A Real-Time Approach to Traffic Situation Representation from Image Processing Data

D. Reichardt
Daimler-Benz AG
F1M/IA
D-70546 Stuttgart
e-mail: reichard@str.daimler-benz.com

Abstract

This paper discusses an approach to provide an environment representation of real traffic scenarios. Image processing is used to get information about the vehicle's position on the road, about detected obstacles in the vicinity and about traffic signs. This incoming information is generated by camera modules which are arranged in a way that a panoramic vision is achieved.
The problem is the noisy input data from different image processing modules which makes autmated driving and adequate driver warning systems very difficult. The main task to solve this problem is to build up an always consistent situation representation in real time. Situation analysis basically consists of data fusion of different object recognition modules. An object matching algorithm has been developed that takes sensor specific deviations and errors into account. The second part is a rule set that examines the current situation for temporal or spacial inconsistencies using uncertainty representation. The resulting situation description is filtered by a set of kalman filters which also provide a single step prediction. Apart from the fusion of obstacle recognition data, the situation analysis provides information about the road and currently valid traffic signs.

1 Introduction

The described method is integrated in the Daimler-Benz demonstrator car VITA II [Ulmer 94] which was built up within the Prometheus program. This autonomous vehicle acquires its environment information by using image processing.
After having shown the general feasibility of vision based autonomous driving, the main task is to improve obstacle recognition. During the Prometheus efforts it became clear that the major challenge of the future will be the field of reliable obstacle detection. VITA II is a platform for different approaches to obstacle recognition. By fusing them, the situation analysis approach presented here, allows the Daimler-Benz demonstrator car VITA II to perform autonomous driving in real traffic.

2 The VITA II Demonstrator

VITA II is a Mercedes 500 SEL passenger car which is equipped with a number of optical sensors. These sensors are arranged in a way that VITA II can perform all around vision which is crucial for autonomous driving. There are three different kinds of recognition modules integrated in the car:

- The **road tracking** module [Behringer 94] that delivers information such as the vehicle position on the road, the curvature or the lanewidth.
- The **traffic sign recognition** module [Estable et al. 94] which delivers information regarding speed limits or overtaking restrictions.
- Four different **obstacle recognition** modules following different approaches to vision based obstacle detection.

Three main methods are used for obstacle recognition:

- The ODT (obstacle detection and tracking) module [Thomanek et al. 94] uses a bifocal camera system and an internal 4D model of the world to increase robustness of the edge based obstacle recognition. ODT is used on two camera platforms looking to the front and to the rear.
- The CT (CarTrack) module [Brauckmann et al. 94] exploits symmetry properties of vehicle views. Furthermore it uses a neural net classifier to distinguish between trucks and personal cars. CT is used on the camera platform looking to the front only.
- The VB (Vision Bumper) and the LOD (Lateral Obstacle Detection) modules [Brauckmann et al. 94] are stereo vision systems detecting any elevation from a ground plane. VB is looking to the front and LOD supervises the areas beside the vehicle.

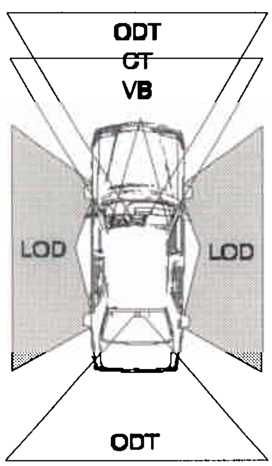

Figure 1: Obstacle Recognition in VITA II

Based on information given by the optical sensors described above, a **behaviour control** (BC) module [Reichardt & Schick 94] decides which manoeuver to perform and communicates commands to the vehicle computer which contains longitudinal and lateral controllers to activate the steering wheel, the brake and the throttle.
All the modules described above run in real time and communicate via a blackboard called dynamic database.

3 The Sensor Fusion Task

Sensor fusion means scene understanding by using information of more than one type. In the present case, the information types differ in sensor location and sensor design as well as in image processing software.
In order to construct an all around scene representation the first task is to combine sensor information of non overlapping regions. Passing cars for example are first detected by a rear sensor, reidentified by a lateral sensor and finally by a front sensor. The fusion system has to cope with blind zones and it has to identify the tracked objects.
The second task is to combine information from different sensors covering overlapping regions. In this case the fusion system has to cope with different and even contradicting measurements and has to solve the correspondence problem since the different sensors deliver multiple objects.
Finally, the scene representation constructed by the sensor fusion should be consistent in time and space. Therefore obstacle movements have to be continuous. Moreover the situation analysis module has to cope with false detections.
An approach that intends to solve the tasks described above is presented in the following.

