Rule-Based Situation Recognition in Video Streams

A Thesis Submitted in Partial Fulfillment of the Requirements for the Award of the Degree

Master of Science in Optics and Photonics

by

Andrew Katumba

at the

Karlsruhe Institute of Technology

27th September 2013

Supervisor: Prof. Dr.-Ing. Christoph Stiller[1]

Supervisor: Dr. rer. nat. Martin Lauer[1]

Advisor: Dr. rer. nat. Wolfgang Hübner

Advisor: David Münch

Fraunhofer Institute for Optronics, System Technologies and Image exploitation

Gutleuthausstr. 1, 76275 Ettlingen

[1]Institute of Measurement and Control - Karlsruhe Institute of Technology

Declaration

Name: Andrew Katumba
Matriculation Number: 1517911

I hereby declare that the work presented here was developed and written entirely by me, and that I have not used any sources or means without declaration in the text.

Ettlingen, 27th September 2013

Andrew Katumba

Abstract

Recognizing situations in video data is one of the most promising and challenging research areas in computer vision today. The challenges mainly arise in the form of errors introduced in the data from the point of acquisition to the moment when identification and recognition of situations is done. The Cognitive Vision System(CVS) based on Fuzzy Metric Temporal Logic and Situation Graph Trees (FMTL/SGT) is a rule-based framework for situation recognition. Representing expert knowledge about expected situations in a scene is done by developing SGTs, and FMTL a logic representation formalism for representing the numerical data about the scene, the rules governing its composition into situations, and the actual querying of situations. To date, the FMTL/SGT CVS has been applied in fields starting with Traffic Control, to Robotics, Smart Work Environments, and Surveillance.

The work in this thesis focuses on activities towards the effective deployment and utilization of such a system in the real world. First, by designing and implementing a flexible architecture for the deployment of such a system. And second, by performing a detailed evaluation of the performance of the FMTL/SGT CVS system in various scenarios.

The contribution of this thesis is, first, a complete design and concrete implementation of an extensible, distributed, real-time-capable architecture with modules for low-level data acquisition and processing, traversal of SGTs to query for potential instantiation, and spot-on visualization of the progression of the reasoning process. The architecture can be adopted and applied in various situation recognition applications. Second, a specification for the performance of a FMTL/SGT CVS system that could be used to gain insight on if such a system is a fit for a specific application for situation recognition.

Contents

Chapter 1

Introduction

1.1 Motivation and Problem Statement

In the quest to build intelligent machines, extracting meaning from visual data is a fundamental task. In particular, understanding human activities through computer vision techniques has benefited from the attention of a diverse array of fields including: surveillance, health services provision, smart work environments, generation of natural language descriptions, and many others.
In the past 20 years, situation recognition has evolved into a multi-disciplinary research area requiring the application of techniques from artificial intelligence, computer vision, statistics to even neuroscience. A recurring approaching in the design of situation recognition systems typically involves three major aspects:

- Acquisition of the raw numeric data from sensors, typically camera systems.
- Extracting low level features from the quantitative data and building a model that represents the on-scene developments.
- Comparing the model to some form of pre-existing database of expected situations to identify the situation occurring in the scene.

However, the goal of extracting relevant events in video data is hindered by the inherent uncertainty in the data acquired from the world accompanied by the errors introduced in the low-level image processing phases that are required to transform this data into a representation that can be manipulated by a computer. These challenges hold the charge to leaving semantic interpretation of video data an open area of research in Computer Vision.

In this work, focus is on developing a pragmatic logic-based architecture for situation recognition, and performing evaluation of the same using video data recorded in real world settings.

1.2 Goals and Contribution

The goal of this thesis is two fold. The primary motive is to develop a specification for the performance of a logic-based Cognitive Vision System in real world scenarios. The results from this process yield insight on the extent to which a rule-based situation recognition system can compensate for errors occurring in the low level acquisition and tracking phases of the system.
The second motive is to develop an architecture for a situation recognition system based on SGTs and FMTL that can be deployed in real world scenarios. The developed architecture clearly separates the concerns of knowledge representation from the data acquisition and reasoning portions of the situation recognition system providing for extensibility and separate evolution.

1.3 Thesis Outline

The structure of this thesis is as follows. Chapter 2 is a discussion on the body of knowledge surrounding the area of situation recognition. Chapter 3 is a detailed description of the brand of Cognitive Vision System used in this work and goes into the details of expert knowledge representation with Situation Graph Trees and reasoning with Fuzzy Metric Temporal Logic. The architecture of the software tools developed and used in this work, as well as the methods used to evaluate the situation recognition system are presented in Chapter 4. The procedure for evaluation of the system is also presented here. Chapter 5 presents the results from the evaluation procedure. Finally, Chapter 6 presents a discussion of the results from the work as well as conclusion and suggestions for future directions.

Chapter 2

Related Work

Situation recognition in video data is a broad and active area of research with numerous unsolved problems. In this chapter, we focus on the work that has been done towards situation recognition.
One way of categorizing literature on situation recognition is into single-layered and hierarchical approaches (Aggarwal and Ryoo, 2011). The different methods for situation recognition are summarized in Figure 2.1.

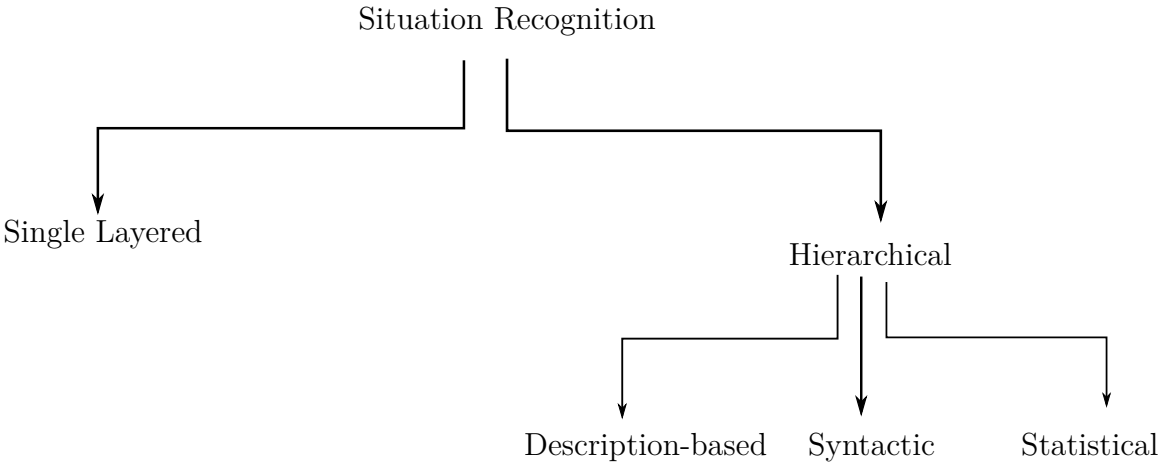


Figure 2.1: Approaches for situation recognition as presented by (Aggarwal and Ryoo, 2011).

2.1 Single Layered Approaches

Single layered approaches work directly with the machine perception data and typically involve some form of machine learning (Aggarwal and Ryoo, 2011) The advantages of these approaches is their relative structural simplicity. However, they are a black box approach and do not provide a clear insight into how they are operating to achieve the results. Moreover, much effort has to go into training data selection and the mechanism used for classification.

2.2 Hierarchical Approaches

Hierarchical methods have one or more additional (reasoning) layers above the machine perception layer. They can further be sub-divided into the general areas of Statistical, Syntactic, and Description-based approaches. Since we utilize a hierarchical method in this work, in the next sections we present details concerning these approaches.

2.2.1 Statistical Approaches

Statistical Approaches are typically hinged on Probabilistic Graphical Models(PGMs) such as Hidden Markov Models (HMMs), Markov random Fields (MRFs), or Dynamic Bayesian Networks (DBNs). Using PGMs, it is possible to model and deduce the joint probability of occurrence of situations from particular observations. An advantage os PGMs is the elegant theoretical framework on which they are based that allows for closed form probability handling from observations to situations.

Robertson and Reid, 2006 presents a system based on HMMs for human behavior recognition in video data using broadcast tennis sequences and surveillance footage as case studies.

In their system, actions are defined as feature vectors. These features are position and velocity (trajectory) data as well as a set of local motion descriptors, and stored in a database that is used for subsequent action recognition attempts (see Figure 2.2). HMMs serve two roles in the system: a) each state in the HMM represents a single activity; b) they provide for smooth transitions between actions in action sequences by keeping track of the current trajectory data, changes in the trajectory data, and the previously observed trajectory data for the action.
Situation recognition is then achieved by searching in a set of pre-defined HMMs for the one with the highest probability of explaining the current sequence of actions. One major drawback of using HMMs for situation recognition is that they require

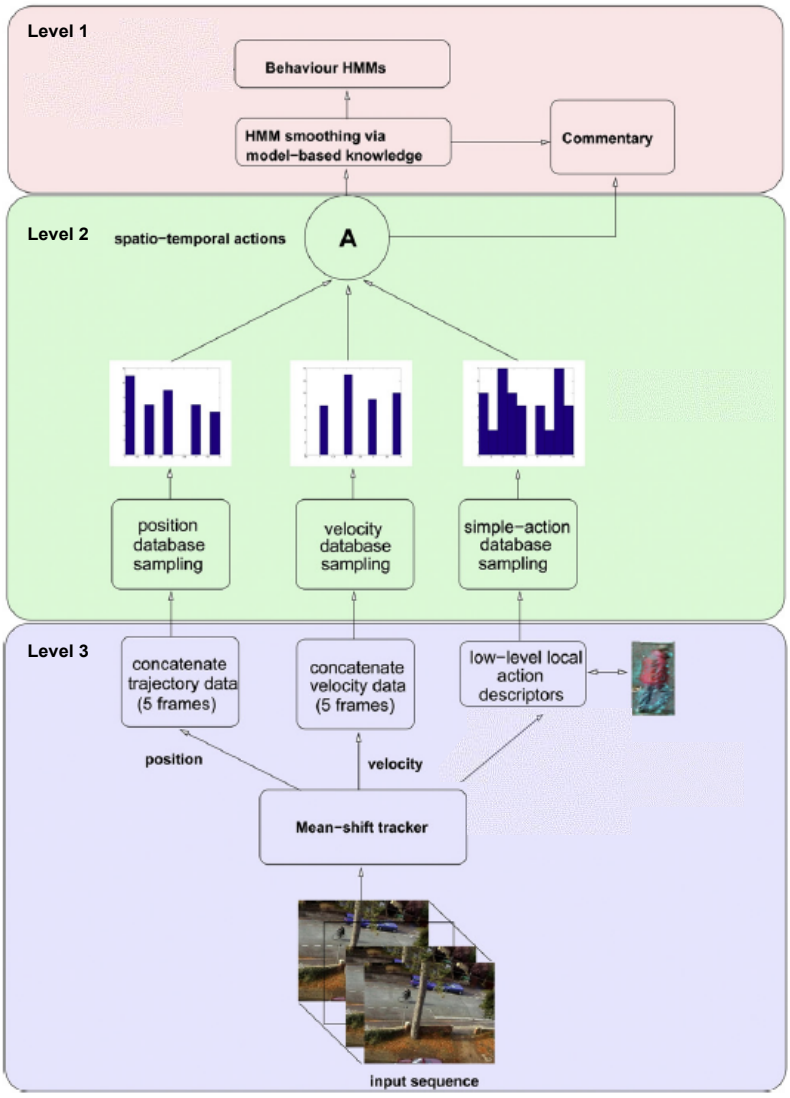


Figure 2.2: Illustration of the work on human behavior recognition by Robertson and Reid, 2006. Level 3 is responsible for extracting low-level features and storing them in a database. Level 2 performs Bayesian fusion of these low-level features to generate spatio-temporal actions. These actions are then combined into action sequences in Level 1 where HMMs are used to smooth them and to perform behavior estimation. Figure from Robertson and Reid, 2006.

independence of the constituents of the feature vectors, which is not always the case especially in real world scenarios. Another shortcoming of standard HMMs is that whereas human behaviour is hierarchical in nature, they do not provide a means for modelling this form of data.

2.2.2 Syntactic approaches

In syntactical approaches, atomic events are combined into complex situations using formal grammars. One implementation of formal grammars are the Context Free Grammars (CFGs) which provide a solid platform for representing structured processes. Using CFGs, composite actions can be depicted in such a way that atomic actions (such as human gestures) serve as terminals while composite actions map to non-terminals (Ye *et al.*, 2012). Production rules provide a means for converting composite actions into atomic actions.
Stochastic Context-Free Grammars (SCFGs) extend CFGs by augmenting production rules with a probabilistic dimension. They have for long been successful applied to the analysis of natural languages primarily due to their suitability in representing hierarchical sentential word structures.
The application of SCFGs to human behavior understanding is contingent on the fact that complex human behaviors are the conglomeration of hierarchies of primitive actions. Implying that by constructing a list of primitive events to be detected, and a set of production rules defining higher-level activities, situations of interest can be extracted from perceived data.
In Kitani *et al.*, 2007, SCFGs are applied to video sequences of typical employee-customer transaction in a store. Primitive actions symbols (such as *TookMoney*, *MovedScanner*, *TookReceipt*) are detected using simple image processing aided by application-specific knowledge. Using these symbols, an optimal grammar describing the interaction is constructed.
SCFGs have been shown to perform well at detecting high level activities as well as dealing with errors in low-level computer vision tasks. However, learning the production rules for SCFGs requires a fairly large amount of training data. Additionally, in complex cases such as multi-agent interactions and overlap between situations, formulate the grammatical rules becomes a central challenge (Ye *et al.*, 2012).

2.2.3 Description-based Approaches

The extraction of substantial situations from the perceived world requires explicit or implicit reference to the time for which and the locations at which situations occurred. Description-based methods encode spatial, temporal and logic properties in a general way. The apply logic methods well established in the Artificial Intelligence

Community (usually as formal logic languages) together with abstract features obtained using appropriate modeling techniques, to provide a platform for situation recognition (Ye *et al.*, 2012). Some of these methods owe their advancement to the development of an explicit temporal logic formalism - Allen's Temporal Logic which makes it possible to specify, constrain, and perform reasoning on temporal sequences between events (Allen, 1983).

Gottfried *et al.*, 2006 demonstrate a description-based approach applied to smart homes. They apply Region Connection Calculus to regions of a room, extracting topological relations between the regions, and acquiring the trajectory data from which activity patterns can be derived and isolated. Figure 2.3 depicts an example of the interplay between the trajectory primitives data, temporal and spatial information that is apparent in their work. Given the acquired region and distance data, combined with the characterized movement trajectory, the trajectory shown at the top of the figure could imply two different situations: a person running to and fro in the room or a deliberate search taking place in the room. This information could not have been apparent if all three modalities considered by description-based methods were not taken into account.

Syntactic Approaches could also be combined with Description-based methods in Markov Logic Networks with a general advantage of being able to systematically deal with uncertainties in the perception data Aggarwal and Ryoo, 2011; Vu *et al.*, 2003.

2.3 The Fuzzy Metric Temporal Logic and Situation Graph Trees Approach

The FMTL/SGT system is an example of description-based approach to situation recognition. It applies Fuzzy Metric Temporal Logic (FMTL) as its logic representation format for both data input to the reasoning system and rules describing generic situations, and Situation Graph Trees to capture the expert knowledge about the domain under observation. The process of recognizing active situations in the data is performed by F-Limette, a reasoning engine for FMTL. Early applications of this system to traffic analysis are detailed in (Arens *et al.*, 2008; Gerber and H.-H. Nagel, 2008; H.H Nagel, 2004). An in-depth treatment of FMTL and SGTs is presented in Chapter 3.
Since the initial demonstration in road traffic analysis, the FMTL/SGT Cognitive Vision System has been applied in a numerous other domains including but not limited to Robotics, Natural Language Description Generation, Football commentary, etc.

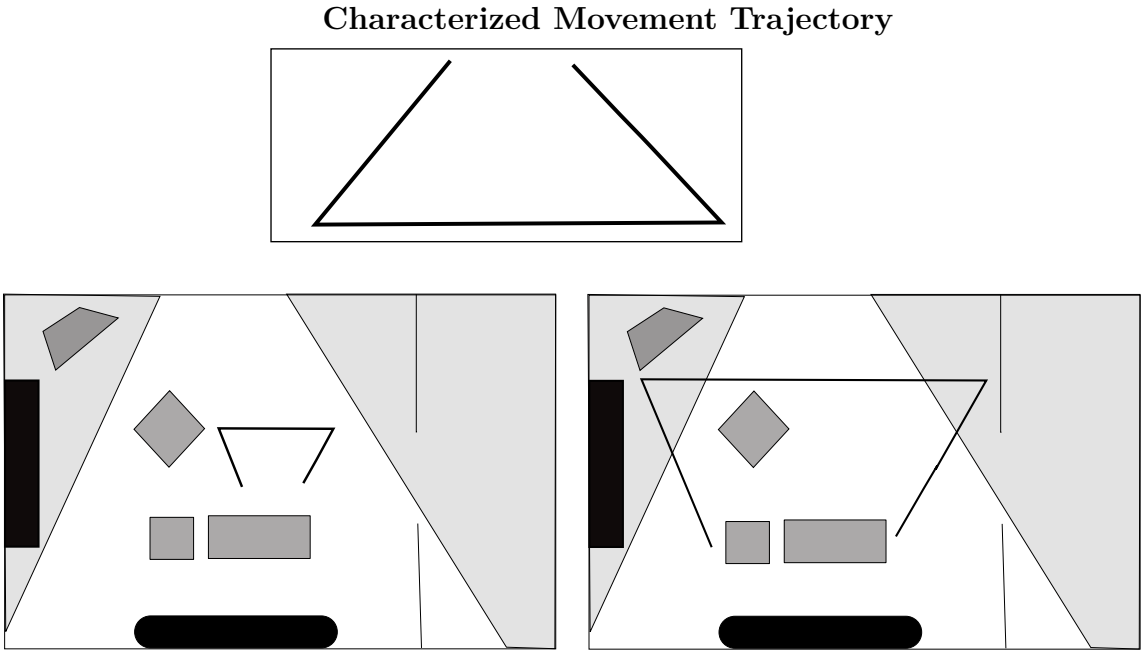


Figure 2.3: Extracting situations based on the combination of trajectory data and distances information(Ye *et al.*, 2012).

2.3.1 Application to Surveillance

In the work of Bellotto, Benfold, *et al.*, 2012, a real-time person tracking system is presented. Their setup, situated in an atrium of an office building, consists of an overhead static wide-angle camera and two Pan-Tilt-Zoom (PTZ) cameras (also referred to as Tracker Active Cameras (TACs)), one on each side of the floor.

An illustration of the system architecture is given in Figure 2.4. The Visual Level constitutes the camera system mentioned above. The Integration Level contains the Supervisor Tracker (SVT) module which encompasses additional modules for the data fusion and camera control. The Conceptual Level is responsible for High-Level-Reasoning. A central piece to the architecture is a SQL-Server database that achieves asynchronous communication between the different layers and their submodules as necessary.

The person tracking procedure starts with the detection of potential human targets in the surveillance area. Images of the scene are acquired from the overhead static camera and processed with the Lehigh Omnidirectional Tracking System (LOTS) algorithm for Background Removal. Estimates of the 3D-position and velocity of

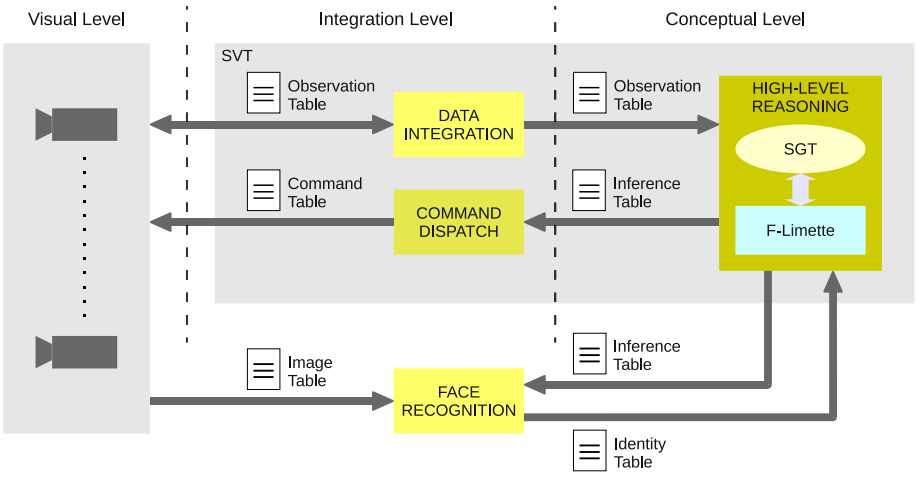


Figure 2.4: Summary of the system architecture for the surveillance system in Bellotto, Benfold, *et al.*, 2012. Figure from: Bellotto, Benfold, *et al.*, 2012.

the targets' heads are then extracted by the data integration module with the help of Kalman filters with a constant-velocity model and nearest-neighbour data association (Bellotto and Hu, 2010).

