in emotion detection.

Short Time Energy (STE)

The size of a signal is important for different applications. We define the signal energy as the area under the squared signal [153].

$$STE = \frac{1}{N}\sum_{n=0}^{N-1}|X(n)|^2$$

Where N describes the total number of samples in a frame or a window, X (n) is a speech signal in a frame.

Pitch Extraction

Fundamentally, this algorithm exploits the fact that a periodic signal, even if it is not a pure sine wave, will be similar from one period to the next. This is true even if the amplitude of the signal is changing in time, provided those changes do not occur too quickly. A pitch detector is basically an algorithm which determines the fundamental period of an input speech signal. Pitch detection algorithms can be divided into two groups: time-domain pitch detectors and frequency domain pitch detectors [154]. Figure 7.5 shows the frequency vs. the pitch[11].

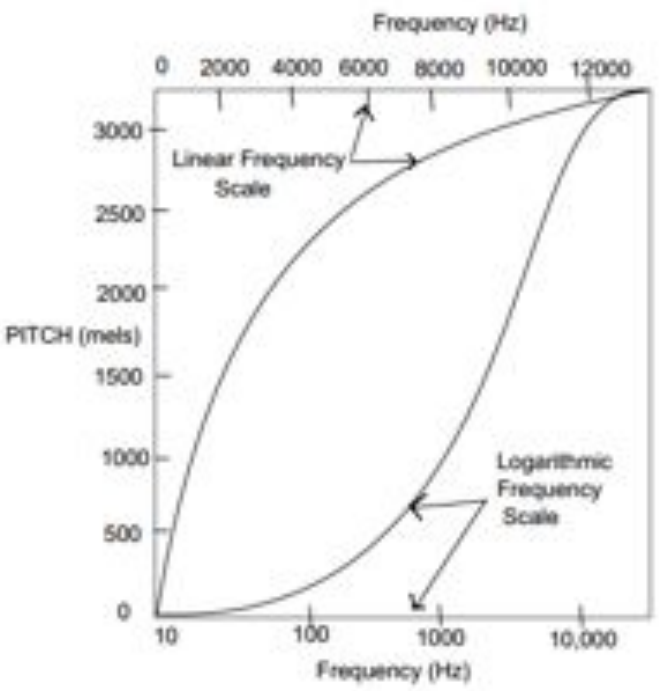

Figure 7.5: The frequency vs. the pitch

The frequency domain pitch detector uses the cepstrum method. This separates the spectral envelope and finds structure by an inverse fourier transform of the log-power

[11]http://www.cs.indiana.edu/ port/teach/641/hearing.for.linguists.Feb27.07.html

spectrum or it can use a histogram for harmonic components in the spectral domain. In the time domain, the correlation based pitch detection uses average magnitude differential function (AMDF) for a speech or residual signal for periodicity detection or the pitch can be calculated based on the center and peak clipping for spectrum flattering and computation simplification. Additionally, the pitch cab detected using a zero-crossing count method applies an iterative pattern in a waveform zero-crossing rate.

Mel Frequency Cepstral Cofficient (MFCC):

MFCC is the most widely used spectral representation of speech. MFCC is based on human hearing perceptions which cannot perceive frequencies over 1000Hz or 1KHz. In other words, MFCC is based on known variation of the human ear's critical bandwidth with frequency. MFCC has two types of filter which are spaced linearly at low frequency below 1KHz and logarithmic spacing above 1KHz [155] [156]. Steps:

1. **Fast Fourier Transform (FFT):** In order to analyze the audio data in the frequency domain, fourier transform is applied to the input signal. Fourier transform can be used using various methods like Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT). FFT has the advantage of quickly generating results. When the input data is divided into frames, the values of each frame are converted into frequency domain using FFT. For the frequency domain algorithms, as the window size changes, the execution time and memory requirements also change [157].

2. **Mel-scaled Filter Bank and Log Processing:** The frequency range in the FFT spectrum is very wide and a voice signal does not follow the linear scale. The magnitude of the filter frequency response is used to get the log energy of that filter. Here, a set of 20 triangular bandpass filters are used. Each filter's magnitude frequency response is triangular in shape and equal to unity at the center frequency and decreases linearly to zero at center frequency of two adjacent filters [157]. The sum of the filtered spectral components is the output of each filter.

3. **Discrete Cosine Transform (DCT):** A Mel-Frequency Cepstral Coefficient (MFCC) is obtained by the conversion of the log Mel spectrum back to time domain. The set of coefficients are called acoustic vectors. Therefore, each input utterance is transformed into a sequence of acoustic vector. Conversion of the Mel spectrum coefficients into the time domain sing the DCT is possible as they are real numbers [157].

4. **Delta Cepstral and Delta Energy** Over time, features related to the change in cepstral features can be added. 12 cepstral features, 13 delta or velocity features and 13 double delta or acceleration features, i.e. a total of 39 features are used [157].

MFCC is the most widely used spectral representation of speech. MFCC parameters are calculated by taking the absolute value of the FFT, warping it to a Mel frequency scale, taking the DCT of the log-Mel spectrum and returning the first 13 coefficients. The function requires the following parameters: signal, sampling frequency, window type, number of coefficients, number of filters in the filter bank, length of a frame and the frame increment.

7.6.5 Classification concept: Bayesian Quadratic Discriminant Analysis

ML method is used to estimate the unknown probability distribution function. For instance, suppose$P(x|\omega_1, \theta)$is the likelihood function with an unknown parameter (θ), the ML method estimates the unknown parameter so that the ML function maximizes. Suppose the following function is the log-likelihood function [158].

$$L(\theta) = lnP(x|\omega_1; \theta)$$

So we take the first derivative with respect to the maximum ML of θ which is related to zero value of the first derivative:

$$\frac{d(L(\theta))}{d(\theta)} = 0 \geq Max \quad L$$

In our case, the mean μ_i and the covariance matrix $\sum i$ are the unknown parameters for the class conditional PDF, by using MLe estimate$\hat{\text{I}}\frac{1}{4}\text{i}$ and $\sum i$ for each class as:

$$\mu_i ML = \frac{1}{N}\sum_{k=1}^{N} x_{ik}$$

Quadratic discernment function:

$$\mu_i ML = \frac{1}{N}\sum_{k=1}^{N}(x_{ik} - \mu_i)(x_{jk} - \mu_j)$$

Let $g1(x) \quad and \quad g2(x)$ be the cost function of classes ω_1, ω_2 so x is classified to ω_1 if:

$$g1(x) > g2(x)$$

The decision surface which separates the two regions is:

$$g1(x)g2(x) = 0$$

In our case, the cost function:

$$g(x) = -\frac{1}{2}(x - \mu_i)^T \sum_{i}^{-1}(x - \mu_i) - \frac{1}{2}log(|\sum_{i}|) + log(P(\omega_i))$$

The decision boundaries are hyper-ellipses or hyper-paraboloids (quadratic) as shown in 2(a) and 2(b).