4 The Situation Analysis System

The design of the situation analysis system follows a *blackboard* architecture [Shapiro 87]. Central part of the situation analysis system is the *scene memory*. It contains the objects describing the current traffic situation. Since there is only one road detection module, its information is just copied into the scene memory. The main task for the situation analysis system is the fusion

of obstacle recognition data. Obstacles detected by the vision modules are matched with objects in the scene memory. If there is no match, new objects are added.
Six different administration processes have access to the scene memory as shown in figure 2:

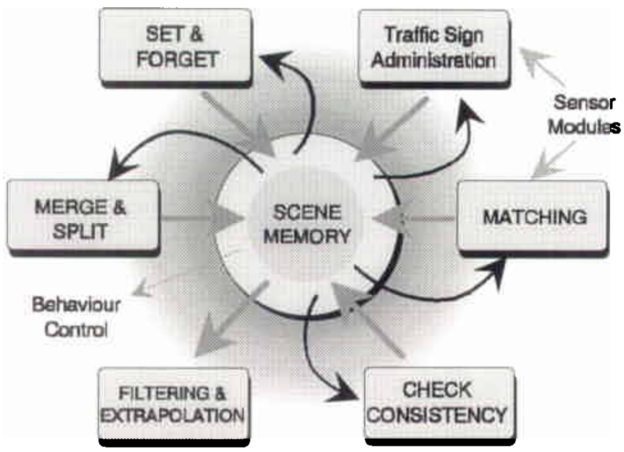

Figure 2: The Situation Analysis System

- The „*matching*" process matches new input data from obstacle recognition modules to objects in the scene memory.
- The „*set & forget*" process decides when to admit a new object and when to dismiss it.
- The „*merge & split*" process decides when to split a tracked object into two and when to merge two tracked objects in the scene memory.
- The „*filtering & extrapolation*" process uses kalman filters. It updates the filters by using the new input coming from the obstacle recognition modules and extrapolates unmatched objects.
- The „*check consistency*" process consists of a ruleset that examins the scene memory for impossible object constellations.
- The „*traffic sign administration*" process uses knowledge about traffic signs to create traffic situation objects reflecting the currently valid traffic regulations.

The administration processes are called sequentially by a simple round robin scheduler. They are designed in a condition-action format scanning the relevant part of the scene memory for objects that represent candidates for them. Since any information is stored locally, every administration process has access to the results of the others.
The entire system is running in real time. Every fourty milliseconds a scene description is passed to the behaviour control module and to the dynamic database. In this way the obstacle recognition systems have access to information where the system assumes an obstacle. This helps them to position their search windows in the detection phase.

obstacle frame		
slot		facetts
distance	m	value, default, reliability, fuzzy-demon
speed	m/s	"
acceleration	m/s^2	"
offset	m	"
lateral speed	m/s	"
confidence	-	value, default
vehicle type	{car,truck}	value, default
blind time	s	value, default
visibility	-	value, default
last sensor	-	value, default
current measurements	-	pointer to list of measuremen default (empty list)
filter	-	pointer to filter that carrie current state
age	s	value, default, start-value-demon
status	-	value, default
identifier	-	value, default

Table 1: Obstacle Representation

4.1 The Scene Representation

The elements of the scene memory are represented by frames. There are three frame classes, a road-frame, a traffic-sign-frame and an obstacle-frame. We will focus on the obstacle frame since it is the main subject of the sensor fusion concept. The road frame as well as the traffic sign frame are designed in a similar way.

Table 1 shows the frame representation of an obstacle. For each element in the scene memory the described structure is created and modified by the administration processes described in the following subsections.

4.2 The Matching Process

The matching process is responsible for the assignment of a new measurement to an existing obstacle which is known by the situation analysis module. The matching process first computes similarity values of measured obstacles and obstacles in the scene memory.

Let S be a set of sensors[1], A a set of attributes, $v\colon A \rightarrow \Re$ the value of an attribute and $r\colon S \times A \rightarrow [0,1]$ a reliability function. The similarity is then computed via a distance function $d\colon S \times S \rightarrow \Re$:

$$d\,(S_1,S_2) = \frac{\sum_{i=1}^{n} f\left(a_i, \left|v(a_{i,S1}) - v(a_{i;S2})\right|\right) \cdot r(S_1,a_i) \cdot r(S_2,a_i)}{\sum_{i=1}^{n} r(S_1,a_i) \cdot r(S_2,a_i)}$$

The distance value is influenced by the mapping $f\colon A \times \Re \rightarrow [0,1]$. The function used in the matching process is defined by the function $sim\colon S \times S \rightarrow [0,1]$:

$$sim(S_1,S_2) = 1 - \left(\frac{d(S_1,S_2)}{1 + d(S_1,S_2)}\right)^s \quad s \in \mathrm{N}^+$$

If the best similarity of a new measurement is lower than a pre-defined acceptance theshold, it is not associated to an existing obstacle but a new one is created. Otherwise it is added as a confirmative measurement in the list of current measurements of the obstacle frame with the highest similarity. For distance functions and similarity computation refer to [Reichardt 92, Weß 91].