The domain knowledge, encoded in an SGT, includes the layout of the scene under surveillance and the positioning details of the cameras. The inference engine, F-Limette, uses the SGT and the input data from the data integration module, together with available composition rules, to calculate which of the Tracker Active Cameras should follow the target (see Chapter 3 for details on both SGTs and FMTL). The general criteria for choosing one camera over the other is the combination of the target's moving direction and its supposed destination area. The output of the reasoning engine is in the form of instructions such as *Track Target* or *Acquire Face Image*, that are channeled through the database to the destination camera.

Upon choosing the TAC, the target person is tracked as follows:

i Steer and zoom the view of the camera towards the target's estimated position.

ii Attempt to perform a face detection in that area, if at all the detection is positive, give the person a unique identifier (ID).

iii Instruct the camera to track the target's head (based on the ID) according to the head's estimated position and velocity.

The face recognition module applies an image filter to deal with sub-optimal images, mainly due to motion blur (the person is in motion and/or camera adjustments) and

the fact that there is no guarantee of always having a front facing view of the target's face. This way, only the best suited images make it to the face recognition phase.

Although the system performs as expected in the application presented in the work, a number of issues have to be addressed. First, the system developers need to figure out a way to perform multiple target surveillance as well as develop a mechanism to halt the tracking of the active camera the instant the target leaves the field of view of the cameras. Second, the layout of the scene under surveillance and the instructions that initiate the motion of the active camera are part of the knowledge encoded in the Situation Graph Tree. This implies that transferring the system to another location, or even changing the Camera System would require the construction of a separate SGT. Until these challenges are addressed, it would generally be laborious to apply the system in any scenario other than the one for which it was developed.

2.3.2 Application to Robotics

Bellotto *et al.*, 2012 apply an FMTL/SGT Cognitive Vision System to the domain social robotics. They demonstrate a system that can be used by to perform high-level interpretation of human motion and to subsequently generate appropriate robot control actions. Their work on a level of conceptual interpretation of human motion, as opposed to the typical topological approach that is usually applied in robotics to quantify human motion.
For the representation of human trajectories, they use a 1-D Qualitative Trajectory Calculus (QTC) which quantifies the relative motion between any two agents, in this case the robot and the human agent under consideration Van de Weghe and De Maeyer, 2005.
The output from the reasoning engine is of two forms: a) *STATUS* corresponding to human motion activity, and b) *COMMAND* representing a control action to be delivered to the robot for execution via the database. The distributed architecture, built on the work introduced in Section 2.3.1, used to achieve the trajectory acquisition, reasoning, and control is given in Figure 2.5.

Even though the results presented in the work are promising, the experiments were performed in a simulated environment, there is need to test whether they can be re-produced in a real world scenario. Additional experiments would also focus on deployment in more complex scenarios where interaction commands may be more complicated.

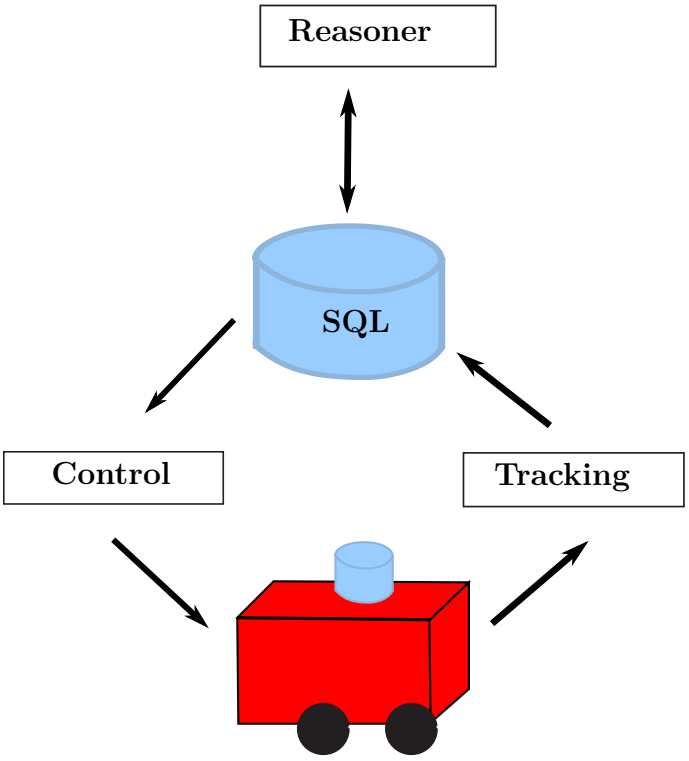


Figure 2.5: Architecture of the system for high-level interpretation of human behavior for a mobile robot. Figure from Bellotto *et al.*, 2012.

2.3.3 Application to Smart Work Environments

Ijsselmuiden *et al.*, 2012 apply an FMTL/SGT Cognitive Vision System to automate the process of behavior report generation in a crisis response control room. In a role playing setting, they study the behavior of control room staff at the State Fire Service Institute North Rhine-Westphalia in response to a train accident.

The multimodal data was acquired as follows: the 6 hours staff exercise was recorded by five cameras were used to record the entire room and the audio data was acquired using four microphones positioned around the room.
The (Hypothetical) machine perception data that is the input to the reasoning process was then obtained by a dedicated manual annotation tool. This allows their evaluation to focus on the reasoning methods without errors that would be introduced if the input data to the system were a model built from the acquired, processed, and fused audio visual data. Spatial-temporal information on person poses, gestures, geometries, and trajectory information, together with data on pertinent

inanimate objects such as clipboards and white-boards, is used to model various group situations as they occur in the room. Evaluation of the reasoning system proceeded by comparing the output from the system to the ground truth annotated with the annotation tool as illustrated in Figure 2.6.
The reports and visualizations generated provide insight on a number of fronts for instance, whether standard operating procedures are being followed, group allocationas and potential re-allocation, individual task completion rates, and resource usage. Generally, this information could be used as a means to increase the effectiveness of learning from recorded staff exercises. It could also be combined with other context information about the real scene to provide a repository of knowledge that could be applied in making critical operation procedures.

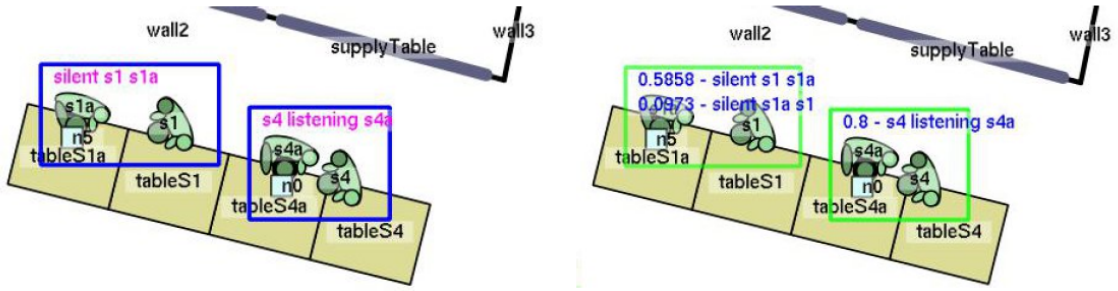


Figure 2.6: An example of manually annotated ground truth results (a) and the corresponding reasoning results (b). The numbers in (b) indicate the degree of confidence in the results from the reasoning process.)

One major drawback of this system is that it requires a lot of time and human effort for the frame-by-frame annotation process rendering it challenging to apply in a scenarios involving a large number of agents. The next evolution of the system would therefore be to replace the annotation tool with a model that reliably incorporates the audio and video data into a form suitable for input to the situation recognition system.

The work's contributions in the area of automatic behavior understanding can be summarized as follows: addition to the knowledge-base for modeling group situations, an evaluation of the FMTL/SGT-based reasoning methods concerning the robustness of of the system on artificial noisy data, and advancement of tools for results and ground truth annotation.

Chapter 3

Background

This chapter presents the fundamental underpinnings of situation recognition based on FMTL and SGTs. First, the Cognitive Vision System, a generic framework on which the FMTL/SGT reasoning system is based is introduced. Then, FMTL is presented as the form for representing expected situations and the situation recognition algorithm, and the traversal rules that are applied to the inference engine as it queries for potential situations occurring in the scene under consideration. Finally, Situation Graph Trees, the hypergraph structures used to model the knowledge base under consideration, are presented.

3.1 Cognitive Vision System Architecture

Building a cognitive system is no straight forward task. Over the past few decades, a number of cognitive architectures have sprung into existence to guide the development of such systems Thórisson and Helgasson, 2012. One such architecture is SOAR that applies a sense-decide-act cycle, and another is NARs that is based on a hierarchical non-mathematical logic framework. Both architectures have their limitations, the former being incapable of real-time operation and the latter being only partially implemented. The architecture utilized in this work is the Cognitive Vision System (CVS) and is described at length in H.H. Nagel, 2000.

We consider an instance of the CVS utilizing FMTL as the formal representational logic formalism, and SGTs as the form used to portray the knowledge, in terms of situations, about the domain under study. To a large extent, the current status of this Cognitive Vision System architecture can be ascribed to the work by David Münch, 2013. This work introduces developments in the architecture to allow for universal knowledge representation. Additionally, it extends it with natural language capabilities to augment the de-facto vision modality. A further refinement on

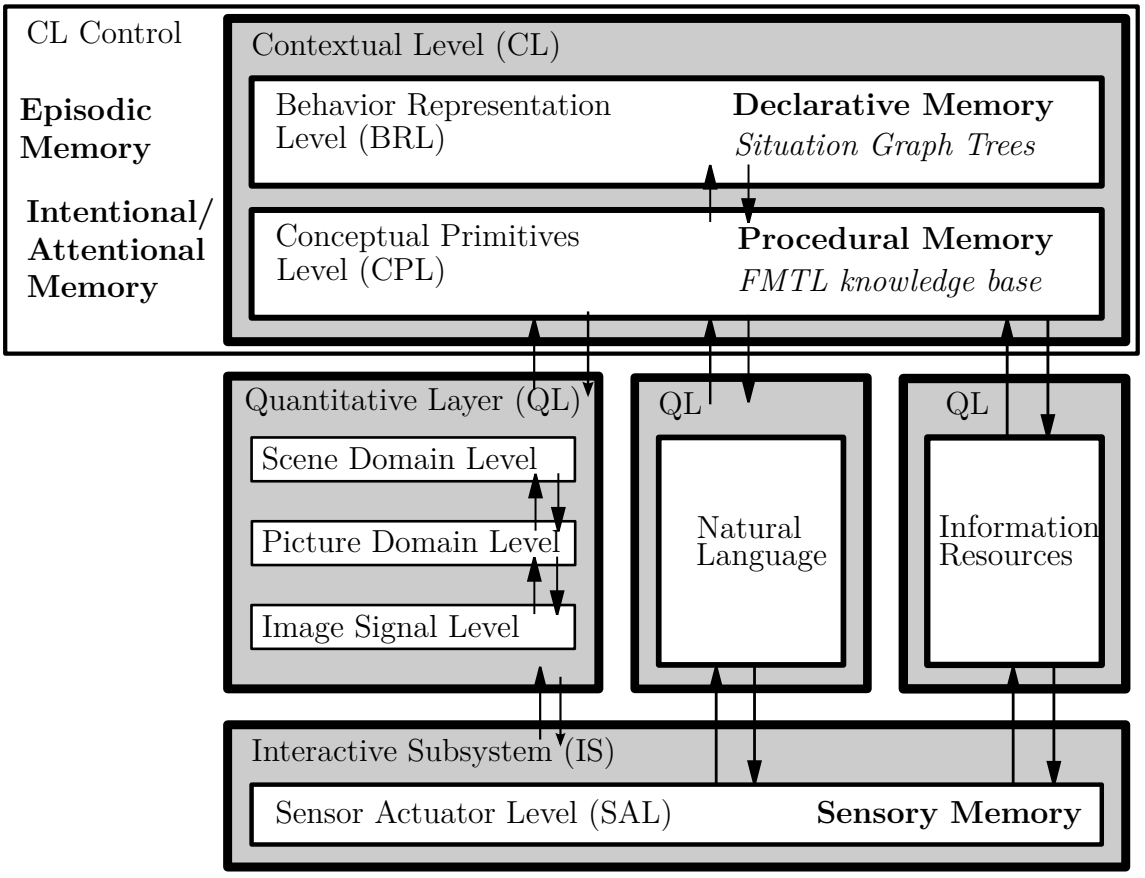


Figure 3.1: The Cognitive Vision System architecture. Figure from David Münch, 2013.

this work by D. Münch, Jüngling, *et al.*, 2011 allows for fuzzy traversal making possible multiple hypothesis inference and guaranteeing consistent propagation of truth values throughout the reasoning process. Complex domains such those presented in Section 2.3 employ this multi-hypothesis search strategy.

Other improvements in the CVS for real-time operation include: parallelization to reduce the run time of the inference process and knowledge sharing (D. Münch, Becker, *et al.*, 2012). And an attempt to combat noisy input data based on a fuzzy uncertainty propagation model, and explicit temporal modeling is detailed in D. Münch, IJsselmuiden, *et al.*, 2012.

Figure 3.1 is a schematic illustration of the CVS architecture. In the architecture, three major levels can be discerned: the Interactive Subsystem, The Quantitative

Level, and the Conceptual Level. Each of the levels consists of one or more sub-layers. In the subsequent discussion, the concept of layer memory is introduced to aid the discussion by drawing parallels with the categorization of human memory in neuro-science.

The bottom-most layer, the **Interactive Subsystem (IS)**, embodies all information exchange with any sensors and actuators used for interacting with the environment. The memory type associated with this layer is Sensory Memory which is a form of short term memory.

The **Quantitative Level (QL)** is positioned in the middle of the layer stack. Components that are specific to the problem being studied are to be found here. A tracking module for computer vision data would be found in this layer, and so would a data-mining engine for inference on cloud-based data sources. For the case of vision-based system, the QL consists of the Scene Domain, the Picture Domain and Image Signal sub-layers.

The **Conceptual Level (CL)** is built from the quantitative information obtained from whatever modules are positioned in the QL. As the inference must be based on semantic information rather than qualitative data, the numerical data is first converted into basic knowledge units (primitives) in the Conceptual Primitives layer (CPL). This knowledge is then expressed in a form on which reasoning can be performed. For this work, this representation is Fuzzy Metric Temporal Logic (FMTL). FMTL provides a mechanism for the conversion of quantitative data together with its associated degree of validity and temporal modality into concepts. Figure 3.2 is a graphical depiction of trapezoidal membership functions that could be used to assign degree of validity values to the discrete concepts: *zero*, *small*, *text*, *normal*, *high*, and *very_high*, relating to the speed of an agent.

The memory associated with the CPL is the Declarative Memory in analog to the human memory of facts and events. The topmost layer in the hierarchy is the Behavior Representation Level (BRL) which contains high-level knowledge about the domain. In this work the BRL is an abstract hyper-graph structure known as a Situation Graph Tree (SGT.) The SGT can very easily be written to FMTL enabling for straight forward interaction of the BRL with the lower CPL layer.

An additional layer in the CL is the Control Unit in the CL. Here, the goals of the inference process are defined. The memory here therefore corresponds to Intentional Memory.

High-Level Inference using the CVS described above is a parallelized agent-based query performed on the knowledge in BRL. The results of inference are stored in an

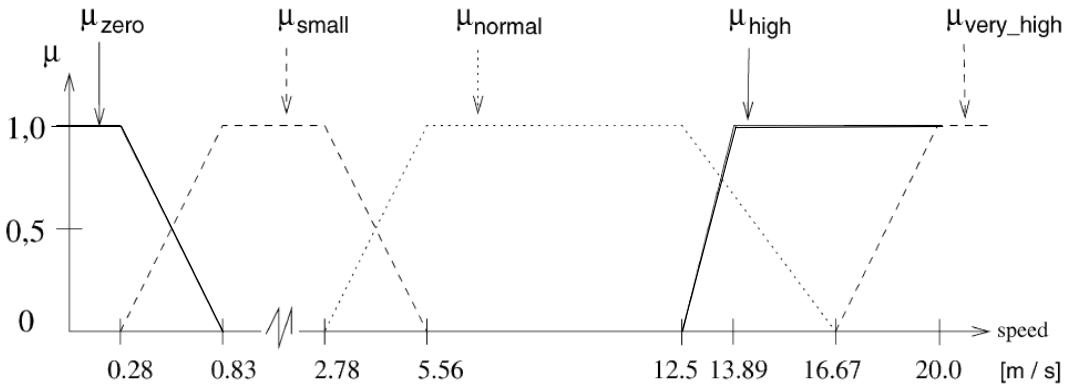


Figure 3.2: Membership functions to assign degrees of validity for concepts related to the speed of an agent. Numerical figures on the horizontal axis are only for illustration purposes. Figure from Gerber and H.-H. Nagel, 2008.

Episodic Memory (long term memory), from where it can be accessed and used in any further processing steps.

3.2 Fuzzy Metric Temporal Logic

Fuzzy Metric Temporal Logic (FMTL) is the language of the inference engine used with FMTL/SGT Cognitive Vision System. The inference engine, F-Limette, is called upon during the reasoning process to yield all situations that can describe the occurrence(s) for a particular instance. This process of query for potential situation instantiations is termed *traversal*.

FMTL extends First Order Logic with: a) A fuzzy component to cater for the degree of confidence in the value of a given predicate, b) A temporal aspect to allow for reasoning on time, and c) a metric on the temporal aspect.

The **Fuzzy Component** of FMTL facilitates the modeling of inherently vague concepts with truth values other than the customary binary "0" and "1" (Arens, 2004). With this representation, concepts such as *Walking too Fast* or *Near one Another* can be modeled. The appropriate truth values, between one and zero, are associated by the means of a membership function drawn on the numeric value of the concept being modeled. For example in the case of *Near one Another*, a trapezoidal membership function (similar to the one represented in 3.2) could be applied on the numerical value of the distance between the two agents under consideration.

Additionally, fuzzy representation comes in handy as a tool to model uncertainty in the input data by simply assigning lower truth values to input data for which we have low confidence (or even "0" if the data is completely missing) such as in D. Münch, IJsselmuiden, *et al.*, 2012.

Besides the fuzzy component, the second extension, the **Temporal Aspect**, provides for the modeling of developments in time other than just a particular timepoint. FMTL also distinguishes itself from Allen's Temporal logic by introducing a metric on time Allen, 1983. This allows for reasoning on about exact differences in time in on top of the categorical "before" and "after". This temporal modality is a cornerstone for modeling both aspects that change over time such as the speed of a person, and for inference on multi-phased situations. Additionally, this treatment of temporal behavior can be used a mechanism for smoothing data against noise, outliers, and ignoring short-lived changes typically associated with real-world data.

3.3 Situation Graph Trees

In order to perform inference on the behavior of agents in a an observed scene, it is imperative to generate knowledge describing he agent and patients with regards to geometries and trajectories. As was discussed in Section 2.2.1 this can be done in a number of ways. While many approaches draw the model directly from actual observations, in many scenarios it is desirable to have the model as an a-priori representation of the expected developments in the scene drawn from experience about the domain. Situation Graph Trees are one such representational form (H.H Nagel, 2004).

In theory, there is no limit to the complexity of the behaviors that can be modeled using SGTs – adding more layers is straight forward. However, this increase in complexity typically translates into traversal time that could present a challenge, for example, if real time performance is a requirement. It is then up to the domain expert to be as concise as possible when drawing up the SGT for the pertinent domain. Situation Graph Trees are tree structured hyper-graphs that can be employed to represent, in schematic form, the knowledge about behaviors occurring over a temporal window in a scene (Arens *et al.*, 2008).

For the traversal phase of the FMTL/SGT Cognitive Vision System, SGTs are encoded in FMTL. This essentially blurs the semantic gap between the knowledge about the scene, the form of the input data required for the reasoning process, and the rules governing how this data is to be aggregated when constructing higher-level concepts.

Note that in this discussion, an agent refers to the subject of interest (usually a specific person,) while a patient is any other entity that the agent interacts with (could be another person or an inanimate object).