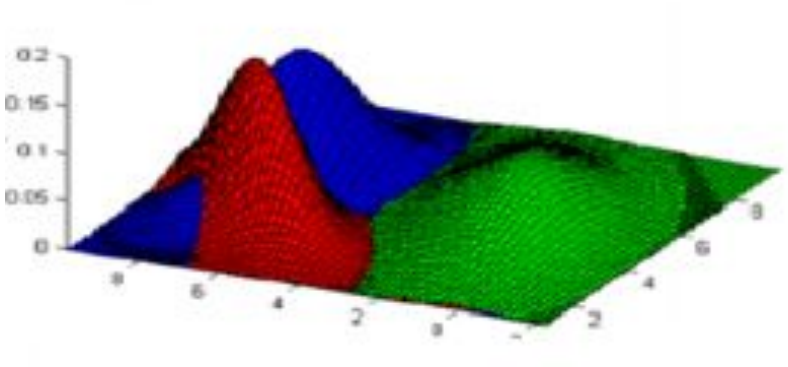

Figure 7.6: The decision boundaries 3D

Table 7.2: The obtained results using three emotions (sad, angry and normal) using BQD

	Normal	Sad	Angry	Total
Training Set	69	52	117	238
Test Set	10	10	10	30
False Positive	1	1	2	4
Detection Ratio	90%	90%	80%	86,67%

7.6.6 Experimental setup, performance results obtained and related comments

In the recent results of speech emotion recognition systems, researchers in [6] use 37 features of the voice streams. They classify 6 types of emotions, their total accuracy was 74% based on a combination of a Support Vector Machine (SVM) and a Rough Set theory and 77,91% based on SVM only. The recognition rates of a normal emotion is 90,50%, an anger emotion is 86% and sadness is 66%. In [11], the number of features is between (30-52), using Berlin database features of the voice streams. They classify 7 types of emotion, their total accuracy was 91.6% based on a Speaker Normalization (SN) and Linear Discriminant Analysis (LDA). The recognition rate of a normal emotion is 77%, an anger emotion is 82% and sadness is 92%.

In [7], authors use 4 statistical moments (mean, maximum, minimum and standard deviation) of 13 features. They classify 4 types of emotion (hot anger, cold anger, neutral and sadness). Their total accuracy was 87% based on a SVMs. In our case study, we use only 12 features to classify between 5 types of emotions based on a Bayesian Quadratic Discriminant Classifier. For an ADAS system, we focus on the 3 classes (sad, normal and angry), because they are strongly related to the representation of the drivers' behavior. Here, we present the experimental results after using different combinations of emotion classes. We use a Bayesian Quadratic Discriminant and Berlin database of emotional speech. For feature extraction, the statistical moments (minimum, maximum, mean and median) of 3 features (MFCC, pitch and energy) are extracted.

Table 7.2 shows what we used for training (69 voice files for normal, 52 for sad and 117 for angry) and for testing (10 for normal, 10 for sad, 10 for angry). 90% of normal and sad signals are correctly classified and 80% of angry. Therefore, 86.67% was the total

Table 7.3: The obtained results using three emotions (happy, sad, angry and normal) using BQD

	Happy	Normal	Sad	Angry	Total
Training Set	60	69	52	117	298
Test Set	10	10	10	10	40
False Positive	7	1	1	3	12
Detection Ratio	30%	90%	90%	70%	70%

Table 7.4: The obtained results using three emotions (fear, sad, angry and normal) using BQD

	fear	Normal	Sad	Angry	Total
Training Set	60	69	52	117	276
Test Set	9	10	10	10	39
False Positive	6	1	3	12	22
Detection Ratio	33.33%	93.33%	91%	84%	80%

result of our classifier.

Table 7.3 shows the influence of the fear emotion when it is added to the classification in general and to the training set of this class. We retrained the QBD classifier and then we used the following test data (9 of fear, 10 of normal, 10 of sad, 10 of angry) to obtain the following results, 6 of 9 afraid voices where false positives, which forms 33.33% of success. This result is low because of the lack of training data (60 only) but the positive side of this experiment is that the recognition ratio of (normal, sad and angry) emotions is increased to 93.33%, 91.66% and 84.32%.

Table 7.4 shows the influence of the happiness emotion when it is added to the classification in general and to the training set of this class. We retrained the QBD classifier and then we used the following test data (10 of happy, 10 of normal, 10 of sad, 10 of angry). We obtained the following results, 7 of 10 happy voices are false positives, which forms 30% of success. This low detection ratio is because of the low number of training sets. We also realize that the recognition ratio decreased (70%) for the angry class, while no change occurred to sadness and normal classes.

7.7 Summary

Speech emotion recognition system will be useful for understanding the state and emotions of a driver. In this work, we come to know that the acoustic information could help to increase the performance of ADAS. However, we have shown that the usage of Bayesian Quadratic Discriminant classifier enables a real-time processing with a low amount of features (12 features). We are able to reduce the calculation cost whilst keeping a high recognition rate. In our future work, we will perform the evaluation over different databases to check the robustness of the algorithms and to see the scalability of the algorithms. Furthermore, this work can be extended in the direction of reducing the acoustic noise.

Chapter 8

Conclusions and future research directions

Autonomous event detection and recognition is an effective approach to reduce the costs of monitoring all over the world. The population has experienced a continuous growth in the last 100 years. Video surveillance systems play an important role in our daily life. New approaches and modern developments are required to reduce risk, increase the safety of society and decrease the costs of monitoring.

In video/audio surveillance systems, weather and environmental conditions can significantly influence the event detection processing performance. The most challenging issue in intelligent video surveillance systems is that all parts of the system should work under any conditions. For a robust and real-time complex event detection in surveillance systems, we need a high performance processing and event detection system. Computing huge amounts of visual information for extracting meaningful data and features needs a special/appropriate approach that can run on embedded platforms.

Chapter 1 addressed the following: the motivation and the general context of this work, a short description of the research questions and objectives of the thesis, the overall research methodology, the scientific and practical significance and contributions of the thesis, a comprehensive summary of the major innovative contributions of the thesis, a list of publications in the frame of this work and finishes with the organization of the thesis.

The first major contribution is concerned with the identification of the relevant requirements of video/audio based surveillance systems. The major functional, design and performance requirements to build a successful surveillance system are defined.

The state-of-the-art technologies, tools and algorithms have been illustrated and the limitations of these approaches have been evaluated. Chapter 2 answered the question:

1. **What are the major functional, design and performance requirements of video-audio based surveillance systems and what are the limitations of the state-of-the-art?**

   Regarding the functional requirements, modern video surveillance systems are using network cameras that give them ability to create and maintain an effective and

reliable IP surveillance system. They are cost effective solutions where users can build a high performance and a scalable wired or wireless IP video surveillance system. Furthermore, they support the system through spatio-temporal event detection functions to verify the previous discussed requirements in question 1.