4.3 The Set & Forget Process

The task of the set & forget process is to distinguish between relevant and irrelevant objects. It also serves as a filter for false detections.

Whenever a new obstacle frame is created by the matching process, it is set to '*candidate*' status which means, that it is not yet part of the scene that will be send to behaviour control.

An acceptance period is assigned to each sensor module together with an acceptance repetition number. The set & forget process checks if the obstacle is confirmed either by another sensor or by repeated recognitions by the same sensor within the acceptance period. It also takes the confidence slot and a visibility slot into account, which are filled by the check consistency process. The status is then set to '*confirmed*', and the obstacle is accepted to be a part of the scene. If there is no sufficient confirmation within the acceptance period, the obstacle is rejected and deleted from the scene memory. This acceptance period is not longer than a small number of video cycles. Otherwise the latency time for immediate reactions on obstacles would not be tolerable.

Once an obstacle is confirmed, its age is set to zero. If it is no longer recognized, the blind time is set to zero. If the blind time period extends a certain limit, the obstacle frame is purged or '*forgotten*'. The accepted blind time depends on the position of the obstacle, its potential risk, and the sensor that should recognize it.

4.4 The Merge & Split Process

Since sensor measurements are sometimes very noisy, it happens that two different obstacles are taken for one or vice versa. This happens especially when an obstacle recognition module mixes up features in the image.

The merge & split process checks the scene memory for obstacle frames which are very similar or even overlapping. If the similarity is higher than a merging threshold, two obstacle frames are fused to a single one.

In the opposite case, an obstacle frame is split into two if the confirming measurements of the sensor modules can be divided into two different and clearly dividable sets over a pre-defined acceptance period.

4.5 Filtering & Extrapolation Process

Before filtering, the measurement data of an obstacle frame has to be fused first.

Let $A^S = (a_0^S, \ldots, a_n^S)$ be the complete attribute vector of an obstacle description created by sensor S and $R^S = (r_0^S, \ldots, r_n^S)$ the associated reliability vector. If

[1] *In this case the internal obstacle data is also considered as sensor information.*

module S does not provide the attribute a_i^S , then r_i^S is set to zero.

The fusion task is obviously easy for disjunct attibute subsets since they are fused by taking data from the providing module. For example, the CT module delivers a classification of the vehicle. This information is not provided by the other modules, so the information is copied to the fusion result.

The situation analysis module has three different modes to perform sensor fusion. The first mode is a simple selection of the sensor information which is assumed to be the best. This assumption is built by an off-line examination of the different sensors and by a reliability value given by the sensor. The additional information is not used in this mode.

The second mode is a classical fusion mode. The fused obstacle is created by a weighted sum of the attribute lists:

$$a_i = \frac{\sum_{j=1}^{|S|} \left(a^{S_j} \cdot r_i^{S_j} \right)}{\sum_{j=1}^{|S|} r_i^{S_j}}$$

These two methods are already integrated in the VITA II demonstrator and verified by a large number of test drives in real traffic.

A third mode which uses uncertainty representation is in a simulation testing and evaluation phase. In this approach each detected obstacle is represented by a position cloud which reflects the estimated position together with a sensor specific assumed error. These clouds are merged similar to the fuzzy inference rule [von Altrock 91]. The result is a new cloud which is then retransformed into a single value. To have access to the result of the merging, the distance slot and the offset slot contain fuzzy-demons which deliver this result if needed. No matter which method is used, the result is a single obstacle that represents the current measurements of the different sensor modules.

The next step is the filtering of the obstacle information. This task is fulfilled by a kalman filter which is associated to the obstacle frame. If the list of current measurements is empty, the filter is not updated and the obstacle information is extrapolated. If there is at least one measurement that confirms the obstacle, a fusion result is achieved as described above which is used as an update for the kalman filter.

4.6 The Check Consistency Process

The check consistency process evaluates constellations in the scene memory. Especially candidate obstacle frames are inspected.