3.3.1 Situation Schemes

In an SGT, each situation is represented by a single situation scheme that describes the state of a given agent or group of agents of interest for a specific time-point. It consists of a a unique name, a state scheme which contains one or more logic predicates that must be satisfied for an agent to be in that situation, and an action scheme that outlines the actions that an agent has to carry out when in that state. Each situation scheme could be a start and/or an end situation. Figure 3.3 is an example of a situation scheme that could be used to represent the act of two people approaching one another.

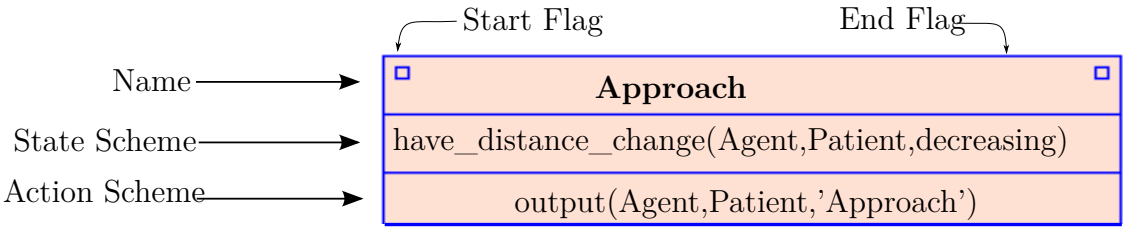


Figure 3.3: Illustration of the component parts of a situation scheme. Both state and action scheme predicates are indicated, and this particular situation is marked as both a start and potential end situation.

3.3.2 Situation Graphs

A Situation Graph (SG) is a container of situation schemes that could be connected together through prediction edges. A Prediction Edge is a statement of temporal resolution. It describes all potential situations for the subsequent time-point. In the example of a situation graph is given in Figure 3.4 where the prediction edges are represented as blue directed edges. A prediction edge could be cyclic, and could optionally specify information for one or more bindings.

A binding represents a variable assignment for a predicate-based search. During the reasoning process a predicate variable could take on a particular value. For that same predicate to be used to search for more potential agents or patients that could

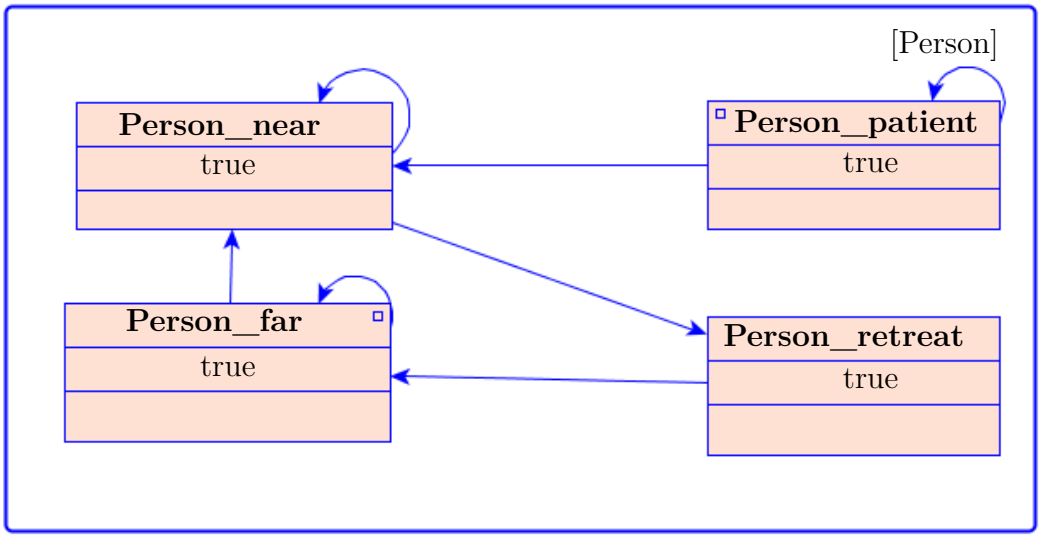


Figure 3.4: A simple situation graph that could be part of an SGT for modeling some aspects of human motion behavior. The Person_patient situation is a start situation, while the Person_far situation is an end situation. Prediction edges are represented by the blue arrows, and any bindings are shown as labels on the pertinent prediction edges

satisfy the predicate, a binding release has to be performed otherwise the predicate will carry the same value(the initial one) it get assigned. Binding release is only required in the context of temporal refinement and therefore binding statements are tied to prediction edges. An example of a binding release is shown in Figure 3.4 for the *Person_patient* situation. In this case the binding is on the cyclic edge of a start situation. At each time step the reasoner tries to identify new patients that could be in a situation with the current agent.

Situations within in a situation graph could be further refined by more specific situation graphs through Specialization Edges. Unlike prediction edges that connect situations at different time-points, specialization edges point to situation graphs with more specific information for the same time-point as the situations which they specialize. This is known as conceptual refinement.

In the process of knowledge representation, all relevant situations are drawn and arranged into situations graphs. Within the SGs, there could be temporal progressions between the situations, and some situations are defined recursively further by entire situation graphs (external to the parent SG). Put together, these components naturally align into a tree-like structure, such as the one depicted in Figure 3.5 - the Situation Graph Tree.

3.3.3 Constructing Situation Graph Trees with SGTyEditor

One advantage of using SGTs in the FMTL/SGT CVS is stated as providing a form that is both intuitive and provable. To support these goals, software tools have been developed for the creation and modification of SGTs, the earliest being the SGTEditor (Arens, 2003) implemented in Java on top of the Diagen diagramming tool. On top of providing SGT creation capabilities, the SGTEditor integrated modules to facilitate the traversal of SGTs with the inference-engine F-Limette.
The latest iteration of SGT creation and manipulation tools, is the SGTyEditor. The SGTyEditor is also a java-based tool built with the aim of making the process of developing an SGT more tractable. Its graphical core is based on the more advanced yEd Graph Editor. It consists of a design surface onto which the different components of an SGT, mentioned above, can be drawn. The action and state schemes for each situation can be set, prediction relations between situations can be represented, and specialization edges between situations and situation graph trees represented. Initial work on the SGTyEditor was done in the work by Bauer, 2012.

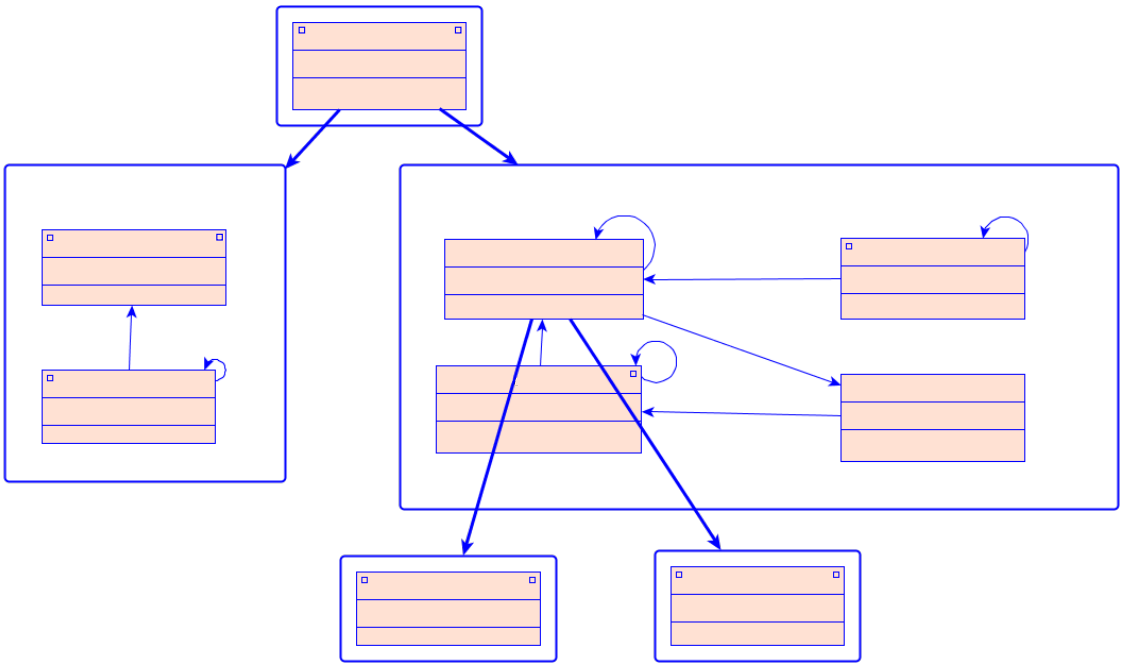


Figure 3.5: A schematic of a simple Situation Graph Tree. Note that the situation names, action and state schema, plus bindings have been excluded for brevity. In the figure Situation Graphs are bounded by blue thick-border rectangles, prediction edges blue thin edges connecting situations in the same Situation Graph, while Specialization edges are indicated by thick edges pointing from a situation to a Situation Graph.

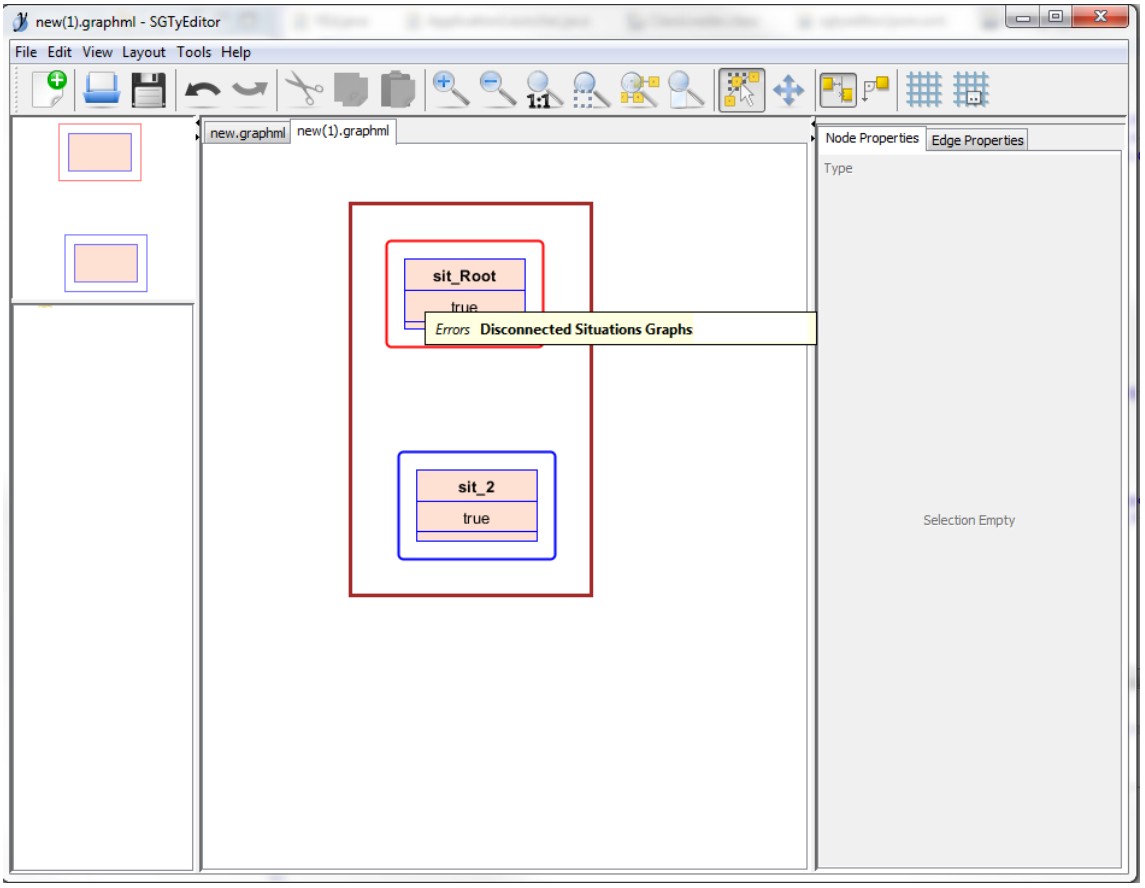


Figure 3.6: SGTyEditor demonstrating on-the-spot SGT validation.

Furthermore, the current edition of the SGTyEditor incorporates both SGT validity constraints as well a real time validation engine that provides on-the-spot feed back on the validity of the current SGT (see Figure 3.6). These additions were added by this author of this thesis in advance of the current work. Once the SGT design process is complete, it can persisted to disk in a form that is ready for input to the reasoning framework.

The developed SGT is then represented in an Ontology (OWL) and stored in the GraphML format (Bauer, 2012).

A key benefit of the SGTyEditor, is that unlike its predecessor, it offers clear separation of SGT creation and validation from traversal. This in essence yields two advantages: a). both tools to evolve separate of each other and b). The SGTyEditor could be shipped to domain experts to represent knowledge for their application and

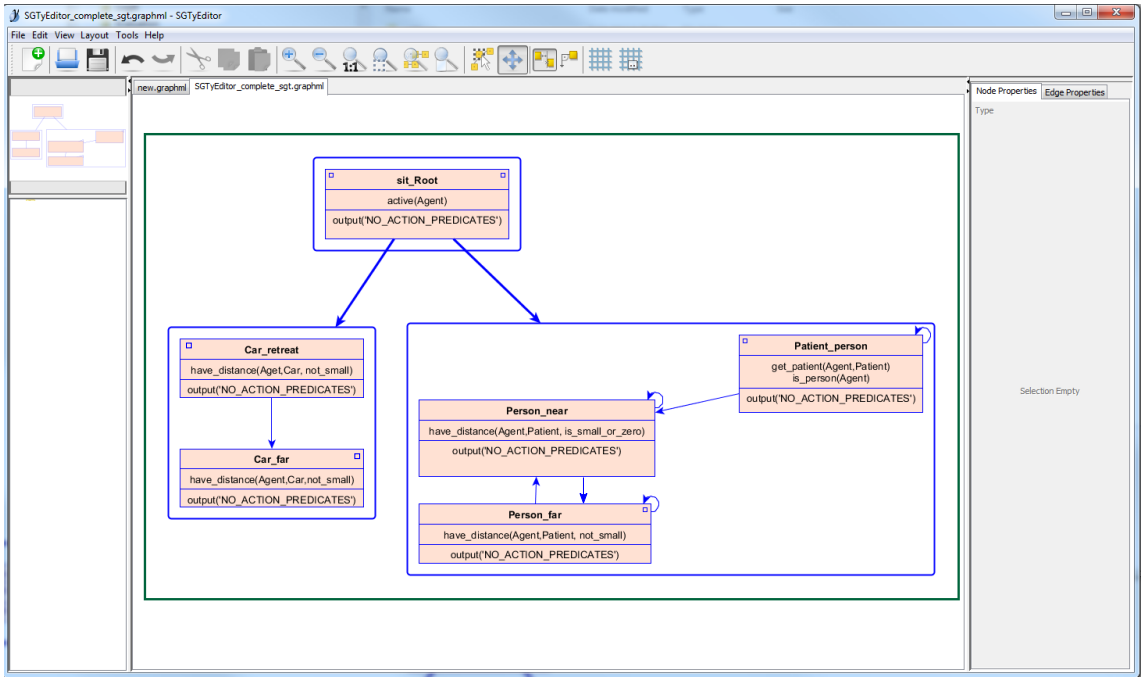


Figure 3.7: The SGTyEditor with a complete simple SGT in the display pane. Some tool panels are collapsed to focus on drawing area.

they need not be bothered about details of the inference process. The SGTyEditor tool is shown in Figure 3.7 with an example of a complete SGT.

3.4 Reasoning with SGTs and FMTL

The procedure for reasoning is initiated by feeding the annotated data, rule files, together with the SGT, all in represented in FMTL, into the reasoning engine, F-Limette. The reasoner then performs traversal time point-by-time point, using Algorithm 1, to find all possible situation instantiations (Arens *et al.*, 2008).

Algorithm 1: Fuzzy Situation Graph Tree Traversal (D. Münch, Jüngling, *et al.*, 2011)

```
   Input: SGT, object
1  if object occurs for the first time then
2      G ⟵ SGT root graph;
3      forall the s|s ∈ G ∧ s is start situation do
4          if s can be instantiated then
5              forall the spec|spec ∈ s ∧ spec is specialization do
6                  s := spec, G := graph containing spec;
7                  start recursion goto line 3;
8              evaluate actions of s;
9  else
10     forall the predSit|predSit is prediction situation of s (the last situation
       of the already known object) do
11         if instantiate predSit successful then
12             s := predSit, G := graph containing predSit;
13             start recursion goto line 4;
14         else
15             if predSit is end situation ∧ predSit ∈ G then
16                 instantiation successful;
17             else
18                 instantiation failed;
```

The traversal process proceeds as follows:

Starting with root situation graph of the SGT root, if the state scheme of the of the root (aslo start) situation scheme is instantiable (see line 3f), the algorithms goes ahead to search for specialization edges. If any are found, the traversal continues recursively (line 6f)starting with the start situation scheme in each specialized situation graph.

If none are found, or none are instantiable, the traversal steps one time point ahead and attempts to instantiate situation schemes as dictated by prediction edges (lines 10f) until an end situation is reached(line 15f). Throughout the traversal, all potential specializations and temporal developments are considered.

Once the traversal of the most specific instantiable specialization in each path is complete, the traversal then works backwards to more general situation until the end situation scheme in the root situation graph is reached, at which point the traversal terminates.

Chapter 4

Architecture and Methodology

4.1 System Architecture Overview

This section contains the core of the architecture developed for situation recognition. An overview of the architecture is illustrated in Figure 4.1.

The Situation Recognition Framework developed as part of this work separates the tools of knowledge representation from those of reasoning. This allows for both to be improved separately. The knowledge representation part of the framework, the SGTyEditor, can be handed to domain experts who can represent the knowledge for the particular scenario without the clutter or even knowledge of the reasoning component. This separation of concerns is important for the adoption of the SGT/FMTL-based situation recognition system in multiple domains.
The architecture of the system consists of the following components:

- **Relational Database Management System (RDBMS)**: This is the MySQL back end database that drives the whole system. Traversal data is stored here, situation instantiations are also persisted here. The database is a key extension point to the system. It essentially allows for the support of infinitely many sources of data for as long as they can be transformed into the format that is expected by the traversal system. As an example, in this work, for some experiments data was drawn from the LPM measurement system, for others from person tracking module, and even from text files for example for the VIRAT video dataset.

- **Knowledge Representation**: This encompasses development of SGTs with the SGTyEditor described in details in Section 3.3.3. And development of the rule base to be used as the criteria for situation instantiation.

- **Reasoning**: Loads an SGT, the related results data, and rule files and initiates the reasoning process. The traversal module therefore also houses a wrapper to the F-Limette inference engine with convenience methods for invoking and getting responses from the reasoner.
- **Analysis**: This consists of: i) A comprehensive extensible **Evaluation Framework**. Once a traversal is done, the ground truth can be loaded and together with the results from the traversal, the metrics for the performance of the system can be drawn. More details on the evaluation procedure are given in Section 5. To ease the evaluation process, the framework provides both of single-shot and a batch mode evaluation submodules. ii) A Visualization Module which gives a real time visual update of the status of the traversal. The ground truth data and any instantiations available are annotated on the image frame corresponding to the time-point currently being queried during traversal.
- **Polling Mechanism**: This is applicable in real-time mode. The module periodically checks the database for any new evidence to be used in the reasoning process.

Since evaluation of real world data is a core of this thesis, in the next section we go into depth on the procedure that was applied to perform evaluation on the different datasets.

4.2 IS modules in the SRF

As has been discussed in earlier sections, the task of automatically determining tracks and geometries of people and objects is a major challenge to situation recognition systems. In this thesis, we integrate a robust person detector and a Local Position Management into the situation recognition system. With this in place, the focus can shift to modeling the knowledge about the domain under investigation and generating the rules for conversion of the tracks from the person detector into concepts that can be used with the reasoner.

4.2.1 Person Tracker

Having a person tracker integrated into a situation recognition system presents the ability to perform high-level reasoning on video-data recorded for numerous scenarios.