When choosing an optimal design for a video surveillance system, it requires the use of a mix of different camera types. Hybrid Network Video Recorders (NVR) and Digital Video Recorders (DVR) support IP cameras and are directly connected to analog cameras. This provides simplicity and reliability.

A data warehouse is a database that can be used for reporting and data analysis. It is a central repository that is created by integrating data from multiple disparate sources (audio or video). The major disadvantage of a data warehouse is that it can be costly to maintain which becomes a problem if the warehouse is underutilized.

The performance requirements of surveillance systems are difficult to achieve because of the trade off between the different requirements.

The main problem with a high recognition rate is that it could require a high power consumption because of the high computation time. Therefore, the design of recognition concepts has to be as accurate as possible and cost effective to run on embedded platforms. Chapter 2 gave the answers to the question:

2. **What are the major methodological approaches for each of the requirements groups of Q1? How far do their satisfactorily solve or not solve the requirements with respect to their limitations?**

Spatio-temporal reasoning is one of the most important challenges in visual event detection systems. Many events and video understanding requires the temporal entities to decide on a specific complex event. Different types of events need a temporal sequence to be recognized, especially in the frame of middle and long term event detection.

The major requirements for real-time reasoning and reasoning under uncertainty have also been considered. Most existing state-of-the-art methods for event/object recognition are model based systems that are expensive computations to run on tiny embedded platforms.

Another challenge is that the detection of objects in a fast computation time is also needed, e.g. in ADAS the driver has no time to think if a dangerous situation occurs.

Reasoning about context based on context ontology supports the representation of both ontological and probabilistic knowledge; we could construct a context knowledge base for the application domain. Reasoning about context information in the domain is supported by three types of reasoning mechanisms: ontological reasoning, rule-based reasoning and Bayesian reasoning.

Chapter 3 has addressed an overview of the following: all existing context models, their classification and the specificity of the spatio-temporal ones, the meaning of knowledge representation, the importance of knowledge representation, ontologies

in relation with context models, the general requirements for ontology based context models, the limitations of context modeling approaches, the meaning of reasoning, the requirements for spatio-temporal reasoning, the limitation of reasoning systems and context modeling approaches. Chapter 4 has suggested the requirements of a spatio-temporal context modeling and answered the question:

3. **What are the requirements of a) spatio-temporal context modeling and related ontologies, b) spatio-temporal reasoning (short term), c) spatio-temporal (long term), d) real-time spatio-temporal reasoning and e) spatio-temporal reasoning under uncertainty?**

The media streams in multimedia sensor networks are often correlated. The system designer has different confidence levels in the decisions obtained. There is a cost in obtaining these decisions which usually includes the cost of a sensor, its installation and maintenance cost, the cost of energy to operate it and the processing cost of the stream. Complex event detection on probabilistic data can be divided into two types:

- Local uncertainty: If an event detection is only concerned with the uncertainty of the entity object itself and is independent from other objects entities.
- Global uncertainty: Whether an object entity satisfies a detection condition depending on other objects or entities.

Vagueness or ambiguity because of the low quality of low level features in a surveillance system are sometimes described as "second order uncertainty," where uncertainty is even about the definitions of uncertain states or outcomes.

In video surveillance systems, two main types of uncertainty have been considered: uncertainty in the inference processes and uncertainty in the data of a sensor's perception caused by weather, fusion or noise coming from sensors.

There are different approaches regarding uncertainty in video surveillance systems. The most famous concepts use Monte Carlo simulations and Bayesian networks.

The recent literature scientists differentiate between various types of uncertainty, e.g. subjective uncertainty, objective uncertainty, epistemic uncertainty and ontological uncertainty. In another taxonomy, uncertainty is classified based on the approach used to measure it . In Chapter 4, we considered the major types of uncertainty and the taxonomies of uncertainty in details.

The presented work pays tribute to this fact by investigating the major types of imprecision that can occur in surveillance systems and by discussing their uncertainty on the decisions made. The integration of different methods for handling imprecision in the decision process is shown. The following questions are answered:

4. **What is uncertainty? What are the different forms of its occurrence in relation to different sensor types and functions? What are**

the different dimensions of uncertainty ? How does the state-of-the-art cope with different dimensions of uncertainty in surveillance systems?

To cover the limitations mentioned in the state-of-the-art in Chapter 4, Chapter 5 has proposed different designs for different context models for audio/video surveillance systems. The first one is built based on the frame of ontology web rule language and Semantic Web rule language.

The second model design is based on the basics of knowledge representation models originating from Answer Set Programming (ASP) as a reasoning environment.

The designed context models are spatio-temporal context models that allow the identification of complex events with respect to spatial and temporal resolutions. Another major contribution is the integration of scene description within an Answer Set Programming (ASP) environment, to enable intelligent reasoning and decision deduction.

Logic programming as a method to represent declarative knowledge in artificial intelligence approaches has proved successful for other rule-based domains so far and will also prove valuable to video/audio surveillance systems. While up to now the computing power has not been available in smart cameras, today's smart cameras already provide a high computing resources, but they are expensive.

The representation of imperfect information in the context-model has already been discussed in the first key question. The management of imperfect information within the reasoning process will now be discussed.

Uncertainty handling is important for a high performance spatio-temporal reasoning process. However, the main idea behind tackling imprecise information during the reasoning process is demonstrated and the application of different approaches for different types of imprecision is outlined with several examples.

This thesis has discussed the consideration of diverse uncertainty forms in the frame of complex event detection through multimedia sensor networks. Uncertainty is the state of having limited knowledge where it is impossible to describe exactly the existing state or to predict the possible outcome.

Related approaches considering uncertainty in event detection are confidence functions in a Boolean data type format, fuzzy modeling approach and Dempster-Shafer approach. These use belief and plausibility functions to describe the reliability features. In Chapter 5, we presented a novel approach which combines Hidden Markov Model (HMM) and Answer Set Programming (ASP).

Regarding event detection on embedded platforms requires a model-free and an inexpensive computational approach in order to have an easy and simple solution, which allows an integration of a FPGA-based (Field Programmable Gate Array) smart camera without the need of a bigger FPGA.

Therefore, the thesis presents a solution based on a foreground-background-segmentation using Gaussian Mixture Models (GMM) to first detect people and then analyze their main and ideal orientation using moments. This allows one to

decide whether a person is staying still or lying on the floor. The system has a low latency and a detection rate of 88% in our case study.

Another key of this algorithm is the use of GMMs for image segmentation. This is not sensitive to the light and small movements in the background of a scene and considers shadow detection that has an influence on the overall event detection process.

Chapter 5 and Chapter 6 have demonstrated the effectiveness of the proposed approaches in comparison to the state-of-the-art and thus, the following question is answered:

5. **What are the novel solutions to the points a, b, c, d and e of Question 3?**

There are different approaches of event detection and recognition and every approach has its advantages and disadvantages. In this thesis, the limitations of the state-of-the-art have been considered deeply. Every Chapter has illustrated the limitations of every approach. The limitations of the methodological approaches for functional, design and performance requirements of surveillance systems.

Additionally, the limitations of short term, long term, real-time and spatio-temporal event detection under uncertainty have been addressed. Consequently, the limitations of the existing context modeling techniques have been discussed. Chapter 3 and Chapter 4 have offered every approach and have given the limitations of every one comprehensively. Therefore, the following question is answered:

6. **What are the limitations of the previous concepts in Q4?**

Emotion detection systems gather participants' emotions as well as proximity and patterns of driver speech by processing the outputs from the sensors. The sensors of an emotion recognition system should be low-cost, low-failure and should be connected in a feasible way. Thus, the emotion detection system can be used to understand the correlation and the impact of interactions and activities of the emotions and behavior of individuals.

The thesis has discussed the taxonomies of uncertainty in audio based event detection. It summarizes the major origins of uncertainty and proposes the minimum required features that should be extracted from the audio data to detect the emotions of humans.

Another type of uncertainty in the model e.g. models could be effected by noise and the noise is possibly represented in the model, therefore, the model has a margin of error where the decision is not always true. Finally, uncertainty in perception e.g. sensors do not return exact or complete information about the world; a system never knows exactly its position.

There is a major question to be asked. How does one deal with uncertainty in audio surveillance systems? The answer consists of two main approaches: the implicit

approach and the explicit approach. The implicit approach deals with uncertainty by building procedures that are robust to uncertainty. The explicit approach deals with uncertainty by building a model of the world that describes uncertainty about its state, dynamics and observations. Then, we reason the effect of actions given the model.

Chapter 7 has considered the following: the basic concepts related to emotion and its involvement in technical systems, the definition of an emotion, the importance of emotion in a variety of technical systems, the consideration of emotion detection as a particular event detection with the illustration of different scenarios.

In addition to this, the requirements of human voice based emotion detection systems, the origin of uncertainty in human voice based emotion detection systems and the general limitations of the related state-of-the-art in human voice based emotion detection have been considered. Thus, the following question is answered:

7. **What are the requirements of emotion detection in the frame of human surveillance? What are the different forms of uncertainty related to emotion detection? What are the limitations of the related state-of-the-art?**

An audio emotion recognition system will be useful for recognizing the emotion in surveillance systems. In Chapter 7, we came to know that the acoustic information combined with the Bayesian Quadratic Discriminant classifier (BQD) and emotion recognition ideas can strengthen the power of the event detection process.

In this thesis, we gave attention to features that predominantly have a role in emotion and omitted other features. The feature selection plays an important role in recognizing the emotion in order to increase the performance in real-time. In this work, we emphasized on performing the evaluation over different emotional databases to check the robustness of the algorithms and to see the scalability of the algorithms. Therefore,

8. **A demo example of an audio based emotion detection has been designed.**

The proposed solutions in this work can be used to build surveillance systems that detect complex events quickly. However, the cost of storing surveillance data remains expensive. The longer data is kept, the more storage is needed and in turn, the higher the cost. The novel event detection reasoning concept systems can help to run on embedded platforms and to store the sufficient required videos and delete others that are not important.

8.1 Outlook

An interesting research area in the future would be the use of Monte Carlo methods. They provide approximate solutions to a variety of mathematical problems by performing statistical sampling experiments. They can be loosely defined as statistical simulation methods, where statistical simulation is defined in quite general terms to be any method that uses sequences of random numbers to perform the simulation. Thus, Monte Carlo

methods are a collection of different methods that all basically perform the same process, e.g. Metropolis-Hastings algorithm and Gibbs-Sampling method. This process involves performing many simulations using random numbers and probability to get an approximation of the answer to the problem [159].

Complexity simulation often gives better physical visibility of a complex system. Suppose a dynamic phenomenon in which the behavior changes over time, e.g. the behavior of objects in a surveillance system. Each change is such an event.

Using universal approximation theories can help to estimate the problem of finding the function that best approximates the data. The quality of an approximation produced by the learning system is measured by the loss function.

For each input that comes from the sensors of a surveillance system, the learning machine should select a model that best describes the data. In other words, a method is needed which approximates the sensor data to the best distribution which is known as a normal behavior in the scene.

There are various model selection methods, e.g. analytical model selection via penalization and model selection via re-sampling. The re-sampling approach has the advantage of making no assumptions on the statistics of the data or the type of target function being estimated. However, its main disadvantage is a high computational effort.

Monte Carlo simulation approaches can do this and have the advantage of choosing the right distribution of data despite of uncertainty in the sensor data [160].

The advantage of Monte Carlo simulation approaches compared to other analytical concepts is that Monte Carlo is easier to deal with in simulations than analytical models. Although, analytical models are deterministic, they usually involve simplifying assumptions to make the model analytically tractable. Such assumptions have to be justified [160].

In the area of video surveillance systems, the search for the scene state providing the maximum posterior probability is required to detect complex events with a high success rate. The detected events can enable the computation of the dynamics likelihood probability using event context.

Another research direction in the area of ADAS and emotion recognition systems could be extensively using thermal imaging (is an extensively used in) to identify and recognize persons under critical illumination conditions. When a subject experiences elevated feelings of alertness, anxiety or fear, increased levels of adrenaline regulate blood flow. The redistribution of blood showed in superficial blood vessels causes abrupt changes in local skin temperature. This is readily apparent in the human face where the layer of flesh is very thin. Therefore, mid- and far-infrared thermal cameras can be used to sense temperature variations or signatures of the face from a certain distance.

List of Abbreviations

ADAS Advanced Driver Assistance Systems
AI Artificial Intelligence
AMDF Average Magnitude Differential Function
ASP Answer Set Programming
ATM Automatic Teller Machine
AVSS Advanced Video and Signal-Based Surveillance Conference
BDES Berlin Database of Emotional Speech
BQD Bayesian Quadratic Discriminant
CCTV Closed-Circuit Television
CFGs Context Free Grammars
CF Certainty Factor
CML Context Modeling Language
CRS Chronicle Recognition System
DFT Discrete Fourier Transform
DSP Digital Signal Processing
DVRs Digital Video Recorders
DVR Digital Video Recorder
ECG Electrocardiogram
EEG Electroencephalogram
EOG Electrooculogram
FACS Facial Action Coding Systems
FASP Fuzzy Answer Set Programming
FED Fuzzy Event Detection
FFT Fast Fourier Transform
FPGA Field Programmable Gate Array
GMMs Gaussian Mixture Models
GMM Gaussian Mixture Model
GM Graphical Modeling
GPS Global Positioning System
HMMs Hidden Markov Models
HMM Hidden Markov Model
IEC International Electrotechnical Commission
IOS International Organization for Standardization
LPODs Logical Programs with Ordered Disjunction
MFCC Mel-Frequency Cepstral Coefficient
MLPs Multilayer Perceptrons
MLP Multilayer Perception

MMSE Minimum Mean Square Error
NTP Network Time Protocol
NVRs Network Video Recorders
NVR Network Video Recorder
OAV Object-Attribute-Value
ORM Object-Role Modeling
OWL Ontology Web Language
PGH Pairwise Geometric Histograms
PTZ Pan Tilt Zoom
QoS Quality of Service
RBF Radial Basis Function
RDF Resource Description Framework
ROC Receiver Operating Characteristic
RTP Real-Time Transport Protocol
RTSP Real-Time Streaming Protocol
SCFGs Stochastic Context Free Grammars
SLD Selective Linear Definite
STFT Short-Time Fourier Transform
SVMs Support Vector Machines
SVM Support Vector Machine
SWRL Semantic Web Rule Language
TCP Transmission Control Protocol
TDFT Time Dependent Fourier Transform
UDP User Datagram Protocol
UML Unified Modeling Language
VMS Video Management Software

List of Figures

List of Tables

Bibliography

[1] C. Piciarelli, C. Micheloni, and G.L. Foresti. Trajectory-based anomalous event detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18 (11):1544–1554, 2008.

[2] PB Farradyne. *Traffic Incident Management Handbook*. Federal Highway Administration Office of Travel Management, 2000.

[3] M. Valera and SA Velastin. Intelligent distributed surveillance systems: a review. In *Vision, Image and Signal Processing, IEE Proceedings-*, volume 152, pages 192–204. IET, 2005.

[4] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, 1997.

[5] S. Maskell and N. Gordon. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. In *Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE*, pages 2–1. IET, 2001.

[6] H. HUNG, S. Venkatesh, and G. West. Tracking and surveillance in wide-area spatial environments using the abstract hidden markov model. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):177–196, 2001.

[7] M. Pontil and A. Verri. Support vector machines for 3d object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(6):637–646, 1998.

[8] Z. Zhi-Hong. Lane detection and car tracking on the highway. *Acta Automatica Sinica*, 29(3):450–456, 2003.

[9] J.M. Ferryman, S.J. Maybank, and A.D. Worrall. Visual surveillance for moving vehicles. *International Journal of Computer Vision*, 37(2):187–197, 2000.

[10] M.E. Weber and M.L. Stone. Low altitude wind shear detection using airport surveillance radars. In *Radar Conference, 1994., Record of the 1994 IEEE National*, pages 52–57. IEEE, 1994.

[11] M. Pellegrini and P. Tonani. *Highway traffic monitoring*. Kluwer Academic, Boston, Mass, USA, 1998.

[12] R. Cucchiara, C. Grana, A. Prati, G. Tardini, and R. Vezzani. Using computer vision techniques for dangerous situation detection in domotic applications. In *Intelligent Distributed Surveilliance Systems, IEE*, pages 1–5. IET, 2004.

[13] D. Greenhill, P. Remagnino, and G.A. Jones. Vigilant: content-querying of video surveillance streams. 2002.

[14] M. Xu, J. Orwell, L. Lowey, and D. Thirde. Architecture and algorithms for tracking football players with multiple cameras. In *Vision, Image and Signal Processing, IEE Proceedings-*, volume 152, pages 232–241. IET, 2005.

[15] Z. Geradts and J. Bijhold. Forensic video investigation. *Multimedia Video-Based Surveillance Systems*, pages 3–12, 2000.

[16] J.K. Money and W.R. Walker. Mobile video surveillance system and method, May 17 2006. US Patent App. 11/383,797.

[17] P.L. Venetianer, A.J. Lipton, A.J. Chosak, M.F. Frazier, N. Haering, G.W. Myers, W. Yin, and Z. Zhang. Video surveillance system employing video primitives, January 11 2011. US Patent 7,868,912.

[18] G.A. Gibson and R. Van Meter. Network attached storage architecture. *Communications of the ACM*, 43(11):37–45, 2000.

[19] G. Yang. *Life cycle reliability engineering*. Wiley, 2007.

[20] G.L. Foresti and P. Mähönen. *Multimedia video-based surveillance systems: Requirements, issues, and solutions*, volume 573. Springer, 2000.

[21] J. Carlson. *An intuitive and resource-efficient event detection algebra*. PhD thesis, Mälardalen University, 2004.

[22] E. Malinowski and E. Zimányi. Introduction. *Advanced Data Warehouse Design*, pages 1–16, 2008.

[23] P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

[24] J.B. Tenenbaum, V. De Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[25] T. Syeda-Mahmood, A. Vasilescu, and S. Sethi. Recognizing action events from multiple viewpoints. In *Detection and Recognition of Events in Video, 2001. Proceedings. IEEE Workshop on*, pages 64–72. IEEE, 2001.

[26] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 984–989. IEEE, 2005.

[27] O. Chomat and J.L. Crowley. Probabilistic recognition of activity using local appearance. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[28] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–123. IEEE, 2001.

[29] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992.

[30] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1371–1375, 1998.

[31] M.S. Ryoo and J.K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1709–1718. IEEE, 2006.

[32] N.A. Rota and M. Thonnat. Activity recognition from video sequences using declarative models. In *ECAI*, pages 673–680. Citeseer, 2000.

[33] G. Medioni, I. Cohen, F. Brémond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(8):873–889, 2001.

[34] S. Batsakis and E.G.M. Petrakis. Sowl: spatio-temporal representation, reasoning and querying over the semantic web. In *Proceedings of the 6th International Conference on Semantic Systems*, page 15. ACM, 2010.

[35] S. Ribaric and T. Hrkac. A knowledge representation and reasoning based on petri nets with spatio-temporal tokens. In *EUROCON, 2007. The International Conference on" Computer as a Tool"*, pages 793–800. IEEE, 2007.

[36] P. Héas and M. Datcu. Modeling trajectory of dynamic clusters in image time-series for spatio-temporal reasoning. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(7):1635–1647, 2005.

[37] U.M. Erdem and S. Sclaroff. Automated placement of cameras in a floorplan to satisfy task-specific constraints. Technical report, Technical Report BUCS-TR-2003-031, Boston University, 2003.

[38] B.A. Abidi. Automatic sensor placement. In *Proc. Intelligent Robots and Computer Vision: Algorithms, Techniques, Active Vision, and Materials Handling*, pages 387–398, 1995.

[39] F. Angella, L. Reithler, and F. Gallesio. Optimal deployment of cameras for video surveillance systems. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 388–392. IEEE, 2007.

[40] H.W. Guesgen and S. Marsland. Spatio-temporal reasoning and context awareness. *Handbook of Ambient Intelligence and Smart Environments*, pages 609–634, 2010.

[41] C. Brewster and K. O'Hara. Knowledge representation with ontologies: the present and future. *Intelligent Systems, IEEE*, 19(1):72–81, 2004.

[42] R. Brachman and H. Levesque. *Knowledge representation and reasoning*. Morgan Kaufmann, 2004.

[43] T. Gruber. What is an ontology. *Encyclopedia of Database Systems*, 1, 2008.

[44] O. Bucur, P. Beaune, O. Boissier, et al. Representing context in an agent architecture for context-based decision making. In *Proceedings of the Workshop on Context Representation and Reasoning (CRR05), Paris, France*, 2005.

[45] T. Strang and C. Linnhoff-Popien. A context modeling survey. In *Workshop Proceedings*, 2004.

[46] P. Moore, B. Hu, X. Zhu, W. Campbell, and M. Ratcliffe. A survey of context modeling for pervasive cooperative learning. In *Information Technologies and Applications in Education, 2007. ISITAE'07. First IEEE International Symposium on*, pages K5–1. IEEE, 2007.

[47] H. Chen, T. Finin, and A. Joshi. An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 18(03):197–207, 2003.

[48] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.

[49] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE, 1994.

[50] T. Gu, X.H. Wang, H.K. Pung, and D.Q. Zhang. An ontology-based context model in intelligent environments. In *Proceedings of communication networks and distributed systems modeling and simulation conference*, volume 2004, pages 270–275, 2004.

[51] C. Simons. Cmp: a uml context modeling profile for mobile distributed systems. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, pages 289b–289b. IEEE, 2007.

[52] A. Artale and E. Franconi. Reasoning with enhanced temporal entity-relationship models. In *Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on*, pages 482–486. IEEE, 1999.

[53] K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in pervasive computing systems. *Pervasive Computing*, pages 79–117, 2002.

[54] H. Kress-Gazit, G.E. Fainekos, and G.J. Pappas. Temporal-logic-based reactive mission and motion planning. *Robotics, IEEE Transactions on*, 25(6):1370–1381, 2009.

[55] A. Artikis and G. Paliouras. Behaviour recognition using the event calculus. *Artificial Intelligence Applications and Innovations III*, pages 469–478, 2009.

[56] K. Henricksen, S. Livingstone, and J. Indulska. Towards a hybrid approach to context modelling, reasoning and interoperation. In *Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning And Management, in conjunction with UbiComp*, 2004.

[57] R. Krummenacher and T. Strang. Ontology-based context modeling. In *Proceedings Third Workshop on Context-Aware Proactive Systems (CAPS 2007)(June 2007)*, 2007.

[58] S. Fuchs. *A Comprehensive Knowledge Base for Context Aware Tactical Driver Assistance Systems*. Shaker, 2009.

[59] L. Snidaro, M. Belluz, and GL Foresti. Representing and recognizing complex events in surveillance applications. In *IEEE Conference on Advanced Video and Signal Based Surveillance, 2007. AVSS 2007*, pages 493–498, 2007.

[60] D.L. McGuinness, F. Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(2004-03):10, 2004.

[61] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, et al. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79, 2004.

[62] M. Uschold, M. Gruninger, et al. Ontologies: Principles, methods and applications. *Knowledge engineering review*, 11(2):93–136, 1996.

[63] J. Cabot, A. Olivé, and E. Teniente. Representing temporal information in uml. *The Unified Modeling Language. Modeling Languages and Applications*, pages 44–59, 2003.

[64] N. Kompridis. So we need something else for reason to mean. *International journal of philosophical studies*, 8(3):271–295, 2000.

[65] S. Liang, P. Fodor, H. Wan, and M. Kifer. Openrulebench: an analysis of the performance of rule engines. In *Proceedings of the 18th international conference on World wide web*, pages 601–610. ACM, 2009.

[66] B. Walczak and DL Massart. Rough sets theory. *Chemometrics and intelligent laboratory systems*, 47(1):1–16, 1999.

[67] A.J. Gonzalez and D.D. Dankel. *The engineering of knowledge-based systems: theory and practice*. Prentice Hall Englewood Cliffs, New Jersey, 1993.

[68] M. Stonebraker, U. Çetintemel, and S. Zdonik. The 8 requirements of real-time stream processing. *ACM SIGMOD Record*, 34(4):42–47, 2005.

[69] W.D. Rowe. Understanding uncertainty. *Risk analysis*, 14(5):743–750, 2006.

[70] M. Niepert, J. Noessner, and H. Stuckenschmidt. Log-linear description logics. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2153–2158. AAAI Press, 2011.

[71] X.H. Wang, D.Q. Zhang, T. Gu, and H.K. Pung. Ontology based context modeling and reasoning using owl. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 18–22. IEEE, 2004.

[72] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[73] I. Pruteanu-Malinici and L. Carin. Infinite hidden markov models for unusual-event detection in video. *Image Processing, IEEE Transactions on*, 17(5):811–822, 2008.

[74] M. Mansouri-Samani and M. Sloman. Gem: A generalized event monitoring language for distributed systems. *Distributed Systems Engineering*, 4(2):96, 1999.

[75] S. Chakravarthy, V. Krishnaprasad, E. Anwar, and SK Kim. Anatomy of a composite event detector. In *Proc. of the 20th International Conference on Very Large Databases, Santiago Chile*. Citeseer, 1993.

[76] N.H. Gehani, H.V. Jagadish, and O. Shmueli. Event specification in an active object-oriented database. In *ACM Sigmod Record*, volume 21, pages 81–90. ACM, 1992.

[77] C. Baral, G. Gelfond, T.C. Son, and E. Pontelli. Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 259–266. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[78] G. Brewka. Logic programming with ordered disjunction. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 100–105. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.

[79] T. Eiter and A. Polleres. Towards automated integration of guess and check programs in answer set programming: a meta-interpreter and applications. *Theory and Practice of Logic Programming*, 6(1-2):23–60, 2006.

[80] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and S. Thiele. A user's guide to gringo, clasp, clingo, and iclingo, 2008.

[81] G. Brewka, T. Eiter, and M. Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12):92–103, 2011.

[82] E.P. Blasch, E. Dorion, P. Valin, and E. Bossé. Ontology alignment using relative entropy for semantic uncertainty analysis. In *Aerospace and Electronics Conference (NAECON), Proceedings of the IEEE 2010 National*, pages 140–148. IEEE, 2010.

[83] C. Liu, D. Grenier, A.L. Jousselme, and E. Bosse. Reducing algorithm complexity for computing an aggregate uncertainty measure. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 37(5):669–679, 2007.