The process uses a ruleset which computes confidence values for current measurements. If a sensed obstacle is overlapped by another obstacle in the scene, the confidence in the measurement is set to a low value. If it is completely covered by another obstacle, its detection is impossible, and so the confidence value for the measurement is set to zero. Obviously the conclusion could be drawn in favour of the other obstacle, too. If the systems has more confidence in the overlapped than in the overlapping obstacle, it can also reject the latter.

To deal with this problem, the obstacle frame has a confidence slot which is administrated by the check consistency process.

The confidence value is dependant on the visibility of an obstacle, the overlapping of two or more obstacles, and contextual information such as freeway exits where the other cars may go off and disappear from the field of view.

The second value which is set by the check consistency process is the visibility of an obstacle. This is used for the computation of the confidence value. It is also necessary for the set & forget process. If an obstacle is not confirmed by any sensor, the reason can be, that it is hidden by another obstacle in the scene. Therefore it should not be removed from the scene memory.

4.7 Traffic Sign Administration

The traffic sign recognition system delivers a code, an additional information and a confidence value for each traffic sign found in the current image. The code is chosen in accordance with the german traffic sign codes [HAV87].

Traffic signs are either valid from the moment they are detected until their validity is cancelled by another traffic sign, or they are valid for a certain distance.

The first case is handled by a dependency network which is evaluated each time a new traffic sign is detected. For the second case the situation analysis module computes the distance from the speed of the autonomous vehicle and removes the corresponding traffic regulation object from the scene memory when the valid zone is passed. For more information refer to [Gachelin 94].

5 Conclusion

A situation analysis module with a frame based situation representation has been developed. Its functionality has been tested successfully in a simulation tool and on the Daimler-Benz demonstrator vehicle VITA II. It runs on a single T805 Transputer in real time and serves as a real world filter to the higher level planning and decision module in VITA II.
Future work will concentrate on a fuzzy set based sensor fusion and the design of a fuzzy interface to the behaviour control module.

6 References

[Behringer 94] R. Behringer, "Road Recognition from Multifocal Vision", in: Proceedings of the Intelligent Vehicles Symposium, Paris 1994

[Brauckmann et al. 94] M.E. Brauckmann, C. Goerick, J. Groß, T. Zielke, "Towards All Around Automatic Visual Obstacle Sensing for Cars", in: Proceedings of the Intelligent Vehicles Symposium, Paris 1994

[Estable et al. 94] S. Estable, J. Schick, F. Stein, R. Janssen, R. Ott, W. Ritter, Y.-J. Zheng, "A Real-Time Traffic Sign Recognition System", in: Proceedings of the Intelligent Vehicles Symposium, Paris 1994

[Gachelin 94] O. Gachelin, "Implantation sur Transputers d'un systeme de guidage pour vehicule autonome et intégration des panneaux de signalisation routière", Mémoire en vue de l'obtention du titre d'ingénieur, C.U.S.T Université Blaise Pascal, Clermont Ferrand 2, ets: Daimler-Benz AG, 1994

[HAV87] „Hinweise für das Anbringen von Verkehrszeichen und Verkehrseinrichtungen", Kirschbaum Verlag 1987

[Reichardt 92] D. Reichardt, „Ähnlichkeitsbasierte Verkehrszeichenerkennung", Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, durchgeführt bei der Daimler-Benz AG, Stuttgart, Forschungsinstitut Mercedes-Benz, F1M/IA Fahrerassistenzsysteme, 1992

[Reichardt & Schick 94] D. Reichardt, J. Schick, "Collision Avoidance in Dynamic Environments Applied to Autonomous Vehicle Guidance on the Motorway", in: Proceedings of the Intelligent Vehicles Symposium, Paris 1994

[Shapiro 87] S.C. Shapiro, „Encyclopedia of Artificial Intelligence", Wiley-Interscience Publications 1987

[Thomanek et al. 94] F. Thomanek, E.D. Dickmanns, D. Dickmanns, "Multiple Object Recognition and Scene Interpretation for Autonomous Road Vehicle Guidance", in: Proceedings of the Intelligent Vehicles Sysmposium, Paris 1994

[Ulmer 94] B. Ulmer, "VITA II - Active Collision Avoidance in Real Traffic", in: Proceedings of the Intelligent Vehicles Symposium, Paris 1994

[von Altrock 91] C. von Altrock, „Über den Daumen gepeilt - Fuzzy Logic: scharfe Theorie der unscharfen Mengen", in: C't Magazin für Computer Technik, März 1991

[Weß 91] S. Weß, „PATDEX/2 - ein System zum adaptiven, fallfokussierten Lernen in technischen Diagnosesituationen, SEKI Working Paper 91-01 (SFB), Universität Kaiserslautern, 1991