Intelligent Robots and Systems. Elsevier, 1994.

Visual Recognition of Obstacles on Roads

Uwe Regensburger and Volker Graefe

Institut für Meßtechnik
Universität der Bundeswehr München
Neubiberg, Germany
Fax (+49 89) 6004-3074

As part of a driver support system for motor vehicles on freeways an obstacle recognition system was developed within the EUREKA project PROMETHEUS over the last four years. The obstacle recognition system uses a single monochrome TV camera and a multi-processor robot vision system. The recognition system includes a road tracker and an obstacle detector that were described in previous publications, and an object classification and tracking module presented here. The module is based on generic 2-D object models. It recognizes obstacles in real time and from distances of 200 to 300 m, sufficient for high-speed driving. The module was extensively tested in real-world scenes on the German Autobahn and on various other roads, including city streets with heavy traffic.

1. INTRODUCTION

A driver support system for road vehicles is a great challenge for the technology of machine vision. The goal is to relieve the driver from monotonous tasks, to warn him of imminent danger (for instance, if he is about to fall asleep), and – possibly – to assume control of the vehicle in precarious situations. The necessary technology is being developed in numerous laboratories all over the world; there is hope that it will eventually lead to a significantly increased traffic safety [Masaki 1992]. Among the characteristics demanded of such a system are high reliability, low costs, and independence of any special infrastructure.

Within the European research project PROMETHEUS the Institute of Measurement Science has developed real-time vision technology that may be used for a driver support system [Graefe 1993]. Freeways were chosen as the principal domain for testing and demonstrating the visual recognition of objects that are relevant for the understanding of traffic situations. The reason for choosing freeways, in spite of the typically high driving speeds, is that the complexity of the traffic situations and the variety of objects are much lower on freeways than on other roads.

Obstacle recognition is an essential subtask of a driver support system. (In the sequel "obstacle" refers to any visible object on the road in front of the ego vehicle that is, or could possibly become, a danger for the ego vehicle.) According to a concept for obstacle recognition proposed by [Graefe 1989] the task should be decomposed into several largely independent subtasks, or modules, including the initial detection of possible obstacle candidates, the classification of obstacle candidates into false alarms (caused e.g. by shadows), and real physical objects, the tracking of obstacles, etc.. A module for detecting obstacles in distances adequate for high-speed driving was introduced by [Solder 1992] and [Solder, Graefe 1993], and road modules by [Tsinas, Graefe 1992] and [Wershofen 1992].

A multitude of methods for recognizing obstacles in monocular image sequences has been described in the literature. Essentially three different classes of methods have been used. Gestalt approaches, e.g. [Davies 1990], aim at a direct reconstruction of the 3-D world as a basis for object recognition. Other approaches, e.g. [Enkelmann 1990], [Carlsson, Eklundh 1990] and [Young, G. et al. 1992] evaluate displacement vector fields and the differences in image motion between points within the road surface and points above the road surface. Due to the large amounts of computation required by these methods, they have not yet been demonstrated in real time and, thus, do not appear suitable for vehicle applications. The third class of methods is based on object models. According to this approach objects are recognized by certain combinations of visible features. This may be accomplished in a number of different ways.

As one example, [Mori et al. 1989] and [Raboisson, Even 1990] first segment the image. Irregularities within the road segment are then taken as indications for objects. Additional criteria (edges, color, etc.) are used to minimize the frequency of false detections, while additional sensors, such as laser range finders [Alizon et al. 1990] or radars [Young, E. et al. 1992], may be used to measure the distance from an object.

[Korn 1989] bases the tracking of vehicles on histograms of edge orientations. [Seitz et al. 1991] use a hierarchically structured multi-layer representation of edge directions for building object models and recognizing objects in images. [Hartmann, Mertsching 1992] describe object images by a "hierarchical structure code."

[Kühnle 1991] and [Zielke et al. 1992] track vehicles on the basis of edges and symmetry. [Graefe, Jacobs 1991], [Efenberger et al. 1992], [Solder 1992], and [Solder, Graefe 1993] detect and track vehicles and obstacles on roads by comparing generic 2-D object models with edges detected in image sequences. [Thomanek, Dickmanns 1992] and [Schmid, Thomanek 1993] use spatio-temporal models for tracking objects.

2. THE CONCEPT OF OBSTACLE RECOGNITION

An object-oriented concept for dynamic robot vision has been proposed by [Graefe 1989; 1992]. It is based on the notion that the world is structured, rather than monolithic, and consists of individual objects. Only a few physical objects in a robot's environment affect the control of the robot in any instant.

Following this concept, the task of object recognition should, therefore, be structured according to the relevant objects. This is done by associating one individual object process within the vision system with each relevant external object. The following object processes are needed for recognizing obstacles on roads (Figure 1):

- Recognition and tracking of the road (because only objects that are actually sizuated on the road are relevant as obstacles)
- Detection of obstacle candidates (the corresponding object is either a hypothetical object that has not yet been found in the image, or an object that has just been detected)
- Tracking of an obstacle; multiple copies of this process may exist in the system to allow multiple obstacles to be tracked simultaneously.

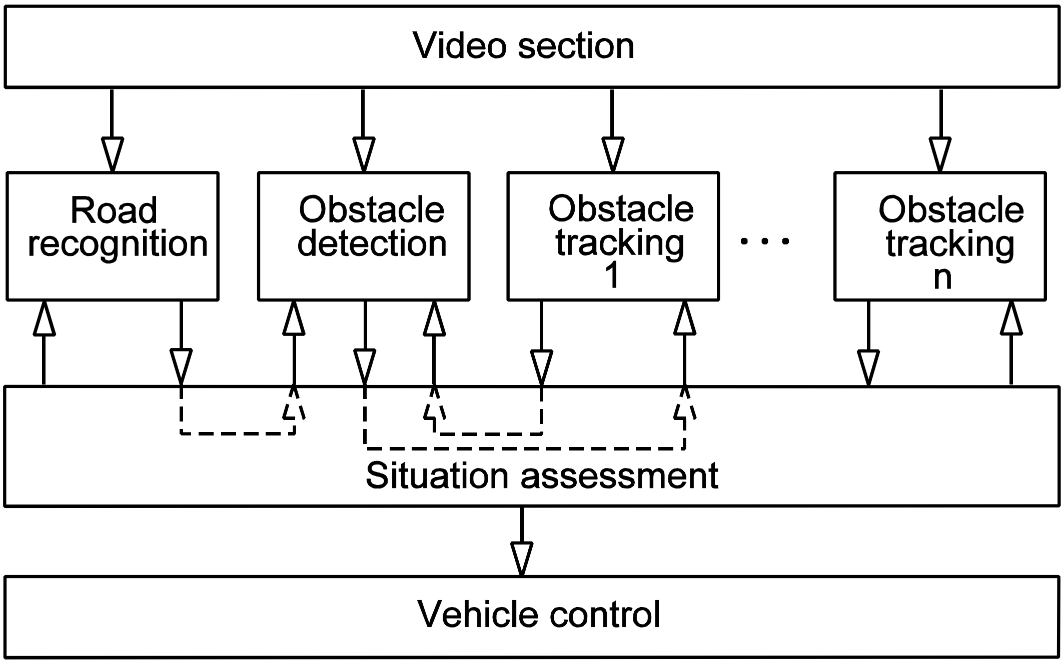

Figure 1
Structure of an object-oriented obstacle recognition system

Each object process receives image data (pixels) from the video section of the vision system and continuously generates a description of its assigned object. All object descriptions are evaluated by the situation assessment module and combined into a situation description. Depending on the perceived situation, vehicle control commands or messages for the human operator may be issued.

Step by step the information contained in the pixels is transformed into more and more abstract forms. Starting with a large number of pixels a chain of data fusion processes eventually leads to a description of an object or a situation. At each step specific knowledge in the form of appropriate models is introduced into the process (Table 1).

At the first step, the feature extraction, groups of pixels are combined into features, such as edges or corners. Knowledge about characteristics of the camera and of the features being searched is used in this process. Feature extraction may be supported by the next higher level in the processing hierarchy, the 2-D object level.

Here, knowledge about the visual appearance of specific physical objects is used to generate hypotheses regarding the nature of the object being seen and expectations as to which of its features should be visible, and where in the image they should show up. Feature extraction is thus not a form of schematic image processing, but part of a model- and knowledge-based hypothesis verification or falsification process; its task is to provide answers to very specific questions as to the presence and location of certain expected features in the image.

A very robust image interpretation may be realized by such an interaction between the feature extraction and the 2-D object levels. It is possible to add additional 3-D or 4-D (spatio-temporal) levels if a description of the 3-D shape of physical objects, or of their motions, is desired.

Table 1
Levels of Modeling and Data Fusion

Model Level	Data Fusion Performed at that Level
feature	groups of pixels → elementary features → complex features
2-D object	groups of features → 2-D objects
4-D object	2-D objects, time, sensor data → 4-D objects
environment and situation	robot state, external objects → situation

3. THE FALSE-ALARM PROBLEM

For reasons of safety an obstacle detection system must be perfectly reliable in the sense that it never fails to detect any obstacle that is actually present. Moreover, in order to allow a timely reaction, obstacles must be detected while they are still far away. These requirements make it practically unavoidable to generate false alarms occasionally, caused, for instance, by shadows, dirt, markings or patches on the road surface.

Although false alarms are not directly dangerous they must be processed and, thus, constitute an unnecessary load for the vision system. To minimize this load, false alarms should be identified and discarded as quickly as possible.

Distinguishing between actual obstacles and shadows is easy in many cases, but sometimes it can be extremely difficult. We use a set of heuristics to reject most shadows within the detector module as quickly as possible. Therefore, all alarms reported by the detection module are classified by matching the visible features that have been detected in the image against generic 2-D models of obstacles. If a fairly good match is possible, the obstacle candidate is accepted as an actual obstacle. This decision is final.

If, on the other hand, the detected features do not match the expectations derived from the model, the obstacle candidate is considered a possible false alarm. A final rejection, however, does not occur at this point. Rather, the detector is requested to report the same obstacle candidate again if it can be detected in one of the subsequent images. