Recognition of Traffic Situations based on Conceptual Graphs

Zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

der Fakultät für Maschinenbau
Karlsruher Institut für Technologie (KIT)
genehmigte

Dissertation

von

DIPL.-ING. NATASA BOROCH

aus Stuttgart

Tag der mündlichen Prüfung:	10.06.2013
Hauptreferent:	Prof. Dr.-Ing. C. Stiller
Korreferent:	Prof. Dr. W. Rosenstiel

Acknowledgments

I would like to express my sincere gratitude to everyone who, by their support, encouragement, help or remarks, have contributed to the fulfillment of this work.

My deepest thanks go to my doctoral supervisor, Prof. Dr.-Ing. Christoph Stiller, for his guidance and confidence throughout the entire process.

I count myself particularly lucky to having pursued this work in the Robert Bosch GmbH in Leonberg in the department for video-based driver assistance systems headed by Marcus Lorei. This department is a great place for an interdisciplinary work, with excellent equipment and helpful, knowledgeable colleagues.

I would like to especially thank my supervisor Dr. Thomas Heger. The motivating and always constructive discussions, a feeling of freedom and trust as well as his warm support I experienced on every stage of my work.

The work would be much more difficult without the friendly support of my colleagues from the group, especially Dr. Thomas Wehking, Carsten Clemens and Christian Renner that shared with me their precious expertise and Jan Jatzkowski and Maik Schulze who made a great job in the implementation and testing phase.

Finally I would like to thank my parents, my husband and son for their love and support.

Stuttgart, July 1st, 2011 Natasa Boroch

Kurzfassung

Videobasierte Fahrerunterstützungssysteme sind ein wichtiger Bestandteil der Fahrzeug-Sicherheitsstrategie. Diese Arbeit nutzt die Ego-Fahrzeug Umgebungsbeschreibung in mehreren Frames, die durch ein Stereosystem generiert werden. Damit sollen sowohl die Beziehungen zwischen Ego-Fahrzeug und anderen Objekten aufgezeigt werden als auch die Beziehungen der Objekte untereinander. Basierend auf diesen Informationen kann das Verhalten der Objekte prognostiziert werden. Damit entsteht die Möglichkeit, den Fahrer des Ego-Fahrzeugs zu warnen und, falls erforderlich, sein Verhalten zu beeinflussen. Eine entscheidende Situation, in welcher der Fahrer gewarnt werden sollte, ist der drohende Zusammenstoß zwischen Ego-Fahrzeug und einem sich bewegendem Objekt.

Diese Arbeit analysiert, in wieweit konzeptuelle Graphen geeignet sind, Problemstellungen der Situationserkennung zu beschreiben. Konzeptuelle Graphen sind eine Untermenge der Prädikatenlogik erster Ordnung, die auf der Graphentheorie basieren. Für ein tieferes Verständnis wird das Prototypen-Modell vorgestellt. Der Szenengraph wird aus den zuvor gewonnenen Informationen über Objekte und Fahrbahnen in Form eines konzeptuellen Graphen erstellt. Die geschieht gemäß der Konzepttyp-Hierarchie, der Beziehungstyp-Hierarchie sowie den Regeln und Einschränkungen. Regeln beinhalten zudem Definitionen von komplexen Konzept- und Beziehungstypen. In einem nächsten Schritt werden die Graphen mittels einer Abfrage, welche die Situation repräsentiert, zusammengeführt. Das Ergebnis ist die Information über die Situationserkennung.

Die Funktionalität des Models wird anhand von realen Verkehrssituationen dargestellt. Im Einzelnen sind dies der Zusammenstoß zwischen Ego-Fahrzeug und Objekt, der Zusammenstoß zwischen zwei sich parallel bewegenden Objekten, bei denen es sich nicht um das Ego-Fahrzeug handelt und schließlich der Fahrspurwechsel des Objekts vor dem Ego-Fahrzeug von der eigenen auf die linke Spur. Dazu wird die Methode zur Berechnung der Wahrscheinlichkeit eines Zusammenstoßes analysiert und ein Algorithmus hergeleitet, in welchen die entsprechenden Parameter eingeflossen sind (Focus of Expansion, Zeit bis zum Zusammenstoß und Distanz zum Objekt). Die Arbeit schließt mit der Beurteilung von konzeptuellen Graphen im Zusammenhang mit den logisch ableitenden, darstellerischen und wissenstechnischen Anforderungen der Szeneninterpretation.

Schlagworte: Fahrerassistenzsysteme, konzeptuelle Graphen, Szenenbeschreibung, Szeneninterpretation, Verkehrssituationen

Abstract

The video-based driver assistance systems are an important part of the vehicle safety strategy. The ego vehicle environment description in several frames provided by a stereo system was used in this work to depict the relations between the ego vehicle and other objects, as well as the relations of those objects between each other. Based on this information the behavior of the objects could be predicted. This gives the possibility for the driver of the ego vehicle to be warned and, if needed, to adjust his behavior. One of the situations when it is crucial for driver to be warned is a collision between ego vehicle and a moving object.

This contribution investigates the suitability of conceptual graphs for situation recognition. Conceptual graphs are a subset of First Order Logic that use graph theory. The prototype model is introduced. The scene graph is created in the form of a conceptual graph according to the concept type hierarchy, relation type hierarchy, rules and constraints using the previously obtained information about objects and lanes. Rules also include definitions of complex concept and relation types. The graphs are then matched using projection with the query conceptual graph, which represents the situation. The information about the detection of situation is given as a result.

The functionality of the model is shown on the real traffic situations, such as a collision between ego vehicle and object, a collision between two parallel moving objects, other than ego vehicle, and a change from own lane to the left one for the object in front of ego vehicle. To attain this the method for the estimation of collision probability is studied and an algorithm incorporating the parameters (Focus of Expansion, Time to Collision and the distance to the object) is proposed. This thesis concludes with an evaluation of the conceptual graphs with the respect to reasoning, representational and knowledge engineering requirements of scene understanding.

Keywords: Driver assistance systems, conceptual graphs, scene description, scene interpretation, collision detection, traffic situations.

Contents

List of Symbols

Symbols

$\mathbf{x} = (x, y, z)^{\mathrm{T}}$	vector
$\mathbf{x}^{\mathrm{T}}$	transposition of vector $\boldsymbol{x}$
$\boldsymbol{R}$	matrix
$\wedge$	conjunction
$\vee$	disjunction
$\cup$	set union
$\cap$	intersection

Abbreviations

2D	two-dimensional
3D	three-dimensional
6D	combination of 2D and 3D over time
ABS	Antilock Braking System
ADAS	Advanced Driver Assistance System
ADTF	Automotive Data and Time triggered Framework
BN	Bayesian Network
BPA	Basic Probability Assignment
CAN	Controller Area Network
CG	Conceptual Graph
CHMM	Coupled Hidden Markov Models
CMOS	Complementary Metal Oxide Semiconductor
CWA	Closed World Assumption
DL	Description Logic
DAS	Driver Assistance System
ECU	Electronic Control Unit
EKF	Extended Kalman Filter

ESP	Electronic Stability Program
FCW	Forward Collision Warning
FOC	Focus of Contraction
FOE	Focus of Expansion
FTA	Fault Tree Analysis
GPS	Global Positioning System
GUI	Graphical User Interface
HMM	Hidden Markov Models
IMO	Independent Moving Object
IR	Infra Red
KB	Knowledge Base
LDA	Lane Detection Algorithm
LDW	Lane Departure Warning
LIDAR	Light Detection and Ranging
LVDS	Low Voltage Differential Signaling
OWA	Open World Assumption
PDF	Probability Density Function
RSR	Road Sign Recognition
RADAR	Radio Detection and Ranging
SGT	Situation Graph Tree
TTC	Time to Collision
TTR	Time to React
WCS	World Coordinate System

1 Introduction

1.1 Motivation

In the last years the number of vehicles on the roads has grown rapidly. Even in the regions with the fastest development of infrastructure, especially in big agglomerations, the density of traffic is very high and safe driving demands maximal concentration and attention from drivers.

The number of fatalities in road accidents in 1970 in Germany was more than 19000 [Off09]. Although the number of vehicles is many times greater than it was in the seventies, the number of fatalities has drastically decreased to less than 4500 in 2008 [Off09]. A significant influence on this development comes from wide application of passive safety systems like ABS or ESP. Further reduction is expected by the wide application of advanced driver assistance systems (ADAS).

The understanding of the situation currently involving the vehicle is an important information for an advanced driver assistance system. The goal of such systems is to increase the passenger safety and comfort by providing the driver with environmental information, such as the current and the expected future behavior of the traffic participants and obstacles. Maneuvers at intersections involving oncoming vehicles especially represent a potential hazard for the own vehicle e.g. due to distractions of other traffic participants [Kä10b].

Such information about vehicle environment can be obtained from different kind of sensors. Many approaches for the collision avoidance or mitigation based on RADAR, LIDAR, video and their combinations can be found in literature. The video based systems provide certain advantages because they can be synergized in hardware implementation with some other video based driver assistance systems, e.g. Lane Departure Warning (LDW) or Road Sign Recognition (RSR) [Bor10b].

1.2 Objectives of the Thesis

The main goal of the thesis is to investigate the suitability of conceptual graphs for situation recognition. For this purpose a prototype of a model that analyzes the information provided by the stereo video system and which provides the high-level description of the ego vehicle environment will be created. The model should be able to recognize different types of real traffic situations, e.g. the collision between two oncoming vehicles, if one of them is overtaking the other, or the lane change of the vehicle driving in front of ego vehicle and could potentially be the core of the system which supports the driver, extending his safety and comfort.

Situation recognition is a part of scene interpretation[1] process. According to Neumann image understanding is the recovery and interpretation of a scene on the basis of images allowing at least one of the following operations [Neu03a]:

- output of a linguistic scene description
- answering linguistic inquiries concerning the scene
- collision-free navigation of a robot in the scene
- regular grasping and manipulating of objects in the scene.

Method developed in this thesis makes contributions to the last three operations, with the special focus on the second one. The output of the model can be seen an answer to the question: “Can the situation, that should be recognized, be detected in the described scene?”.

An evaluation of the method will be made based on reasoning, representational and knowledge engineering requirements of scene understanding [Hum09]. Those requirements are described in the following.

A scene interpretation should be based on the collective knowledge, because the recognition of different collision situations requires the gathered knowledge about detected lanes and objects in the scene. Also the reasoning should be task-driven, so only the relevant parts of the scene are examined during answering a particular query. For example, when the vehicle is changing from left to ego lane, right lane is not considered. In this way the situation recognition process is faster. The joint use of those two principles requires a combined bottom-up/top-down reasoner.

For the scene understanding operations a representation of semantic knowledge is necessary. Since a part of the semantic knowledge is programmed by knowledge engineers, the transparency of the representation, easiness of maintenance and extendability are in focus. The terms should be explained clearly and there should be no place for ambiguities, that e.g. two knowledge engineers understand behind a concept definition different meanings.

[1] The terms: image understanding, scene understanding and scene interpretation are here used as synonyms.

The representation should be able to incorporate wide spectrum of qualitative and quantitative data from different abstraction layers. The attributes, like size and position, need to be represented as well as the information that e.g. one car is preceding the another. This is supported by a hierarchical and relational representation.

Furthermore, a representation must allow incomplete data. The values of some parameters can be unknown or the input can be uncertain, since the sensor output is sometimes not present or it is noisy. A representation must allow a generic parameter space since the number and type of objects is not known a priori and changes over the time.

Since the system needs to deal with large amount of knowledge and the situation recognition should be fast, good engineering techniques are required. This can be achieved by using of object-oriented programming, effective algorithms, graphical user interface (GUI), for both user and knowledge engineer, and extensive knowledge engineering literature. Although prototypical implementation cannot satisfy all of those criteria, this work will show the direction in which the implementation can be developed to be able to do that.

An adequacy of conceptual graphs in fulfilling those requirements is evaluated. The other objective of this work is to verify the prototype's functionality in typical traffic situations that may lead to collision, based on the real video sequences. To achieve this target, the novel algorithm for collision detection is proposed and implemented. The output of the algorithm, the collision probability between ego vehicle and the object, is delivered as an object attribute to the model.

1.3 Thesis Outline

The thesis consists of seven chapters. Chapter 1 gives a short introduction to the motivation and the objectives of the thesis. In Chapter 2 basic concepts of optical flow and stereo geometry are presented. Different approaches for collision detection and scene description and interpretation, as well as the theory of conceptual graphs are also described in Chapter 2. A discussion about the presented approaches is conducted at the end of the chapter.

Chapter 3 gives the information about measurement methods and the way the video sequences are realized. The description of parameters for developed collision detection algorithm and the algorithm itself are given in Chapter 4. A model for ego vehicle environment description and interpretation based on conceptual graphs is described in Chapter 5.

In Chapter 6 the results are presented and discussed. Chapter 7 briefly summarizes the most important results of this work and gives an evaluation of the method. At the end potential subjects for the future are listed.

2 Literature Review

2.1 Optical Flow

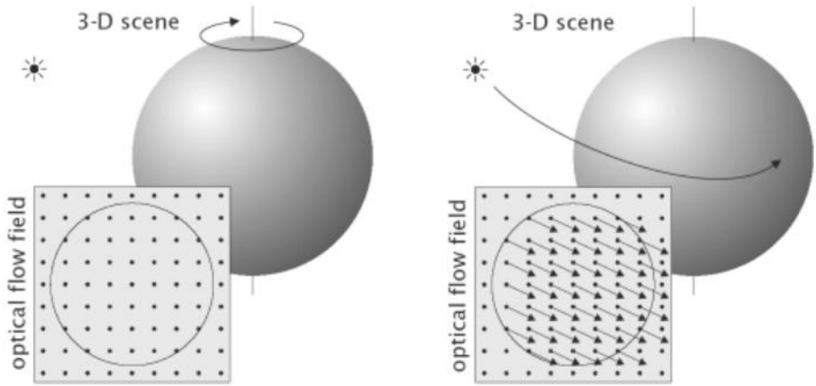
Optical flow describes the motion of gray values in the image particularly assuming constant illumination. Optical flow is also described as an approximation of the motion field, which is a 2D projection of a relative motion between camera and objects in 3D scene ([Bal82]). The moving patterns cause temporal varieties of the image brightness. It is assumed that all temporal intensity changes are due to motion only, but there are some exceptions ([Bar94], [Bar04]). A difference between optical flow and motion field can be described on one example depicted in Fig. 2.1 ([Jä99]):

- Fig. 2.1(a) shows a rotating sphere under a light source fixed in position and intensity. There will be no changing intensity and hence no optical flow even though the sphere is in motion.
- Alternatively, if the sphere is stationary (Fig.2.1(b)), but the light source's position and/or intensity changes, optical flow can be measured even though there is no motion.

Further assumptions, like constant light source, small changes in relative distance from camera and light source to objects between two images and rigidness of objects make sure that the optical flow can be approximated with the motion field.

If an object executes a relative translational motion toward the camera, an optical flow field expands. The intersection point of the lines containing optical flow vectors of the object is called focus of expansion (FOE) (Fig. 2.2). In the case of a relative object movement away from the camera the intersection point is called focus of contraction (FOC) respectively. If there are independent moving objects (IMOs) in the camera field of view, the relative motion to the camera can result in FOEs or FOCs additional to the one effected by the egomotion. That is why the FOE can be used for segmentation of IMOs from the static background. Exceptions are objects moving exactly parallel to the egomotion.

During the calculation of optical flow in the image sequence, one comes across a correspondence problem: which pixels of the image correspond to which pixels of



(a) (b)

Figure 2.1: Difference between optical flow and motion field: (a) A rotating sphere with fixed illumination source causes no optical flow. (b) A stationary sphere with non-fixed illumination source causes optical flow.

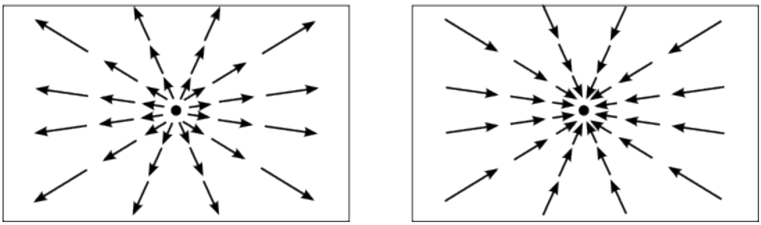


Figure 2.2: Optical flow field if the motion is pure translation: In the left figure the vectors originate from FOE and in the right they terminate at FOC

the next frame of the image sequence. Generally, it is not possible to determine the correlated pixels unambiguously, which the following example illustrates (Fig. 2.3). Fig. 2.3(a) shows a circle, which does not have unique features that could be tracked and Fig. 2.3(b) the grid where shifts left or right, as well as shifts that are a multiple of the distance between grid lines, are not detectable before the whole grid is visible. It can also come to ambiguity when a part of an object is visible in one frame and in the next occluded, because the correspondence can not be found. There are two main approaches to answer this issue:

1. compute the optical flow and use its geometrical properties to deduce three-dimensional information about the scene and the motion

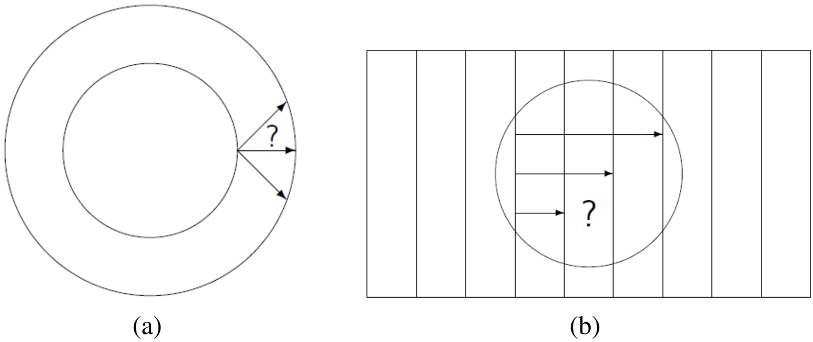


Figure 2.3: Correspondence problem: (a) no unique features, (b) periodical texture

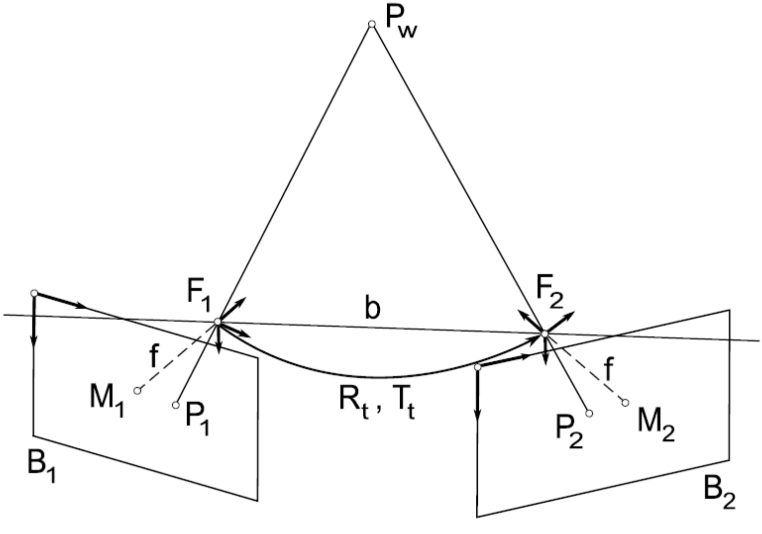


Figure 2.4: Stereo geometry

2. convert the motion problem to a stereo problem and find the correspondence between a number of feature points in the image in two consecutive frames (the feature points should have local variation in two directions)

2.2 Stereo Geometry

Fig. 2.4 ([Ste08]) shows a typical construction of a stereo system. It is assumed that both cameras are separately calibrated leading to intrinsic parameters and also calibrated to each other relative to their position which results in extrinsic parameters. Intrinsic parameters indicate characteristics like camera constant, principal point and pixel pitch and extrinsic the transformation of the coordinate system of one camera into the coordinate system of another ([Har03]). A point P_w in world coordinate system and its projections P_1 and P_2 on the image planes B_1 and B_2, respectively, is considered. F_1 and F_2 represent the optical centers and M_1 and M_2 the principal points of the image planes. Camera constant f is defined as the orthogonal distance between the projection center and the image plane and it is the same for both cameras. The position and orientation of the cameras to one another are described through the rotational component R_t and the translational component T_t which yields:

$$P_2 = R_t \cdot P_1 + T_t. \tag{2.1}$$

The origin of the image plane lies in its upper left corner and every image point P can be described with its horizontal component u and vertical component v. The following explanation is given for a special case when optical axes are parallel and the image planes are just translated for the value of baseline b. World coordinate system (WCS) is defined with the origin in F_1 and x-axis forward, y-axis pointing to the left and z-axis up. If the lens distortion is neglected, the projection $P_1 = (u_1, v)^T$ of the point $P_w = (X, Y_1, Z)^T$ on the image plane of the first camera with regard to the principal point $M_1 = (u_0, v_0)^T$ can be determined as follows:

$$P_1 = \begin{pmatrix} u_1 \\ v \end{pmatrix} = \begin{pmatrix} f \cdot \frac{Y_1}{X} + u_0 \\ -f \cdot \frac{Z}{X} + v_0 \end{pmatrix} \tag{2.2}$$

Likewise the horizontal component of the projection $P_2 = (u_2, v)^T$ of the point $P_w = (X, Y_2, Z)^T$ defined in WCS, with the origin in F_2 with regard to $M_2 = (u_0, v_0)^T$ can be determined:

$$u_2 = f \cdot \frac{Y_2}{X} + u_0 \tag{2.3}$$

Disparity δ is defined as the difference between horizontal components of projections and its unit is pixel (pel):

$$\delta = u_1 - u_2. \tag{2.4}$$

Using the knowledge about disparity z-coordinate of the point P_w can be determined:

$$X = \frac{f \cdot b}{\delta}. \tag{2.5}$$

Having the depth information it is also possible to calculate x-coordinate and y-coordinate of the point P_w:

$$P_w = \begin{pmatrix} X \\ Y_1 \\ Z \end{pmatrix} = \begin{pmatrix} X \\ \frac{u_1 - u_0}{f} \cdot X \\ \frac{v_0 - v}{f} \cdot X \end{pmatrix} \tag{2.6}$$

2.3 Collision Detection

The basis for collision detection is a so called "Sailor's Test". If the angle under which an object B is seen from an object A remains constant over time, and the apparent object size is growing, then a collision will take place. This is equivalent to the fact that the FOE of the relative motion between the camera and the moving object lies within the growing picture of an object in the image.

As mentioned in Section 1.1 there are different approaches for collision detection in driver assistance systems. The differences on one hand, lie in the sensors that are used for the perception of the environment, and on the other, in the application field, so that they are based on different requirements and assumptions. In this section some of those approaches will be introduced.

2.3.1 Multi-Sensor based Collision Detection

Vankatesh et al. in [Ven05] introduce for Neuro-Fuzzy Vehicle Collision Prediction System a method that uses laser for obtaining information about surroundings. With the help of one more sensor that measures a speed of ego vehicle, the system determines the distance between ego vehicle and object, the relative speed to each other as well as the direction of the object relative to ego vehicle. Obtained information is passed to a two-stage prediction system. The first stage clusters these data in order to prioritize them based on their relevance. The second stage is a neuro-fuzzy sub-system, which processes these data applying the rules which analyze the possibility of a collision. After this the driver is warned accordingly.

In [Jan02] Jansson et al. look at three different types of sensors, that are used, separately or combined, for gathering the data about the road environment in their

Forward Collision Avoidance System: millimeter RADAR,IR RADAR and camera. The calculation of collision probability is done by forming the joint PDF of the host's and the other objects' position relative to each other, taking into account measurement uncertainty and driver maneuvers. The probability is calculated by integrating the joint PDF over the area which corresponds to a collision (the area where the two objects physically overlap).

Hillenbrand et al. in [Hil06] describe an approach for detection of collision situations. Time to react (TTR), the time left for the driver of the ego vehicle to begin a maneuver that circumvents a collision with the object, is used as a metric to assess the criticality of traffic situations. To approximate TTR they look at a small set of selected maneuvers that cover the different types of physical constraints: maximum braking, maximum acceleration, minimum radius steering to the left, and minimum radius steering to the right.

In [Kae09] Kaempchen et al. explain their collision prediction algorithm. It simultaneously considers all physically possible trajectories of the object and ego vehicle and it can be applied to all different scenarios including rear-end collisions, collisions at intersections, and collisions with oncoming vehicles. The algorithm checks all different combinations of trajectories from the host vehicle and the opponent vehicle. They are defined by a tuple of angles $[\gamma_{host}, \gamma_{opp}]$ in Kamm's circle. Two lines are constructed from the direction of the relative velocity vector and the two corners of the host vehicle's front. These lines intersect with the opponent vehicle and define an area. The collision index is defined by the size of this area. Considering all different combinations of γ_{host} and γ_{opp}, a matrix can be composed in which regions of a central crash, almost collision avoidance and collision avoidance can be detected.

Takahashi et al. [Tak00] give an overview of the system for prevention of rear-end collisions and prevention of lane departure which consists of:

- Lane-Keep Assistance System, that uses a CCD camera to recognize the lane markers, assisting the driver to keep lane position and prevent departure from lane;
- ACC with Stop & Go System, that uses a millimeter-wave RADAR to track the vehicle in front by keeping the appropriate distance between the two cars during the operation including start and stop;
- Collision Velocity Reduction System, that applies the brakes sharply to reduce the vehicle's collision velocity when the system has estimated that a collision with the vehicle in front is likely to happen. When the distance from the vehicle in front is progressively reducing, this system urges the

> driver to exercise caution by way of a warning sound. If the driver fails to respond and proceeds to close in on the vehicle in front, to the point where the distance between the vehicles is dangerously close, it urges the driver to avoid the vehicle ahead by steering and applying the brakes by way of a warning sound and automatically commences braking.

Salim et al. in [Sal07] propose an intersection collision detection system that is able to adapt to different types of intersections by acquiring the collision patterns of the intersection through data mining. Collision patterns that are specific to that intersection are stored in a knowledge base to select vehicles which are exposed to a high risk of collision. A pair-wise vehicle contention algorithm is used for collision detection. A future collision is detected if the difference in the time for both vehicles to reach the collision point is less than a certain parameter, representing the collision region, depending on the vehicle size.

In the article [Men04] Mendes et al. address the development of an anti-collision system based on a laser scanner, for low speed vehicles. A database with the objects being tracked at each interval of data processing is created. Using the velocity estimates, the time of the collision with the vehicle is computed for each database object. The worst case time-to-collision and the correspondent predicated impact point on the vehicle, are sent to the path-following controller, which provides collision avoidance behavior using this information.

Fritsch et al. ([Fri08]) use a 3-phase danger handling scheme depending on the distance and relative speed of a recognized obstacle. When the time-to-collision is less than $3s$ a visual and acoustic warning is issued and the brakes are prepared. If teh dangerous situation is not resolved by the human driver, in the second phase the brakes are engaged with a deceleration of $0.25g$ followed by hard braking of $0.6g$ in the third phase. The overall architecture is organized in a brain-like fashion by separating the identification of new objects from the continuous tracking of previously detected objects. Through tuning the attention system to interesting objects, the system analyzes only relevant parts of the scene. By performing an EKF-based fusion of different depth sources, good depth estimates are obtained for all objects on the street.

In the article [Bot10] a collision detection algorithm based on the curvilinear-motion model for trajectory estimation is presented. The algorithm takes into account the ego vehicle's driving state and the high-level representation of surrounding objects. It is checked for all the possible combinations of ego vehicle and object trajectories whether a collision should occur in the near future. Hereby, a maximum prediction time must be set, e. g., two seconds. The first time instance when a collision occurs is stored for the considered combination. If no crash is detected in the maximum prediction time, then the value "‘Inf"’ is stored for the considered trajectory combination. A crash is considered unavoidable in a scenario in which every possible combination of ego vehicle and object trajectories lead to collisions. In order to guarantee both real-time constraints and effective collision detection, a method for the pre-selection of potential collision opponents, based on the Random Forest classifier, is employed.

Labayrade et al. ([Lab07], [Lab05]) describe the REduce Speed of Collision Under Emergency (RESCUE) collision-mitigation system based on data fusion between stereovision and laser scanner. The first subsystem is a generic obstacle-detection system designed to adapt to all types of road geometry, including nonplanar surfaces. The second one is a warning-area generator that builds a warning area corresponding to the path of the equipped vehicle. Obstacles outside this warning area are not taken into account. This warning area also allows the handling of narrow roads and tight turns. The third subsystem is the very automatic-braking system, which is used to dissipate the kinetic energy of the vehicle before a collision. The decision unit sends a warning message to the automatic-braking system if the certainty of the track after the fusion step is above 0.7, $0 < TTC \leq 1s$ and the intersection between the track and the warning area is not empty.

In the article [Che07] Cheng et al. propose a road situation analysis based on traffic rules, vehicle dynamics, driver behavior, sensor uncertainty and vehicle state. A multiple-obstacle detection and tracking algorithms using multiple sensors including RADAR, LIDAR, and a camera are used, and future collisions are assessed by computation of local tracks of moving obstacles using extended Kalman filter, maximum likelihood estimation to fuse distributed local tracks into global tracks, and finally, computation of future collision distribution from the global tracks.

Chang et al. ([Cha04]) use stereo system for the imminent collision detection. First, possible imminent collision threats are detected upon their presence in the stereo depth image and relevant properties of the threat, such as size, height, width, location, and the motion of the treat are estimated. Finally, the threat car trajectory and the potential colliding spot on the threat car are predicted.

VisionSense system described in [vD05] is equipped with CCD cameras integrated in both outside mirrors, and microwave radar, mounted in the vehicle rear bumper. It combines lateral collision warning with a vehicle-to-vehicle communication system and assists drivers while making a lateral movement with warning signals, if it detects a vehicle in the driver's blind spot. Besides the TTC value, the distance headways between the subject and the three nearest vehicles is estimated. A conflict is defined as a TTC shorter than 3 seconds or distance headway shorter than 5 meters.

In the article [Sos07] an object is considered as an obstacle if the relative speed and distance between ego vehicle and object is less than the distance in which the vehicle could stop without hitting the object in front of it. This algorithm can be used with IR, LIDAR or RADAR.

There are some approaches that use ultrasonic sensor. Song et al. ([Son04]) designed and implemented an ultrasonic sensor system for lateral collision avoidance of vehicles at low speeds. Multiple sensors are installed along the side of the ego vehicle. The developed sensor system is useful for detecting vehicles, motorcycles, bicycles and pedestrians that pass by the lateral side of a vehicle. The warning is issued if the distance to an obstacle is less than $4m$. Agarwal et al. ([Aga09]) also installed multiple sensors, but on the front of the vehicle. Those sensors are used for a short-range collision warning and parking assistance systems where the vehicle speed is relatively low.

2.3.2 Video based Collision Detection

The most popular data source for collision detection is video implemented in the vehicle as a mono (single camera) or stereo (two cameras) system.

In [Sun05] Sun et al. introduce the collision-avoidance strategy based on the vision perception and implemented by a fuzzy decision making mechanism. The proposed Fuzzy Inference System is based on fuzzy logic and approximate reasoning. Fuzzy Knowledge Base contains the domain knowledge composed of fuzzy database described by linguistic variables for input and output, and fuzzy rule base is composed of many fuzzy production rules to describe the behavior of system. The essential work of approximate reasoning is to calculate the strength of fired rules, compute the degree of each fired rule's conclusion and defuzzify the output's real value. The three inputs here are relative velocity, absolute velocity and relative distance. The outputs are two kinds of warning approaches. Considering the real time requirement, these three input factors are combined to define a degree of exceeding safe-distance (DESD) as the ratio of the difference between safety and relative distance and safety distance. If $DESD \geq 0$, the possibility of collision exists. The output factors are replaced by a safety coefficient (SC) to judge the degrees of warning and to reduce or increase the velocity of a following vehicle and three fuzzy rules are defined: if DESD is low then SC is high, if DESD is medium then SC is medium and if DESD is high then SC is low.

There are several approaches using a single camera for forward collision detection. Dagan et al. [Dag04] describe a Forward Collision Warning (FCW) system that by means of TTC threshold decides if the detected object presents an immediate danger for ego vehicle. If TTC falls below the threshold, the decision is made using the trajectories of the right and left object edge that are calculated based on the projection equation. This system can be combined with LDW system to determine the lateral position of other vehicles. The method presented in the article [Mar08] is based on TTC calculation taken from an optical flow analysis. The number of pixels that have a small TTC is analyzed and if the number of pixels goes beyond a certain established threshold, the system will indicate that the ego vehicle is facing a dangerous situation. Article [Chu03] shows a vehicle safety crashworthy alarm expert system based on the time-safety model and driving rule. If the maintained distances to the vehicle in front of ego vehicle driving in the same direction differ a lot, it's accidental and not the driver's driving habits. On the other hand, if the maintained distances differ only a bit, it' s the driver's driving habits, and should be analyzed to assess if the driving-habits are safe. If the average distance is greater than the safe distance, the driving habits are safe. Otherwise, they are dangerous. Liu et al. ([Liu08]) use headway distance estimation model to detect potential forward collision.

In the last few years some approaches that use stereo systems also appeared. In the article [Ned08] a FCW approach based on a 3D Elevation Map provided by a Dense Stereo Vision System is presented. Nedevschi et al. detect the obstacle delimiters extracted from the elevation map that represents a description of the scene, derived from the raw dense stereo information. The elevation map cells are classified into drivable points, curb points and object points. Taking into account the ego vehicles parameters, such as wheel speed and yaw rate, they evaluate the ego vehicle trajectory and associate to it a driving tunnel. A warning is generated when the obstacle delimiters intersect the current driving tunnel at an unsafe distance.

Sailor's test is used by Woelk et al. [Woe04] for side collision detection with the help of camera and GPS. They describe three cases when this kind of detection fails:

- if there is no relative motion between the camera and the moving object and therefore no FOE of the relative motion between ego vehicle and the moving object (in this case there is also no collision);
- if the FOE of the relative motion and the FOE of the camera motion coincide;
- if the two FOEs and the object are collinear and the object is not located in between the two FOEs.

Sailor's test in its basic form is only valid when it comes to collision between the object and the camera. The geometry of the scene and the size of ego vehicle and the object must be considered because the camera is mounted on the ego vehicle. Taking this into account Metzler et al. [Met06] developed a system for early detection of side collision where the camera is sidewards positioned. A critical area is defined in the image, where the size of ego vehicle is regarded. If the relative motion of the object is directed to this critical area and the object is close enough to ego vehicle, a collision is detected. Therefore TTC is calculated, so the worst case distance estimation of the object is obtained under assumption of the chosen minimum object velocity.

2.4 Situation Recognition

There are many different approaches for situation recognition in literature. In this section some of them will be described. They are based on three different types of models: deterministic, probabilistic and fuzzy. In deterministic models the recognized situation is precisely determined through known relationships among states and events. Probabilistic models predict the situation by calculating the probability of all possible situations based on temporal and spatial parameters. A fuzzy model consists of a finite set of fuzzy relations that form an algorithm for recognizing the situation from some finite number of past inputs and outputs.

2.4.1 Deterministic Approaches

Arens and Nagel ([Are02], [Nag05]) deal with systems which rely on an explicated knowledge base which comprises knowledge about the geometry of the depicted scene and admissible trajectories of moving vehicles. Conceptual knowledge is required in order to transform quantitative results related to geometric properties into textual descriptions of vehicle maneuvers and their context. Here it is given in the form of Situation Graph Tree (SGT) (Fig. 2.5). The behavior of an agent is described in terms of situations an agent can be in. Transitions between these situations express temporal changes from one situation to another. Situation scheme describes the situation in two parts: a state scheme, which denotes the state of an agent and his environment with logic predicates, and an action scheme, which denotes actions the agent is supposed or expected to execute if each of state atoms is satisfied (i.e. situation scheme can be instantiated). The expectation of which situation the agent will instantiate at the next point in time can be expressed by prediction edges. Situation schemes together with prediction edges build situation graphs. These graphs are directed and can comprise cycles. Each path from start situation to an end situation defines a sequence of situations represented by the situation graph. Multiple specializations of one situation scheme are possible simply by connecting parent scheme with several other situation graphs. One situation graph can only specialize exactly one or none situation scheme. Therefore,

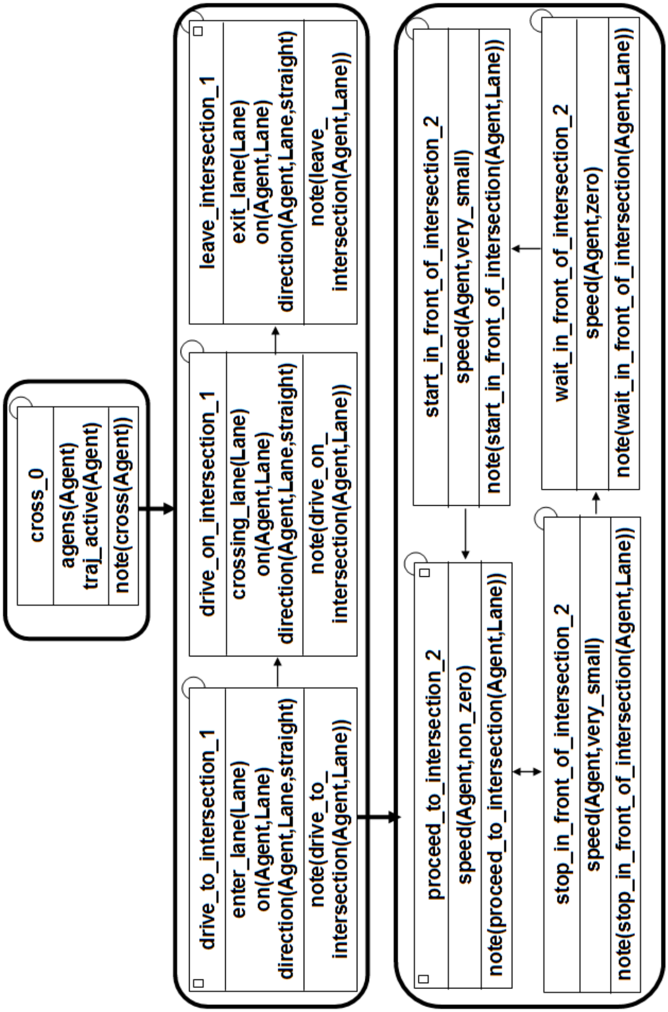


Figure 2.5: Part of the situation graph tree describing the behavior of vehicles on an inner city road intersection. Circles in the upper right corner of situation schemes indicate a self-prediction of that scheme. Small rectangles to the left or to the right mark that scheme as a start- or end-situation, respectively. [Are02]

situation schemes connected recursively to situation graphs build a situation graph tree. The recognition of situations is performed by a graph traversal. It is started in the root situation graph. Here, a start situation is searched for which each state atom is satisfied for the actual inspected point of time. If no such situation scheme exists, the traversal fails. Otherwise the observed agent is instantiating that situation. If this situation scheme is specialized further by any situation graph, these graphs are again searched for start situation schemes with state atoms also satisfied in the actual point of time. For each point of time an agent is instantiating situation schemes on several levels of detail, each on a path from a scheme in the root graph to the most special scheme.

In the approach of Lattner et al. ([Lat05]) a knowledge base is the central component for higher-level functionality. The architecture presented here is refinement of Dickmanns' architecture ([Dic02],[Dic05]) described in Section 2.4.3. A qualitative mapping module abstracts from the quantitative data and stores symbolic facts in the knowledge base (KB). During each cycle new facts are inserted into the KB, or existing intervals are extended (if a relation is still valid). In order to find out if an existing interval must be extended, the KB is queried of the existence of such relation directly before the current mapping interval. The KB allows storing relevant information for describing traffic situations: object classes, topological information, spatial relations, road network, speed, distance, traffic situation and background knowledge. For situation assessment different situation patterns must be defined. Patterns are abstract descriptions of situations where certain conditions hold. The patterns are based on the qualitative representation in the knowledge base. Complex patterns can be composed of the different basic predicates. Different pattern matching modules observe the KB at regular intervals and notify their initiators if certain patterns are detected. In each matching cycle all patterns are evaluated and all valid assignments are derived by an inference engine.

In [Mie04] Miene et al. claim that an abstraction to a qualitative description leads to more stable representations as similar situations at the quantitative level are mapped to one qualitative representation. The recognition and prediction of motion situations is based on the qualitative motion description. Their approach to qualitative motion description includes absolute motion of single objects in combination with the changes in their pairwise spatial relations over time. On a quantitative level the objects' absolute and relative motion is described by four types of time series: the motion direction and speed of the intelligent vehicle itself and the other moving object perceived, and the spatial direction and distance for each pair of objects. In a first abstraction step each time series is segmented into time intervals of homogeneous motion values. In a second step the attribute values describing the intervals are mapped onto qualitative classes for direction, speed or distance, respectively. They distinguish eight direction classes and five speed respective distance classes, which are organized in distance systems. The resulting motion description builds the basis for a qualitative interpretation of the dynamic scene. Domain knowledge concerning the function or type of objects involved in a situation, leads to more appropriate interpretations.

Neumann et al. ([Neu08],[Neu03b]) show that aggregates composed of multiple parts and constrained primarily by temporal and spatial relations can be used to represent high-level concepts such as object configurations, occurrences, events and episodes. A dynamic scene is captured by several cameras and processed essentially bottom-up to the level of geometric scene descriptions. It is assumed that at this level the scene is described by (partial) views of objects ("blobs"). Furthermore it is assumed that moving blobs can be tracked and grouped into blob motions. Blobs and blob motions constitute the visual evidence which is used for high-level interpretations. The conceptual framework for interpretations is provided in terms of scene models which range from single object models to complex occurrence models. Scene models are linked to the records of a vision memory and are considered the result of a learning process. Scene interpretation is modeled as a stepwise process which exploits the taxonomical and compositional relations between aggregate concepts while incorporating visual evidence and contextual information. Three basic interpretation steps can be identified: aggregate instantiation, instance refinement and instance merging. Aggregate instantiation is the act of inferring an aggregate from parts, also known as part-whole reasoning. There are two variants of instance refinement: instance specialization and instance expansion. Instance specialization means tightening properties and constraints, either along the specialization hierarchy or by checking objects for possible roles in aggregates. Instance expansion is the step of instantiating the parts of an aggregate if the aggregate itself is already instantiated. New instances may be generated at any level and in any branch of the compositional hierarchy depending on visual evidence, context information and current interpretation state. Hence, different sequences of interpretation steps may lead to identical instances which must be merged. The conceptual structure of multiple-object occurrences, in particular temporal and spatial relations, can be expressed in a Description Logic (DL) which meets specific representational requirements, including feature chains, the same-as construct, and a concrete domain extension for the representation of temporal and spatial constraints. Currently, there is no operational DL system which offers all of these language features.

In [Vac07] Vacek et al. present an approach that relies on case-based reasoning to predict the evolution of the current situation and to select the appropriate behavior. Case-based reasoning allows utilizing prior experiences in the task of situation assessment. All knowledge for describing situations is modeled in OWL-DL, which is a decidable fraction of first order logic. The domain knowledge is modeled with concepts, roles and assertions. Situations are represented by instances of concepts and roles. Background knowledge is available as rules. The quantitative data from the perception is mapped to a qualitative description because symbolic information is more feasible for reasoning processes. Part of this transformation is the evaluation of topological and spatial relations between objects. After the transformation, additional relations are evaluated by applying rules of the background knowledge. A situation consists of the road network of the local scene, all objects in this scene, behavior estimation of other traffic participants, the mission goal of ego vehicle and all relations. The current situation is described by a case and experience and problem solving knowledge is generated by the retrieval of known cases. These cases are then evaluated and the adaptation of existing problem solving knowledge to the current situation leads to the selected behavior. In order to benefit from newly acquired knowledge, the new case is retained in the case-base. This is done when a new situation is reached and the previous solution can be confirmed.

In [Foe06] Foedisch et al. contend that image- or map-based and symbolic information should be combined in the internal world model of the system, and that this information should be stored at multiple levels of resolution, which is consistent with the 4D/RCS reference architecture for vehicle control. 4D/RCS is a hierarchical architecture supporting knowledge representation at different levels of abstraction. Traditionally, the lowest levels of the architecture primarily contain state variables such as actuator positions and states of sensors for detecting motion. Higher levels of the hierarchy contain map-based information, with decreasing resolution and increasing spatial extent as one proceeds higher up the hierarchy. A combination of map-based representations and object knowledge bases are used, which contain names and attributes of environmental features such as road edges, holes, obstacles, ditches, and targets. These maps represent the shape and location of terrain features and obstacle boundaries. Still higher up the hierarchy is symbolic information referring to the location of vehicles, targets, landmarks, and local terrain features such as buildings, roads, woods, fields, streams, fences, ponds, etc. Groups of objects are represented on the top levels of the hierarchy. The goal of the road recognition process is to find associations between feature items and (primitive) road models, described by road width and lane structure, and eventually an interpretation of the scene. The features available from a priori databases lend themselves to the formation of a graph based representation of a road network. Once a graph has been formed, any number of search techniques may be applied to find a cost optimal trajectory through the graph. Typical cost criteria include such items as path length and road type.

An approach to make a situation assessment presented in [MH93] is to use a hierarchical plan description language to model possible driving maneuvers at qualitative and quantitative levels. These descriptions are put together in a knowledge base, representing a traffic behavior model. On one hand, the plan descriptions are used as behavior patterns for the plan recognition component, and, on the other, they have to provide knowledge about the execution of maneuver for the conflict detection component. Each agent in a traffic world has one or more plans how to reach their goal to arrive at a specific place. They are classified at strategic, tactical and operational levels. At the strategic level, an agent has to decide where to go and which route to take. At the tactical level, there are short-term plans and goals: the agent must choose and perform appropriate driving maneuver (e.g. overtake, turn left or right at the junction, etc.). A tactical plan is a partially ordered set of elementary driving actions, such as brake, accelerate, change lane, etc. A plan may contain sequential, parallel and conditional actions. These elementary driving actions belong to the operational level. Hazardous situations occur at the operational level as well as at the tactical level. At the operational level they result from an incorrect execution of elementary actions, while critical situations at the tactical level result from conflicts between the tactical plans of agents. Plans without conflicts are the plans which can be executed completely independently from each other or the plans that are chronologically and spatially synchronized.

Duchow et al. in [Duc06] propose estimation architecture for a city intersection detection that uses expressive knowledge representation to reduce estimation problem to a few feature detections and hypothesis verification. The information from digital maps is extended with estimated parameters from the image, e.g. number of lanes and lane width. An intersection modeling can be deductively derived from the ontology-based knowledge representation introduced for a discourse area Roads&Junctions. Concepts and relations of the discourse area Roads&Junctions are modeled under following aspects: semantic, geometry, mereology, topology and taxonomy ([Hum09]). A set of the generated intersection geometry hypotheses as well as the estimated vehicle position are tested in the hypotheses tester which is based on a probabilistic model of a local orientation feature. The model describes two classes of images: ones that are classified as correct for the given hypothesis and others that do not match the given hypothesis. For a local orientation feature an observer model is defined. Using Bayes rule a correctness of a hypothesis for a given image can be proved.

Dickmanns ([Dic02],[Dic05]) describes “Expectation-based, Multi-focal, Saccadic vision system”(EMS-vision) that is integrated in an overall architecture for autonomous mobile robots. Scene representation is done with object-oriented generic models and explicit homogeneous coordinate transformations (HCT) in a scene tree. The scene tree includes all objects and reference systems as nodes. Transformation variables between two nodes represent the edges. There are special agents for visual perception of members of object classes (roads with intersections, obstacles, other vehicles, landmarks). They feed a dynamic object database with their best estimates for the states of objects at video rate. All perceptual and behavioral capabilities are explicitly represented in the system allowing flexible activation. These maneuver representations may also be used for understanding the motion of other subjects. A situation is defined as a sum of all facts and influences that are important for a behavior decision. Situation assessment is performed looking at time histories of relevant other objects/subjects in conjunction and by extrapolating maneuvers started into the future (intent recognition) and behavior decision is derived from these results.

2.4.2 Probabilistic Approaches

The method of dangerous situation recognition in DAS using fault tree analysis was discussed in [Che06]. This is only a study and it has not been yet used in practice. Fault tree analysis (FTA) is a universal method which is widely used to analyze system's reliability and safety. By structuring fault tree, this method can give logical process from basic failure events (they are often damage or failure of system elements) to top event (accident), and can calculate probability of top event occurring when one or some of basic failure events occur. The fault tree is structured here for a sub-system of traffic system which is called traffic situation and includes vehicle, road and environment. Top event of traffic system is certainly a traffic accident. To a specific traffic situation, there are many types of accident. Method of classifying accidents is based on the causation difference of traffic accidents. The basic method for calculation of possibility of top event occurring is by means of calculating possibility of summation of events in all minimum cut sets. There were two types of bottom events of fault tree of a traffic situation: component failure and component using state. Possibility of the two type bottom events can both be evaluated with dynamical detection to state variables of vehicle, road and environment. The situation recognition method consists of three parts:

1. classification of the roads by the pattern of traffic accident occurring on them and definition of a set of traffic situations and structure fault trees for each traffic situation,

2. detection of state variables of system components and creation of a dynamical evaluating model of a possibility of bottom events,

3. determination of a strategy for dealing with a dangerous situation, including possibility threshold of a situation being dangerous, threshold and grade of warning, threshold of assistance control going into effect and method of warning.

Käfer et al. ([Kä10b], [Kä10a]) propose a framework for recognition of intersection situations involving two oncoming vehicles. The basis for this framework is a long-term motion prediction method (2-4 s ahead), which is applied to each vehicle separately. The motion patterns of vehicles are represented by trajectories, which are defined as ordered tuples combining states (position, yaw angle, velocity, yaw rate) with a time stamp. A motion database consisting of observed trajectories is built. As a measure for the similarity between trajectories they use the quaternion-based rotationally invariant longest common subsequence metric. A particle filter framework simultaneously tracks a large number of motion hypotheses and assigns a likelihood value to them. Knowledge about the lane geometry extracted from a map is used for penalizing unlikely predictions such as those crossing the edge of the road. The interaction behavior of the traffic participants is taken into account by a situation classifier. It is assumed that the vehicles are able to turn right (R), turn left (L), or drive straight on (G), which results in a set of possible situation classes LL,LG,LR,RL,RG,RR,GL,GG,GR, where in each pair the *i*th character denotes the motion of the *i*th vehicle.

Schubert et al. in [Sch10] introduce the situation assessment and decision algorithm based on a Bayesian network (BN) that enables the integrated handling of uncertainty from the perception to the decision stage. The main feature of BNs is the ability to perform probabilistic reasoning. Knowledge about the state of certain nodes (which is generally called evidence) can be incorporated into the network to calculate the probability distributions of all other nodes under the condition of the entered evidence. It is possible to deduce possible reasons for observed effects, or vice versa. Uncertain evidence (likelihood evidence or soft evidence) can also be entered into the network. To account for the uncertainties of the perception evidence, likelihood functions need to be defined for nodes that are directly influenced by the perception results. The parameters OwnLane, LaneLeft and LaneRight are used for modeling, if keeping or changing the lane can be considered safe. Thus, it has to be determined which vehicles travel on the lane under consideration, which of those vehicles is the closest, and if this vehicle is within a certain safety margin. The deceleration to safety time is proposed as a threat measure. It denotes the deceleration that has to be applied to a vehicle to maintain a certain safety time (with respect to another vehicle). The latter can be regarded a safety distance that depends on the absolute velocity of the vehicle.

In [Hua94] Huang et al. use a dynamic belief network to make inferences about traffic events such as vehicle lane changes and stalls. Belief networks are directed acyclic graphs in which nodes represent random variables usually discrete and arcs represent causal connections among the variables. Associated with each node is a probability table that provides conditional probabilities of the node's possible states given each possible state of its parents. When values are observed for a subset of the nodes, posterior probability distributions can be computed for any of the remaining nodes. This updating takes place using a compiled form of the belief network that is more suitable to propagating the influence of evidence to other nodes. For each vehicle in a traffic scene there is a separate belief network corresponding to it. The influence of nearby vehicles on the current vehicle is incorporated by assigning some nodes to those vehicles.

Large et al. in [Lar04] present an iterative planning approach that addresses obstacles motion prediction and vehicle motion planning. The case of an autonomous vehicle evolving in a confined environment observed by video cameras is considered. The environment is monitored by video cameras in order to learn the typical motions of the moving obstacles. Once the learning stage is completed, the future motion of any given obstacle can be predicted. The concept of Non-Linear Velocity Obstacle is used to estimate efficiently the safety of a vehicle's motion in the predicted environment. Non-Linear Velocity Obstacle is defined as the set of all linear velocities of the robot, that are constant on a given time interval and that induce a collision with an obstacle before the end of this time interval. This process is iteratively repeated to incrementally build a search tree, until a complete trajectory to the goal is found, or until the available computing time is out. The tree is updated to reflect the environment changes every time a trajectory is computed.

Hu et al. in [Hu06] introduce a motion analysis system, which is composed of four main modules: tracking of multiple objects, learning of statistical motion patterns, detection of anomalies, and prediction of future behaviors. The module for tracking multiple objects is implemented by clustering foreground pixels in each image frame and comparing pixel distributions between successive images. The outputs of this module are object trajectories and features (such as color, size, etc.). These outputs form the sample data for learning motion patterns. After enough sample data are acquired, object motion patterns are learned using a hierarchical trajectory clustering based on the fuzzy K-means algorithm. In the motion patterns, each pattern is represented by a chain of Gaussian distributions and statistical descriptions of typical motion patterns are then formed. In the module for anomaly detection, the probabilities of the matching between observed behaviors and the learned motion patterns are calculated first and then the probability values of abnormality of the observed behaviors. In the module for behavior prediction, partial trajectories are matched to the learned motion patterns, and future behaviors are stochastically inferred.

In [Alt08] methods for algorithmic verification are used for detection of dangerous situations. First, the set of all possible states which a system can reach is computed, starting from a set of initial conditions for a given set of input trajectories. The values of the initial states (position, velocity) of the traffic participants and their

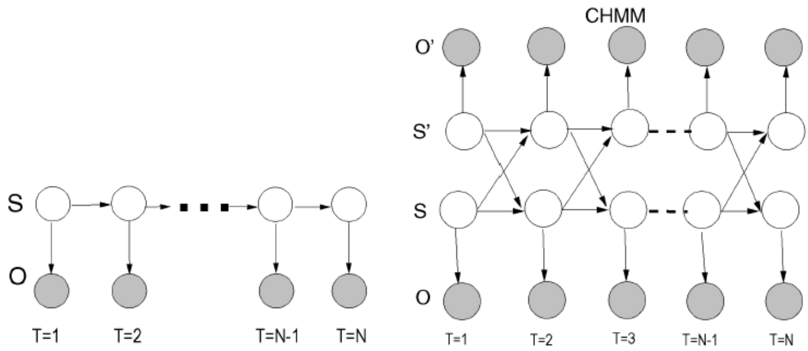


Figure 2.6: Graphical representation of HMM and CHMM rolled-out in time

possible inputs (acceleration values) are restricted to bounded sets. This restriction is considered by computing the probability distributions in a way, in which the possible states of each traffic participant are guaranteed to stay within the region of nonzero probability. This region together with the probability distribution is referred to as the stochastic reachable set. The reachable set is then intersected with regions in the state space that are unsafe. If the intersection is empty, the system is safe. For an online verification algorithm operating in a vehicle, unsafe sets are road areas which are occupied by surrounding traffic participants.

In [Sch08] Schneider et al. propose a method for interpreting the information about both own state and the driving environment by a detection of relevant driving situations and maneuvers. A driving situation is defined as the observable fragment of the spacious and temporal constellation of the traffic oriented factors of the road user's working environment. A driving maneuver is an action or an action sequence of a road user, that transfers a current driving situation into a new situation. A quality measure of the situation decision depending on the sensor inaccuracies is determined. The uncertainty in the system is composed on one hand of sensor-based inaccuracies of the environment detection, and on the other (hand) of fuzzy situation perception and maneuver realization of the driver. Tree-structured probabilistic networks (Bayesian Networks) serve to determine the situations and maneuvers with a certain probability based on fuzziness. The probability is interpreted as degree of belief or certainty for the existence of a specific situation or maneuver.

Oliver et al. ([Oli00b], [Oli00a]) developed machine models of driver behavior that incorporate elements of situational awareness for tactical driving. Graphical models, Hidden Markov Models (HMMs) and Coupled Hidden Markov Models (CHMMs), have been trained using the experimental driving data to create models of seven different driver maneuvers: passing, changing lanes right and left, turning right and left, starting and stopping. Hidden Markov Models (HMMs) offer a probabilistic framework for modeling processes that have structure in time. An HMM is essentially a quantization of a system's configuration space into a small number of discrete states, together with probabilities for transitions between states. A single finite discrete variable indexes the current state of the system. Any information about the history of the process needed for future inferences must be reflected in the current value of this state variable (Fig. 2.6). Nodes marked with O represent the observations $o(t)$, nodes marked with S represent the hidden state variable $s(t)$, horizontal arcs represent the transition matrix $P_{s(t)|s(t-1)}$ and parameters associated with the vertical arcs determine the probability of an observation given the current state $P_{s(t)}(o(t))$. Using the car, driver's gaze and road lane data HMMs are built for each of the maneuvers to be recognized. Since HMMs can not represent the systems that have structure both in time and space, Brand et al. ([Bra97]) introduced Coupled Hidden Markov Models (CHMMs) for modeling two interacting processes (humans or cars). In this architecture state chains are coupled via matrices of conditional probabilities modeling causal (temporal) influences between their hidden state variables. Instead of a symmetric CHMM, an asymmetric CHMM architecture is proposed, where the surrounding traffic affects the behavior of the driver, but not vice-versa. The predictive power of those models is of 1 second before the maneuver commences.

In [Joh96] Johnson et al. describe a model of the probability density functions (PDFs) of possible instantaneous movements and trajectories within a scene. The model is automatically generated by tracking objects over long image sequences. The PDFs are represented by the distribution of prototype vectors which are placed by a neural network implementing vector quantization. The temporal nature of trajectories is modeled using a type of neuron with short-term memory capabilities. Recognition of simple and complex events can be achieved by attaching semantics or meaning to areas of the distributions. This is a matter of labeling the relevant network nodes, and retrieving the information when the nodes are activated. Trajectory prediction can be achieved in a similar way by labeling nodes who's prototypes represent complete trajectories with information acquired automatically in a further learning phase. Partial trajectories can then activate the node representing the most similar complete trajectory.

[Bro05] presents a framework for efficiently generating a probability distribution for the motion of multiple objects in a road scene, a technique for analyzing the probability of collision in the future and a method for finding the safest path to avoid the danger. The aim of the reasoning algorithm is to generate a probability distribution for the future motion of all cars in the scene. This is calculated by considering all possible control inputs for all objects. All the control inputs which lead to a collision are eliminated, and a goal function assigned to each object is used to rank all the safe inputs in order of desirability. The output is the best control input, or a probability distribution over all control inputs, of the likelihood that any two objects collide. For an approximation of a total probability of a collision Monte Carlo random sampling was used. Based on this framework Eidehall et al. in [Eid08] defined a treat assessment algorithm to detect general treats, i.e., threats that are not necessarily connected to a particular safety function. It has an ability to detect indirect treats, i.e., threats from objects that may not be on a direct collision course but are forced onto a collision course by other elements in the traffic situation. Since drivers most often try to avoid collisions, this fact was incorporated into the algorithm by creating a mix of two distributions: one where other vehicles try to avoid the ego vehicle and one where they do not see or consider the ego vehicle. These distributions are then weighted based on the visibility relation between the ego vehicle and the objects.

Hülnhagen et al. in [Hü10] suggest an approach to model and recognize various driving situations. Probabilistic finite-state machines are used to model complex driving maneuvers as sequences of basic elements. These basic elements are specified by a set of rules in a fuzzy logic system. They can be reused in multiple models for different driving maneuvers. A Bayes filter approach is employed to recognize a driving maneuver by computing the probability of each basic element in the context of the maneuver model.

[Amd10] gives a description of a Dempster-Shafer approach for situation refinement. Dempster-Shafer evidential theory is a probability-based data-fusion-classification algorithm, which combines evidence from different sources and arrives at a degree of belief that takes into account all the available evidence. Using this approach seven host maneuvers are classified: lane change, overtake, free flow, cut in, merging, following vehicle in path and following vehicle in next lane. A basic probability assignment (BPA) function is defined for each information source, which returns the probability (mass) of the maneuver for a given value of the source. Every time scan, the mass evidence, belief, and plausibility for all the information sources are calculated. Then, they are combined recursively. Finally, a vector of evidence masses and beliefs of all host maneuvers is produced. The maneuver of the greatest mass is identified as the classification result.

Fuerstenberg et al. ([Fue05], [Fue06]) present the concept of the project INTERSAFE which includes the accurate localization of the ego vehicle achieved by LASER and video using individual feature-level maps of the intersection, as well as tracking and classification of other road users and obstacles, providing additional data for the path prediction and risk assessment. In a first step potential conflicts for the host vehicle at the intersection are identified. The detected conflicts are not necessarily dangerous if the paths of the vehicles will follow the conflict-free paths. Therefore, it is important to perform a plan-recognition in order to identify the possible maneuvers for each traffic participant and their degree of probability. The inputs for this process are precise lane information, the position and dynamics information of all vehicles, and possibly, the indicator information of the ego vehicle and the other vehicles transferred by car-to-infrastructure-to-car communication. These probabilistic plans are combined with the conflicts computed in the first step. The combination forms a consolidated scenario interpretation for the current scene that serves as basis for the risk assessment. The risk assessment takes assigned probabilities of the driving behavior of the vehicles as weighting factors and computes a risk level for each possible conflict in time and space domain. A fuzzy rule-base is built to reproduce "human thinking" for risk assessment, so that an adequate strategy can be formulated in an easy way. Important input variables are speed, acceleration and distance to the conflict point, but also the type of the opponent, that can be translated to a severity index for a possible crash. The results of the risk assessment are different risk levels for all possible conflicts in the scene.

2.4.3 Fuzzy Approaches

In [Pel02] Pellkofer und Dickmanns explain that symbolic meanings of objects are calculated and stored by so-called situation aspects. A situation aspect is defined as a linguistic variable and consists of the following parts:

- a name, that the human viewer can associate a symbolic statement which describes an aspect of the situation,
- the linguistic values, that are fuzzy sets represent the intrinsic symbolic information,
- data processing, that buffers and pre-processes required data
- the membership functions, that use the pre-processed data to calculate the fulfillment of values,
- the data logic and the processing logic and
- the period of validity, that specifies how long the meanings are valid.

The symbolic information is used by a set of fuzzy IF-THEN-rules. In the IF-part of each rule, the linguistic values of specific situation aspects are tested. In the THEN-part of each rule a list of suitable capabilities of autonomous vehicle with its parameters and alternatives are specified. The rules that are fulfilled most, determine which capabilities are going to be used.

2.5 Conceptual Graphs

In this work the approach for scene description and interpretation based on conceptual graphs is developed. In this section the theory of conceptual graphs is explained and the advantages and disadvantages of this method are discussed.

2.5.1 Theory of Conceptual Graphs

Sowa ([Sow76]) introduced conceptual graphs (CGs) as new logic-based knowledge representation formalism. In Sowa's words, the purpose of conceptual graphs is "to express meaning in a form that is logically precise, humanly readable, and computationally tractable". In conceptual graph theory there is a distinction between ontological and factual knowledge. The ontological knowledge is represented by support, which is encoded in hierarchies. The factual knowledge is represented by a labeled bipartite graph whose nodes are taken from the support.

In this work a simplified version of support $S = (T_C, T_R, I)$ is considered. The concept types are part of a concept type hierarchy T_C. The latter is provided with a partial-ordering relation $\leq$. A type higher than certain type in the hierarchy is called its super-type and the one lower in the hierarchy is called its sub-type. Relation types are organized into a similar hierarchy T_R, which is separated from the concept type hierarchy. The orders on hierarchy correspond to an "A-Kind-Of" relation between types. The partial-ordering relations $\leq$ can be interpreted as a categorical generalization relation. The semantic of each relation type r is given as a signature $S_r = (r, n, C_1, \dots, C_n)$, where n is the arity of the relation type r and $C_1, \dots, C_n$ are the greatest concept types in the hierarchy H_c which can be linked by the relation type r. I is a set of individual markers.

Formally ([Cro07b]), a simple CG is a triple $CG = (S, G, \lambda)$:

- S is a support;
- $G = (V_C, V_R, E)$ is an bipartite, connected, finite and oriented graph; $V = V_C \cup V_R$ is the node set of G, V_C is a finite nonempty set of conceptual

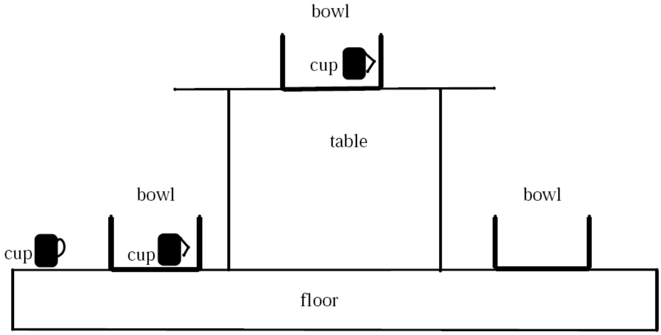


Figure 2.7: An example of a scene.

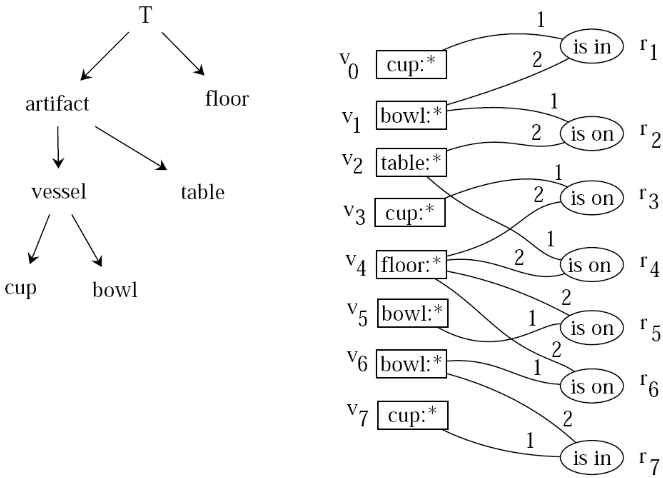


Figure 2.8: Corresponding CG-style scene graph of the scene presented in Fig.2.7.

nodes, V_R is a finite set of relation nodes; E is the set of edges $\{v_r, v_c\}$ where $v_r \in V_R$ and $v_c \in V_C$;

- $\lambda : V \to S$ is a labeling function; if $v \in V_C$ then $\lambda(v) = (type_v, ref_v)$ where $type_v \in T_C$ and $ref_v \in I \cup \{*\}$; if $r \in V_R$ then $\lambda(r) \in T_R$.

Concept nodes represent entities, attributes, states or events, and relation nodes show how the concepts are interconnected. A concept, represented graphically by a box, has a concept type and a referent. A particular referent is "*", which stands for a generic individual in the domain. A conceptual relation, represented graphically by an oval, has only a type. An arc pointing toward an oval marks the first argument of the relation, and an arc pointing away from an oval marks the second argument. If a relation has only one argument, the arrowhead is omitted. If a relation has more than two arguments, the arrowheads are replaced by integers $1, \ldots, n$. If the scene given in Fig.2.7 is considered, the concept type hierarchy T_C is depicted on the left side of Fig.2.8 ([Cro07b]). The factual information provided by Fig.2.7is given by the labeled bipartite graph on the right side.

The theory of conceptual graphs offers a possibility of constraints. Constraints can model conceptual knowledge or express concrete or inferred knowledge about a scene. They arise from logics (e.g. "the vehicle is in the left neighbor lane" implies "the vehicle is left from ego vehicle"), physical laws (e.g. different solid objects may not occupy the same place at the same time) or conventions (e.g. temporal and spatial constraints for overtake maneuver). If they are hard, they must be satisfied and a constraint violation prohibits a solution. If they are soft, they should be satisfied and a constraint violation worsens the quality of a solution.

Sowa [Sow84] showed that the conceptual graph formalism is a subset of first order logic. Every conceptual graph makes an assertion that can be mapped into a logical formula. The most used logical operators are the conjunction $\wedge$ and the existential quantifier $\exists$, but the negation $\neg$ is also possible. However, conceptual graphs allow exploration of new directions in first order logic research by applying knowledge and tools from graph theory. New conceptual graphs may be obtained from existing conceptual graphs by using four elementary operators defined by the formalism:

- Copy: If w is a conceptual graph then the copy u of w is also a conceptual graph.
- Restriction: A graph is restricted when a concept type is replaced by a subtype, or when a generic referent * is changed to an individual referent i.
- Simplification: When two concepts are linked by two identical relations, then one of the relations can be deleted.
- Join: Two graphs with the same identical concept can be joined to form another graph by sharing that concept.

If a graph u is obtained from another graph v by applying one or more of these four operators, then the graph u is a specialization of the graph v and the graph v is a generalization of the graph u. A distinct operator provided with the formalism is the projection operator. This operator projects a graph into another if there is a subgraph in the second which is a copy of the first, up to some concept restrictions. Projection can be used as a matching function between two conceptual graphs.

Conceptual graphs have found many applications. In [Rac01] an intuitive man-machine interface for the communication of image processing tasks, specifically segmentation, was developed. Conceptual graphs were used for the symbolic image description in the hierarchical region-based image representation. Representation of text contents in information retrieval has been a topic of several research groups, e.g. [Gen97], [Nic95], [Tha06]. Article [DK05] shows the advantages of Conceptual Graph formalism for the Semantic Web through several real-world applications in the framework of Corporate Semantic Webs. In [Mis03] conceptual graph representations are used for source code retrieval. Conceptual graphs can also be employed to describe the data sources as a decision support system, e.g. for medical diagnosis ([Cro07a]) or detection of microbial risks in food products ([Tho03]).

2.5.2 Discussion

Probabilistic models, like Bayesian Networks (see [Sch10]) or Markov Models (see [Oli00b]), offer a strong reasoning framework with a possibility of the parameter learning. In many cases probabilistic models also have efficient algorithms, some exact and some approximate, such as Markov Chain Monte Carlo sampling. They lack in the representational power, since objects and relations between them cannot be represented. Therefore, the difference between the knowledge about the domain, like concept or relation definitions, and the knowledge about individuals and relations between them cannot be expressed. Also the network topology needs to be manually modified in the case of application change.

Most of deterministic models are logic-based. All logic-based models lack the expressiveness with regard to uncertain knowledge. In most approaches all possible scene objects are known a priori, but they consider the fact that the visual knowledge is in general incomplete.

Conceptual graphs belong to the logic-based approaches, but they also benefit from the graph theory and graph algorithms. Graphical representation allows the construction of useful user interfaces and knowledge engineering literature. Conceptual graphs have also following advantages:

1. they have graph based reasoning mechanisms which allow versatile querying algorithms, i.e. the model based on them is independent of an application;

2. they are expressive enough to be able to represent the rules associated with extracted data;

3. they are easy to plug in on top of existing ontologies due to the distinction between ontological knowledge (the support) and factual knowledge (bipartite graph);

4. the support and the rule knowledge base can be easily extended and maintained without changes in the reasoning algorithm.

Naturally, there are certain drawbacks.

The decision process is deterministic, i.e. the situation is recognized or not. It is only possible to project the whole query graph and get the positive answer if the situation represented by the query graph is recognized. To be able to get also partial answers if the situation is not recognized, e.g. "The situation is recognized with the probability of 0.8.", the weights based on the importance for the situation can be associated to the concepts and relations. In this way the projected subgraph with the largest sum of weights is considered and this results in the probability of the situation.

Although the uncertainty of the representation and recognition process can not be expressed, there is one dimension that can be taken into account and that is the uncertainty of the extracted object attributes. For each numeric value a range can be given in the scene graph and later compared with the query graph.

The history of behavior is not given, i.e. the decision is based only on one scene graph. The issue can be partially solved, if more scene graphs are saved and the decision is based on them. This allows the user to create queries that spread on several consequent scene graphs. The decision is sometimes made under insufficient information. Nevertheless, when the new information arrives, the previous decision is not revised, but a new situation is recognized.

Semantic ambiguity can be present, since not all logical formulas can be represented. The quantifiers, such as "at least" or "exactly", cannot be expressed. Nevertheless, for the application described in this work those quantifiers are not needed.

The developed method based on conceptual graphs is described in Chapter 5.

3 Measurement System

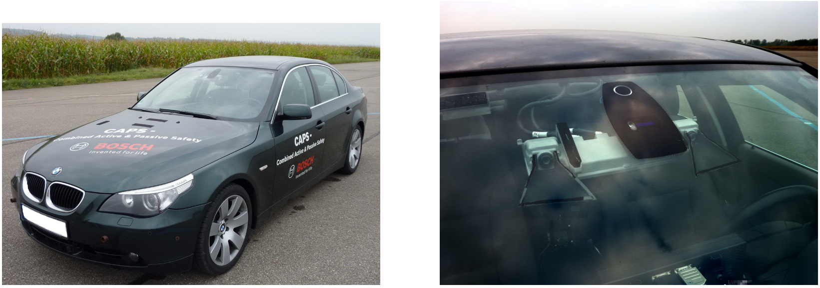
3.1 Realization of Video Sequences

The video sequences and the needed measurements are realized with an experimental vehicle. The used experimental vehicle is a serial produced BMW series 5 car, which is equipped with an additional infrastructure. Additional to the standard CAN bus system there are private CAN networks, installed to enable the communication over gateways between the development Electronic Control Unit (ECU), the computer and the car network. A car computer, as well as all actuators and monitors, are run by a separate voltage supply. Figure 3.1(a) shows the used experimental vehicle. A windshield with the stereo system can be seen in Fig. 3.1(b). Both CMOS cameras are connected through LDVS-Interface to ECU ([Weh10]). ECU communicates with the car computer using proprietary debug interface.

3.2 Measurement Programs

3.2.1 Lane Detection

The simplified form of lane detection algorithm is presented in Fig. 3.2. The grayscale image is processed line by line in certain spacings and the intensity development is analyzed. Based on this the dark/light and light/dark changes can be detected, the same as in the lane markings. The second-order derivative of the intensity is built. It essentially captures the rate of change in the intensity gradient. Thus, in the ideal continuous case, detection of zero-crossings in the second derivative captures local maxima in the gradient ([Ren06]). From each second-order derivative a measuring point is created. In the next step those points are clustered and from those clusters lines are formed. However, not only lane markings are detected in this process. Tar joints on the streets or guard rail can also be among detected lines. Because of this the relevant lines are chosen and grouped based on belonging to a certain lane: ego lane, right neighbor lane, left neighbor lane, etc. The result can be seen in Fig. 3.3. The extraction of lane attributes will be shown in Section 5.3.



(a) (b)

Figure 3.1: (a) An experimental vehicle; (b) Integrated stereo system.

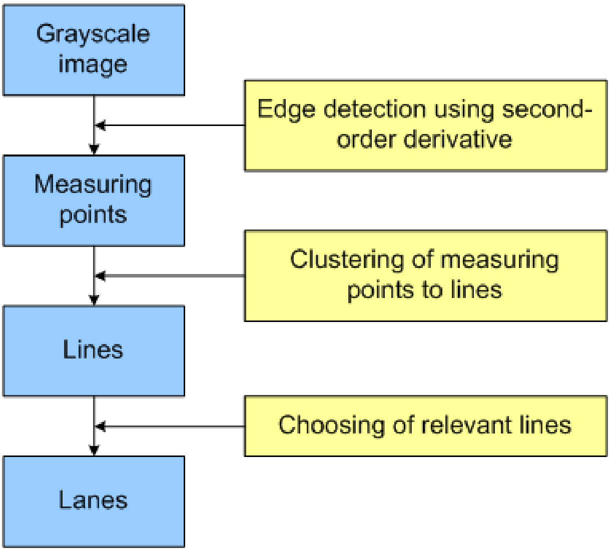


Figure 3.2: Simplified lane detection algorithm

3.2.2 Object Detection

The object detection is based on 6D-Stereo approach described in the PhD thesis of Thomas Wehking [Weh10], which is based on the 6D-Stereo approach by Franke et al. [Fra05]. This algorithm is run on the ECU.

The data used for this method is extracted from optical flow, the fundamentals

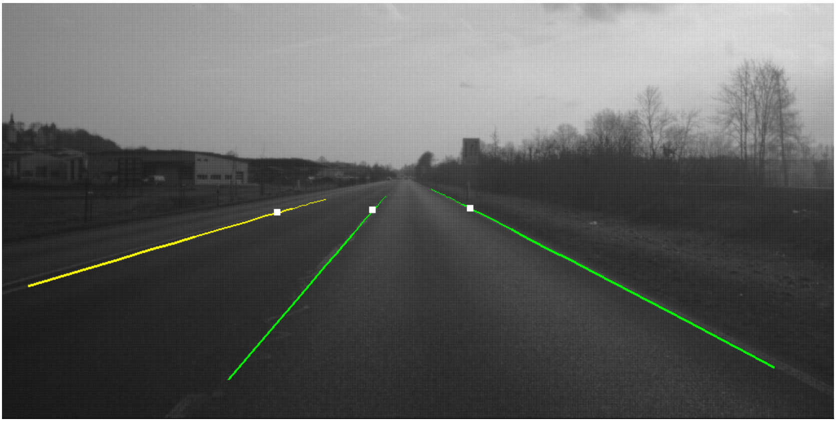


Figure 3.3: The result of lane detection

Figure 3.4: Example of the optical flow [Teu09]

of which are given in Section 2.1, and disparity is explained in Section 2.2. The examples of optical flow and disparity are given in Fig. 3.4 and Fig. 3.5, respectively. Using Haar-Wavelets the signatures are created for left, right and consecutive images to map the corresponding points for calculating the optical flow or the disparities.

The 6D-Stereo segmentation process is built by two segmentation processes. In the first stereo segmentation process the regions with a special accumulation per distance level are segmented from u-disparity and v-disparity histograms. The second segmentation process is based on Connected-Component-Labeling where the three-dimensional scatter-plot with integrated motion information is segmented by processing the data row by row and looking for linked regions. Using those two segmentation methods at the same time leads to the redundant detection of objects. In this way it is possible to make the difference between the movable and static

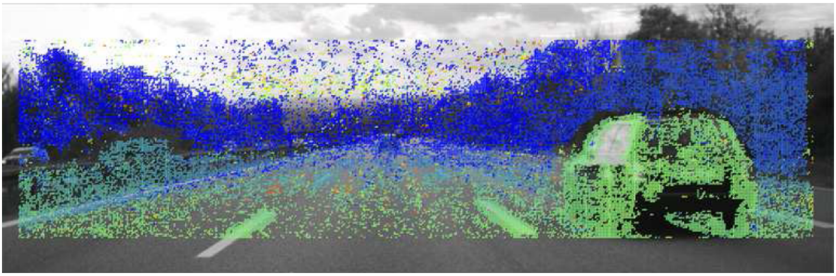


Figure 3.5: Example of the disparity [Teu09]

object at the same distance. Those detected objects deliver enough information for extracting of attributes: the direction and position relative to ego vehicle, the size of object, as well as the speed.

3.2.3 Object Classification

An approach of shape-based classification of pedestrians and vehicles based on fuzzy logic is used. The fuzzy logic is a part of the multi-valued logic and the degree of truth can take values from the interval $[0, 1]$. The following description of the different steps in application of fuzzy logic is based on [Mik99]. The input is a set of numerical data $\varphi_i \in \chi_i$ $(i = 1, \ldots, N)$. The numerical data φ_i are mapped by membership functions $\mu_{ij} : \chi_i \longrightarrow [0, 1]$ $(i = 1, \ldots, N;\ j = 1, \ldots, J_i)$ to a membership value[1] describing how much φ_i corresponds to the linguistic term[2] A_j of the linguistic variable V_i. So every term of a linguistic variable has its own fuzzy function.

The main concept of fuzzification in this approach is using statistical data of human and vehicle measures. Because of this, data such as height, width and length of people and vehicles are collected. Vehicles were sorted by classes of cars (e.g. compact class, van), but the smooth transition from one class to another referring to these data led to their summarization in one fuzzy set called *CAR-like* with the exception of trucks and buses. They are different from *CAR-like* and so they are summarized in a fuzzy set *TRUCK-like*. The fuzzy set *PEDESTRIAN-like* is based on human measures obtained from [Til02].

The membership functions are piecewise defined and consist of constant parts connected by exponential defined pieces. The beginning and the end of the intervals,

[1] synonyms: degree of membership, grade of membership

[2] synonym: linguistic value

where the functions are defined as a constant greater then 0, are determined by the extrema of the statistics mentioned above, while the length of the exponential defined intervals is dependent on the accuracy of the object detection. In that way, each object having measures similar to the statistics should be assigned to a membership value greater then 0 by the membership function.

The generalized form of such a membership function μ is:

$$\mu(\varphi) = \begin{cases} c_1 & & x \leq & a_1 \\ g(\varphi, 1) & a_1 & < x < & a_1 + \epsilon_1 \\ c_2 & a_1 + \epsilon_1 & \leq x \leq & a_2 \\ \vdots & & & \\ c_i & a_{i-1} + \epsilon_{i-1} & \leq x \leq & a_i \\ g(\varphi, i) & a_i & < x < & a_i + \epsilon_i \\ c_{i+1} & a_i + \epsilon_i & \leq x \leq & a_{i+1} \\ \vdots & & & \\ g(\varphi, n-1) & a_{n-1} & < x < & a_{n-1} + \epsilon_{n-1} \\ c_n & a_n + \epsilon_n & \leq x & \end{cases}$$

where c_i is the constant value returned by μ at the interval $[a_i, a_i + \epsilon_i]$ and ϵ_i has to accomplish the condition $\epsilon_i \leq a_{i+1} - a_i$. The function g is used to connect the constant pieces of μ and has the following form:

$$g(\varphi, i) = \min(c_i, c_{i+1}) + |c_{i+1} - c_i| \cdot \exp\left(\frac{\varphi - a_i - \epsilon_i}{\varphi - a_i}\right)^{sgn(c_{i+1} - c_i)}$$

Figure 3.6 shows membership functions for the linguistic variables WIDTH, HEIGHT, SPEED and ASPECT_RATIO used for fuzzification. The membership functions for the linguistic variables WIDTH, HEIGHT and ASPECT_RATIO are based on statistical data of human and vehicle measures as previously described. The fuzzification of the velocity differs from the other fuzzification functions. In principle, at low velocity the detected object moves like a slow vehicle or a walking person. The more velocity increases, the less a pedestrian is expected and expectation of vehicles remains at the same level. The velocity fuzzification functions are chosen using the fact, that a vehicle in most cases moves faster than “walking speed” and most pedestrians stand or walk instead of jog or run. Furthermore the membership value at velocity 0 is set to 0.3 because in this case the significance of velocity is decreased, since the number of plausible classes, which an object can be classified to, increases. For example, a stationary object with the form of a pedestrian could also be a traffic sign or a traffic light.

In the inference part linguistic rules are evaluated to create output fuzzy sets from the fuzzified input sets. The inference can be divided into three parts: aggregation,

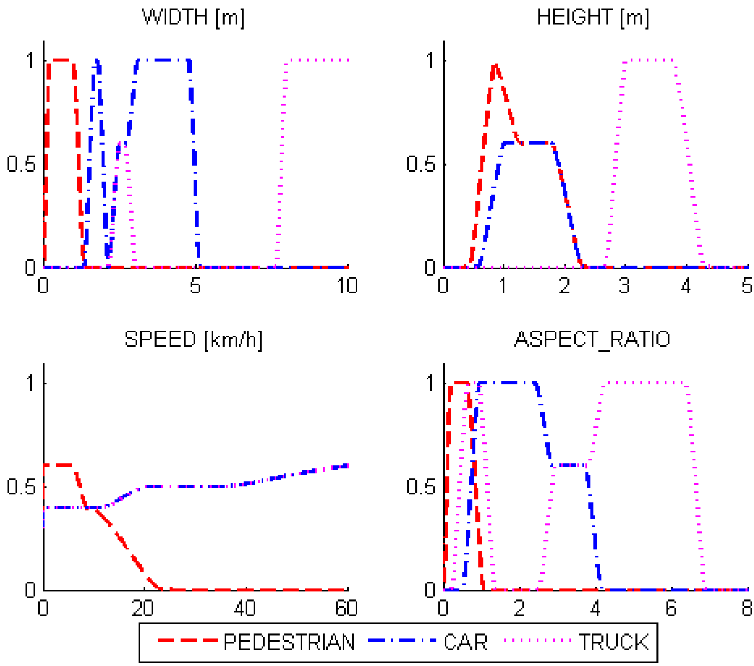


Figure 3.6: Membership functions

activation and accumulation. A linguistic rule starts with one or more premises (IF-part), that are combined by fuzzy operators (some examples are given in Table 3.1), and ends with a conclusion (THEN-part). At least the premises have to consist of linguistic terms, whereas the conclusion can have one of the following forms: it can consist of one or more linguistic conclusion-parts (*Mamdani*-type) or it is a function of the input membership values (*Takagi-Sugeno*-type). Potential operators are conjunctive (T-norms), disjunctive (T-conorms or S-norms) or average operators. There are also hybrid operators, which are a combination of the above. Some well known operators are presented in Table 3.1.

The step of activation is used to add weight factors ranged from 0 to 1 to the results of the linguistic rules. In that way it is possible to qualify each linguistic rule by a degree of confidence. The accumulation unites the results of activation, i.e. combines the degrees of membership received by the conclusions of the linguistic rules, and creates an output fuzzy set.

In the end, the output fuzzy set has to be transformed to a numerical value. The

	$\wedge$	$\vee$
Min/Max	$\min(\mu(\varphi_1), \mu(\varphi_2))$	$\max(\mu(\varphi_1), \mu(\varphi_2))$
Lukasievicz	$\max(0, \mu(\varphi_1) + \mu(\varphi_2) - 1)$	$\min(\mu(\varphi_1) + \mu(\varphi_2), 1)$
Product	$\mu(\varphi_1) \cdot \mu(\varphi_2)$	$\mu(\varphi_1) + \mu(\varphi_2) - (\mu(\varphi_1) \cdot \mu(\varphi_2))$

Table 3.1: Some conventional operators

simplest way to do this is the maximum-defuzzification, returning the argument at which the membership function of the output fuzzy set has its maximum. If there is more than one argument satisfying the maximum-condition, there are different ways to get a definite solution, e.g.:

- Left maximum (LM) returns the smallest argument of all maxima
- Mean of maximum (MOM) returns the mean of the arguments of all maxima
- Right maximum (RM) returns the largest argument of all maxima

In this approach some rules that combine the different linguistic variables are evaluated. Equation 3.1 shows the utilized rules.

$$\begin{array}{l} WIDTH \wedge HEIGHT \\ (WIDTH \vee HEIGHT) \wedge ASPECT\_RATIO \\ WIDTH \wedge HEIGHT \wedge SPEED \end{array} \tag{3.1}$$

The realization of $\wedge$ and $\vee$ is done with the Min/Max-operators presented in Table 3.1. Because the rules are similar to each other, the same weight is assigned to each rule. The step of activation is ignored, but to correspond to the theory, it can also be interpreted as all weights are set to 1. At accumulation the results of the activation (which are in fact the results of the rules) are now combined to get one membership value for every class. This composition of activation results is realized with a geometric average. The geometric average returns 0 in the moment that one rule returns 0. This is a desired behavior because of the following reasons:

- The membership functions are constructed in such a way, that the function only returns 0, if the object attribute really does not apply to the linguistic term. Thereby even the accuracy of the object detector is considered.
- The rules combine the given attributes and if one rule returns that the object cannot be part of a class, the decision is considered correct.

Finally, in contrast to the theory of fuzzy control, the membership value is not translated to a numeric one, but to a class label. The Max-operator is used to defuzzificate the results of accumulation. An object should be assigned to the class it is the most similar to and the Max-operator supplies the class that is the most similar to the object in accordance to the measures.

4 Collision Detection

One of the goals of the thesis was to develop a collision detection method that does not depend on the situation in which the ego vehicle is involved. An object should be visible and a relative motion between ego vehicle and the object should be translation in order to be able to use the developed method for collision detection. It can work with a monocular system, as well as a stereo system, taking one of the cameras as a reference. In the work stereo system was applied because it delivers more reliable position estimation. This method ([Bor10b]) is based on several parameters: focus of expansion, Time-to-Collision and distance to the object. In the following sections the key parameters are introduced.

4.1 Focus of Expansion

The object between time $t = 0$ and $t = \Delta t$ shows only translation $\Delta\mathbf{T} = (\Delta X, \Delta Y, \Delta Z)^T$ relative to the camera, where $\Delta\mathbf{T}$ is given in DIN 70000, [DIN94], world coordinate system at $t = 0$. Taken the object point $P_t = (X_t, Y_t, Z_t)^T$ given in world coordinate system in time $t \in \{0, \Delta t\}$, the relation $P_{\Delta t} = P_0 + \Delta\mathbf{T}$ can be deducted. Calculation is possible when the object point is visible in the image for $t = 0$ and $t = \Delta t$. The optical flow for this object point is seen as the displacement $\mathbf{v}_p = p_{\Delta t} - p_0$ of its projection between $t = 0$ and $t = \Delta t$:

$$\mathbf{v}_p = p_{\Delta t} - p_0 = \begin{pmatrix} x^p_{\Delta t} \\ y^p_{\Delta t} \end{pmatrix} - \begin{pmatrix} x^p_0 \\ y^p_0 \end{pmatrix}. \tag{4.1}$$

Using basic form of linear equation $y = mx + b$ the parameters of the line containing $\mathbf{v}_p$ are calculated. Analog method is used for the object point $Q \neq P$. FOE is defined as the intersection point of above mentioned lines $\mathbf{p}_{\text{FOE}} = (x_{\text{FOE}}, y_{\text{FOE}})^T$ and given as ([Jat09]):

$$p_{\text{FOE}} = \begin{pmatrix} a_x \frac{\Delta X(\sin\psi\cos\varphi + \sin\theta\cos\psi\sin\varphi) + \Delta Y(\cos\psi\cos\varphi - \sin\theta\sin\psi\sin\varphi) - \Delta Z\cos\theta\sin\varphi}{\Delta X\cos\theta\cos\psi - \Delta Y\cos\theta\sin\psi + \Delta Z\sin\theta} + x^0_b \\ a_y \frac{\Delta X(\sin\psi\sin\varphi - \sin\theta\cos\psi\cos\varphi) + \Delta Y(\cos\psi\sin\varphi + \sin\theta\sin\psi\cos\varphi) + \Delta Z\cos\theta\cos\varphi}{\Delta X\cos\theta\cos\psi - \Delta Y\cos\theta\sin\psi + \Delta Z\sin\theta} + y^0_b \end{pmatrix} \tag{4.2}$$

where $\theta,\ \psi,\ \varphi \in [\frac{-\pi}{4}, \frac{\pi}{4}]$ are the pitch, yaw and roll angle, respectively, describing the rotation of camera in world coordinate system, $a_x = -k_x f$ and $a_y = -k_y f$,

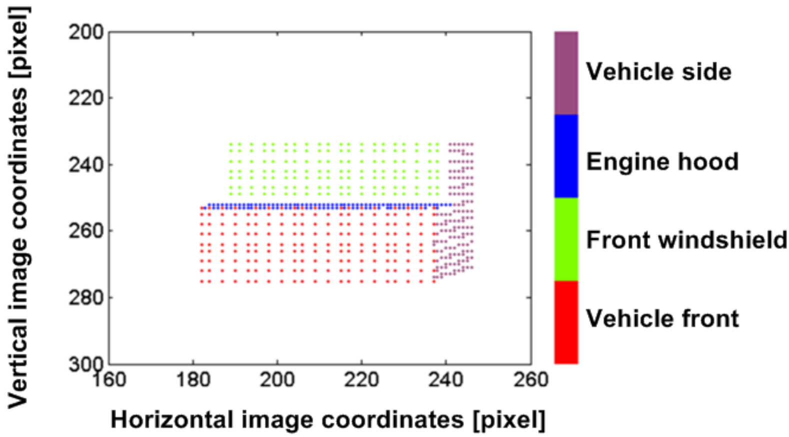


Figure 4.1: Color-coded image points of a vehicle model $35.2\ m$ in front of the camera. Most points are placed on the vehicle surfaces parallel to the camera plane.

k_x and k_y - horizontal and vertical scaling factor, f - camera constant, and (x_b^0, y_b^0) principal point of the camera. If $\theta = \psi = \varphi = 0$, Eq. 4.2 can be simplified:

$$p_{\text{FOE}} = \begin{pmatrix} a_x \frac{\Delta Y}{\Delta X} + x_b^0 \\ a_y \frac{\Delta Z}{\Delta X} + y_b^0 \end{pmatrix} \tag{4.3}$$

Due to discretization error and possible measurement error of optical flow, the intersection point of optical flow lines is ambiguous. Numeric methods are applied to achieve the best possible estimation of the intersection point. One of them is a binning method. In this method the intersection points and their frequency for as many as possible optical flow lines are calculated. The intersection point with the highest incidence frequency is assumed to be FOE.

The object that is a passenger car model which has the length $4.5\ m$, width $2.0\ m$ and height $1.5\ m$ and the engine hood $2.0\ m$ long and $0.8\ m$ high is observed (Fig. 4.1). It is driving toward the camera with the relative velocity $30m/s$ and is 3 m moved along Y-axis. Fig. 4.3 shows the results of FOE calculation for this vehicle based on its optical flow that is depicted in Fig. 4.2.

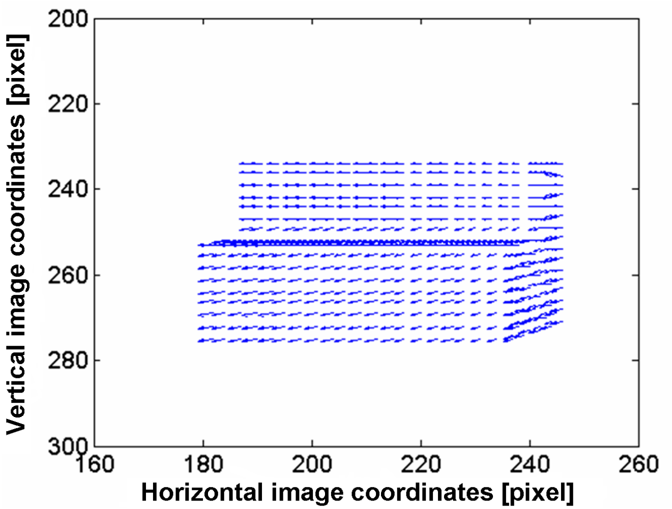


Figure 4.2: Optical flow of a vehicle model $35.2\ m$ in front of the camera.

4.2 Time to Collision

The object as the two-dimensional flat surface parallel to the image plane is observed. Using the projection equation $a = fA/d$, where A is vertical or horizontal size of the object, a its projection on image plane and d distance between the plane through the optical center and the object, and its derivation in time, it is possible to calculate at current time $t = t_0$ Time to Collision (TTC) when the object collides with the camera:

$$TTC = -\frac{d(t_0)}{\frac{\partial d(t)}{\partial t}} = \frac{a(t_0)}{\frac{\partial a(t_0)}{\partial t}}. \tag{4.4}$$

This is determined under the assumption that the relative velocity $\mathbf{v}_{rel}$ between the camera and the object is constant and negative when the object and camera move toward each other. The first derivative of a in time can be approximated with forward difference quotient:

$$\frac{\partial a}{\partial t}_{|t=t_0} \approx \frac{a_{t_1} - a_{t_0}}{\Delta t} \qquad \text{with } \Delta t = t_1 - t_0, \tag{4.5}$$

which in combination with Eq. 4.5 yields:

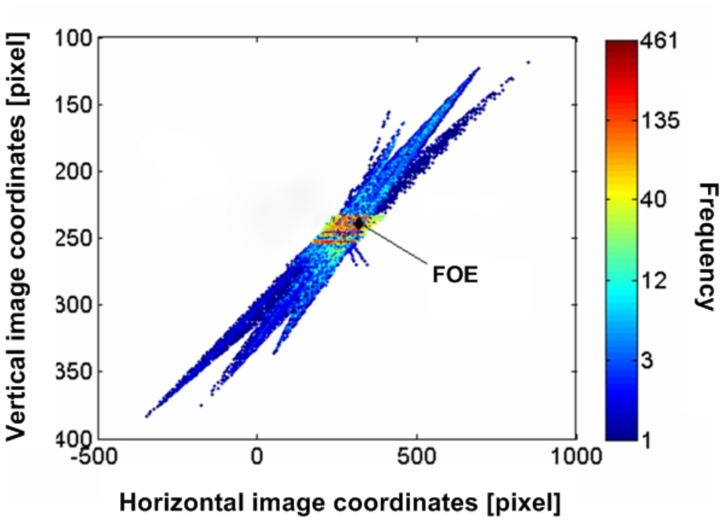


Figure 4.3: The frequency distribution of the optical flow intersections plotted by a logarithmic color scale.

$$TTC = \frac{a_{t_0} \Delta t}{a_{t_1} - a_{t_0}}. \tag{4.6}$$

The size a_t can be seen as the difference between its border points at certain time: $a_t = a_t^1 - a_t^0$ with $a_t^1 \geq a_t^0$ and for two consecutive frames the corresponding component of optical flow $\Delta a^i,\ i \in \{0, 1\}$, the difference between the same border points at different times: $a_{t_1}^i = a_{t_0}^i + \Delta a^i$. Based on this TTC can be calculated by means of optical flow at a certain time:

$$TTC = \frac{\Delta t}{\frac{\Delta a^1 - \Delta a^0}{a_{t_0}^1 - a_{t_0}^0}}. \tag{4.7}$$

Alternative to this the motion field of an object can be considered. Using projection equation it can be computed as follows:

$$\begin{aligned} \mathbf{v}_p &= \begin{pmatrix} \frac{\partial \frac{fY_t}{X_t}}{\partial t} \\ \frac{\partial \frac{fZ_t}{X_t}}{\partial t} \end{pmatrix} \\ &= \frac{f}{X_t^2} \begin{pmatrix} \frac{\partial Y_t}{\partial t} X_t - Y_t \frac{\partial X_t}{\partial t} \\ \frac{\partial Z_t}{\partial t} X_t - Z_t \frac{\partial X_t}{\partial t} \end{pmatrix} \end{aligned}$$

$$= \frac{1}{X_t} \begin{pmatrix} \frac{f \partial Y_t}{\partial t} - x_t \frac{\partial X_t}{\partial t} \\ \frac{f \partial Z_t}{\partial t} - y_t \frac{\partial X_t}{\partial t} \end{pmatrix}, \tag{4.8}$$

where $p_t = (x_t, y_t)^T$ is the projection of the object point P_t. Since the object is rigid and the object surface is parallel to the image plane the camera constant f, coordinate X_t and the derivatives $\frac{\partial X_t}{\partial t}$, $\frac{\partial Y_t}{\partial t}$ and $\frac{\partial Z_t}{\partial t}$ are the same for all object points at certain time t. In this case the linear correlation between motion field and image coordinates is given. Due to the fact that the optical flow is an approximation of the motion field, the denominator in Eq. 4.7 can be understood as a slope of the linear function in Eq. 4.8 and estimated using linear regression.

In Fig. 4.1 it is clearly visible that in the case of oncoming traffic on the flat street the amount of vehicle side optical flow vectors is significantly smaller than the amount of vehicle front optical flow vectors. For that reason its influence on TTC calculation using slope is small.

4.3 Collision Probability

Since neither the distance d nor the relative velocity $\mathbf{v}_{rel}$ give information about the lateral position between the object and ego vehicle, TTC alone cannot be the index of collision. Two cases when ego vehicle holding the camera and another object move toward each other in parallel tracks can be compared. In the first case the object is exactly in front of the camera and in the second the object moves laterally. TTC is the same in both cases, but in the second the object and ego vehicle are not on the collision course.

The orthogonal projection of 3D scene on the plane where $Z = c$ in WCS with a constant $c \in \mathbb{R}$ is shown in Fig. 4.4. Most traffic objects can be approximated with a rectangle on this plane. The vertices $\mathbf{E}_i = (E_x^i, E_y^i)^T$ of ego vehicle and $\mathbf{O}_i = (O_x^i, O_y^i)^T$ of another object, $i \in \{1, \ldots, 4\}$, in coordinate system whose Y-axis is parallel to the rear axle and center lies in the optical center of the camera, are given. If the position vectors of the points are identified with the point coordinates, the ego vehicle and another object at time t can be described as point sets using rectangle parametrization:

$$\mathcal{E} = \{\mathbf{X} = \mathbf{E}_4 + \lambda(\mathbf{E}_1 - \mathbf{E}_4) + \tau(\mathbf{E}_3 - \mathbf{E}_4)\} \tag{4.9}$$

$$\mathcal{O} = \{\mathbf{X} = \mathbf{O}_1 + \chi(\mathbf{O}_2 - \mathbf{O}_1) + \upsilon(\mathbf{O}_4 - \mathbf{O}_1)\} \tag{4.10}$$

where $\lambda, \tau \in [0, 1]$ and $\chi, \upsilon \in [0, 1]$. Thereby are the set $\mathcal{E}$ time- independent and the set $\mathcal{O}$ time-dependent because of the constant translation of the object points

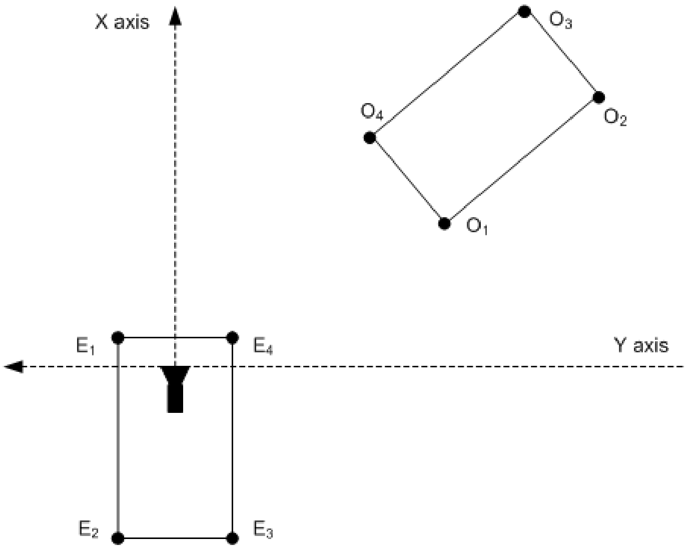


Figure 4.4: Orthogonal projection of a traffic scene with the ego vehicle and another object to a plane parallel to the flat road. The form of the vehicles is approximated by rectangles.

with the relative velocity $\mathbf{v}_{rel}$. To fulfill the condition that the relative movement is pure translation, the optical flow is corrected in respect of rotation that is caused by ego vehicle. The time at which the collision begins can be defined as:

$$TTC = \min\{t > 0 \mid \partial\mathcal{E} \cap \partial\mathcal{O}(t) \neq \emptyset\} \tag{4.11}$$

The velocity vector can be obtained under constraints $(\lambda \in [0,1] \ \wedge \ \tau \in \{0,1\}) \vee (\lambda \in \{0,1\} \ \wedge \ \tau \in [0,1]))$ and $((\chi \in [0,1] \ \wedge \ \upsilon \in \{0,1\}) \vee (\chi \in \{0,1\} \ \wedge \ \upsilon \in [0,1])$:

$$\mathbf{v}_{rel} = \frac{1}{TTC}(\mathbf{E}_4 + \lambda(\mathbf{E}_1 - \mathbf{E}_4) + \tau(\mathbf{E}_3 - \mathbf{E}_4) - \mathbf{O}_1 - \chi(\mathbf{O}_2 - \mathbf{O}_1) - \upsilon(\mathbf{O}_4 - \mathbf{O}_1)) \tag{4.12}$$

The relative angle α of the object movement toward ego vehicle in respect of X-axis, limited on the interval $]-\frac{\pi}{2}, \frac{\pi}{2}[$, can be determined with:

$$\alpha = \arctan \frac{v_x^{rel}}{v_y^{rel}} \tag{4.13}$$

Since the scaling of the relative velocity vector is of no importance for the calculation of direction, v_y^{rel} can be set to 1. Due to the strict monotony of arctan, relative

Figure 4.5: Testing of a side collision using a dummy vehicle when a collision occurs

angle interval $[\alpha_{min}, \alpha_{max}]$ when it comes to a collision, can be defined with the minimal and maximal possible value of v_x^{rel} taking into consideration Eq.4.12. This interval depends on the coordinates of vertices of the ego vehicle and the object. Based on this the minimal and maximal collision angle are determined for velocity vector:

$$\begin{aligned} \mathbf{v}_{rel}^{min} &= \mathbf{O}_4 - \mathbf{E}_3 \\ \mathbf{v}_{rel}^{max} &= \mathbf{O}_2 - \mathbf{E}_1. \end{aligned} \tag{4.14}$$

Angles $[\alpha_{min}, \alpha_{max}]$ are calculated using the distance estimation and the image coordinates of the object box edges and later projected as a danger zone in the image:

$$x_{min} = f \tan \alpha_{min} \qquad \text{and} \qquad x_{max} = f \tan \alpha_{max} \tag{4.15}$$

In Fig.4.5 detected object (white box), object danger zone (big red box) and object FOE (small red box) are visualized. Since the object FOE lies in the danger zone, the collision will occur if the vehicle continues to move in the same manner. In Fig.4.6 the collision will not occur because the object FOE lies outside of the danger zone and is not visible in the image.

Similar to binning method, the intersection points of optical flow lines are counted, but in contrast to it the percentage of intersection points $\mu \in [0, 1]$, which lie in the

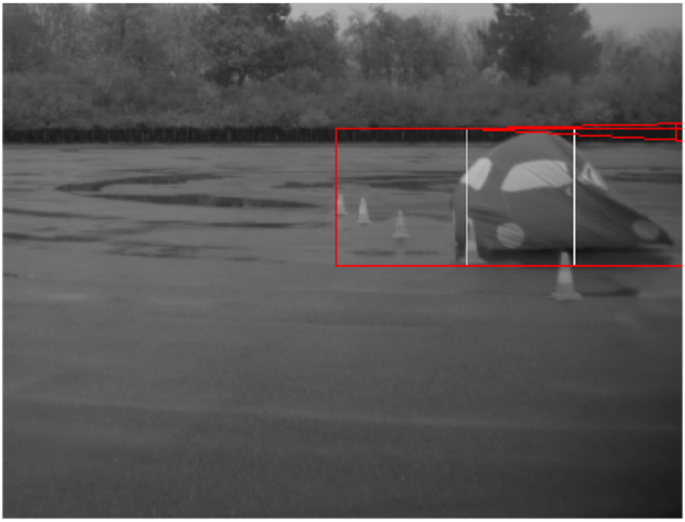


Figure 4.6: Testing of a side collision using a dummy vehicle when a collision does not occur

danger zone, in the total number of intersection points is determined, not the FOE itself. Finally this is used for determination of collision probability κ:

$$\kappa = \delta_{long} \cdot \delta_{lat} \cdot \mu. \tag{4.16}$$

Parameters δ_{long} and δ_{lat} describe quality criteria for the distance estimation and the object's lateral position calculation, respectively. Since the distance estimation has influence on the calculation of the object's lateral position, both parameters are chosen to be the same. To consider the error proneness of δ_{long} and δ_{lat}, but not to restrict the possible range of κ too much, for the first approach the maximal values for the parameters are assumed: $\delta_{long} = \delta_{lat} = 0.9$. This yields the maximal value for the collision probability 0.81. This is achieved when all intersection points lie in the danger zone. More detailed investigation of the quality criteria is needed in order to be able to calculate collision probability more precisely.

Tracking the changes in TTC can help in evaluation of distance estimation, e.g. too high detected object box's bottom edge can be detected. If the distance in consecutive frames progresses opposite to corresponding TTC, the distance is linear extrapolated based on estimated values in N previous frames. Negative TTC means that the object and ego vehicle move from each other, so FOE in the danger zone represents the necessary, but not the sufficient condition for a collision. In this case the collision probability is set to be 0.1.

Figure 4.7: Example frame from the test sequence. Red filled rectangle represents the object that has high probability of collision with ego vehicle if both vehicles continue to behave in the same manner.

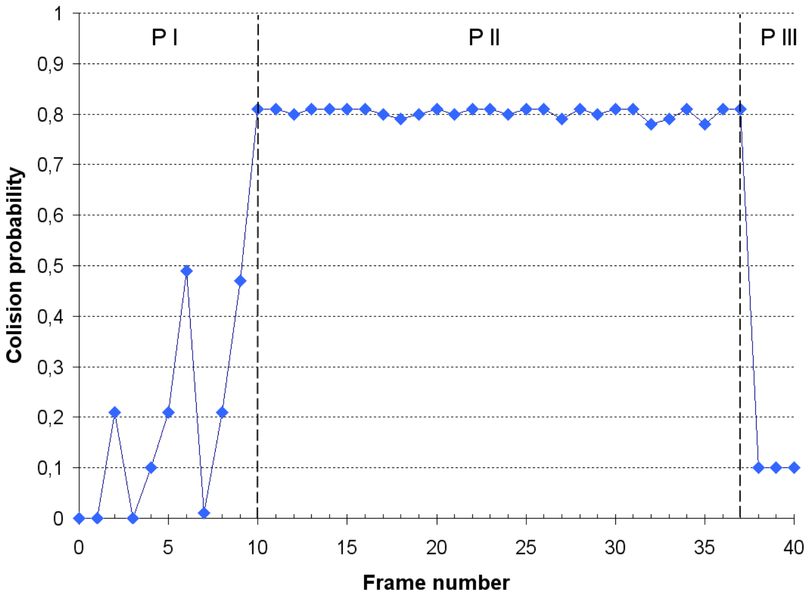


Figure 4.8: Collision probability development for the presented test sequence in Fig. 4.7.

The following example represents the application of the described approach. The situation is observed where the vehicle is approaching from the right side orthogonally to the street on which ego vehicle is driving (Fig. 4.7). Segmentation based on optical flow and disparities provides the object box. At the end of the sequence the detected object box becomes very unstable. This behavior can be explained with the fact that the vehicle in this period of time is very close to ego vehicle and the vehicle brakes and disappears from the image.

For this sequence a collision probability until shortly after the vehicle starts to brake is calculated and depicted in the Fig.4.8. The threshold for high collision probability is empirically set to be 0.6, based on the observations made during the test drives. In the first phase (P I) the collision probability scatters between 0 and 0.5. This is caused by the alternating speed of the vehicle at the beginning of the test. The values in the second phase (P II), when the collision probability is high, run very smoothly, because they depend only on the percentage of intersection points in the danger zone. The third phase begins and the collision probability drops. This is caused by braking of the vehicle to avoid real collision.

5 Vehicle Environment Model Description and Interpretation

A model based on conceptual graphs has been created to represent an environment of the ego vehicle. It provides a possibility to have an overview of a scene, deduct relationships between its elements and recognize potentially dangerous situations. The discussion about the usability of conceptual graphs is performed in 2.5.2.

5.1 System Architecture

The system architecture is depicted in Fig. 5.1. First, objects and lanes are detected using measurement programs as described in Sections 3.2.1 and 3.2.2. Objects are assumed to be moving straight with a constant speed. Position, size, distance and time to collision (TTC) are obtained as attributes of each object and direction and speed are determined for movable objects additionally. Since a lane marking is represented by a clothoid, its offset, angle, curvature and curvature change rate (CCR) are extracted.

After shape-based object classification explained in Section 3.2.3 the graph that describes the scene is generated and saved in CG Repository. The scene graph is created in CG Generator in the form of a conceptual graph according to the Concept Type Hierarchy, Relation Type Hierarchy and Rules and Constraints using the previously obtained information about objects and lanes. Rules and Constraints also include definitions of complex concept and relation types. The graphs from CG Repository are then matched in CG Matcher using projection with the Query CG with respect to the Concept Type Hierarchy, Relation Type Hierarchy and Rules and Constraints and the Result CG is calculated ([Bor10a]).

5.2 Input Data

There are some characteristics of the input data that need to be examined in order to be able to understand the scene, e.g. if the input data is partial or complete or if the number of scene objects is a priori known or not. In this case the input data

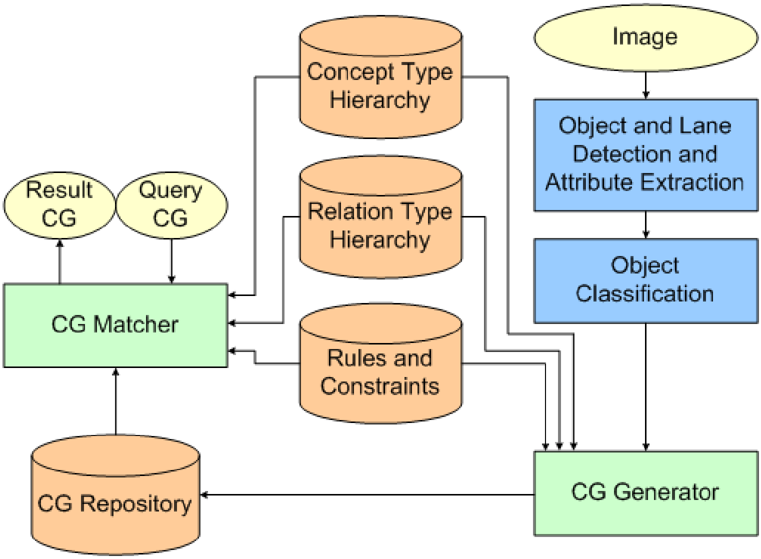


Figure 5.1: System architecture

is partial. The detected set of objects can be incomplete due to a limited field of view, occlusions and imperfect sensors. Only in closed domain all relevant scene objects are known a priori. In most cases every new scene brings with itself new actors, so the number of objects is not a priori known.

Knowledge representation systems that adopt a Closed World Assumption (CWA), like database systems, treat the absence of information as a negative information ([Hum09]). Therefore, the truth values "false" and "unknown" cannot be distinguished. Since the input data is partial, an Open World Assumption (OWA) is required, where the information is considered partial by default and no truth value is deduced for absent information. Since there is a need for classification of certain constructs, the model is required to be at least locally complete. Although there are many possibilities to describe the ego vehicle environment, only certain concepts and relations are defined as they are relevant for the scene, e.g. the road can have many lanes, but only three lanes are specially addressed (described in Section 5.3). Also the unknown constructs are taken into account and properly classified, but no new concept or relation is defined.

5.3 Model Support

5.3.1 Concepts

To represent the vehicle environment a set of concept types is defined and ordered into the hierarchy shown in Fig. 5.2. All concept types can be divided into five categories defined as first level of super-types:

- “MeasTime” defining the time stamp of image frame considered,
- “Value” characterizing the allowed value types used,
- “Visual” showing the possible classes of objects and their placement on the road plane,
- “Component” defining the vector components,
- “Attribute” listing the visual’s attributes that were mentioned at the beginning of the section.

The branch that starts with the concept type “Visual” is closely observed. The concept type “Visual” has two sub-types: “Object” and “Plane”. The concept type “Object” is defined as an object determined by object detection and the concept type “Plane” as a driving surface determined by lane detection. The definitions of the sub-types of concept types “Object” and “Plane” are summarized in Table 5.1.

5.3.2 Relations

The possible conceptual relations are ordered into the hierarchy (Fig. 5.3). Each relation is binary which means that it has only two arguments. If the signature of a relation type is not given in the figure, then it is the same as the signature of its super-type. Besides the relations needed for the functionality of the conceptual graphs, e.g. “scene”,“next”, “chrc” and “part”, there are relations that show the position of objects with regard to the position of the ego vehicle, the location of the object on road plane and its sub-types, the behavior of objects with regard to the ego vehicle and the order between values.

DIN 70000 world coordinate system is used for the representation of object’s position. It is fixed to ego vehicle and its origin is the middle of ego vehicle front axis (Fig.5.4). The position of ego vehicle is defined as $(0, 0, 0)$. The position of object $P = (X, Y, Z)^T$ is defined as the position of object’s middle point. If the depth

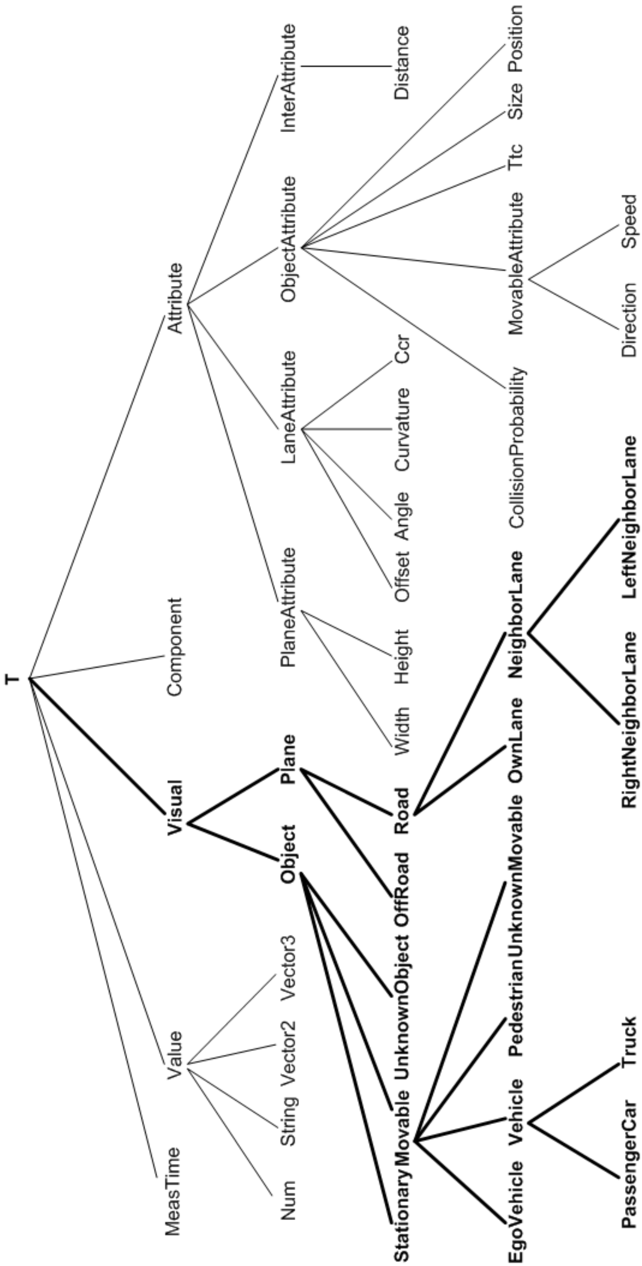


Figure 5.2: Concept type hierarchy

Concept	Definition
Stationary	“Object” that has the absolute value of speed less than $0.01m/s$
Movable	“Object” that has the absolute value of speed greater or equal than $0.01m/s$
UnknownObject	“Object” whose speed cannot be calculated
EgoVehicle	“Movable” introduced as ego vehicle by user
Vehicle	“Movable” that can be classified as “CAR-like” or “TRUCK-like” (Section 3.2.3)
Pedestrian	“Movable” that is classified as “PEDESTRIAN-like” (Section 3.2.3)
UnknownMovable	“Movable” that can not be classified
PassengerCar	“Movable” that is classified as “CAR-like”
Truck	“Movable” that is classified as “TRUCK-like”
OffRoad	“Plane” outside of two outermost detected lane markings left and right from ego vehicle
Road	“Plane” inside of two outermost detected lane markings left and right from ego vehicle
OwnLane	“Road” between the first markings left and right from ego vehicle
NeighborLane	“Road” that can be “RightNeighborLane” or “LeftNeighborLane”
RightNeighborLane	“Road” between the first and second lane marking on the right
LeftNeighborLane	“Road” between the first and second lane marking on the left

Table 5.1: List of concept types representing detected image areas and their definitions

of the object can not be calculated because only one side of the object is visible, x-coordinate of the object’s size is set to be 0.

If the y-coordinate of the object is more than the sum of the half width of object and the half width of ego vehicle greater than the y-coordinate of ego vehicle, than the relation “left” is correct. If the y-coordinate of ego vehicle is more than the sum of the half width of object and the half width of ego vehicle greater than the y-coordinate of the object, than the relation “right” is correct. The relation "same" is correct, if the difference is somewhere in between. If it is not possible to calculate the correct position, the relation "pos" is true.

There are also different relations concerning types of behavior between object and ego vehicle marked bold in Fig. 5.3. The conditions for their existence are summarized in Table 5.2. A part of the decision about the situation is already at this point determined.

Relation	**Condition**
motion	object's speed greater than 0
intersecting	"motion" $\wedge$ (object coming from left or right)
intersectingRight	"intersecting" $\wedge$ (object coming from right)
intersectingLeft	"intersecting" $\wedge$ (object coming from left)
parallelMotion	"motion"$\wedge$(direction the same or opposite to ego vehicle's)
preceding	"parallelMotion" $\wedge$ (direction the same as ego vehicle's)
oncoming	"parallelMotion" $\wedge$ (direction opposite to ego vehicle's)
precedingAdjacent	"preceding" $\wedge$ ("left" $\vee$ "right")
precedingFront	"preceding" $\wedge$ "same"
oncomingAdjacent	"oncoming" $\wedge$ ("left" $\vee$ "right")
oncomingFront	"oncoming" $\wedge$ "same"

Table 5.2: List of relation types representing behavior types and their conditions

Since the clothoid model for lane marking representation is used, the curvature C is defined as the reciprocal value of the curve radius ([Dic92]). It is limited to changing linearly over arc length on each road segment:

$$C = C_0 + C_1 * L. \tag{5.1}$$

The lane marking is defined in border coordinate system as shown in Fig. 5.4, where the offset Y_{BW} and the angle Ψ_{BW} are also depicted. If the angle Ψ_{BW} is positive, ego vehicle is driving to the left. The heading is defined as the first integral of the curvature. If the heading changes are small (less than $15°$), the y_B can be represented as:

$$y_B = C_0 * L^2/2 + C_1 * L^3/6. \tag{5.2}$$

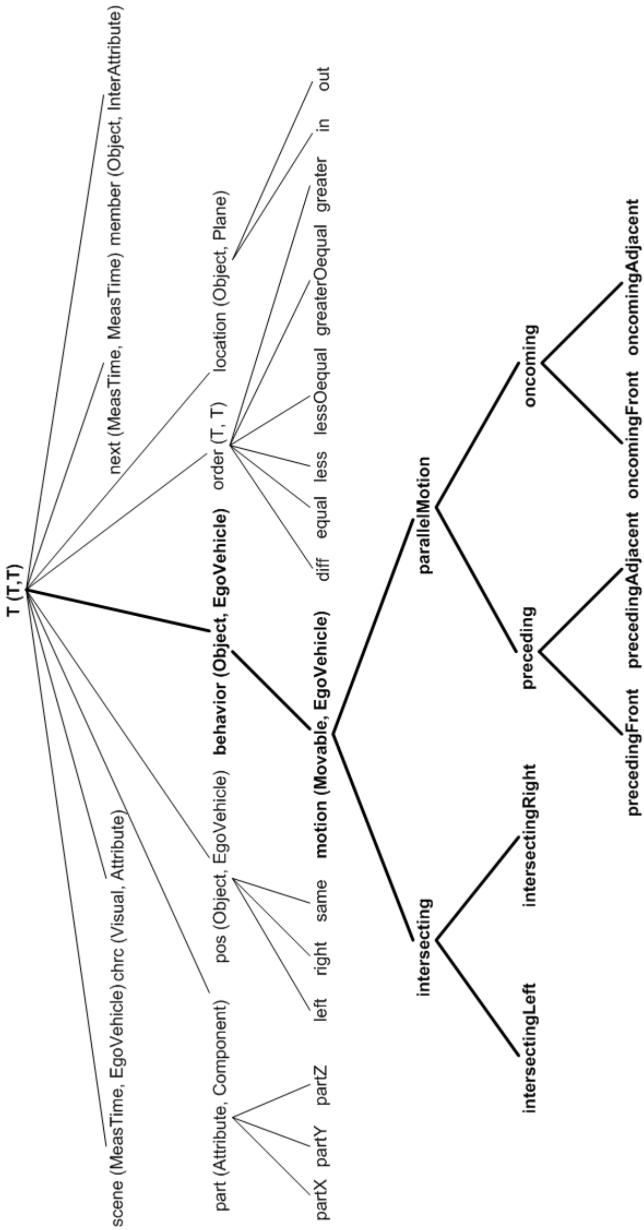


Figure 5.3: Relation type hierarchy

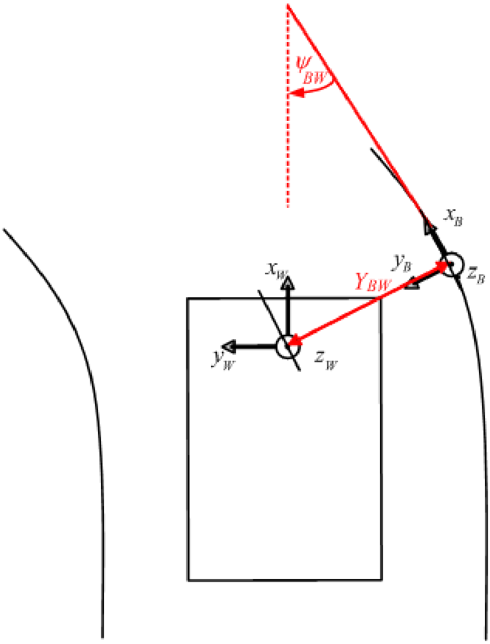


Figure 5.4: DIN world coordinate system and border coordinate system

The arc length L can be linearized:

$$L = arcsin(C_0 * x_B)/C_0. \tag{5.3}$$

Transformation from border coordinates to DIN world coordinates is described with:

$$\mathbf{x}_W = \boldsymbol{R}_{BW}^T \left[\begin{pmatrix} x_B \\ y_B \\ z_B \end{pmatrix} - \begin{pmatrix} 0 \\ Y_{BW} \\ 0 \end{pmatrix} \right], \tag{5.4}$$

where

$$\boldsymbol{R}_{BW} = \begin{bmatrix} cos\Psi_{BW} & -sin\Psi_{BW} & 0 \\ sin\Psi_{BW} & cos\Psi_{BW} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.5}$$

DIN world coordinates of the left and right lane markings on the same distance from origin as the object can be calculated using equations 5.2, 5.3 and 5.4. If the

Figure 5.5: Scene, where two vehicles are moving in front of ego vehicle

y-coordinate of the object lies between the y-coordinates of the left and right lane markings, then the relation “in” between object and the lane is true. Otherwise, the relation “out” is true. If none of those two relations between object and the plane or any of this sub-types exists, the relation “location” is than true.

5.3.3 Individual Markers

Individual markers are also a part of the model support. Since it is not possible to predict all needed individual markers at the beginning, a new individual marker is automatically created when a new object is detected. The object attributes are compared in order to find out if the object has been already known.

5.4 Scene Description

A scene where two passenger cars are preceding ego vehicle is given in Fig. 5.5. The detected passenger car on the right is marked with orange object box because of a high collision probability (0.79) and left detected passenger car is marked with green object box because of a low collision probability (0.21).

A conceptual graph that describes this frame is depicted in Fig. 5.6. The conceptual graph saved in CG Repository consists of three scene conceptual graphs describing three consecutive frames that are joined in one by relation “next”. The explanation will start from the concept node “EgoVehicle” with individual 100 in the middle of the marked part. It is connected with relations “precedingFront” and “precedingAdjacent” with concepts “PassengerCar” with individual 0 and “PassengerCar” with individual 1, respectively. The individuals have the values of object IDs. The concept “PassengerCar:1” is linked with relation “left” and the

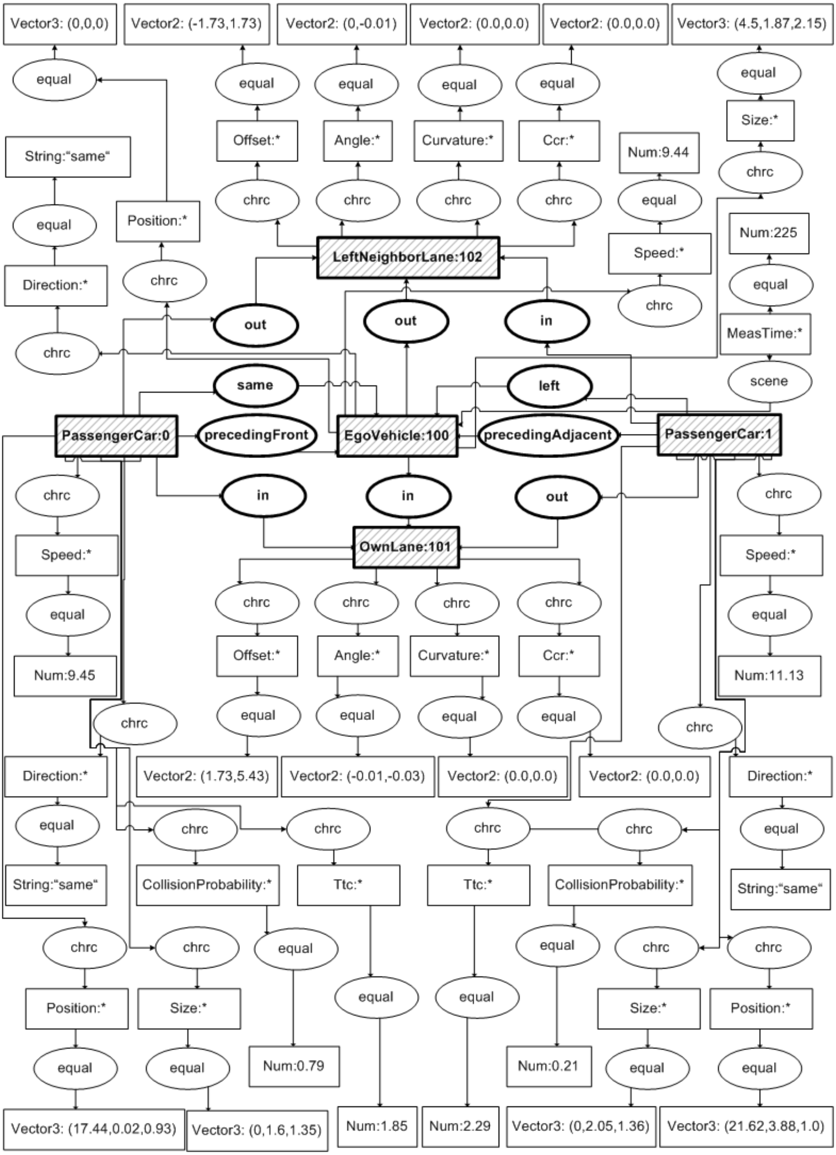


Figure 5.6: Scene conceptual graph describing one frame. It demonstrates the high complexity of the representation.

concept “PassengerCar:0” with relation “same” to the concept “EgoVehicle:100”. The relations “in” join the concepts “EgoVehicle:100” and “PassengerCar:0” with the concept “OwnLane:101” and the concept “PassengerCar:1” with the concept “LeftNeighborLane:102”. The relations “out” join the concepts “EgoVehicle:100” and “PassengerCar:0” with the concept “LeftNeighborLane:102” and the concept “PassengerCar:1” with the concept “OwnLane:101”. The rest of concepts show the attributes and their values.

The algorithm for generating a scene graph depends mostly on the number of detected objects and the number of attributes that describe those objects, but the number of lanes and the number of attributes that describe them cannot be neglected. Between each object at least one relation is determined. The concept hierarchy branch starting from the concept type "Object" is traversed to determine which concept type can be assigned to each object. The concept hierarchy branch starting from the concept type "Plane" is traversed to determine which concept type can be assigned to each road part. The same is done for the relation types between those determined concepts.

Therefore, the complexity of this algorithm can be defined as $O(n_1 * m_1 + n_1 * n_1 + n_1 * n_2 + n_2 * m_2)$, where n_1 is the number of detected objects, m_1 is the number of object attributes, n_2 is the number of detected lanes and m_2 is the number of lane attributes.

5.5 Scene Interpretation

5.5.1 Query Conceptual Graph

The scene can be analyzed using different queries according to the need of the user. Those queries must be constructed with regard to Rules and Constraints that are previously stored. There are some rules and constraints that apply to more than one frame. Therefore not only one scene description is saved but also two previous scene graphs.

The definition of a complex concept type that is stored in Rules is given in Fig. 5.7. It is the sub-type of concept type “InterAttribute”. It is defined between two movable objects if they are moving parallel to the ego vehicle trajectory, the y-coordinate of their position is similar and the distance between them is reduced in two consecutive frames. The part of the definition marked blue gives the conditions from previous frame. Two “Movable” concepts are with relations “parallelMotion” connected to the concept “EgoVehicle”. The relations “chrc” link the concepts “Position” to the concepts “Movable” and the relations “partY” the

concepttype CollisionMovableParallel (*x) **is**

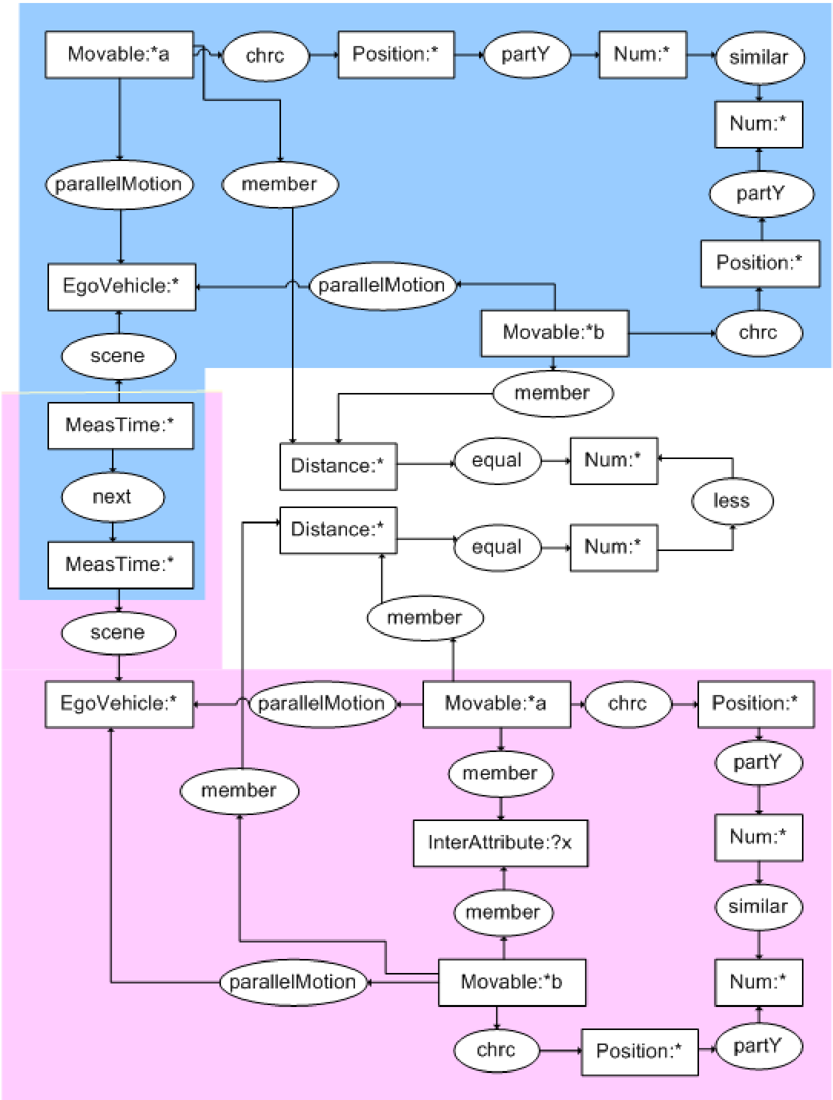


Figure 5.7: Definition of a complex concept type CollisionMovableParallel

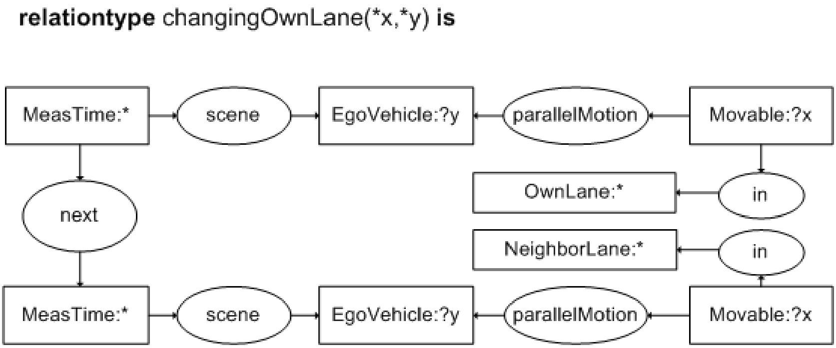


Figure 5.8: Definition of a complex relation type ChangingOwnLane

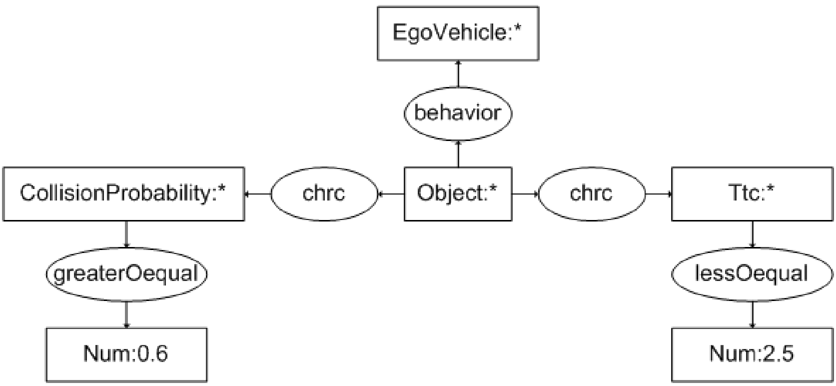


Figure 5.9: Query CG for collision between ego vehicle and object

Figure 5.10: Query CG for changing own lane

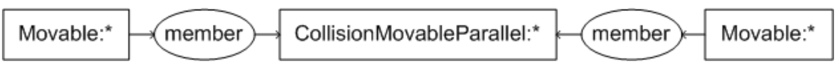


Figure 5.11: Query CG for collision between two parallel moving objects

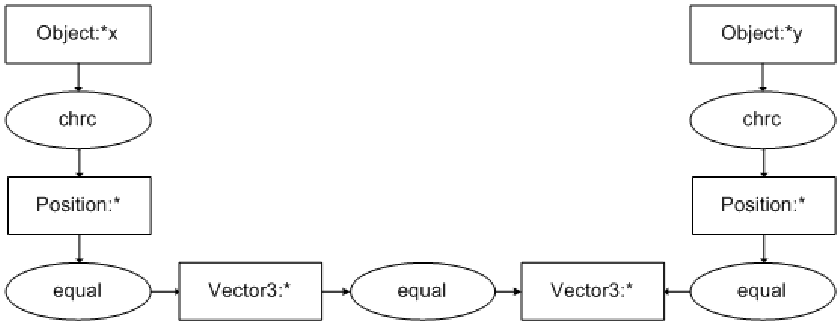


Figure 5.12: Minus constraint

concepts "Position" and "Num" which represent the values of y-coordinates. The relation "similar" between concepts "Num" is true if the difference between them is not greater than 2. The conditions from current frame are depicted in the pink area and they are analog to the previous description and concern two "Movable" concepts with the same individuals. The last part of the rule shows two "Distance" concepts connected with the relation "equal" to "Num" concepts which give the values of the distance between two movable objects in two consecutive frames. If the value from the current scene is smaller than the value from the previous, the relation "less" is defined.

There is also a complex relation type saved in Rules (Fig. 5.8). It is the sub-type of relation type "motion" defined between concept types "Movable" and "EgoVehicle". It shows that, if a movable object is moving parallel to the ego vehicle trajectory and it is in one frame in the lane where ego vehicle is driving and in the consecutive frame in the neighbor lane, the relation is true. The concepts "MeasTime" are connected with the relation "next" which serves as a link between two frame descriptions. In both frames the relation "parallelMotion" connects the concepts "EgoVehicle" and "Movable". Whereas in the previous frame the relation "in" is the link between the concept "Movable" and the concept "OwnLane", the relation "in" in the current frame joins the concept "Movable" and the concept "NeighborLane". The concept "Movable" has the same individual in both frames. A similar complex relation can be defined for changing the left lane or the right lane. In this way it could be possible to detect the lane change in general, but not to predict it.

Additionally two constraints are defined, one belonging to minus constraints (Fig. 5.12) and one to plus constraints (Fig. 5.13). The minus constraint expresses the fact that two objects cannot have the same position at the same time. If this graph

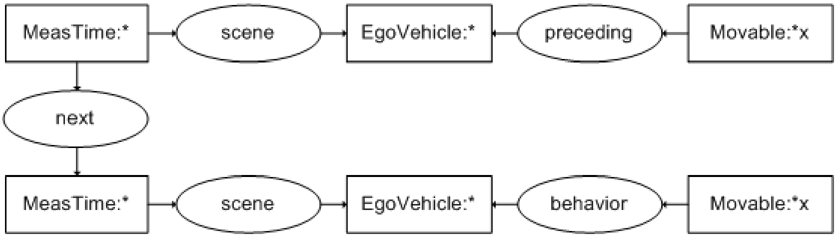


Figure 5.13: Plus constraint

is found in the scene graph, no solution is delivered. The plus constraint expresses the fact if the object is preceding ego vehicle in one frame, the object is present in the scene in the next frame. If this graph is matched, a solution is delivered. If this graph is not matched, this graph is joined with the current scene graph. This can help if the objects from one to the next frame disappear.

To issue a warning for a collision between ego vehicle and object, a query conceptual graph that combines collision probability and TTC is constructed (Fig. 5.9). It says that if TTC is not greater than $2.5s$ and collision probability is not less than 0.6, a possible collision is detected. A query for a situation when a movable object is changing from lane where ego vehicle is driving to another lane (Fig. 5.10), and a query for a detection of a collision between two movable objects (other than ego vehicle) driving parallel to ego vehicle (Fig. 5.11) are constructed.

5.5.2 Graph Matching

The conceptual graph stored in the CG repository is created during the run-time by joining the current scene CG with the previous using the relation “next”. That graph and the query graph are saved not only with the general conceptual graph information, but also with a $c-r-c$ list. $c-r-c$ is a 3-tuple, $T =< c_i, r, c_j >$, where c_i, c_j are concept nodes (i and j are not equal) and r is a conceptual relation, where c_i is the first and c_j the second argument of the relation r. The overall running time for the projection operation is known to be NP class problem [Che92].

The algorithm presented in this work is based on an idea proposed by Pfeiffer et al. [Pfe07]. Some modifications are made to reduce the time for finding and creating the projections. The projection of the query graph onto the CG repository is based on looking at all triples that are in the query graph and checking for a complete subgraph match of the query graph. Subsequently, after all the matches of triples are found, the actual projection graphs are built.

The algorithm is shown in Section A.1. It is split into two sections: the answering of the decision question: “Is there a projection?” and the construction of projections. New data structure *MappedTripleList* that consists of the list of matched triples for each matched relation of the query graph is introduced. This data structure improves the performance by making preprocessed information accessible for the phase where the projection graphs are build. In this case it was better to use relations instead of concepts, as described in [Pfe07], because it was faster to get the negative answer to the decision question if there is one relation in the query graph that could not be matched. In most of the scene graphs the concept types are similar, but the relation types are different.

The size of the scene graph has a big effect on the first part of the matching algorithm, because it is normally much larger than the query graph. Not only the size, but also the content has an influence on the execution time. If there are a lot of triples in the scene graph that match one triple in the query graph, as it is the case, the process of building projection by checking all possibilities that can form a subgraph of the scene graph is getting much longer.

The amount of time for the projection creation is also controlled by the number of triples in the query graph. If the query graph includes a complex concept, the complex type definition is seen as a piece of the query graph. To make the algorithm faster, the number of relations in a query graph is limited. For example, the complex concept type “CollisionMovableParallel” is divided into three parts as depicted in Fig. 5.7 and for each of them a query is created. The first query marked blue represents the question for one frame: “Are there two objects that are moving parallel to ego vehicle and the y-coordinate of their positions is similar?”. The second, marked pink, represents the same question for the following frame. The third query consists of non-marked part of the rule, “Movable” concepts and “member” relations from other two parts and represents the question :”Is the distance between the objects in the following frame smaller than in the previous?”.

5.5.3 Result Conceptual Graph

The query CG for collision between ego vehicle and object depicted in Fig. 5.9 is applied. In this case the concept node “PassengerCar” with the individual 0 is the concept with the super-type “Object” that is with the relation “precedingFront”, whose type is the sub-type of the relation “behavior”, connected to the concept "EgoVehicle" with the individual 100. It is connected with the relation “chrc” to the concept “CollisionProbability”, whose value is equal 0.79, which is greater than the given threshold 0.6. The concept “PassengerCar” is also with the relation “chrc” to the concept “Ttc” attached. Its value is equal 1.85, which is less than

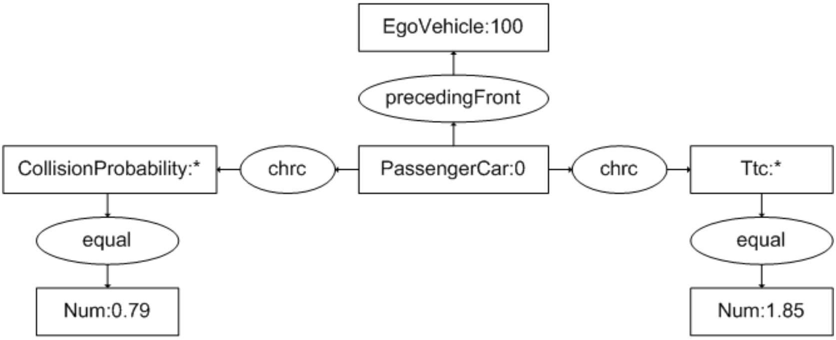


Figure 5.14: Result CG for collision between ego vehicle and object

2.5. The Result CG is depicted in Fig.5.14.

6 Model Validation

The system has been realized on the PC in the programming language C++ using Automotive Data and Time triggered Framework (ADTF). ADTF is provided by Elektrobit and is used in the area of driver assistance systems. It supports the modular structure and makes the changes in the system architecture easier.

6.1 Use Cases

The model is validated on four use cases. Selected use cases represent the most typical traffic scenarios. For each situation several real video sequences involving ego vehicle and one or two additional cars were recorded and analyzed. Some of them show the scenarios when the situation that is examined exists and they are marked as positive. The others were taken to show that, if the situation does not happen, the match is also not found. They are identified as negative.

- In the use case 1 (Fig. 6.1) a vehicle is driving with a constant speed on a street that is orthogonally intersecting ego lane, and braking before it reaches a crossroad. A situation "collision between ego vehicle and object" is analyzed. To see if it comes to a collision, a query depicted in Fig. 5.9 is applied. For a positive scenario the speed of the passenger car is $20km/h$, its distance from the crossroad $40m$, the speed of ego vehicle $50km/h$ and its distance from the crossroad $100m$ when they start driving with the constant speed. The negative scenario includes the passenger car that is driving $30km/h$ and there is no possibility that it will come to a collision.

- Use case 2 represents a scenario when a vehicle is driving in front of ego vehicle (Fig. 6.2). As in the previous case, a query shown in Fig. 5.9 is used. When the speed of a vehicle is smaller than the speed of ego vehicle, it could come to a collision. In the negative scenario a vehicle is driving faster than the ego vehicle, so the collision is not possible.

- A scenario when a vehicle is overtaking a vehicle driving parallel to ego vehicle in the different direction on the left neighbour lane is taken for the use case 3 and displayed in Fig. 6.3. The overtaking vehicle is driving with the speed $40km/h$ and ego vehicle with the speed $30km/h$. Two different situations are examined:

- The query 1 shown in Fig. 5.9 is used for examining the collision situation between ego vehicle and any object. Taking into account the speed of the overtaking object and ego vehicle, for the verification purposes it can be calculated that it could come to a collision if the overtaking vehicle is still in own lane, when the distance between it and ego vehicle is less than $50m$. In the negative scenario the passenger car overtakes the other passenger car much earlier, so the whole overtaking process is finished before the distance between ego vehicle and passenger car reaches approximately $50m$.
- The query 2 describes a situation when a collision between two parallel moving objects other than ego vehicle could occur. It is depicted in Fig. 5.11 and Fig. 5.7. In the positive scenario the overtaking vehicle is too close to the other vehicle and its speed is just a bit greater than the speed of the other vehicle. It does not come to a collision between two parallel moving vehicles if the overtaken vehicle is driving much slower than the other vehicle, is braking a bit and the other vehicle is overtaking further from it.

• A part of a scenario when a vehicle driving in front of ego vehicle is overtaking another vehicle in front of it is use case 4 (Fig. 6.4). The first process of the overtaking is changing from own lane to the left one. This situation is examined by the query for changing own lane depicted in Fig. 5.10. In the positive scenario the passenger car direct in front of ego vehicle changes the lane and in negative it starts to change the lane, but the driver changes his mind and returns to own lane.

6.2 Results

The results for the positive scenarios are summarized in Table 6.1. It shows ,for each situation, the number of taken video sequences ("Number"), in how many of them the situation was found ("True") and in how many was not ("False"). Table 6.2 for the negative scenarios gives also, for each situation, the number of taken video sequences ("Number"), in how many of them the situation was not found ("True") and in how many it was ("False").

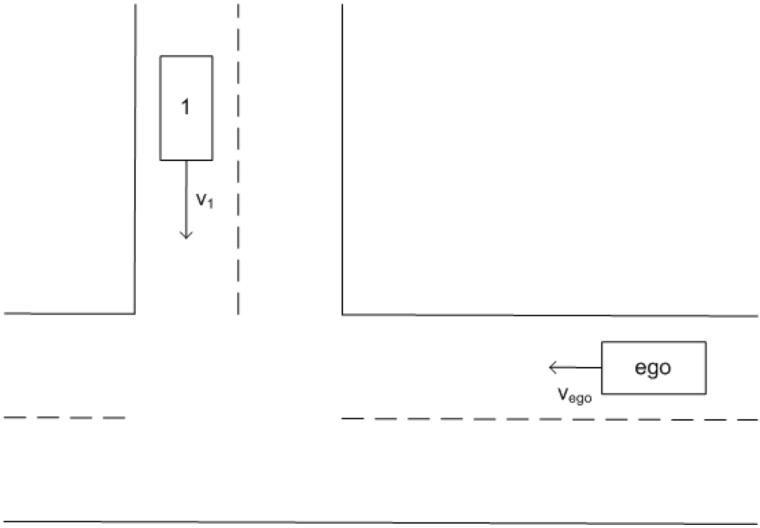


Figure 6.1: Use case 1: A vehicle is intersecting from right.

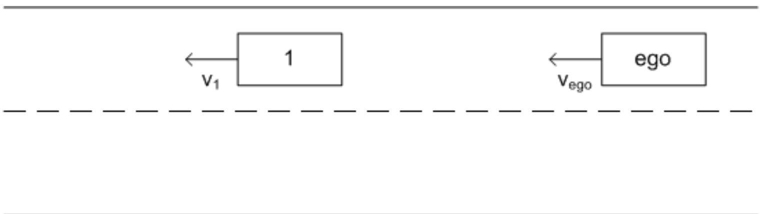


Figure 6.2: Use case 2: A vehicle is preceding in front of ego vehicle.

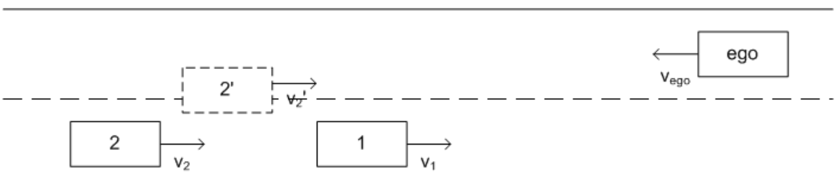


Figure 6.3: Use case 3: A vehicle is overtaking a vehicle oncoming left from ego vehicle.

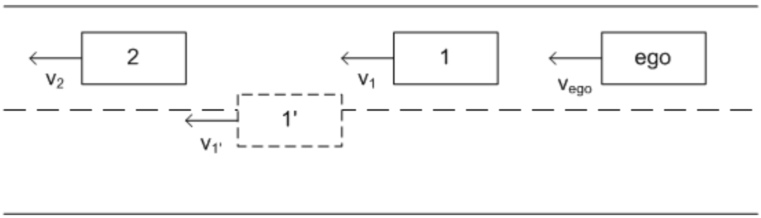


Figure 6.4: Use case 4: A vehicle that is preceding in front of ego vehicle is moving to other lane.

Situation	**Number**	**True**	**False**
Collision between ego vehicle and object Use Case 1	20	20	0
Collision between ego vehicle and object Use Case 2	20	20	0
Collision between ego vehicle and object Use Case 3	10	10	0
Collision between two parallel moving objects Use Case 3	10	9	1
Changing own lane Use Case 4	10	10	0

Table 6.1: Positive scenarios

Situation	**Number**	**True**	**False**
Collision between ego vehicle and object Use Case 1	10	0	10
Collision between ego vehicle and object Use Case 2	10	10	0
Collision between ego vehicle and object Use Case 3	10	5	5
Collision between two parallel moving objects Use Case 3	10	10	0
Changing own lane Use Case 4	10	10	0

Table 6.2: Negative scenarios

- An example result for one of true sequences for the collision between ego vehicle and object in use case 1 is shown in Fig. 6.5. The passenger car marked red is the object with the high probability of collision with ego vehicle and small TTC.

- One of the scenes where a match is found in use case 2 is presented in Fig. 6.6. Since the collision probability for the passenger car in the image equals 0.81 and TTC $2.1s$, the result is determined and the passenger car in question is marked red.

- In the scene from use case 3 shown in Fig. 6.7 the collision probability of the passenger car with the id 0 is 0.81 and TTC $2.0s$, so the concept that represents the passenger car with the id 0 is a part of the result conceptual graph for the query 1.

- One of the results in use case 3 for the query 2 is given in Fig. 6.8. A significant part of result CG in the current scene description looks like this:

    1. concept “EgoVehicle” with the individual 100
    2. concept “PassengerCar” with the individual 0 has the super-type “Movable”
    3. concept “PassengerCar” with the individual 1 has the super-type “Movable”
    4. relation “oncomingFront” with the super-type “parallelMotion” connects 2 and 1
    5. relation “oncomingAdjacent” with the super-type “parallelMotion” connects 3 and 1
    6. concept “Position” with generic individual
    7. concept “Position” with generic individual
    8. relation “chrc” connects 2 and 6
    9. relation “chrc” connects 3 and 7
    10. concept “Num” with the individual 1.7 represents the value of y-coordinate [m]
    11. concept “Num” with the individual 3.7 represents the value of y-coordinate [m]
    12. relation “partY” connects 6 and 10
    13. relation “partY” connects 7 and 11

Since the difference between individuals of “Num” concept is not greater than 2 the relation “similar” is true. This is the same for the previous frame. Moreover, the value of the concept “Distance”, which is connected with both concepts “PassengerCar” with the relation “member”, is smaller than the value of the concept “Distance” in the previous frame. This yields a result conceptual graph, and the graphical representation of both “PassangerCar” concepts is marked red in the image.

- A result for the situation changing own lane in one scene in use case 4 is shown in Fig. 6.9. A part of the result CG coming from the current frame can be described like this:

  1. concept “EgoVehicle” with the individual 100
  2. concept “PassengerCar” with the individual 0 has the super-type “Movable”
  3. relation “precedingAdjacent” with the super-type “parallelMotion” connects 2 and 1
  4. concept “LeftNeighborLane” with individual 102 has the super-type “NeighborLane”
  5. relation “in” connects 2 and 4

  In the previous frame the relation “in” was the connection between the concept “PassengerCar” with the individual 100 and the concept “OwnLane” with individual 101. Since the whole query can be matched, the result CG is created. The red marked object is the concept “PassengerCar” from the result.

6.3 Discussion

The traffic situation recognition success rate for positive scenarios is presented in Fig. 6.10. As it can be seen, the average success rate for positive scenarios is 98%.

For the situation collision between ego vehicle and the passenger car in the positive scenarios in some frames the collision probability was sporadically falling under the threshold of 0.6. As the result of bad weather conditions, e.g. fog and low sun, the object detection was not always correct. The inaccuracy of the object detection was around 5% ([Weh10]). On the other hand, optical flow is very sensitive on small changes in speed and direction of the object (see Section 2.1). During the situation recording the driver was not always able to keep the constant speed and

Figure 6.5: One of the scenes from use case 1 when a match is found. The distance of the ego vehicle to the crossroad is $46m$.

Figure 6.6: One of the scenes from use case 2 when a match is found. The distance of the ego vehicle to the other vehicle is $31m$.

Figure 6.7: One of the scenes from use case 3 when a match with the query for collision between ego vehicle and object is found. The distance between them is $22m$.

Figure 6.8: One of the scenes from use case 3 when a match with the query collision between two parallel moving objects is found. The distance between them is $4.4m$.

Figure 6.9: One of the scenes from use case 4 when a match with the query for changing own lane is found

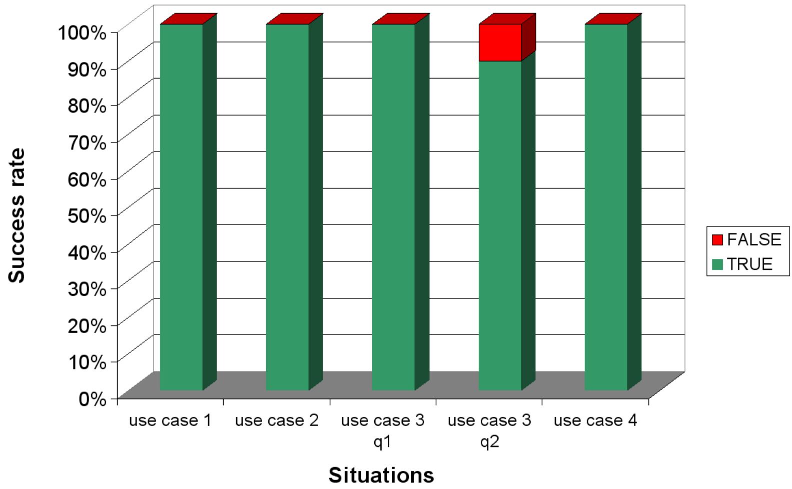


Figure 6.10: Success rate of traffic situations recognition for positive scenarios

direction of the vehicle. By saving the conceptual graph that describes a current scene and two previous frames, and finding a match in those three graphs this behavior could be minimized.

In use case 3 for the situation collision between two parallel moving objects (q2) in the positive scenarios, although the passenger car was in front of other passenger car and the distance between was getting smaller in time, the match was not found. This can be explained with the fact that for distances greater than approximately $60m$, the error in distance calculation can be greater than $1m$ ([Weh10]). Because of this, the positions of objects were not properly determined, so the calculated distance between objects was not getting smaller.

The success rate of traffic situation recognition for negative scenarios is summarized in Fig. 6.11.

In the negative scenarios in use case 1, in all of them the match was found at the end of the video sequences when the passenger car is braking and it is very close to ego vehicle. At one point it comes to collision detection because the collision probability gets higher and in its calculation the acceleration of the object is not taken into account. It is assumed that the object is continuing the driving with the same speed. To point out that the issue was caused by the choice of scenario,

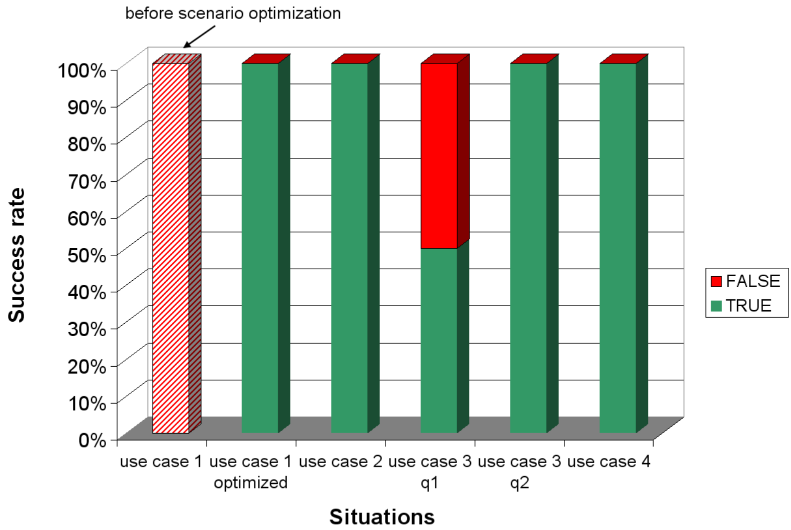


Figure 6.11: Success rate of traffic situations recognition for negative scenarios

Figure 6.12: Example of a scene where an incorrect segmentation leads to incorrect recognition in “use case 3 q1”

the results in Fig. 6.11 are marked with pattern. The negative scenarios were optimized, so the passenger car was driving through the crossroad in front of ego vehicle and 5 new sequences were recorded. In none of those 5 sequences the match was found, confirming correct functionality of the prototype. In order to make sure that in the reality the collision would not occur more test sequences would be needed.

In the negative scenarios of use case 3 for the collision between ego vehicle and object (q1), the match in false video sequences was found at the end of the video sequences when the passenger car is close to ego vehicle and disappears from the image. Instead of detection of object as one object box, segmentation process resulted in more parts with different sizes of object boxes and collision probabilities (Fig. 6.12). One of those collision probabilities was greater than 0.6. The object box that corresponded to it covered only a small part of the vehicle and was unstable. It gave incorrect calculation of object size, position and TTC, which led to incorrect calculation of collision probability.

Situation	**Generator**	**Matcher**
Collision between ego vehicle and object Use Case 1	1	15
Collision between ego vehicle and object Use Case 2	1	15
Collision between ego vehicle and object Use Case 3	3	55
Collision between two parallel moving objects Use Case 3	3	1950
Changing own lane Use Case 4	2	40

Table 6.3: Average run-time in [ms] for CG Generator and CG Matcher when the match is found (calculated using 2.13 GHz Pentium-M 770 processor with 1024 MB RAM)

The average run-time of components CG Generator and CG Matcher, for the frames when the match is found, was measured and is presented in Table 6.3. Since the run-time of CG Matcher for some situations is greater than the time between two frames ($40ms$), this system cannot be deployed in an electronic control unit (ECU) of current generation, although the success rate of situation recognition is very high.

7 Conclusion

7.1 Summary

In the thesis the suitability of conceptual graphs for situation understanding in advanced driver assistance systems was investigated. For this purpose a prototype of a model that analyzes the information provided by the stereo video system and which provides the high-level description of the ego vehicle environment was created. The model recognizes different types of real traffic situations and could potentially be the core of the system which supports the driver, extending his safety and comfort.

The model is based on conceptual graphs, which are expressive and offer high flexibility to analyze the extracted data. The important feature of CGs is that the relations between other objects besides ego vehicle can be represented. This quality, which is not available in many approaches presented in the literature, e.g. Situation Graph Tree ([Are02], [Nag05]), can be used to detect a potential risk for ego vehicle caused by other traffic participants. Such exemplary situation: the collision between two oncoming vehicles, if one of them is overtaking the other, was taken into account in the model verification phase described in this work. Although the implementation is made for video-based driver assistance systems, this approach is universal and can be used for driver assistance systems based on other types of sensors.

In this work, the estimation method of a collision danger was proposed for pure translation between ego vehicle and the object. It was based on the fact that if collision should occur between the camera and an object, the FOE of their relative motion lies within the growing projection of the object in the image. Additionally, the size of ego vehicle and the object was considered, so the critical area around the object projection in the image could be determined.

The functionality of the model prototype has been verified on four use cases, selected to represent the most typical traffic scenarios. For each situation several video sequences were recorded and analyzed. Some of them show the scenarios when the situation that is examined exists and they are marked as positive. The others were taken to show that, if the situation does not happen, the match is not found. They are identified as negative.

In the use case 1, a vehicle is driving with a constant speed on a street that is orthogonally intersecting ego lane and braking before it reaches a crossroad. Use case 2 represents a scenario when a vehicle is driving in front of ego vehicle. A scenario when a vehicle is overtaking a vehicle driving parallel to ego vehicle in the different direction on the left neighbor lane is taken for the use case 3. A scenario when a vehicle driving in front of ego vehicle is overtaking another vehicle in front of it is considered in use case 4. For first three use cases a situation "collision between ego vehicle and object" is analyzed. In the third use case a situation when a collision between two parallel moving objects, other than ego vehicle, is also investigated. A change from own lane to the left one for the object in front of ego vehicle is examined in the fourth use case.

The testing results can overall be described as positive. The traffic situation recognition success rate for positive scenarios was close to 100%. The success rate of traffic situation recognition for negative scenarios is summarized in Fig. 6.11. After optimization of "use case 1" to make sure that the collision in reality would not take place, the model delivered a correct answer in average 90% of tested sequences. As expected, based on theory, the negative influence of bad weather conditions causing variation in optical flow was observed, not leading to significant difficulties in matching process. Disturbing effects of the changes in optical flow caused by small changes in speed and direction of the object during the situation recording, were successfully minimized by saving the conceptual graph that describes a current scene and two previous frames and finding a match in those three graphs.

7.2 Evaluation

An evaluation of the developed method is based on the requirements presented in Section 1.2. A scene description and interpretation are carried out when the information from all sources are gathered and processed. This assures better conclusions than in the case when each source is considered separately. A decision is made also under insufficient information, but it cannot be revised in the future when a new information appears. In this case new decision is made based on new evidence.

The fact that conceptual graphs are a subset of first order logic is proven to be beneficial. If the application is changed, there is no need to alter the whole representation and reasoning process, as it is the case in propositional languages. Unfortunately, there are some ambiguities in the semantics of conceptual graphs, since not all quantifiers can be expressed, e.g. "at least" or "exactly". The Open World

Assumption and the Open Domain Assumption which are required for scene understanding are fulfilled. They are required for scene understanding. The graphical representation of both quantitative and qualitative knowledge makes a spatial reasoning easier and the possibility to represent a relation between more than two objects is also given.

The representation is modular. Since model support, concept and relation definitions, rules, constraints and queries are stored apart and separated from the reasoning process, the corrections or extensions in the model can be done at minimal expense. They are also well readable.

As described the conceptual graphs have many positive aspects, like modularity and readability, but for small body of scene knowledge they are not as efficient as quantitative models. These positive aspects become visible when the scene knowledge base gets large enough, so the number of if-then clauses needed for reasoning blows up and the result of reasoning can become unreliable. The conceptual graphs have proven to be suitable representation for situation understanding fulfilling many of its conditions.

It is also possible to develop an user interface that creates a query graph based on the query in the natural language. Since only a prototype is described in this work, this quality did not come to its own, bur the usage of object-oriented paradigm and graph algorithms did. Although the situation recognition rate was out of the scope of this thesis, it can be stated that the system's speed is not sufficient for real-time applications. A significantly greater computer performance and approximate graph matching algorithm is necessary for such utilization.

7.3 Outlook

The future work on this topic could include investigation on the effect on collision probability, when the relative motion between ego vehicle and an object is not pure translation and the acceleration of the objects is taken into account. By optimization of external measurement programs delivering input to the model, the method output could become more robust on bad weather conditions and small changes in speed and direction. In the current version of the method the quality measures for the distance estimation and the object's lateral position calculation were assumed to be constant. Higher number of test cases should be analyzed to determine the quality criteria and in this way extend the precision of the collision probability calculation.

The functionality of the system could be enhanced by definition of additional concepts, relations and constraints, which would equip the system to answer new queries based on available data. The extension of the model related to the probability of the situation recognition would be valuable. The idea to solve this issue with the weights for concepts and relation depending on the importance in the situation could be potentially applied. Additionally, the implementation of the approximate graph matching algorithm to make the recognition process faster could be one of the subjects of future work. Nevertheless, the quality of the output would be lower comparing to the presented method.