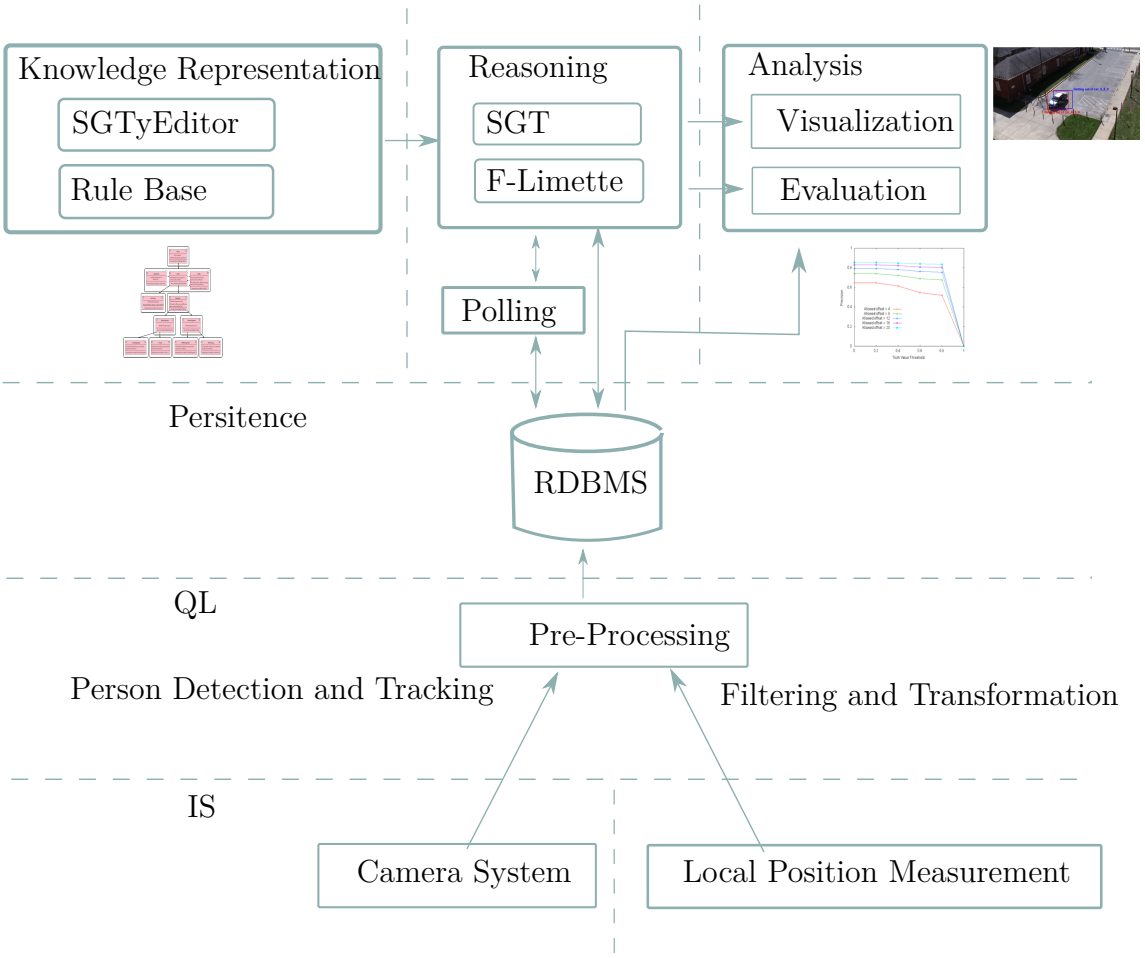
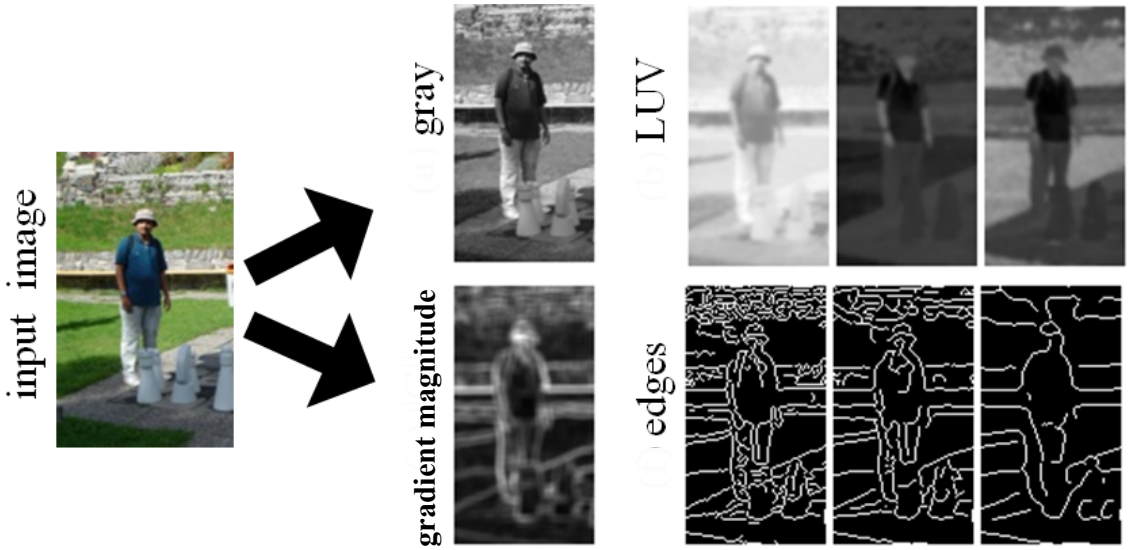
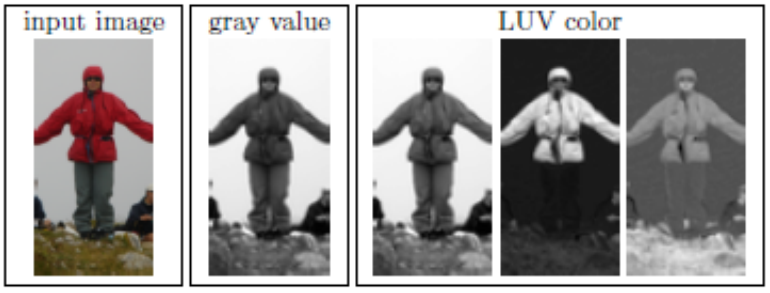


Figure 4.1: Architecture of the situation recognition system developed in this work. The lowest level contains modules developed for the Interactive Subsystem (IS) of the CVS, one for person detection and another for handling trajectory data from the Local Position Measurement system. In the Quantitative layer (QL) above the IS, the pre-processing module that transforms the numerical data acquired in the IS into a form that can be easily converted to concepts is introduced. The upper most layer adds an Analysis module that is responsible for evaluation and visualization of the system. The different modules of the architecture can store and retrieve data, and communicate through a Relational Database Management System.

The person tracker was used to extract geometries and tracks from the video sequences of the IOSB VCA dataset. The tracker uses a transmodal classifier that is able to detect persons in visible and infrared images. The classifier is based on integral channel features such as gray value, LUV color and gradient magnitude that are extracted from image channels. (Dollár *et al.*, 2009). Examples of both image channels and integral features are given in Figure 4.2.



(a)



(b)

Figure 4.2: (a). Examples of image channels (Dollár *et al.*, 2009) from which the integral channel features (b) used to build the weak classifiers for the tracker in Kieritz *et al.*, 2013.

The classifier is trained using (Viola and Jones, 2001) and applies a soft-cascade mechanism to quickly detect regions in the image that do not need to be considered

when searching for persons.

4.2.2 Local Position Measurement Data Module

Although, the person tracker presented in Section 4.2.1 is robust, it may not always provide the most accurate ground truth information. The work in this thesis introduces the ability to acquire and perform inference on data from a Local Position Measurement system, that accurately determines the locations and speeds of people during the recording of scenes. The LPM allowed the precise determination of ground truth data during the recording process. This essentially implies accurate ground truth for the scene under observation. The details on how the LPM system is setup and configured to record person geometries and trajectories are given in Section 5.2 and Appendix A.1.2.

4.3 Evaluation Framework

The evaluation framework incorporated into the architecture described in Section 4.1 above employs the same method followed in Oh *et al.*, 2011 - an interval based approach.

4.3.1 Event Level Matching Criteria

The Event Matching Criteria specifies all the conditions that need to be met before a detection (D) is said to match a ground truth event (G). The criteria is as follows:

i) **Spatial Match:** Detection D is regarded as a match for ground truth G if the intersection ratios for every bounding box pairs per frame are over 10%. Equation 4.1 gives the intersection ratios.

$$\text{Intersection Ratios} = \frac{\#\text{ of intersected pixels}}{\text{Total }\#\text{ of pixels in Bounding Box}} \tag{4.1}$$

ii) **Temporal Match:** Down-stream and up-stream temporal intersections between D and G should be more than 10%. A spatial match between D and G is a pre-requisite for a temporal match. The two required temporal intersection rations are computed by dividing the duration of the temporal intersection by both durations of D and G.

iii) **Label Match:** On top of satisfying the spatial and temporal matching criteria, the event labels assigned to both D and G should be the same for a complete match, i.e. if D is labeled *Grouping* then so must G.

With this criteria, recognized situations (events) can be sub-categorized into true positives (TPs), false positives (FPs), false negatives (FNs), and true negatives (TNs).

4.3.2 Evaluation Metrics

The following interval-based event matching criterion and event-level metrics are considered.

i) **Precision:** The precision is computed as follows:

$$\text{Precision} = \frac{\text{\# of true positives (TPs)}}{\text{Total \# of Detections (TDs)}} \tag{4.2}$$

ii) **Recall:** Also called the Probability of Detection (PD), the recall is computed using the formula:

$$\text{Recall} = \frac{\text{\# of true positives (TPs)}}{\text{Total \# of Ground Truth Events (T)}} \tag{4.3}$$

iii) **F-Score:** Captures the summary capability of the situation recognition system and is computed as the harmonic between the Precision and Recall.

$$\text{F-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

The output, in both text and JSON formats, allows for charting and any other analysis as may be required. Additionally, the framework is built with scaling in mind; evaluation of any additional datasets is straight forward.

4.3.3 Correct Detection and False Alarms

In the following discussion, bounding boxes representing detected situations are drawn in red while those for the ground truth are drawn in blue. Detected situations are labeled with D followed by a number e.g. D1, similarly query activities are labeled with Q followed by a number.

It is also worth mentioning that in each case, sufficient temporal overlap is a requirement.

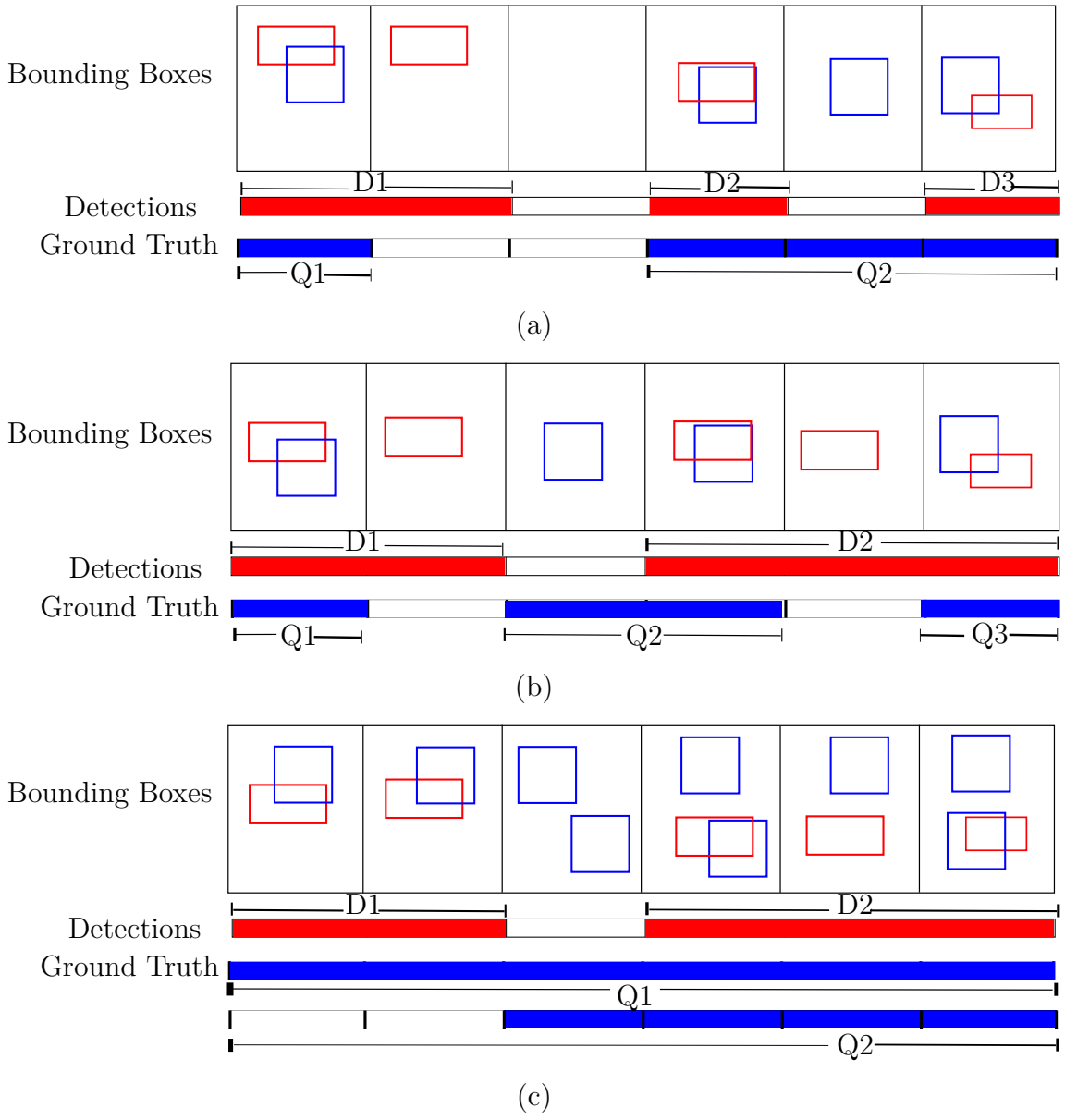


Figure 4.3: Criteria for counting correct detections. Bounding boxes for the ground truth are in blue while those from the situation recognition (detected) are in red Oh *et al.*, 2011.

i) **Correct Detections**

a) A situation from the ground truth may be matched by my detected situations. This counts as a single hit for the ground truth situation (see Figure

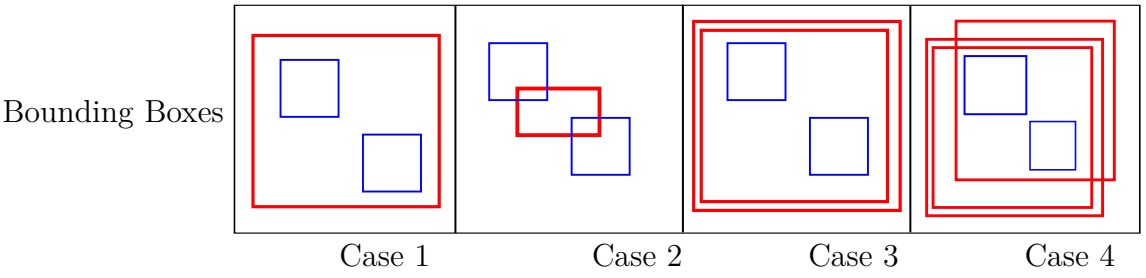


Figure 4.4: Criteria for scoring correct detections for the case of overlapping bounding boxes Oh *et al.*, 2011.

4.3 (a)).

b) As illustrated in Figure 4.3 (b), a detected situation could match multiple situations in the ground truth. In this case, one detection contributes to multiple correct detections.

c) Multiple detections and ground truths can occur concurrently as depicted in 4.3 (c). Here the scoring for all detections and ground truths is done according to the scoring criteria in (a) and (b).

d) For the case of overlapping boundary cases, the following four cases shown in Figure 4.4 :

Case 1: A single situation detection encloses two ground truth situations: both detected situations count towards a correct detection

Case 2: A single detection overlaps two ground truth situations: both detections count towards a correct detection.

Case 3: Two detections enclose two ground truth events: both detections count towards a correct detection.

Case 4: Three detections enclosing two annotated query events: two detections count towards correct detections.

ii) **False Detections** occur when detected situations do not map to any existing ground truth. An illustration of such a scenario is given in Figure 4.5.

For this work, the evaluation framework explained above is extended with an additional frame offset dimension. This is necessary as the multi-hypothesis search for situation instantiations leads to situations either being instantiated earlier or even later than they actually occur.

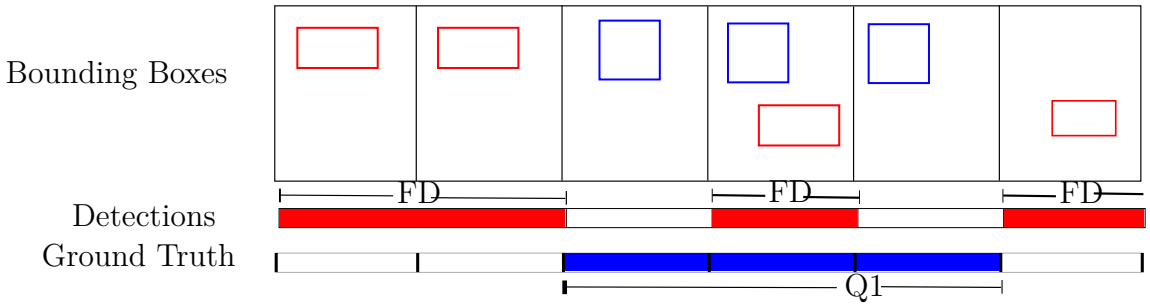


Figure 4.5: Basis for scoring false detections Oh *et al.*, 2011.

4.4 Real-Time Operation

In real-time mode, the Local Positioning System (detailed in Appendix A.1.2) produces information on tracks and geometry of the people and/or objects involved in the particular scene. An active camera system is used to get the video data of the same scene. This data is stored into a MySQL database as it comes in. A polling mechanism detects the availability of new data, and a trigger kicks off the F-Limette inference is to perform traversal situation queries on this new information. The results, representing all potential situation instantiations, each with the corresponding degree of validity, are stored into the database and can later be fed into the integrated evaluation sub-system to get metrics on how well the inference is progressing. This way the Situation Recognition Framework can be used to detect situations that are occurring in the scene of interest. Use of the MySQL database makes it is possible to have the computer vision sub-system and the reasoning sub-system in locations that are remotely situated relative to each and still obtain near real-time performance.

Chapter 5

Results and Evaluation

This section presents the results of performing evaluation on three different video datasets; two standard datasets, and one dataset that was recorded as part of the work for this thesis. Additionally, for the case of the custom dataset, activities pertinent to the creation of the dataset and obtaining the ground truth are presented.

The evaluation procedure in the output of the SGT/FMTL cognitive system is fundamentally different from a typical multi-class classification problem. Evaluation for this system, requires that the generic situation representation must be correctly instantiated; all required actors and objects together with their correct configuration must present in the reasoning result as well as in the ground truth.

It is worthwhile to mention that this thesis does not concern itself with the knowledge representation phase (SGT creation) of the situation recognition hierarchy. Knowledge for the situation evaluated in this work was already developed in previous works and was hence just used in this work so as to focus on the evaluation of the situation recognition system in its entirety.

5.1 Standard Dataset Evaluation

The evaluation in this work focused on situations involving human behavior, these events are relevant for numerous application including: surveillance applications, smart work environments, team activities e.g. sports, and many others. The reference datasets considered are the BEHAVE Interactions Test Case Scenarios dataset and the VIRAT dataset. These two datasets were chosen because they provide a wide variety of events and scenes. In the following sections, we present the datasets in detail, together with the results from the evaluation process as described in Section 4.3.

Video	Number of Frames	Label
1	9,075	VIRAT_S_000002
2	20,940	VIRAT_S_000003
3	17,640	VIRAT_S_000004

Table 5.1: Video Sequences evaluated for the VIRAT Dataset.

5.1.1 VIRAT Dataset

The VIRAT dataset is composed of 16 outdoor scenes of events carried out by non-actors in the real world, and aerial datasets collected with unmanned aircraft. With over 20 hours of video data, the dataset can be used as the basis for a grounded evaluation process. VIRAT's true benefits lie in its variety of events involving multiple agents, scenes with stationary and moving vehicles and recordings at various locations Oh *et al.*, 2011.

5.1.1.1 Evaluation Setup

The SGT representing the knowledge about the situations for the VIRAT dataset is given in Figure 5.1.

The evaluation was done for the Load_Object, Unload_Object, Get_Into_Car and Get_Out_Of_Car Events. The video clips chosen for the evaluation are given in Table 5.1. In the discussion, we refer to the videos by the number given in the *Video* column of the table.

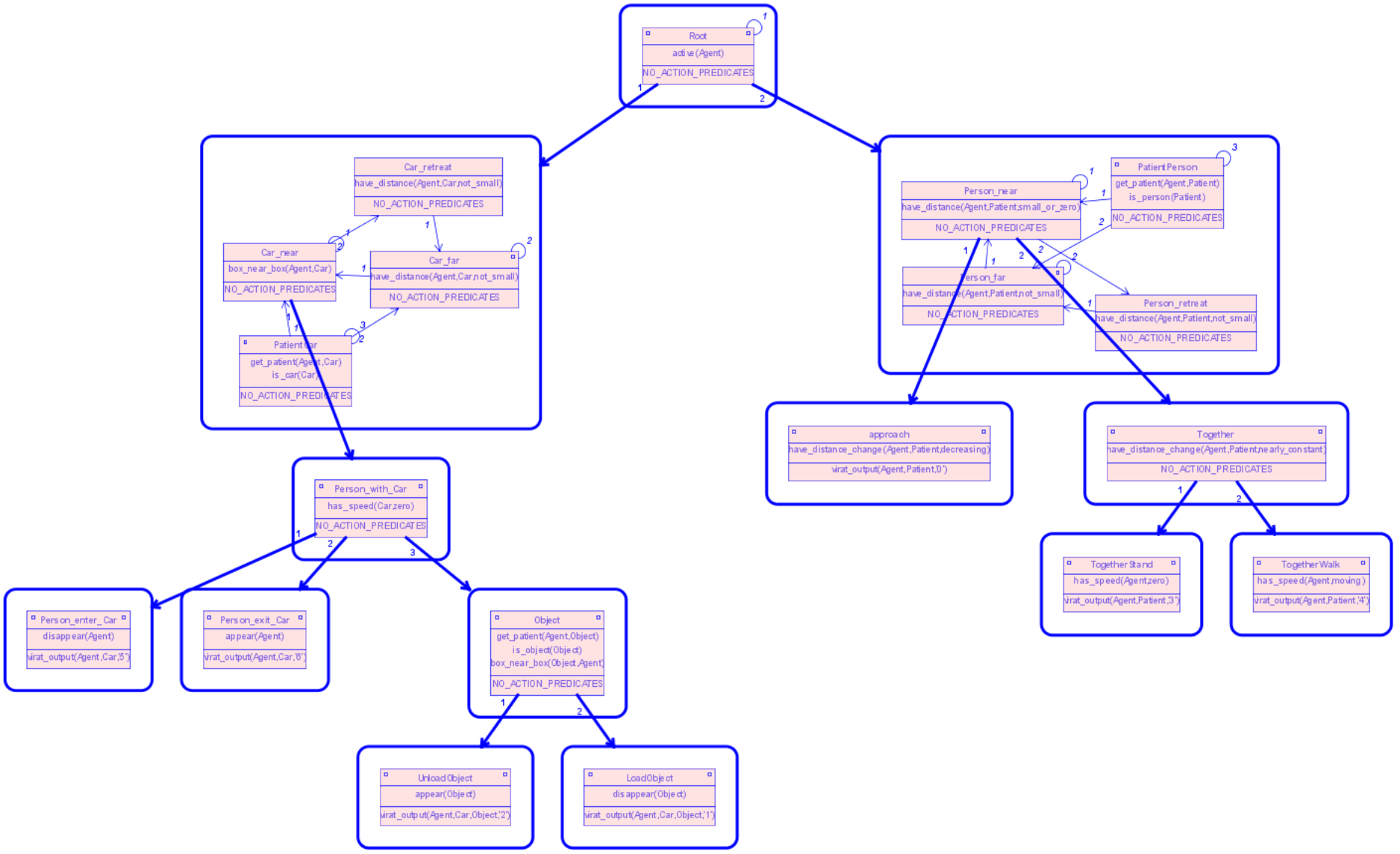


Figure 5.1: Situation Graph Tree representing the events of interest for the VIRAT dataset.

5.1.1.2 Evaluation Results

The evaluation was performed for: a) all chosen events, and b) for single events.

Figure 5.2 shows the precision plotted against truth value threshold for Video 1 for allowed offsets ranging from 4 to 20 in steps of 4. From the graph, it is evident that the precision increases with allowed offset. This is explained by the fact that allowing for a larger offset increases the chance of all correct detections: those detected earlier than expected in the ground truth, and those detected later than indicated in the ground truth. Similarly, recall curves for the same video and allowed offsets are given in Figure 5.3. A similar conclusion about the trend for the recall can be drawn as for the precision.

The results till now prompted a further evaluations for allowed offset values of 1s, 2s, 3s, and 5s. The corresponding precision and recall curves are given in Figures 5.4 and 5.5 respectively. From the Figures, beyond an offset of 2s, we observe perfect recall and precision above 90%.

Similar results were obtained for Video sequences 2 and 3. Figure 5.6 is a plot of precision and recall values for all three videos for an allowed offset of 1s.

The evaluation was then performed for single situations with video sequence 2. Figure 5.7 is a plot of the precision for all 4 events considered for evaluation while Figure 5.8 is the graph for the recall. All precision value are above 80 % and recall values have an average of about 60 %.

5.1.2 BEHAVE Interactions Test Case Scenarios

The BEHAVE Interactions Test Case Scenarios dataset (from here on BEHAVE dataset) is a multi-agent dataset consisting of people acting out numerous interactions. The dataset consists of 90,000 video frames and was recorded at a frame rate of 25fps with a resolution of 640×480 pixels. Ground Truth data is available for most of the videos (Blunsden and Fisher, 2010). The dataset is made up of 10 different scenarios that are:

i) ***InGroup*** : People in a group and not moving very around significantly.

ii) ***Approach*** : Two people or two groups one approaching the other or approaching each other.

iii) ***WalkTogether*** : Any number of people walking together.

iv) ***Meet*** : Meeting between two or more people.

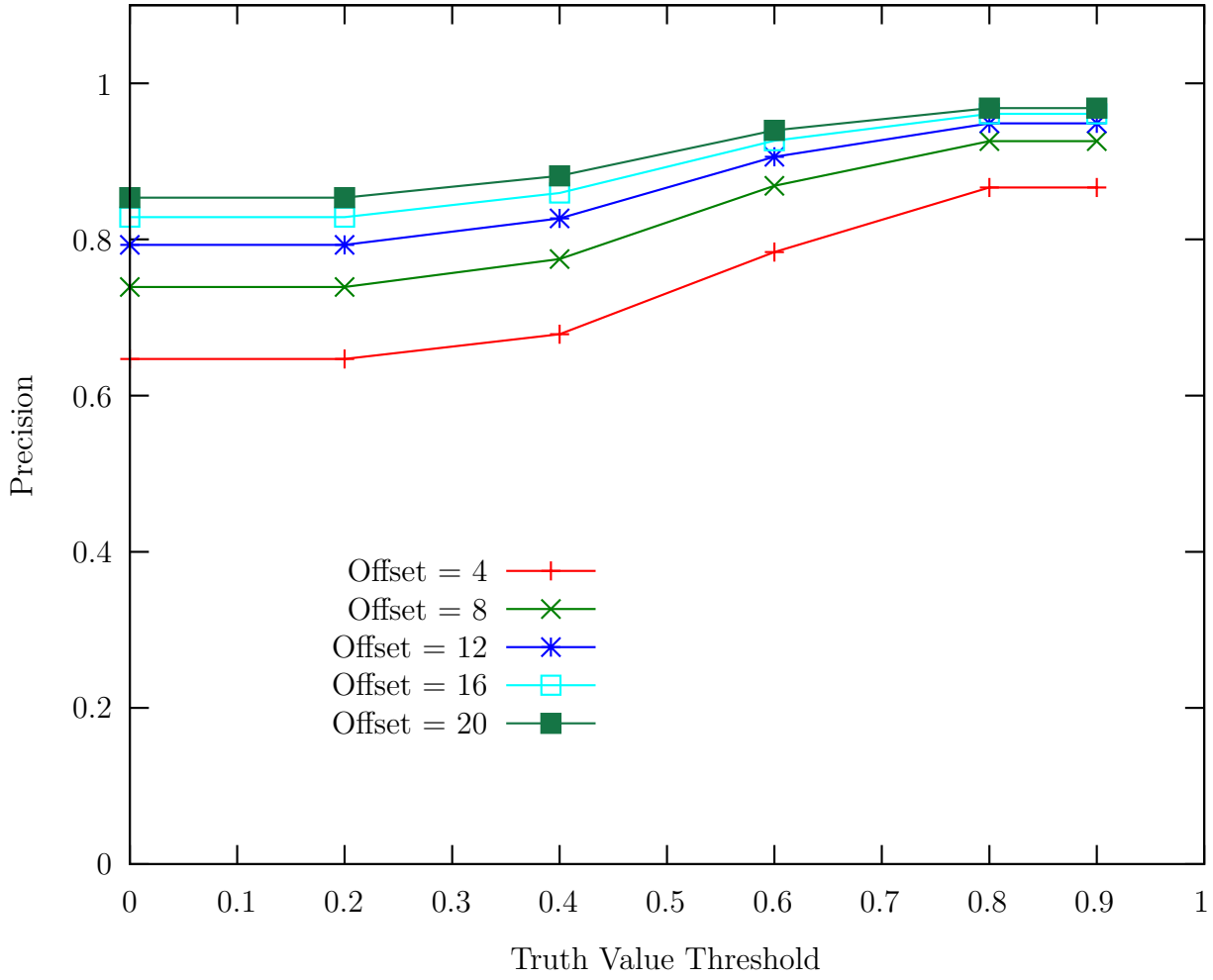


Figure 5.2: Precision values for Video 1 of the VIRAT dataset for allowed offsets ranging from 4 to 20.

v) ***Split*** : Splitting of two or more people from one another.

vi) ***Ignore*** : Ignoring one another.

vii) ***Chase*** : One group chasing another.

viii) ***Fight*** : Two or more groups fighting each other.

ix) ***RunTogether*** : A group of people running together.

x) ***Following*** : One person being followed by another.

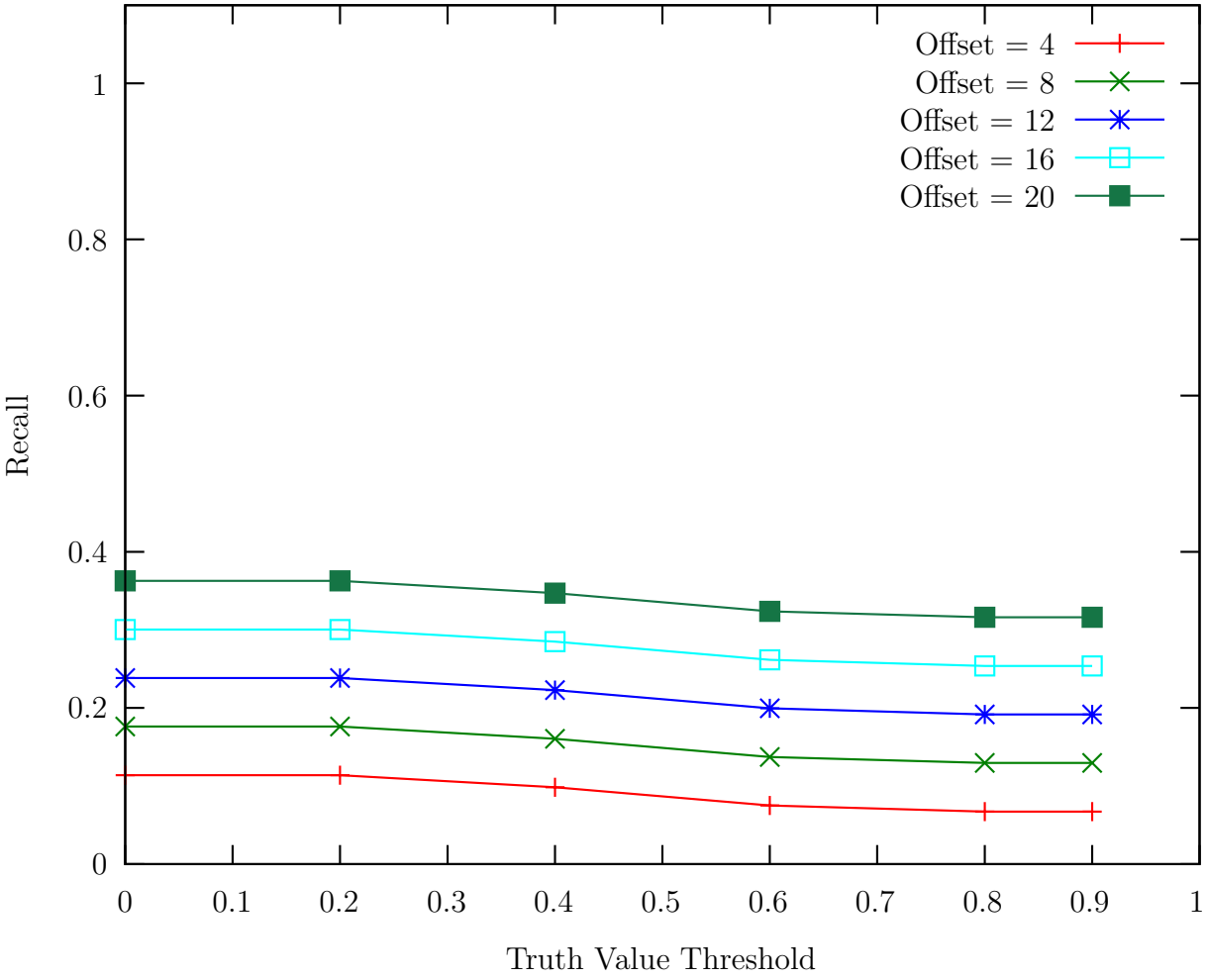


Figure 5.3: Recall values for Video 1 of the VIRAT dataset for allowed offsets ranging from 4 to 20.

The knowledge base for the events in the BEHAVE dataset is captured in a Situation Graph Tree shown in Figure 5.9.

5.1.2.1 Evaluation

Results were obtained for one of the eight available videos sequences (frames 1 - 11200) of the BEHAVE dataset.

In this work, the evaluation was performed for three events: *Approach*, *WalkTogether*, and *Run Together*. Unlike the VIRAT Dataset (whose results are discussed in Section

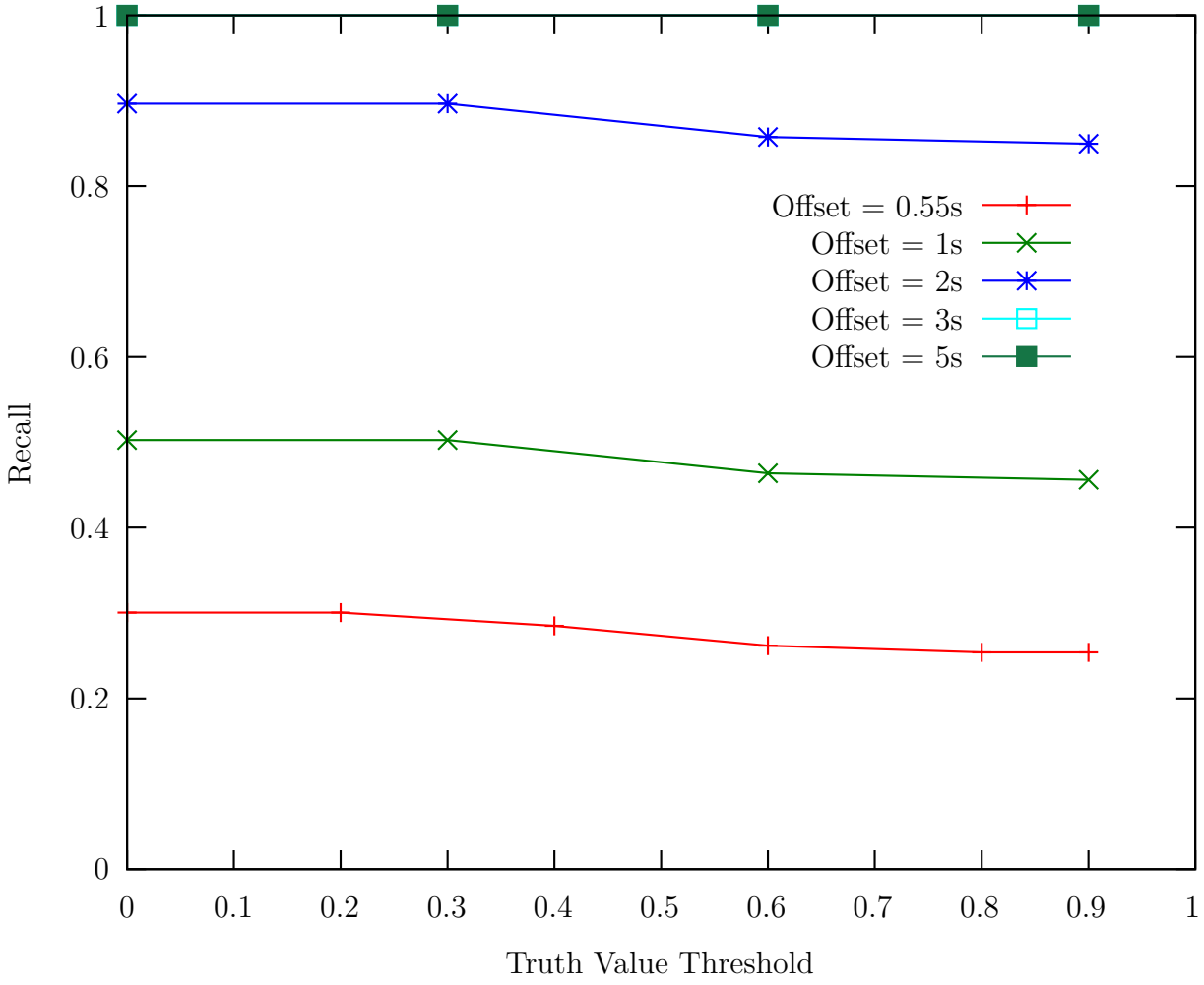


Figure 5.4: Recall values for Video 1 of the VIRAT dataset for allowed offsets ranging from 4 to 20 frames.

5.1.1.2), the BEHAVE dataset has a large number of events per video sequence. This translates into a longer traversal time and a lengthier duration to run the frame-by-frame interval-based evaluation.

The Precision, Recall, and F-Score values, obtained for a frame offset of 10, are given in Figure 5.10. in the Figure, the most striking issue are the low recall values (as compared to those from the VIRAT Dataset). By visual inspection, it is obvious that the issue is the large number of events per frame coupled with small deviations in the ground truth bounding box sizes. These, together, create challenges for the situation recognition system. An additional challenge is the large number of people

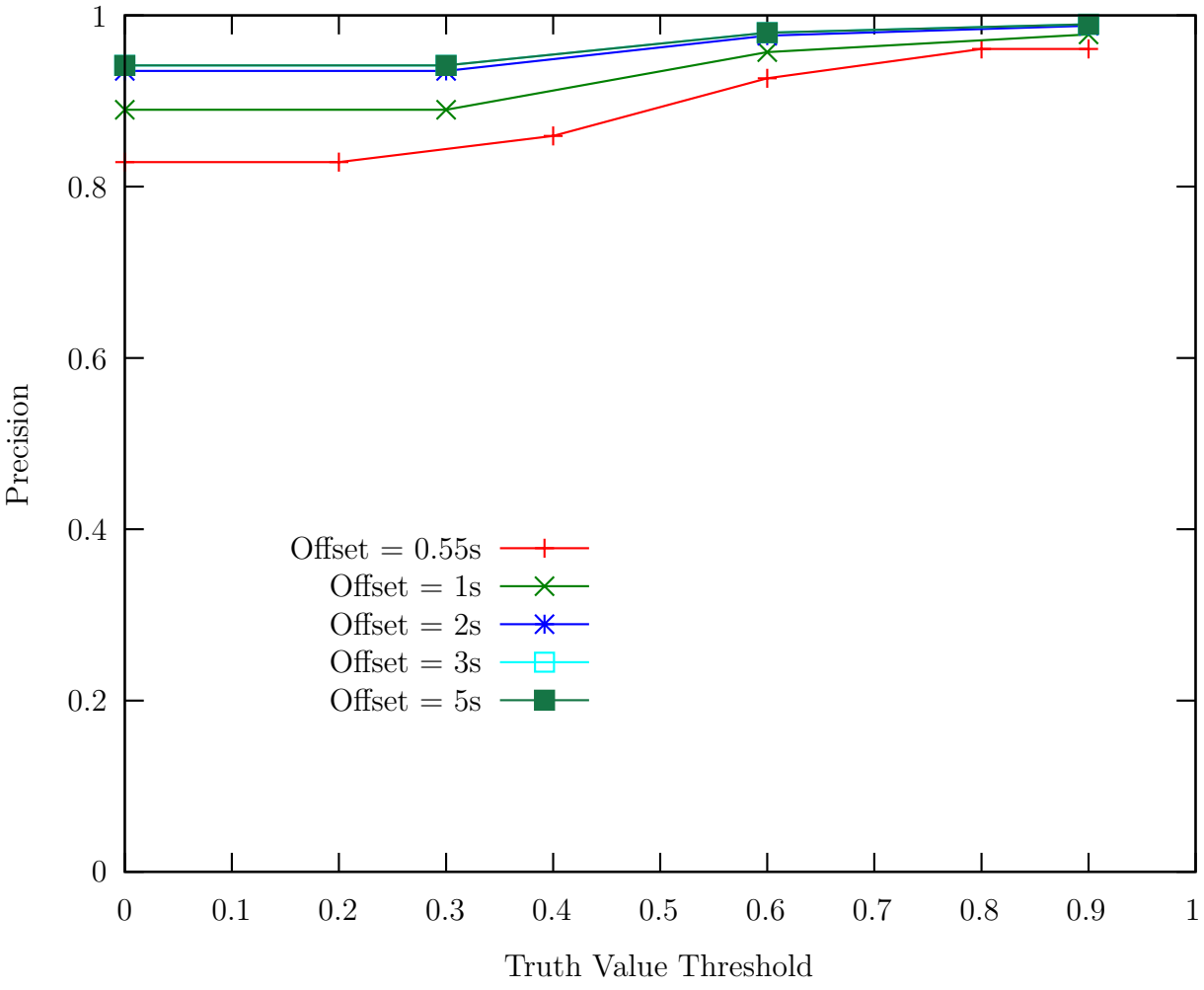


Figure 5.5: Precision curve for Video 1 of the VIRAT dataset for allowed offsets ranging from 4 to 20.

in the scene. For example, if one person is approaching a group of people, it is usually the case that the reasoning system tries to instantiate the *Approach* event for every person in the group rather than just a single situation as demarcated in the ground-truth (see Figure 5.11 (c)). Another scenario that could lead to accumulation of false positives can be seen in Figure 5.11 (d). Here, due to the close proximity of the people involved in the scene, the situation recognition system cycles through a number of situations: *WalkTogether* for more than two people, *WalkTogether* for only 2 people, and *Approach*, while the ground truth is annotated for only one *WalkTogether* situation.

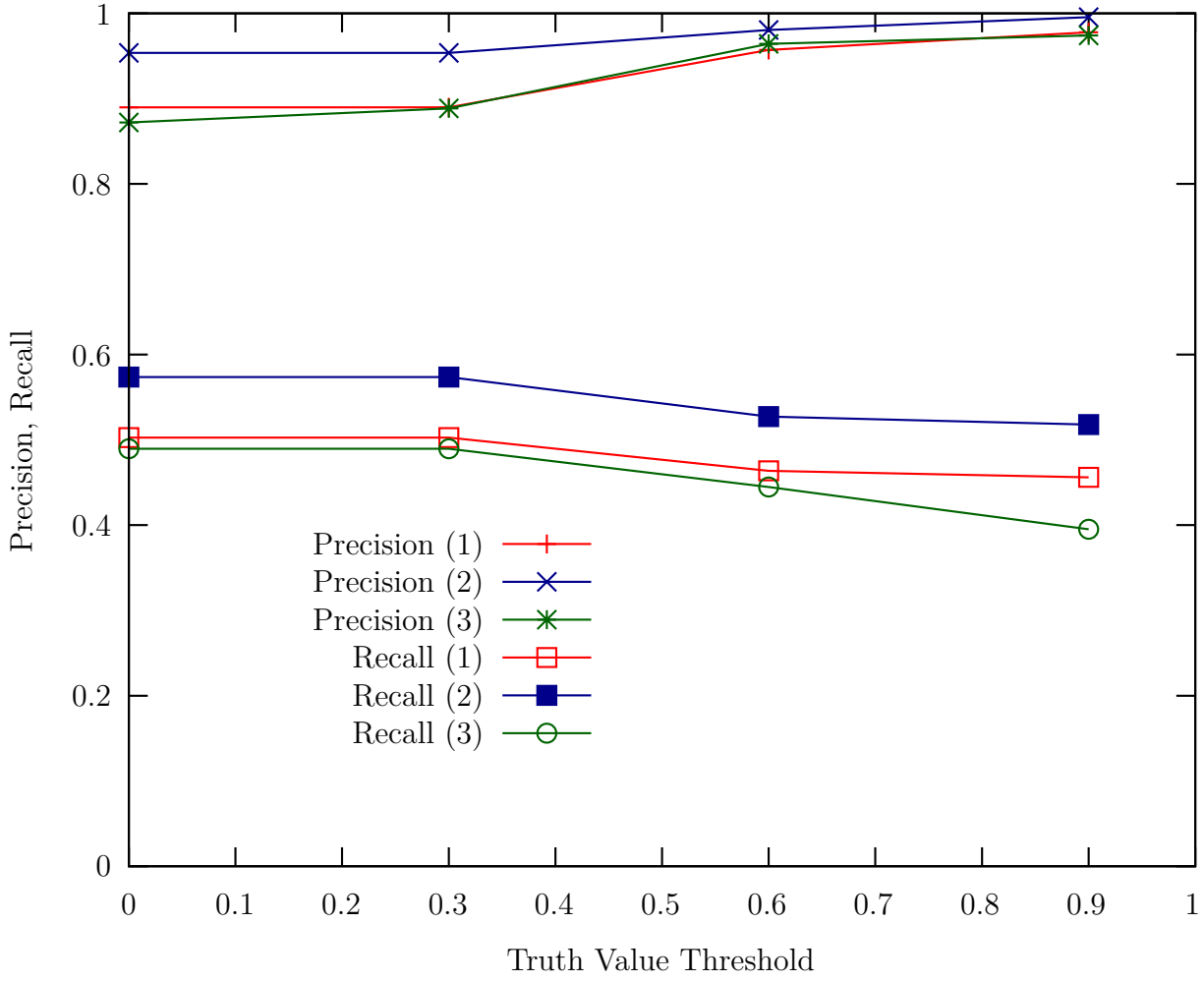


Figure 5.6: Precision and Recall against truth value threshold for evaluation results of all 3 VIRAT video sequences. The number in the bracket map to the video sequence number.

5.2 IOSB VCA Dataset

The IOSB VCA Dataset was recorded as part of the work for this thesis to as a means to obtain video data for a more extensive evaluation. The procedure for creation of the dataset: related hardware, software, and other modalities are presented in appendix A.

Similar to the BEHAVE Dataset described in Section 5.1.2, the IOSB VCA dataset is made of situations involving human behavior. The 20 situations captured in the

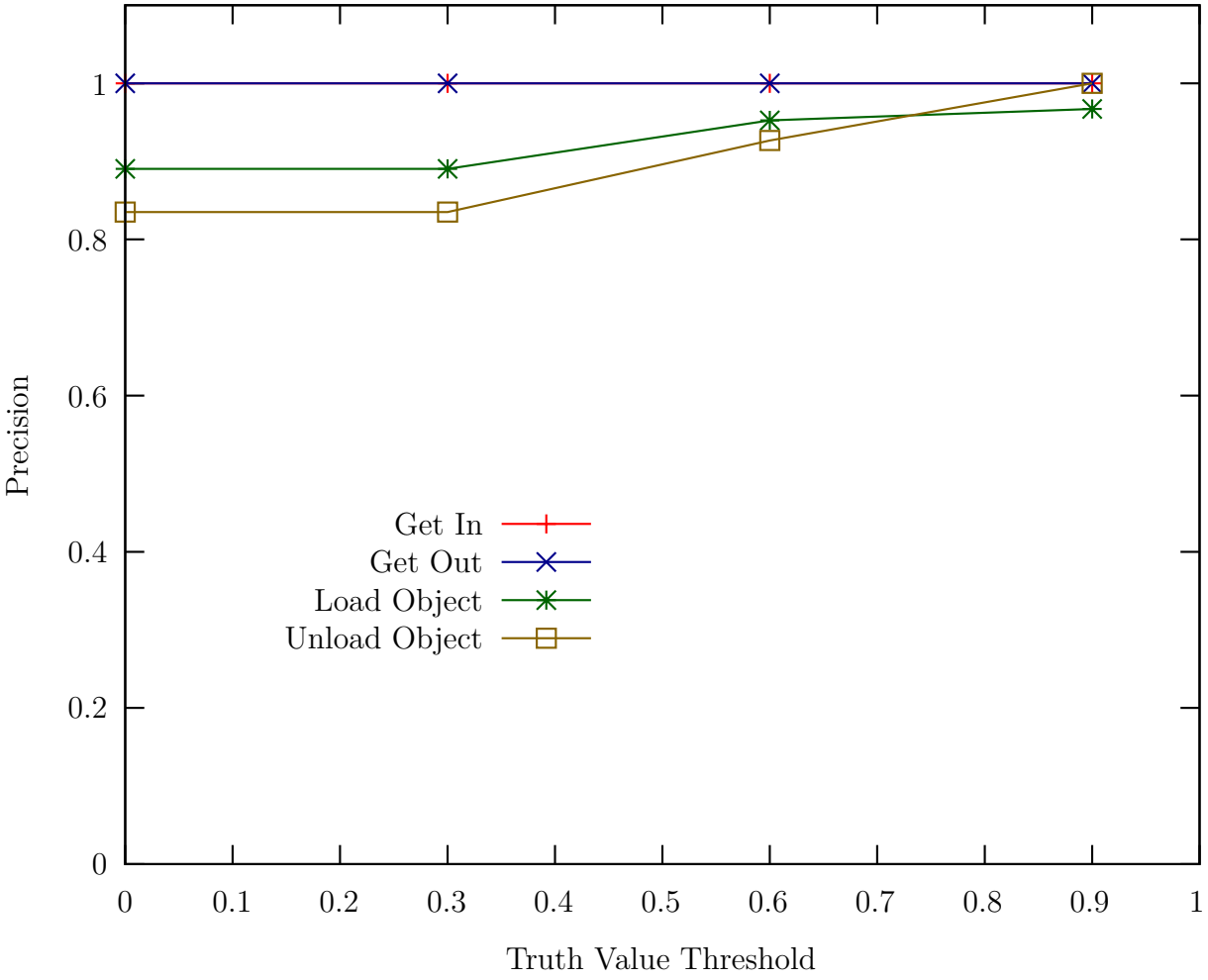


Figure 5.7: Precision plotted against truth value threshold for evaluation results video 2 of the VIRAT dataset for single event evaluation. Allowed offset = 1s.

dataset can be grouped into the following categories:

i) Single Person (Walking, Running, …)

ii) Two Person (Walking Together, Meeting, Approaching, Ignoring, …)

iii) Group Situations (Grouping, UnGrouping, …)

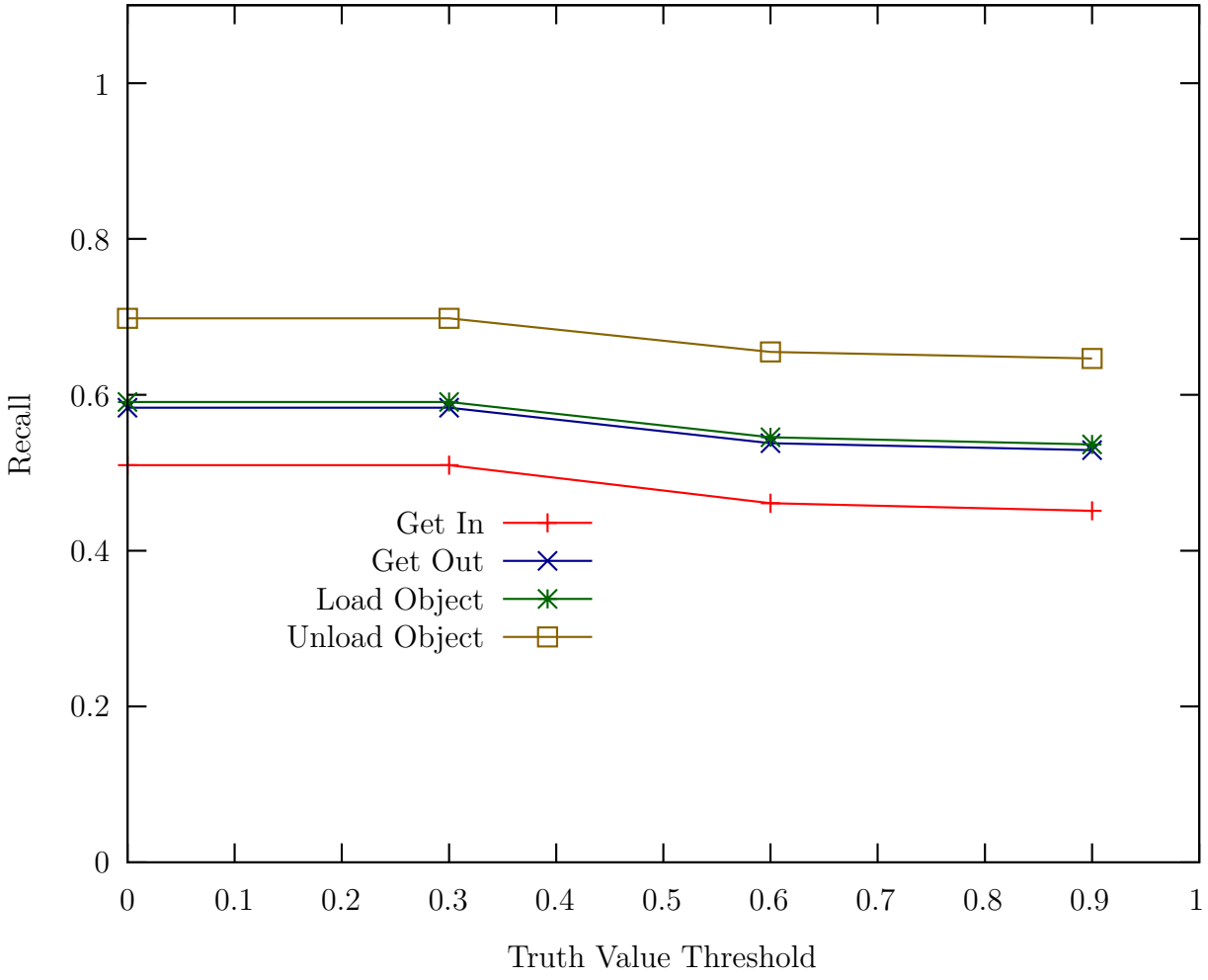


Figure 5.8: Recall plotted against truth value threshold for evaluation results video 2 of the VIRAT dataset for single event evaluation. Allowed offset = 1s.

5.2.1 Obtaining the Ground Truth

One particular characteristic of the IOSB VCA dataset is the automatic acquisition of the ground truth data. The recording of the dataset was done with a camera system running in synchronism with the ABATEC Local Positioning System (LPM) (see Figure 5.12 for a summary). The system uses a combination of base sStations and transponders (worn by all persons in taking part in a particular situation) to determine the exact positions of people in the scene. Detailed technical specifications and the procedure for obtaining the ground truth are given in Appendix A.1.2

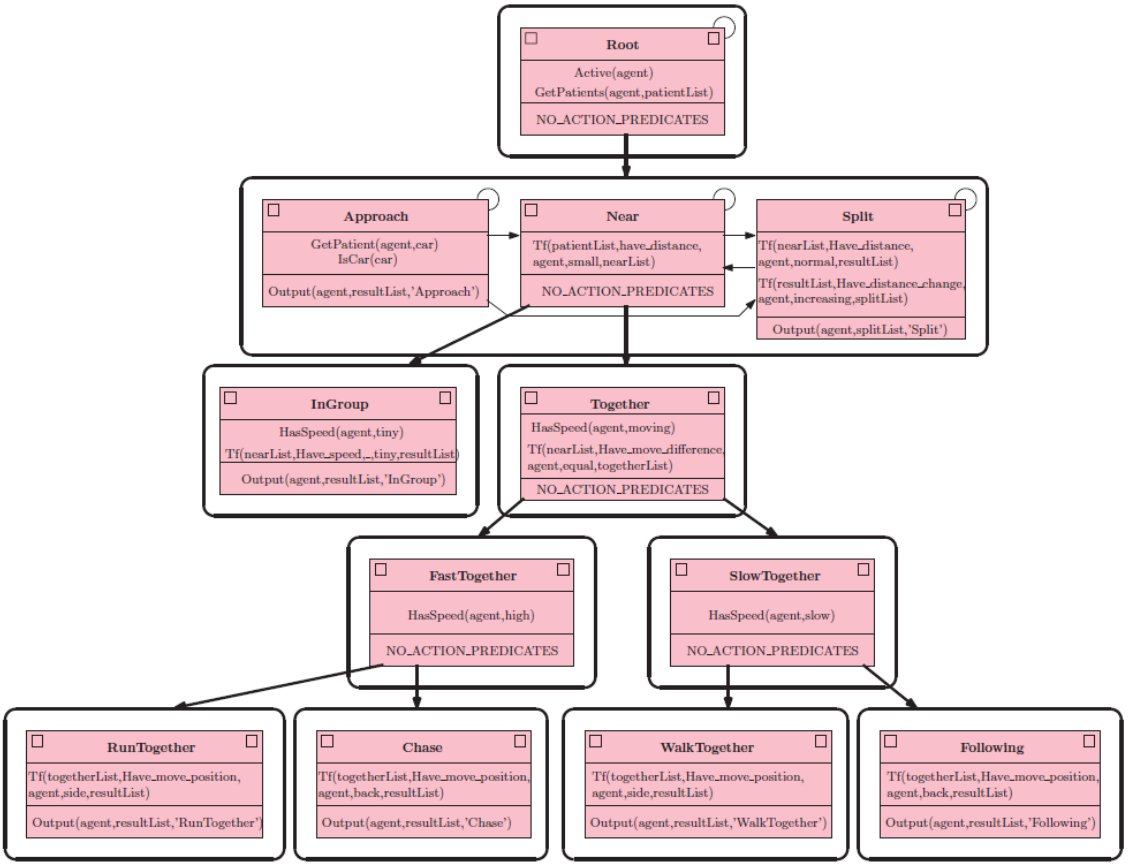


Figure 5.9: Situation Graph Tree representing the events of interest for the BEHAVE dataset

Automated, accurate ground truth is associated with the following benefits:

- Avoiding the large number of hours (or even months) spent on manual annotation of ground truth data.
- Get rid of errors in the dataset that are typically introduced due to manual annotation.
- The dataset itself can act as benchmark for evaluating situation recognition methods and other computer vision tasks such as person detection detection and tracking, whose performance may be challenging to evaluate otherwise.

A sample image involving two agents running together is shown in Figure 5.13.

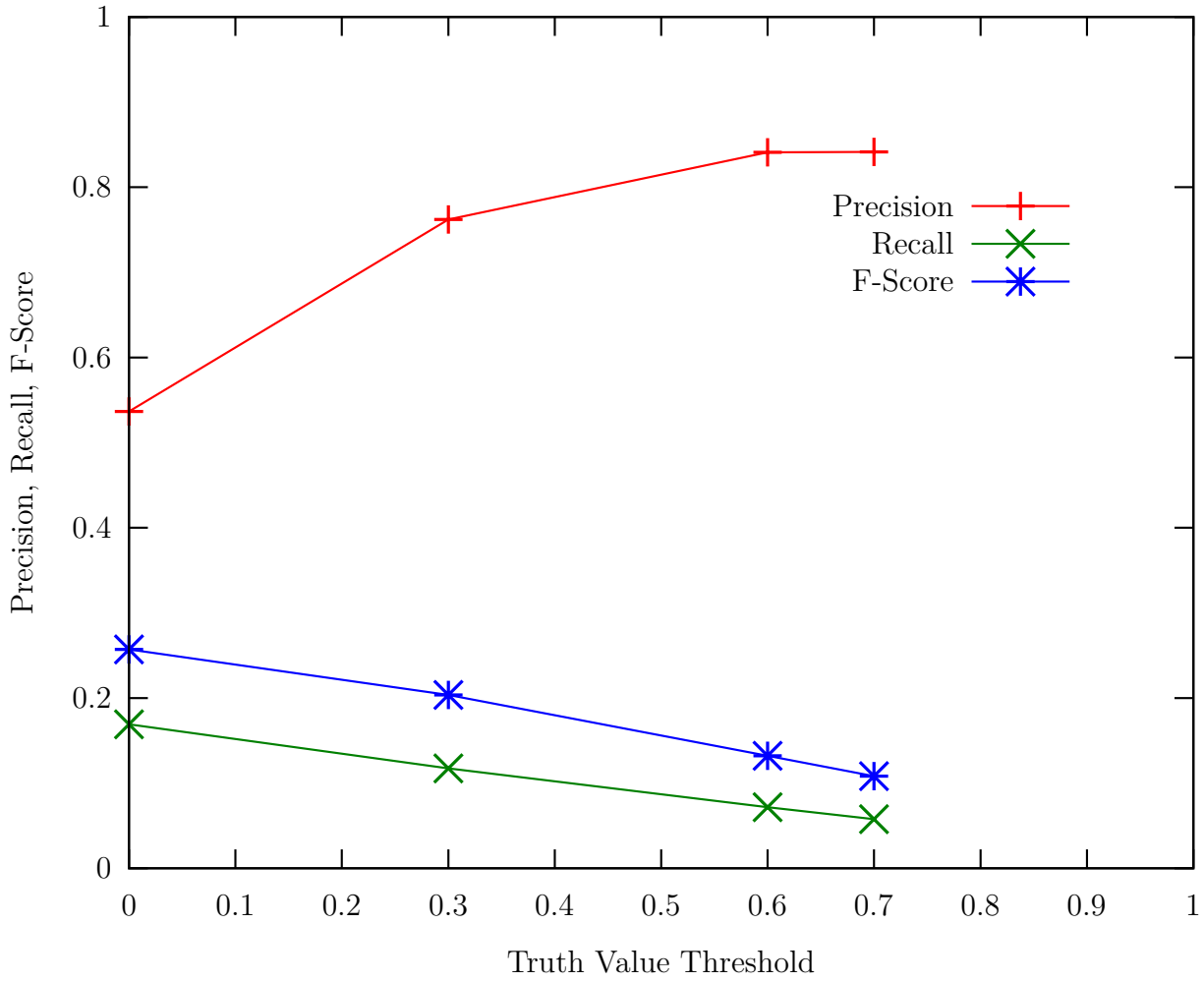
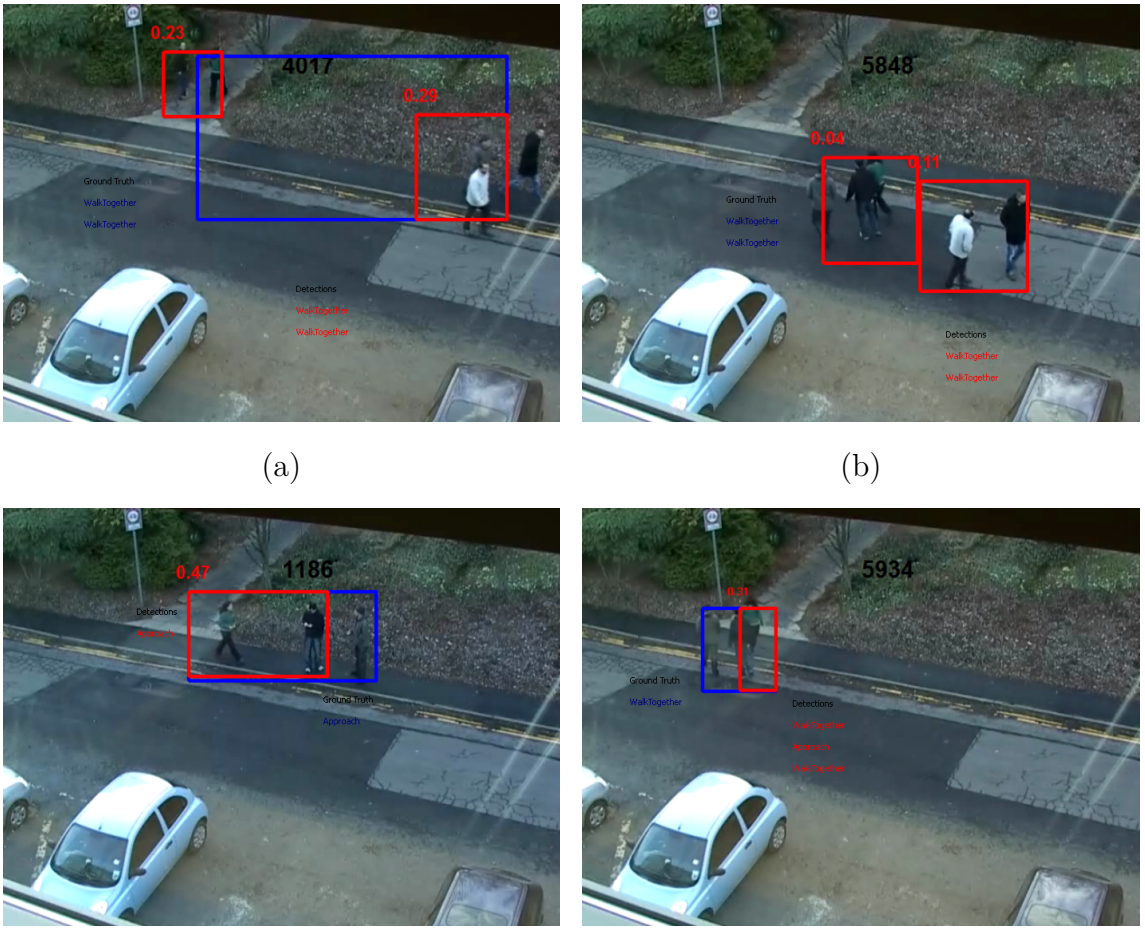


Figure 5.10: Precision, Recall and F-Score against Threshold for a frame offset of 10.



(a) (b)

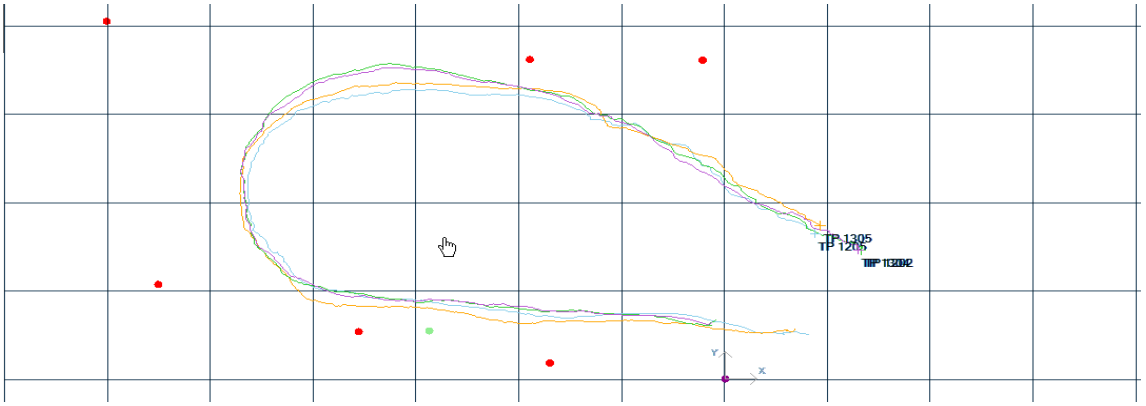
(c) (d)

Figure 5.11: (a) and (b) are examples of true positives showing two different instantiation of the *WalkTogether* situation. (c) and (d) are false positives due to the involvement of many agents and close proximity of people in motion respectively.

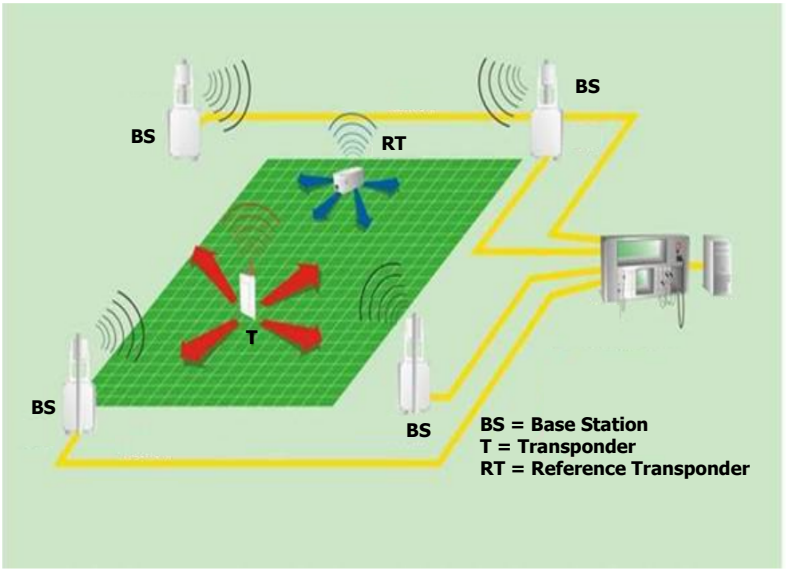
With the images sequences for each scene obtained from the camera system, and the trajectory data extracted from the LPM system. The following additional post-acquisition processing tasks had to be performed on the data before it could be used:

i Remove the effect of random variations in location data by passing it though a Kalman Filter.

ii Perform a time resolution, based on the frames per second of the images, to synchronize the LPM data with the images.



(a)



(b) (c)

Figure 5.12: Summary of the tools used to record the IOSB dataset.(b) is the Local Position Measurement System and (c) is the camera system. (a) Represents an example of a trajectory for the LPM during the recording of a scene involving two people running together (see Figure 5.13). Note: each person wear two transponders.

iii Project the data from the transponders into bounding boxes on the data. This required the setup of a camera projection model to map the real world coordinates into image coordinates.

An overview of the pre-processing steps is shown in Figure 5.14.

Figure 5.13: Sample scene taken from the IOSB VCA dataset with a situation involving two people running.

An initial attempt to provide the location information for people in the images is shown in Figure 5.17.

5.2.1.1 Evaluation

Since the situations of interest captured in the IOSB VCA dataset are the same as those for BEHAVE dataset, the same Situation Graph Tree, shown in Figure 5.9 could be used to perform inference on the IOSB VCA data. This is an example of the domain agnostic nature of SGTs as an expert knowledge representation structure. The evaluation was performed for the *Following* situation. The traversal was done directly off the Kalman filtered LPM data (see Section 5.2.1) for the recorded for the video involving the scene. The evaluation was done for offsets ranging from 4 to 20 and truth value thresholds ranging from 0.0 to 0.9. The recall is given in Figure 5.15. The results point to an early cutoff of the recall values at a truth value threshold of 0.6. This can be explained by degrees of validity for the results not going beyond 0.6 which could be a consequence of the current rule files for the dataset being a work in progress. Similarly, the results for the precision are plotted in Figure 5.16.

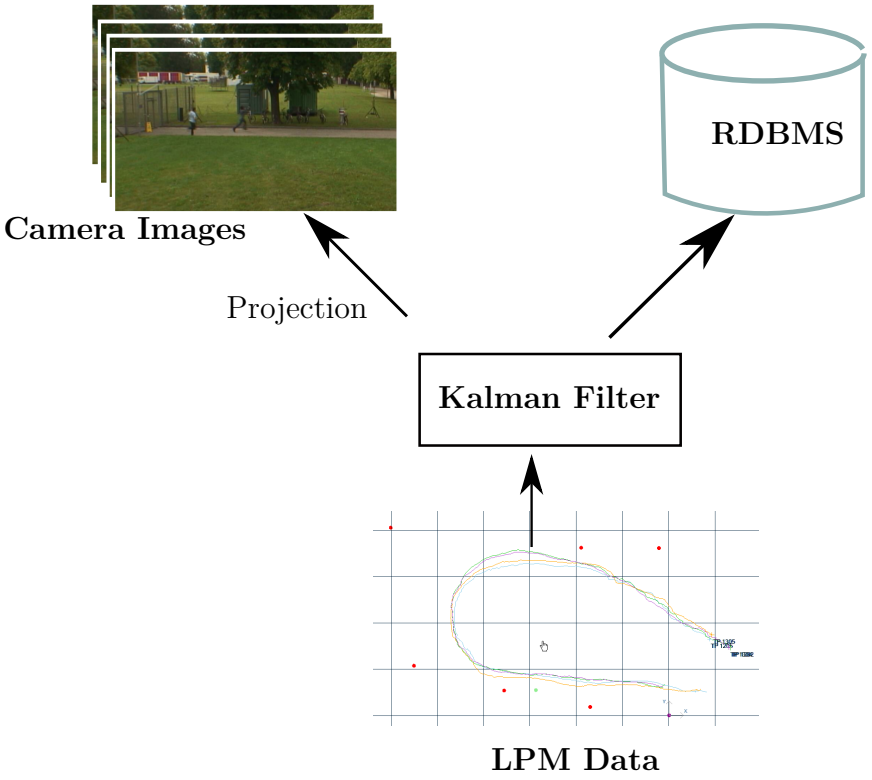


Figure 5.14: Preprocessing for the IOSB VCA data. The data is first smoother by Kalman filtering and is stored in the database for subsequent use for situation recognition and is also projected into image coordinates to provide the person locations in the image for each frame.

5.3 Integration of the Person Tracker

Initial results for running the person tracker on the scene of two people following each other is given in Figure 5.18. The results look promising, but there is still some adaptation work that has to be done before the tracker can be fully integrated.

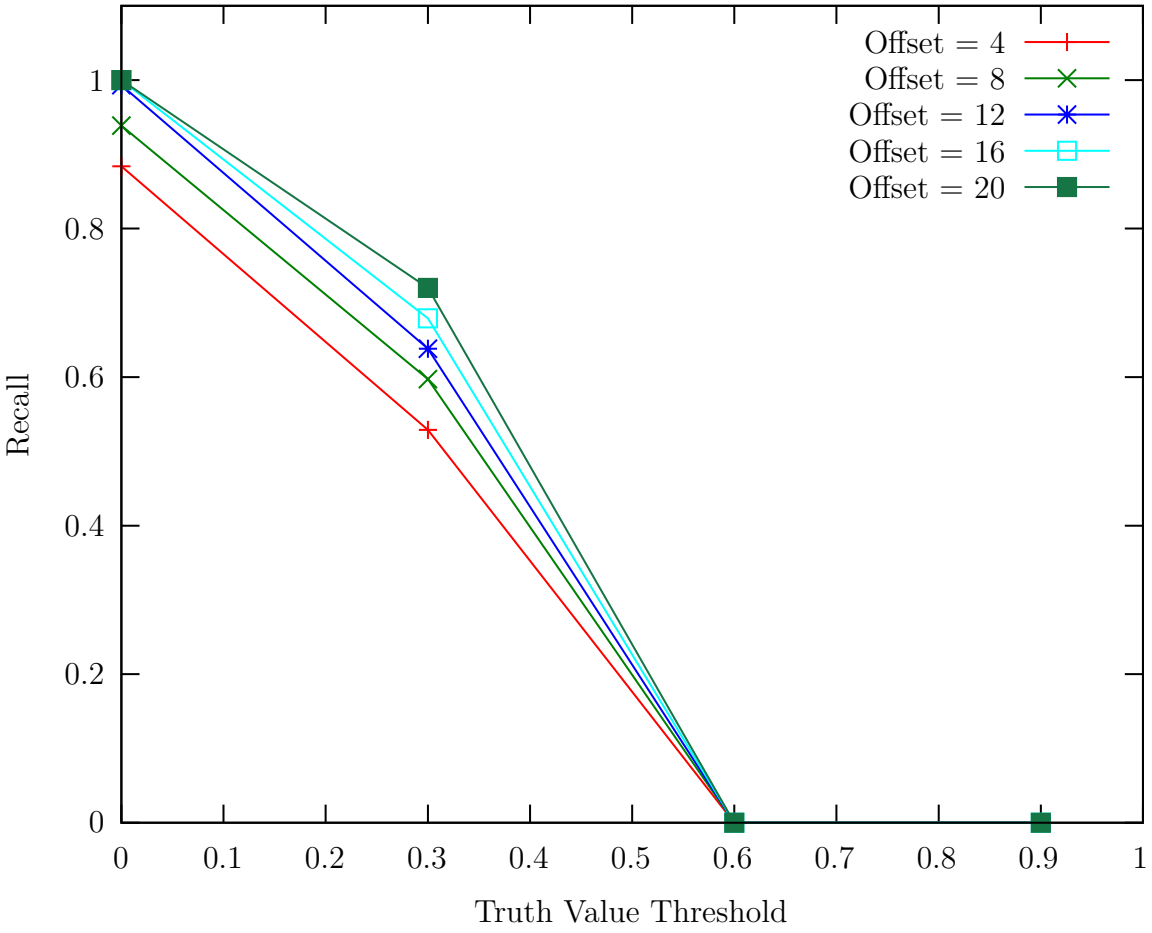


Figure 5.15: Recall versus Threshold for one scene of the *Following* situation from the IOSB VCA dataset.

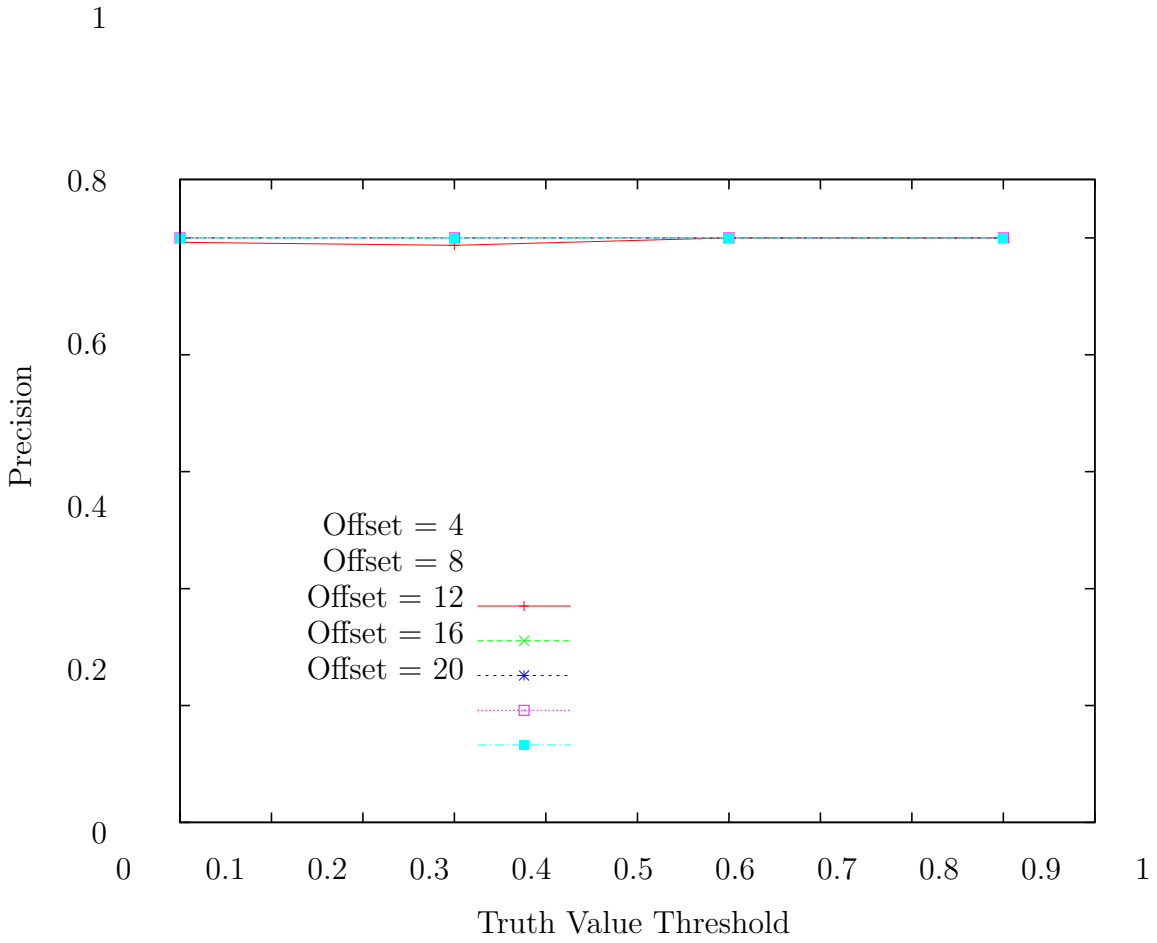


Figure 5.16: Precision versus Threshold for one scene of the *Following* situation from the IOSB VCA dataset.

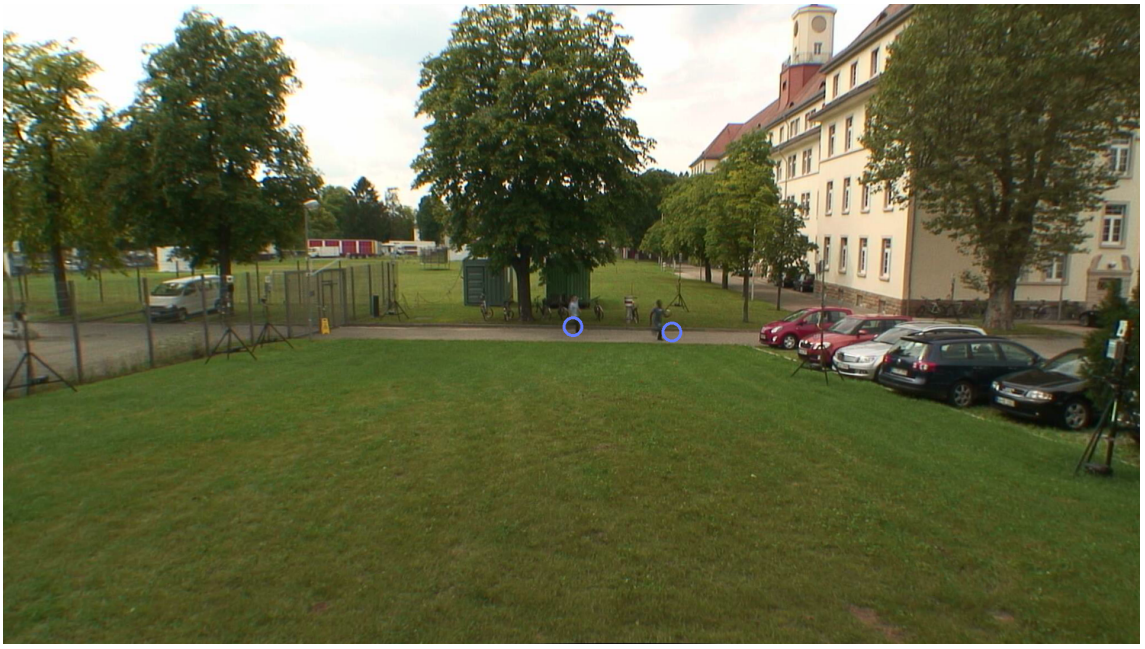


Figure 5.17: Illustration of initial results for obtaining the ground truth from the LPM data. The light blue circles are positioned at the position of the feet of the people in each frame.

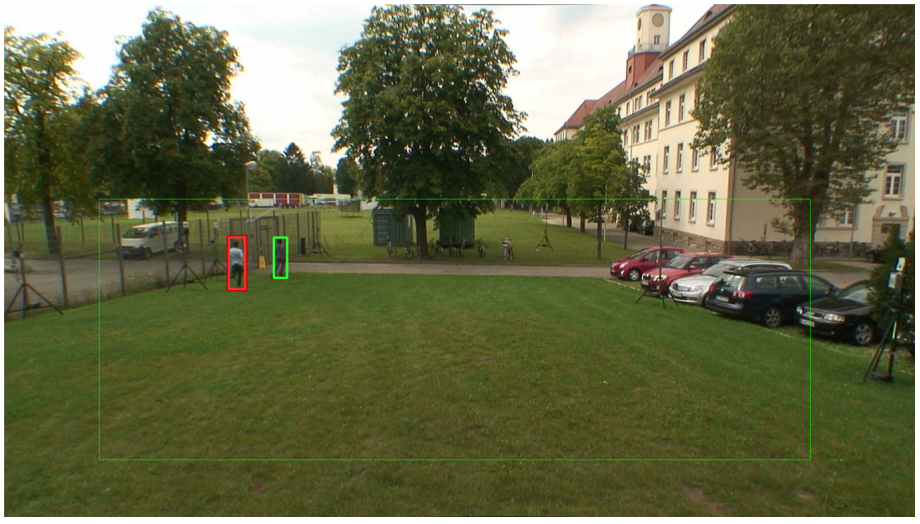


Figure 5.18: An example of running the person tracker on a video.

Chapter 6

Summary and Conclusion

6.1 Discussion of Results

The work in this thesis focuses on steps towards situation recognition in un-constrained video streams. The long term goal is to have a situation recognition system that takes input data that is acquired reliably via computer vision methods. Eventually, any other input data beyond computer vision could be added easily.

A major component of the work is the evaluation of the FMTL/Cognitive Vision System on real world data. For this, the BEHAVE and VIRAT datasets were chosen as they present large diversity in scenes, and naturalism (video recordings of people in natural interactions rather than actors following scripts). Additionally, to the best of my knowledge, the work in this thesis is the first time a frame-by-frame evaluation, rather than a compressed version of the video sequence (with reduce frames per second), has been done on this situation recognition system on these particular datasets.

The evaluation for the VIRAT dataset was done for four different situations involving person-car interactions for three video sequences. Initial experiments considered low allowed offsets. These initial results prompted evaluations with larger offsets (in the order of seconds). With this change, we observe that precision values go up with the size of the offset, and so do the recall values. Additional experiments considered the precision and recall values for one video of the dataset with each of the four situations evaluated separately.

Due the high density of events in each video sequence, evaluation of the BEHAVE dataset was only performed for one video and three situations. The low recall of the situation recognition system stands out in this case, due to the multi-hypothesis search nature of the FMTL/SGT Cognitive Vision System, and the often missing

ground truth annotation for some of the events. However, even with input data of this nature, one can argue for applications where high precision is important.

The development of the IOSB VCA dataset is another major contribution of this thesis. The custom in-house dataset that was recorded with automatically determined ground truth for evaluation with the FMTL/SGT CVS. The unique way in which the ground truth for the dataset was obtained allows for evaluation results to be obtained for example even in the case when the people involved in a particular situation are occluded: either by each other or by other objects. Through various pre-processing steps, the data could be converted to a form usable with the situation recognition system. The same SGT for the knowledge from the BEHAVE dataset could be used for reasoning on the IOSB VCA data further providing evidence for the domain transferability of SGTs. An evaluation was carried out for the *Following* situation. The results show high precision for the system and a cutoff for the truth value threshold at 0.6. In an application, this is not a concern since to get a high recall for the system, one will then only have to pick a truth value threshold that is less than 0.6.

A major extra output of this work is a extensible real-time-capable Situation Recognition Architecture that represents contribution towards the integrates existing separate tools using in situation recognition system into one complete user-friendly package. Some of the key modules the architecture introduces are: analysis (evaluation and visualization) that gives insight into the situation recognition process during after each run, a central Relational Database Management System (RDBMS) that acts as the glue for the entire system by ensuring reliable storage and retrieval of input machine perception data, and any other information from other modules. Together with all the other components introduced in Section 4.1, this architecture brings us closer to a situation recognition system that is field deployable.

6.2 Future Work

The evaluation in this work could be extended to provide a specification for situation identification to cater for: situation interruption, agents performing more than one task at the same time, bigger number of agents. The conversion of the IOSB VCA dataset has only just begun as presented in this thesis. Future should finalize this task and more importantly start applying it for evaluation of situation recognition and other computer vision techniques that could benefit from it. Initial steps were taken to integrate a person tracker into the situation recognition architecture, a full integration of the system would be beneficial, for both offline and online operation of the recognition system. This would allow for semantic feedback to the lower-levels

of the CVS, such as the Camera System, so as to influence the machine perception process with the aim of acquiring more specific and more details data.

Appendix A

IOSB Dataset

A.1 Technical Equipment

A.1.1 Cameras

For both indoor and outdoor day time recordings, an Axis Q1755 Network camera was used, see Fig A.1 for datasheet. Night time outdoor recordings were done with an Axis P5534 PTZ Network Camera. The camera datasheet is given in Figure A.2.

Camera	
Models: Indoor	AXIS Q1755 60 Hz; AXIS Q1755 50 Hz
Models: Outdoor	AXIS Q1755-E 60 Hz; AXIS Q1755-E 50 Hz
Image sensor	1/3" progressive scan CMOS 2 megapixel
Lens	f=5.1 - 51 mm, F1.8 - 2.1, autofocus, automatic day/night Horizontal angle of view: 48.1° - 5.1° M37x0.75 mounting thread for optional lens adaptor
Minimum illumination	Color: 2 lux at 30 IRE, F1.8 B/W: 0.2 lux at 30 IRE, F1.8
Shutter time	1/10000 s to 1/2 s
Zoom	10x optical and 12x digital, total 120x

Figure A.1: Axis Q1755 Camera datasheet.

Camera	
Models	AXIS P5534 60 Hz; AXIS P5534 50 Hz
Image sensor	1/3" progressive scan CCD 1.3 megapixel
Lens	f=4.7 – 84.6 mm, F1.6 – 2.8, autofocus, automatic day/night Horizontal angle of view: 55.2° - 3.2°
Minimum illumination	Color: 0.74 lux at 30 IRE F1.6 B/W: 0.04 lux at 30 IRE F1.6
Shutter time	1/10 000 s to 1/4 s
Pan/tilt/zoom	E-flip, Auto-flip, 100 preset positions Pan: 360° (with Auto-flip), 0.2° - 300°/s Tilt: 180°, 0.2° – 300°/s 18x optical zoom and 12x digital zoom, total 216x zoom
Pan/tilt/zoom functionalities	Limited guard tour Control queue On-screen directional indicator

Figure A.2: Axis P5534 PTZ Dome Camera datasheet.

A.1.2 Local Position Measurement

First to calibrate the LPM system, the coordinates for 8 base stations (BSs), positioned around the scene (see Figure A.3), and one reference transponder (RT) were determined precisely using a tachymeter. The tachymeter readings are given in Table A.1. One of the BSs, the master BS, triggers the RT, which emits periodically a chirp to provide a common time base Resch *et al.*, 2012. After all BSs have been synchronized, the measurement transponders (MTs) can be activated. In each scene, every agent was strapped with two MTs (over the shoulders, one on each side of the head). The measurement for each base station for each transponder was then sent to a central computation unit via WLAN, and used to determine the details for the trajectory of that agent.

Base Station	x(m)	y(m)	z(m)
1	-8.54	0.9	12
2	-27.57	5.34	0.0
3	-8.54	0.9	2.15
4	-27.57	5.34	1.77
5	-24.94	33.14	0.0
6	-1.12	18.01	0.0
7	-24.94	33.14	1.49
8	-1.12	18.01	1.36

Table A.1: Tachymeter readings for the base stations using with the LPM system.

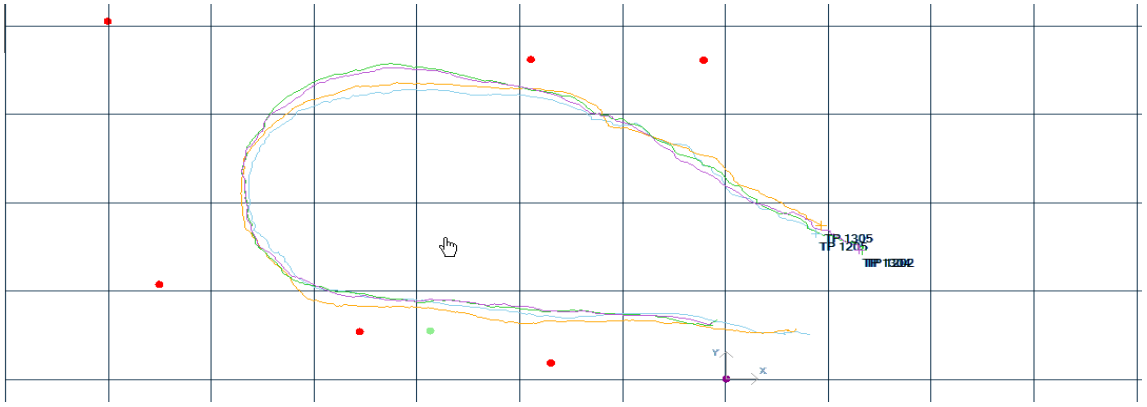


Figure A.3: Sample trajectory recorded with LPM. Red dots indicate the positions of the base stations.

A.1.3 LPM Dataformat

The structure of the raw LPM data is as follows:

```
typedef struct _LPMLongLine
{
  // 26 int: allg. Daten
  //  6 int: Daten zu je einer BS
```

```
  // mit 20 BS ergeben sich 146 int = 584 Byte
  int Timestamp;  // in 0.1 milliseconds
  int TranspID;
  int Quality;
  int Telemetry;
  int PosX;       // in mm
  int PosY;       // in mm
  int PosZ;       // in mm
  int SpeedX;
  int SpeedY;
  int SpeedZ;
  int AccelX;
  int AccelY;
  int AccelZ;
  int RawX;       // in mm
  int RawY;       // in mm
  int RawZ;       // in mm
  int Heading;
  int Roll;
  int Pitch;
  int TrackStatus;
  int FilterMode;
  int W_Offset_Bancroft;
  int W_Offset_EKF;
  int W_Point_EKF;
  int tNumberBS;
  int CellID;
  int HUBPort_[20];
  int PowerLevel_[20];
  int PeakQuality_[20];
  int Chi2_[20];
  int TimeDiff_[20];
  int BSTelemetry_[20];
} LPMLongLineStruct;
typedef struct _LPMLongData
```

List of Figures

Bibliography

Aggarwal, J. K. and M. S. Ryoo

2011 "Human Activity Analysis: A Review", *ACM Computing Surveys*. (Cited on pp. 3, 4, 7.)

Allen, James F.

1983 "Maintaining knowledge about temporal intervals", *Commun. ACM*, 26, 11 (Nov. 1983), pp. 832-843. (Cited on pp. 7, 17.)

Arens, M.

2003 *SGTEditor Reference Manual*, 1.2, Institut für Algorithmen und Kognitive Systeme, Fakultät für Informatik der Universität Karlsruhe (TH). (Cited on p. 20.)

2004 *Repräsentation und Nutzung von Verhaltenswissen in der Bildfolgenauswertung*, Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Dissertationen zur Künstlichen Intelligenz (DISKI) 287, Akademische Verlagsgesellschaft Aka GmbH. (Cited on p. 16.)

Arens, M., R. Gerber, and H-H. Nagel

2008 "Conceptual Representations Between Video Signals and Natural Language Descriptions", *Image and Vision Computing*, 26, 1, pp. 53-66. (Cited on pp. 7, 17, 24.)

Bauer, S.

2012 *Comparing Ontologies and Situation Graph Trees in Cognitive Vision Systems*, MA thesis, Karlsruhe Institute of Technology. (Cited on pp. 20, 22.)

Bellotto, N. *et al.*

2012 "Robot control based on qualitative representation of human trajectories", *AAAI Symposium on Designing Intelligent Robots: Reintegrating AI.*. (Cited on pp. 10, 11.)

Bellotto, N., B. Benfold, H Harland, H-H. Nagel, N. Pirlo, I. Reid, and C. Sommerlade E.and Zhao

2012 "Cognitive visual tracking and camera control", *Computer Vision and Image Understanding*, 116, 3, <ce:title>Special issue on Semantic Understanding of Human Behaviors in Image Sequences</ce:title>, pp. 457-471, ISSN: 1077-3142. (Cited on pp. 8, 9.)

Bellotto, N. and H. Hu

2010 "Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of Bayesian filters", *Autonomous Robots*, 28, 4, pp. 425-438. (Cited on p. 9.)

Blunsden, S. and R.B Fisher

2010 "The BEHAVE video dataset: ground truthed video for multi-person behavior classification", *Annals of the BMVA*, 2010, 4, pp. 1-12. (Cited on p. 40.)

Dollár, P., Z. Tu, P. Perona, and S. Belongie

2009 "Integral Channel Features", in *BMVC*, 4, vol. 2, p. 5. (Cited on p. 30.)

Gerber, R. and H.-H. Nagel

2008 "Representation of occurrences for road vehicle traffic", *Artificial Intelligence*, 172, 4 5, pp. 351-391, ISSN: 0004-3702. (Cited on pp. 7, 16.)

Gottfried, B., H. W. Guesgen, and S. Hübner

2006 "Spatiotemporal reasoning for smart homes", in *Designing Smart Homes*, Springer, pp. 16-34. (Cited on p. 7.)

Ijsselmuiden, J., A-K. Grosselfinger A-KGrosselfinger, M. Arens, and R. Stiefelhagen

2012 "Automatic Behavior Understanding in Crisis Response Control Rooms", in *Ambient Intelligence*, ed. by Fabio Patern , Boris Ruyter, Panos Markopoulos, Carmen Santoro, Evert Loenen, and Kris Luyten, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 97-112, vol. 7683, ISBN: 978-3-642-34897-6. (Cited on p. 11.)

Kieritz, H., W. Hübner, and M. Arens

2013 "Learning transmodal person detectors from single spectral training sets", in *Optics and Photonics for Counterterrorism, Crime Fighting and Defence IX*. (Cited on p. 30.)

Kitani, K.M., Y. Sato, and A. Sugimoto

2007 "Recovering the Basic Structure of Human Activities from a Video-Based Symbol String", in *Motion and Video Computing, 2007. WMVC '07. IEEE Workshop on*, pp. 9-9. (Cited on p. 6.)

Münch, D., S. Becker, W. Hübner, and M. Arens

2012 "Towards a Real-Time Situational Awareness System for Surveillance Applications in Unconstrained Environments", in *Future Security*, ed. by Nils Aschenbruck, Peter Martini, Michael Meier, and Jens T lle, Communications in Computer and Information Science, Springer Berlin Heidelberg, pp. 517-521, vol. 318, ISBN: 978-3-642-33160-2. (Cited on p. 14.)

Münch, D., J. IJsselmuiden, A-K. Grosselfinger, M. Arens, and R. Stiefelhagen

2012 "Rule-Based High-Level Situation Recognition from Incomplete Tracking Data", in *Rules on the Web: Research and Applications*, ed. by Antonis Bikakis and Adrian Giurca, Lecture Notes in Computer Science, Springer Berlin - Heidelberg, vol. 7438, pp. 317-324, ISBN: 978-3-642-32688-2. (Cited on pp. 14, 17.)

Münch, D., K. Jüngling, and M. Arens

2011 "Towards a Multi-purpose Monocular Vision-based High-Level Situation Awareness System", Anglais, in *International Workshop on Behaviour Analysis and Video Understanding (ICVS 2011)*, p. 10. (Cited on pp. 14, 24.)

Münch, David

2013 *Towards an Applied Cognitive Vision System*, Technical Report, Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, Ettlingen. (Cited on pp. 13, 14.)

Nagel, H.H.

2000 "Image sequence evaluation: 30 years and still going strong", in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 1, 149-158 vol.1. (Cited on p. 13.)

Nagel, H.H

2004 "Steps Toward a Cognitive Vision System", *AI Magazine*, 25, 2, pp. 31-50, ISSN: 0738-4602. (Cited on pp. 7, 17.)

Oh, Sangmin, A. Hoogs, A. Perera, N. Cuntoor, Chia-Chih Chen, Jong Taek Lee, S. Mukherjee, J.K. Aggarwal, Hyungtae Lee, L. Davis, E. Swears, Xioyang Wang, Qiang Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, Bi Song, A. Fong, A. Roy-Chowdhury, and M. Desai

2011 "A Large-Scale Benchmark Dataset for Event Recognition in Surveillance Video", in *CVPR*, pp. 3153-3160. (Cited on pp. 31, 33-35, 38.)

Resch, A., R. Pfeil, M. Wegener, and A. Stelzer

2012 "Review of the LPM local positioning measurement system", in *Localization and GNSS (ICL-GNSS), 2012 International Conference on*, pp. 1-5. (Cited on p. 62.)

Robertson, N.N and I. Reid

2006 "A general method for human activity recognition in video", *Computer Vision and Image Understanding*, 104, 2 3, <ce:title>Special Issue on Modeling People: Vision-based understanding of a person s shape, appearance, movement and behaviour</ce:title>, pp. 232-248, ISSN: 1077-3142. (Cited on pp. 4, 5.)

Thórisson, K.R and H. P. Helgasson

2012 "Cognitive Architectures and Autonomy: A Comparative Review", *Journal of Artificial General Intelligence*, 2, 2, pp. 1-30. (Cited on p. 13.)

Van de Weghe, N. and P. De Maeyer

2005 "Conceptual neighbourhood diagrams for representing moving objects", in *Perspectives in Conceptual Modeling*, Springer, pp. 228-238. (Cited on p. 10.)

Viola, P. and M. Jones

2001 "Rapid Object Detection using a Boosted Cascade of Simple Features", *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1, p. 511, ISSN: 1063-6919. (Cited on p. 30.)

Vu, V-T., F. Bremond, and M. Thonnat

2003 "Automatic video interpretation: a novel algorithm for temporal scenario recognition", in *Proceedings of the 18th international joint conference on Artificial intelligence*, Acapulco, Mexico, pp. 1295-1300. (Cited on p. 7.)

Ye, J., S. Dobson, and S. McKeever

2012 "Situation identification techniques in pervasive computing: A review ", *Pervasive and Mobile Computing*, 8, 1, pp. 36-66, ISSN: 1574-1192. (Cited on pp. 6-8.)