[84] J. Yin, D.H. Hu, and Q. Yang. Spatio-temporal event detection using dynamic conditional random fields. In *IJCAI*, volume 9, pages 1321–1327, 2009.

[85] G.J. Klir and B. Yuan. Fuzzy sets and fuzzy logic: Theory and applications. *Possibility Theory versus Probability Theory, Prentice Hall*, pages 200–207, 1995.

[86] S. Parsons. *Qualitative methods for reasoning under uncertainty*, volume 13. Mit Press, 2001.

[87] D.V. Lindley. *Understanding uncertainty*. Wiley-Interscience, 2006.

[88] B.G. Buchanan and E.H. Shortliffe. *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley series in artificial intelligence)*. Addison-Wesley Longman Publishing Co., Inc., 1984.

[89] G. Shafer. *A mathematical theory of evidence*, volume 76. Princeton university press Princeton, 1976.

[90] JSR Jang. *Fuzzy inference systems*. Upper Saddle River, NJ: Prentice-Hall, 1997.

[91] O. Cappé, E. Moulines, and T. Rydén. *Inference in hidden Markov models*. Springer, 2005.

[92] C. Baral. Logic programming and uncertainty. *Scalable Uncertainty Management*, pages 22–37, 2011.

[93] N. Meghanathan, B.K. Kaushik, and D. Nagamalai. *Advances in Networks and Communications: First International Conference on Computer Science and Information Technology, CCSIT 2011, Bangalore, India, January 2-4, 2011. Proceedings*, volume 132. Springer Verlag, 2011.

[94] Y. Wang, S. Velipasalar, and M. Casares. Cooperative object tracking and composite event detection with wireless embedded smart cameras. *Image Processing, IEEE Transactions on*, 19(10):2614–2633, 2010.

[95] G. Gravier, C.H. Demarty, S. Baghdadi, and P. Gros. Classification-oriented structure learning in bayesian networks for multimodal event detection in videos. *Multimedia Tools and Applications*, pages 1–17, 2012.

[96] W. Khreich, E. Granger, R. Sabourin, and A. Miri. Combining hidden markov models for improved anomaly detection. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[97] H. Zhou, D. Kimber, and L. Wilcox. Unusual event detection via collaborative video mining, August 30 2011. US Patent 8,009,193.

[98] M.D. Naish, E.A. Croft, and B. Banhabib. Object surveillance using reinforcement learning based sensor dispatching. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 71–76. IEEE, 2004.

[99] S.T. Li and Y.C. Cheng. A hidden markov model-based forecasting model for fuzzy time series. *WSEAS Transactions on Systems*, 5(8):1919–1924, 2006.

[100] J. Yang and Y. Xu. Hidden markov model for gesture recognition. Technical report, DTIC Document, 1994.

[101] *Mortality figures for accidental falls*, 1998. Office of National Statistics.

[102] Christophe Bobda, Ali Akbar Zarezadeh, Felix Mühlbauer, Robert Hartmann, and Kevin Cheng. Reconfigurable architecture for distributed smart cameras. In *International Conference on Engineering of Reconfigurable Systems and Algorithms*, 2010.

[103] J. Spehr, M. Gövercin, S. Winkelbach, E. Steinhagen–Thiessen, and F. Wahl. Visual fall detection in home environments. *6th Int. Conference of the Int. Soc. for Gerontechnology Pisa, Italy*, 2008.

[104] Jens Spehr. *Beaufsichtigung von Personen im häuslichen Umfeld: Grundlagen und Konzepte zur Verwendung einer Fischaugenkamera*. Vdm Verlag Dr. Müller, 2007.

[105] Adam Williams, Deepak Ganesan, and Allen Hanson. Aging in place: fall detection and localization in a distributed smart camera network. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 892–901, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-702-5. doi: http://doi.acm.org/10.1145/1291233.1291435.

[106] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1:pp. 511, 2001.

[107] A. P. Ashbrook, N. A. Thacker, and P. I. Rockett. Multiple shape recognition using pairwise geometric histogram based algorithms. *Fifth International Conference on Image Processing and its Applications*, vol. 5:90–94, 1995.

[108] Jean-Yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. Technical report, Intel Corporation, Microprocessor Research Labs, 2004.

[109] *Q24M Kamerahandbuch*, 2010. MOBOTIX AG.

[110] Yogesh Raja, Stephen J. McKenna, and Shaogang Gong. Segmentation and tracking using color mixture models. In *ACCV '98: Proceedings of the Third Asian Conference on Computer Vision-Volume I*, pages 607–614, London, UK, 1997. Springer-Verlag. ISBN 3-540-63930-6.

[111] L. Kotoulas and I. Andreadis. Image analysis using moments. In *5th Int. Conf. on Tech. and Automation*, pages 360–364, 1998.

[112] J. Kilian. Simple image analysis by moments. Technical report, OpenCV library documentation, 2001.

[113] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *Proceedings of the Fourth International IEEE Conference on Intelligent Transportation Systems*, 2001.

[114] P. Smets. Imperfect information: Imprecision, and uncertainty. *Uncertainty Management in Information Systems*, 1996:225–254, 1996.

[115] Y. Lin and G.A. Cunningham III. A new approach to fuzzy-neural system modeling. *Fuzzy Systems, IEEE Transactions on*, 3(2):190–198, 1995.

[116] Fernyhough J., Cohn A.G., and Hogg D.C. Constructing qualitative event models automatically from video input. *Image and Vision Computing*, 18(2):81 – 103, 2000.

[117] T. Gu, X.H. Wang, H.K. Pung, and D.Q. Zhang. An ontology-based context model in intelligent environments. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, volume 2004. Citeseer, 2004.

[118] H.J. Baek, H.B. Lee, J.S. Kim, J.M. Choi, K.K. Kim, and K.S. Park. Nonintrusive biological signal monitoring in a car to evaluate a driver's stress and health state. *TELEMEDICINE and e-HEALTH*, 15(2):182–189, 2009.

[119] I.C. Christie and B.H. Friedman. Autonomic specificity of discrete emotion and dimensions of affective space: A multivariate approach. *International Journal of Psychophysiology*, 51(2):143–153, 2004.

[120] S.K.L. Lal, A. Craig, P. Boord, L. Kirkup, and H. Nguyen. Development of an algorithm for an eeg-based driver fatigue countermeasure. *Journal of Safety Research*, 34(3):321–328, 2003.

[121] E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. Bartlett, and J. Movellan. Drowsy driver detection through facial movement analysis. *Human–Computer Interaction*, pages 6–18, 2007.

[122] V. Slavova, H. Sahli, and W. Verhelst. Multi-modal emotion recognition-more "cognitive" machines. *New Trends in Intelligent Technologies*, page 70, 2009.