A Appendix

A.1 Graph Matching Algorithm

```
Get all relation vertices from Scene and Query graphs
Pset ← empty
for j ← 0, numsecondrelations do    //all relations in Query graph
    mappedRelation[j] ← false
    MappedTripleList[j] ← empty
end
for i ← 0, numfirstrelations do    //all relations in Scene graph
    for j ← 0, numsecondrelations do    //all relations in Query graph
        if (type(ri) == type(rj) or (type(rj) is in the list of supertypes(ri) then
            if (type(rj.1) == type(ri.1) or type(rj.1) is in the list of supertypes(ri.1)) and
            (type(rj.2) == type(ri.2) or type(rj.2) is in the list of supertypes(ri.2)) then
                if (individual(ri.1) == individual(rj.1) or individual(rj.1) == 0) and
                (individual(ri.2) == individual(rj.2) or individual(rj.2) == 0) then
                    if (relations connected to rj.1 are subset of relations connected to ri.1) and
                    (relations connected to rj.2 are subset of relations connected to ri.2)
                        mappedRelation[j] ← true
                        T ← (ri.1- ri- ri.2)
                        MappedTripleList[j] ← MappedTripleList[j] ∪ T
                    end if
                end if
            end if
        end if
    end for
end for
foundmatch ← true
for j ← 0, numsecondrelations do
    if !mappedRelation[j] then
        foundmatch ← false
        break
    end if
end for
if foundmatch == true then
    combination(PSet,CombiTriples,MappedTripleList,numsecondrelations,0)
end if
```

```
if (!Pset==empty) then
    return Pset    //set of projections returned
else
    return 0    //no projection returned
end if
```

```
function combination(PSet,CombiTriples,MappedTripleList,numsecondrelations,kk)
begin
if (!MappedTripleList[kk]==empty)
  for i ← 0, numMappedTripleList[kk] do    //all triples in MappedTripleList[kk]
    CombiTriples[kk] ← MappedTripleList[kk][i]
    if (!kk == numsecondrelations-1) then
      combination(PSet,CombiTriples,MappedTripleList,numsecondrelations,kk+1)
    else
      P ← new subgraph projection from CombiTriples    //build new projection
      Pset ← Pset ∪ P
    end if
  end for
end if
end
```

List of Tables

List of Figures

Bibliography

[Aga09] V. Agarwal, N. V. Murali and C. Chandramouli: „A Cost-Effective Ultrasonic Sensor-Based Driver-Assistance System for Congested Traffic Conditions“. *IEEE Transactions on Intelligent Transportation Systems* **10** (3), pp. 486–498, September 2009.

[Alt08] M. Althoff, O. Stursberg and M. Buss: *Stochastic Reachable Sets of Interacting Traffic Participants*. In *Proceedings of the 2008 IEEE Intelligent Vehicle Symposium*, pp. 1086–1092, 2008.

[Amd10] A. Amditis, E. Bertolazzi, M. Bimpas, F. Biral, P. Bosetti, M. D. Lio, L. Danielsson, A. Gallione, H. Lind, A. Saroldi and A. Sjögren: „A Holistic Approach to the Integration of Safety Applications: The INSAFES Subproject Within the European Framework Programme 6 Integrating Project PReVENT“. *IEEE Transactions on Intelligent Transportation Systems* **11** (3), pp. 554–566, September 2010.

[Are02] M. Arens and H.-H. Nagel: *Representation of Behavioral Knowledge for Planning and Plan-Recognition in a Cognitive Vision System*. In *KI 2002: Advances in Artificial Intelligence, 25th Annual German Conference on AI, KI 2002, Aachen, Germany, September 16-20, 2002, Proceedings*, M. Jarke, J. Koehler and G. Lakemeyer (eds.), vol. 2479 of *Lecture Notes in Computer Science*, pp. 268–282, Springer Berlin Heidelberg, 2002.

[Bal82] D. Ballard and C. Brown: *Computer Vision*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

[Bar94] J. L. Barron, D. J. Fleet and S. S. Beauchemin: „Performance of Optical Flow Techniques“. *International Journal of Computer Vision* **12**, pp. 43–77, 1994.

[Bar04] J. Barron and N. Thacker: *Tutorial: Computing 2D and 3D Optical Flow*. Tina Memo 2004-012, Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, Manchester, 2004.

[Bor10a] N. Boroch and T. Heger: *Vehicle Environment Description and Interpretation Using Conceptual Graphs*. In *Proceedings of the IEEE Intelligent Transportation Systems Conference 2010*, September 2010.

[Bor10b] N. Boroch, J. Jatzkowski and T. Heger: *Estimation of Collision Risk using Monocular On-Board Camera*. In *IFAC Symposium Advances in Automotive Control*, July 2010.

[Bot10] M. Botsch and C. Lauer: *Complexity Reduction Using the Random Forest Classifier in a Collision Detection Algorithm*. In *Proceedings of the 2010 IEEE Intelligent Vehicle Symposium*, pp. 1228–1235, 2010.

[Bra97] M. Brand, N. Oliver and A. Pentland: *Coupled hidden Markov models for complex action recognition*. In *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 994–999, 1997.

[Bro05] A. Broadhurst, S. Baker and T. Kanade: *Monte Carlo Road Safety Reasoning*. In *Proceedings of the 2005 IEEE Intelligent Vehicle Symposium*, pp. 319–324, 2005.

[Cha04] P. Chang, T. Camus and R. Mandelbaum: *Stereo-Based Vision System for Automotive Imminent Collision Detection*. In *Proceedings of the 2004 IEEE Intelligent Vehicle Symposium*, pp. 274–279, 2004.

[Che92] M. Chein and M. laure Mugnier: „Conceptual Graphs: fundamental notions“. *Revue d'Intelligence Artificielle* **6**, pp. 365–406, 1992.

[Che06] J. Chengwu, J. Guiyan, L. Ximin and D. Tongqiang: *Dangerous Situation Recognition Method of Driver Assistance System*. In *Proceedings of the 2006 IEEE International Conference on Vehicular Electronics and Safety*, pp. 169–173, 2006.

[Che07] H. Cheng, N. Zheng, X. Zhang, J. Qin and H. van de Wetering: „Interactive Road Situation Analysis for Driver Assistance and Safety Warning Systems: Framework and Algorithms“. *IEEE Transactions on Intelligent Transportation Systems* **8** (1), pp. 157–167, March 2007.

[Chu03] L. Chuanjin, Q. Xiaohu, H. Xiyue, C. Yi and Z. Xin: *A monocular-vision-based driver assistance system for collision avoidance*. In *Proceedings of the IEEE Intelligent Transportation Systems Conference 2003*, pp. 463–468, October 2003.

[Cro07a] M. Croitoru, B. Hu, S. Dashmapatra, P. Lewis, D. Dupplaw and L. Xiao: *A Conceptual Graph Description of Medical Data for Brain Tumour Classification*. In *Conceptual Structures: Knowledge Architectures for Smart Applications; Proceedings of the 15th International Conference on Conceptual Structures*, U. Priss, S. Polovina and R. Hill (eds.), vol. 4604 of *Lecture Notes in Artificial Intelligence*, pp. 140–153, Springer Berlin Heidelberg, 2007.

[Cro07b] M. Croitoru and K. van Deemter: *A Conceptual Graph Approach to the Generation of Referring Expressions*. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2456–2461, January 2007.

[Dag04] E. Dagan, O. Mano, G. P. Stein and A. Shashua: *Forward Collision Warning with a Single Camera*. In *Proceedings of the Intelligent Vehicles Symposium*, pp. 37–42, IEEE, 2004.

[Dic92] E. Dickmanns and B. Mysliwetz: „Recursive 3-D Road and Relative Ego-State Recognition“. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14** (2), pp. 199–213, February 1992.

[Dic02] E. Dickmanns: *The development of machine vision for road vehicles in the last decade*. In *Proceedings of the IEEE Intelligent Vehicle Symposium 2002*, vol. 1, pp. 268–281, June 2002.

[Dic05] E. Dickmanns: *Vision: Von Assistenz zum Autonomen Fahren*. In *Fahrerassistenzsysteme mit maschineller Wahrnehmung*, M. Maurer and C. Stiller (eds.), pp. 203–237, Springer Berlin Heidelberg, 2005.

[DIN94] DIN: *Strassenfahrzeuge; Fahrzeugdynamik und Fahrverhalten; Begriffe; ISO 8855 Ausgabe 1991, modifiziert*. Beuth Verlag Berlin, 1994.

[DK05] R. Dieng-Kuntz and O. Corby: *Conceptual Graphs for Semantic Web Applications*. In *Proceedings of the 13th International Conference on Conceptual Structures*, F. Dau, M.-L. Mugnier and G. Stumme (eds.), vol. 3596 of *Lecture Notes in Artificial Intelligence*, pp. 19–50, Springer Berlin Heidelberg, 2005.

[Duc06] C. Duchow, B. Hummel, A. Bachmann, Z. Yang and C. Stiller: *Akquisition, Repräsentation und Nutzung von Wissen in der Fahrerassistenz*. In *Informationsfusion in der Mess- und Sensortechnik 2006*, pp. 223–237, 2006.

[Eid08] A. Eidehall and L. Petersson: „Statistical Threat Assessment for General Road Scenes Using Monte Carlo Sampling". *IEEE Transactions on Intelligent Transportation Systems* **9** (1), pp. 137–147, March 2008.

[Foe06] M. Foedisch, R. Madhavan, S. Balakirsky, M. Shneier and C. Schlenoff: *Symbolic Representations for Autonomous Vehicle Perception and Control in Urban Environments*. In *Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision*, pp. 1–6, 2006.

[Fra05] U. Franke, C. Rabe, H. Badino and S. Gehring: *6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception*. In *Pattern Recognition*, W. Kropatsch, R. Sablatnig and A. Hanbury (eds.), vol. 3663 of *Lecture Notes in Computer Science*, pp. 216–223, Springer Berlin Heidelberg, 2005.

[Fri08] J. Fritsch, T. Michalke, A. Gepperth, S. Bone, F. Waibel, M. Kleinehagenbrock, J. Gayko and C. Goerick: *Towards a Human-like Vision System for Driver Assistance*. In *Proceedings of the 2008 IEEE Intelligent Vehicle Symposium*, pp. 275–282, 2008.

[Fue05] K. C. Fuerstenberg and T. Weiss: *Feature-Level Map Building and Object Recognition for Intersection Safety Applications*. In *Proceedings of the 2005 IEEE Intelligent Vehicle Symposium*, pp. 490–495, 2005.

[Fue06] K. C. Fuerstenberg and B. Roessler: *Advanced Intersection Safety - The EC project INTERSAFE*. In *Proceedings of the 2006 IEEE Intelligent Vehicle Symposium*, pp. 89–93, 2006.

[Gen97] D. Genest and M. Chein: *An Experiment in Document Retrieval Using Conceptual Graphs*. In *Conceptual structures: Fulfilling Peirce's Dream; Proceedings of the 5th International Conference on Conceptual Structures*, D. Lukose, H. Delugach, M. Keeler, L. Searle and J. Sowa (eds.), vol. 1257 of *Lecture Notes in Artificial Intelligence*, pp. 489–504, Springer Berlin Heidelberg, 1997.

[Hü10] T. Hülnhagen, I. D. andd Andreas Tamke, T. Dang and G. Breuel: *Maneuver recognition using Probabilistic Finite-State Machines and Fuzzy Logic*. In *Proceedings of the 2010 IEEE Intelligent Vehicle Symposium*, pp. 65–70, 2010.

[Har03] R. Hartley and A. Zisserman: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, United Kingdom, Second. ed., 2003.

[Hil06] J. Hillenbrand, A. M. Spieker and K. Kroschel: „A Multilevel Collision Mitigation Approach-Its Situation Assessment, Decision Making, and Performance Tradeoffs“. *IEEE Transactions on Intelligent Transportation Systems* **7** (4), pp. 528–540, December 2006.

[Hu06] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan and S. Maybank: „A System for Learning Statistical Motion Patterns“. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28** (9), pp. 1450–1464, September 2006.

[Hua94] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell and J. Weber: *Automatic Symbolic Traffic Scene Analysis Using Belief Networks*. In *Proceedings of the 12th National Conference on Artificial Intelligence*, pp. 966–972, 1994.

[Hum09] B. Hummel: *Description Logic for Scene Understanding at the Example of Urban Road Intersections*. PhD thesis, Universität Karlsruhe (TH), 2009.

[Jä99] B. Jähne and M. Ulrich (eds.): *Handbook of Computer Vision and Applications*, vol. 2. Academic Press, San Diego, CA, 1999.

[Jan02] J. Jansson, J. Johansson and F. Gustafsson: *Decision Making for Collision Avoidance Systems*. In *Society of Automotive Engineering*, 2002.

[Jat09] J. Jatzkowski: *Kollisionswahrscheinlichkeiten aus monokularen Bildfolgen*. Master's thesis, WWU Münster, 2009.

[Joh96] N. Johnson and D. Hogg: „Learning the Distribution of Object Trajectories for Event Recognition“. *Image and Vision Computing* **14**, pp. 583–592, 1996.

[Kä10a] E. Käfer, C. Hermes, C. Wöhler, F. Kummert and H. Ritter: *Recognition and Prediction of Situations in Urban Traffic Scenarios*. In *Proceedings of the 2010 International Conference on Pattern Recognition*, pp. 4234–4237, 2010.

[Kä10b] E. Käfer, C. Hermes, C. Wöhler, H. Ritter and F. Kummert: *Recognition of Situation Classes at Road Intersections*. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 3960–3965, May 2010.

[Kae09] N. Kaempchen, B. Schiele and K. Dietmayer: „Situation Assessment of an Autonomous Emergency Brake for Arbitrary Vehicle-to-Vehicle

Collision Scenarios“. *IEEE Transactions on Intelligent Transportation Systems* **10** (4), pp. 678–687, December 2009.

[Lab05] R. Labayrade, C. Royere and D. Aubert: *A Collision Mitigation System using Laser Scanner and Stereovision Fusion and its Assessment*. In *Proceedings of the 2005 IEEE Intelligent Vehicle Symposium*, pp. 441–446, 2005.

[Lab07] R. Labayrade, C. Royere and D. Aubert: „Experimental Assessment of the RESCUE Collision-Mitigation System“. *IEEE Transactions on Vehicular Technology* **56** (1), pp. 89–102, January 2007.

[Lar04] F. Large, D. Vasquez, T. Fraichard and C. Laugier: *Avoiding Cars and Pedestrians Using Velocity Obstacles and Motion Prediction*. In *Proceedings of the 2004 IEEE Intelligent Vehicle Symposium*, pp. 375–379, 2004.

[Lat05] A. D. Lattner, J. D. Gehrke, I. J. Timm and O. Herzog: *A Knowledge-based Approach to Behavior Decision in Intelligent Vehicles*. In *Proceedings of the 2005 IEEE Intelligent Vehicles Symposium*, pp. 466–471, 2005.

[Liu08] J.-F. Liu, Y.-F. Su, M.-K. Ko and P.-N. Yu: *Development of a Vision-Based Driver Assistance System with Lane Departure Warning and Forward Collision Warning Functions*. In *Proceedings of the Digital Image Computing: Techniques and Applications*, pp. 480–485, December 2008.

[Mar08] E. Martínez, M. Díaz, J. Melenchón, J. A. Montero, I. Iriondo and J. C. Socoró: *Driving assistance system based on the detection of head-on collisions*. In *Proceedings of the 2008 IEEE Intelligent Vehicle Symposium*, pp. 913–918, 2008.

[Men04] A. Mendes and U. Nunes: *Situation-based Multi-target Detection and Tracking with Laserscanner in Outdoor Semi-structured Environment*. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 88–93, 2004.

[Met06] J. Metzler, M. Grinberg and D. Willersinn: *Assessment of the FOE for Side Impact Alert*. In *3rd International Workshop on Intelligent Transportation*, IEEE, 2006.

[MH93] R. Mock-Hecker and M. Zeller: *Recognizing hazardous situations among vehicles in a traffic worlds*. In *Proceedings of the 4th Annual Conference on AI, Simulation, and Planning in High Autonomy Systems*, pp. 205–210, 1993.

[Mie04] A. Miene, A. D. Lattner, U. Visser and O. Herzog: *Dynamic-preserving Qualitative Motion Description for Intelligent Vehicles*. In *Proceedings of the 2004 IEEE Intelligent Vehicles Symposium*, pp. 642–646, 2004.

[Mik99] R. Mikut, S. Böhlmann, B. Cuno, J. Jäkel, A. Kroll, T. Rauschenbach, B.-M. Pfeiffer and T. Slawinski: *Fuzzy-Logik und Fuzzy Control - Begriffe und Definitionen*. In *Proc. 9. Workshop "Fuzzy Control", GMA-UA 5*, 1999.

[Mis03] G. Mishne: *Source Code Retrieval using Conceptual Graphs*. Master's thesis, University of Amsterdam, 2003.

[Nag05] H.-H. Nagel and M. Arens: *'Innervation des Automobils' und Formale Logik*. In *Fahrerassistenzsysteme mit maschineller Wahrnehmung*, M. Maurer and C. Stiller (eds.), pp. 89–116, Springer Berlin Heidelberg, 2005.

[Ned08] S. Nedevschi, A. Vatavu, F. Oniga and M. M. Meinecke: *Forward Collision Detection using a Stereo Vision System*. In *Proceedings of the 4th International Conference on Intelligent Computer Communication and Processing*, pp. 115–122, 2008.

[Neu03a] B. Neumann: *Bildverstehen - ein Überblick*. In *Handbuch der künstlichen Intelligenz*, G. Görz, C.-R. Rollinger and J. Schneeberger (eds.), pp. 815–838, Oldenburg Wissenschaftsverlag, 2003.

[Neu03b] B. Neumann and T. Weiss: *Navigating through logic-based scene models for high-level scene interpretations*. In *Proceedings of the 3rd International Conference on Computer Vision Systems*, pp. 212–222, 2003.

[Neu08] B. Neumann and R. Möller: „On Scene Interpretation with Description Logics". *Image and Vision Computing* **26**, pp. 82–101, January 2008.

[Nic95] N. Nicolov, C. Mellish and G. Ritchie: *Sentence Generation from Conceptual Graphs*. In *Conceptual Structures: Applications, Implementation and Theory; Proceedings of the 3rd International Conference on*

Conceptual Structures, G. Ellis, R. Levinson, W. Rich and J. F. Sowa (eds.), vol. 954 of *Lecture Notes in Artificial Intelligence*, pp. 74–88, Springer Berlin Heidelberg, 1995.

[Off09] F. S. Office: *Statistical Yearbook 2009 For the Federal Republic of Germany*. Federal Statistical Office, Wiesbaden, Germany, 2009.

[Oli00a] N. Oliver and A. P. Pentland: *Graphical Models for Driver Behavior Recognition in a SmartCar*. In *Proceedings of the 2000 IEEE Intelligent Vehicle Symposium*, pp. 7–12, 2000.

[Oli00b] N. M. Oliver, B. Rosario and A. P. Pentland: „A Bayesian Computer Vision System for Modeling Human Interactions“. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, pp. 831–843, August 2000.

[Pel02] M. Pellkofer and E. D. Dickmanns: *Behavior Decision in Autonomous Vehicles*. In *Proceedings of the 2002 IEEE Intelligent Vehicle Symposium*, pp. 495–500, 2002.

[Pfe07] H. D. Pfeiffer and R. T. Hartley: *A Comparison of Different Conceptual Structures Projection Algorithms*. In *Conceptual Structures: Knowledge Architectures for Smart Applications; Proceedings of the 15th International Conference on Conceptual Structures*, U. Priss, S. Polovina and R. Hill (eds.), vol. 4604 of *Lecture Notes in Artificial Intelligence*, pp. 165–178, Springer Berlin Heidelberg, 2007.

[Rac01] J. Racky: *Aktive Bildverarbeitung*. PhD thesis, Universität Kaiserslautern, 2001.

[Ren06] C. Renner: *Fahrerabsichts- und situationsangepasste Warnstrategie für eine Spurverlassenswarnung*. Master's thesis, Eberhard Karls Universität Tübingen, 2006.

[Sal07] F. D. Salim, S. W. Loke, A. Rakotonirainy, B. Srinivasan and S. Krishnaswamy: *Collision Pattern Modeling and Real-Time Collision Detection at Road Intersections*. In *Proceedings of the IEEE Intelligent Transportation Systems Conference 2007*, pp. 161–166, September 2007.

[Sch08] J. Schneider, A. Wilde and K. Naab: *Probabilistic Approach for Modeling and Identifying Driving Situations*. In *Proceedings of the 2008 IEEE Intelligent Vehicle Symposium*, pp. 343–348, 2008.

[Sch10] R. Schubert, K. Schulze and G. Wanielik: „Situation Assessment for Automatic Lane-Change Maneuvers“. *IEEE Transactions on Intelligent Transportation Systems* **11** (3), pp. 607–616, September 2010.

[Son04] K.-T. Song, C.-H. Chen and C.-H. C. Huang: *Design and Experimental Study of an Ultrasonic Sensor System for Lateral Collision Avoidance at Low Speeds*. In *Proceedings of the 2004 IEEE Intelligent Vehicle Symposium*, pp. 647–652, 2004.

[Sos07] R. Sosa and G. Velazquez: *Obstacles Detection and Collision Avoidance System Developed with Virtual Models*. In *Proceedings of the 2007 IEEE International Conference on Vehicular Electronics and Safety*, pp. 1–8, 2007.

[Sow76] J. F. Sowa: „Conceptual Graphs for a Data Base Interface“. *IBM Journal of Research and Development* **20** (4), pp. 336–357, July 1976.

[Sow84] J. F. Sowa: *Conceptual Structures*. Addison-Wesley, Reading, MA, 1984.

[Ste08] C. Steger, M. Ulrich and C. Wiedemann: *Machine Vision Algorithms and Applications*. WILEY-VCH Verlag GmbH @Co. KGaA, Weinheim, 2008.

[Sun05] T.-Y. Sun, S.-J. Tsai, J.-Y. Tseng and Y.-C. Tseng: *The Study on Intelligent Vehicle Collision-Avoidance System with Vision Perception and Fuzzy Decision Making*. In *Proceedings of the 2005 IEEE Intelligent Vehicles Symposium*, pp. 112–117, 2005.

[Tak00] A. Takahashi and N. Asanuma: *Introduction of HONDA ASV-2(Advanced safety Vehicle-Phase 2)*. In *Proceedings of the 2000 IEEE Intelligent Vehicle Symposium*, pp. 694–701, 2000.

[Teu09] M. Teutsch: *Fusion von 6D-Stereo- und Radardaten zur Segmentierung und Attributierung von Objekten im Straßenverkehr*. Master's thesis, Forschungszentrum Informatik, Universität Karlsruhe, 2009.

[Tha06] D. Thammasut and O. Sornil: *A Graph-Based Information Retrieval System*. In *Proceedings of the International Symposium on Communications and Information Technologies, 2006.*, pp. 743–748, 2006.

[Tho03] R. Thomopoulos, P. Buche and O. Haemmerle: „Representation of weakly structured imprecise data for fuzzy querying“. *Fuzzy Sets and Systems* **140** (1), pp. 111–128, 2003.

[Til02] A. Tilley and H. D. Associates: *The Measure of Man and Woman: Human Factors in Design*. John Wiley & Sons, Inc., New York, USA, 2002.

[Vac07] S. Vacek, T. Gindele, J. M. Zöllner and R. Dillmann: *Situation classification for cognitive automobiles using case-based reasoning*. In *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, pp. 704–709, June 2007.

[vD05] T. van Dijck and G. A. van der Heijden: *Visionsense; An Advanced Lateral Collision Warning System*. In *Proceedings of the 2005 IEEE Intelligent Vehicle Symposium*, pp. 296–301, 2005.

[Ven05] K. Venkatesh, A. Ramesh, M. Alagusundaram and J. V. Sahana: *NEFCOP: A Neuro-Fuzzy Vehicle Collision Prediction System*. In *CIMCA '05: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 1, pp. 1079–1083, IEEE, 2005.

[Weh10] T. Wehking: *Echtzeitfähiges 6D-Stereo-Video-System als Basis für zukünftige Komfort- und Sicherheitsfunktionen in der Fahrerassistenz*. PhD thesis, Eberhad-Karls-Universität Tübingen, 2010.

[Woe04] F. Woelk, S. Gehrig and R. Koch: *A Monocular Image Based Intersection Assistant*. In *Proceedings of the Intelligent Vehicles Symposium*, pp. 286–291, IEEE, 2004.