[123] H. Gu, Q. Ji, and Z. Zhu. Active facial tracking for fatigue detection. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 137–142. IEEE, 2002.

[124] L. Kessous, G. Castellano, and G. Caridakis. Multimodal emotion recognition in speech-based interaction using facial expression, body gesture and acoustic analysis. *Journal on Multimodal User Interfaces*, 3(1):33–48, 2010.

[125] J.D. Woodward Jr, C. Horn, J. Gatune, and A. Thomas. Biometrics: A look at facial recognition. Technical report, DTIC Document, 2003.

[126] N. Sebe, I. Cohen, and T.S. Huang. Multimodal emotion recognition. *Handbook of Pattern Recognition and Computer Vision*, 4:387–419, 2005.

[127] Y.H. Yang, Y.C. Lin, Y.F. Su, and H.H. Chen. A regression approach to music emotion recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):448–457, 2008.

[128] E. Ghiani, N. Locci, and C. Muscas. Auto-evaluation of the uncertainty in virtual instruments. *Instrumentation and Measurement, IEEE Transactions on*, 53(3):672–677, 2004.

[129] M.J. Korczynski and A. Hetman. A calculation of uncertainties in virtual instrument. In *Instrumentation and Measurement Technology Conference, 2005. IMTC 2005. Proceedings of the IEEE*, volume 3, pages 1697–1701. IEEE, 2005.

[130] M. El Ayadi, M.S. Kamel, and F. Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, 2011.

[131] G. Betta, C. Liguori, and A. Pietrosanto. Propagation of uncertainty in a discrete fourier transform algorithm. *Measurement*, 27(4):231–239, 2000.

[132] R. López-Cózar, Z. Callejas, M. Kroul, J. Nouza, and J. Silovskỳ. Two-level fusion to improve emotion classification in spoken dialogue systems. In *Text, Speech and Dialogue*, pages 617–624. Springer, 2008.

[133] C.H. Wu and W.B. Liang. Emotion recognition of affective speech based on multiple classifiers using acoustic-prosodic information and semantic labels. *Affective Computing, IEEE Transactions on*, 2(1):10–21, 2011.

[134] T. Pilutti and A.G. Ulsoy. Identification of driver state for lane-keeping tasks. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 29(5):486–502, 1999.

[135] W.S. Wijesoma, K.R.S. Kodagoda, and A.P. Balasuriya. Road-boundary detection and tracking using ladar sensing. *Robotics and Automation, IEEE Transactions on*, 20(3):456–464, 2004.

[136] R. Bittner, K. Hána, L. Poušek, P. Smrka, P. Schreib, and P. Vysokỳ. Detecting of fatigue states of a car driver. *Medical Data Analysis*, pages 123–126, 2000.

[137] I. Luengo, E. Navas, and I. Hernáez. Combining spectral and prosodic information for emotion recognition in the interspeech 2009 emotion challenge. *Proc. Interspeech, Brighton*, pages 332–335, 2009.

[138] C.C. Lee, E. Mower, C. Busso, S. Lee, and S. Narayanan. Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, 53(9): 1162–1171, 2011.

[139] L. Todorovski and S. Džeroski. Combining classifiers with meta decision trees. *Machine Learning*, 50(3):223–249, 2003.

[140] S. Yacoub, S. Simske, X. Lin, and J. Burns. Recognition of emotions in interactive voice response systems. In *Proceedings of Eurospeech*, pages 729–732, 2003.

[141] S. Wu, T.H. Falk, and W.Y. Chan. Automatic speech emotion recognition using modulation spectral features. *Speech Communication*, 53(5):768–785, 2011.

[142] J. Zhou, G. Wang, Y. Yang, and P. Chen. Speech emotion recognition based on rough set and svm. In *Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference on*, volume 1, pages 53–61. IEEE, 2006.

[143] V. Krueger and S. Zhou. Exemplar-based face recognition from video. *Computer Vision ECCV 2002*, pages 361–365, 2006.

[144] PS Hiremath, A. Danti, and CJ Prabhakar. Modelling uncertainty in representation of facial features for face recognition. *Face Recognition*, 10:183–218, 2007.

[145] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2578–2585. IEEE, 2012.

[146] L.B. Almeida. The fractional fourier transform and time-frequency representations. *Signal Processing, IEEE Transactions on*, 42(11):3084–3091, 1994.

[147] A. Stark and K. Paliwal. Mmse estimation of log-filterbank energies for robust speech recognition. *Speech Communication*, 53(3):403–416, 2011.

[148] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero. A minimum-mean-square-error noise reduction algorithm on mel-frequency cepstra for robust speech recognition. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 4041–4044. IEEE, 2008.

[149] R.F. Astudillo, D. Kolossa, P. Mandelartz, and R. Orglmeister. An uncertainty propagation approach to robust asr using the etsi advanced front-end. *Selected Topics in Signal Processing, IEEE Journal of*, 4(5):824–833, 2010.

[150] R. Togneri and D. Pullella. An overview of speaker identification: Accuracy and robustness issues. *Circuits and Systems Magazine, IEEE*, 11(2):23–61, 2011.

[151] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss. A database of german emotional speech. In *Proc. Interspeech*, volume 2005, 2005.

[152] M. Grimm, K. Kroschel, and S. Narayanan. The vera am mittag german audio-visual emotional speech database. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 865–868. IEEE, 2008.

[153] A.H. Omar. *Audio segmentation and classification*. PhD thesis, Informatik og Matematisk Modellering, Danmarks Tekniske Universitet, 2005.

[154] T. Vogt, E. André, and J. Wagner. Automatic recognition of emotions from speech: a review of the literature and recommendations for practical realisation. *Affect and Emotion in Human-Computer Interaction*, pages 75–91, 2008.

[155] J.R. Johnson. Introduction to digital signal processing. *Introduction to digital signal processing, by JR Johnson. Englewood Cliffs, NJ, Prentice Hall, 1989, 426 p.*, 1, 1989.

[156] D. Sundararajan. *The discrete Fourier transform: theory, algorithms and applications*. World Scientific Publishing Company Incorporated, 2001.

[157] I.R. Murray and J.L. Arnott. Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion. *The Journal of the Acoustical Society of America*, 93:1097, 1993.

[158] S. Srivastava, M.R. Gupta, and B.A. Frigyik. Bayesian quadratic discriminant analysis. *Journal of Machine Learning Research*, 8(6):1277–1305, 2007.

[159] J. Pengelly. Monte carlo methods. *University of Otago*, 2002.

[160] C. Baudrit and D. Dubois. Comparing methods for joint objective and subjective uncertainty propagation with an example in a risk assessment. In *International Symposium on Imprecise Probabilities and Their Application (ISIPTA 05), Pittsburg (USA, Pennsylvanie)*, pages 31–40, 2005.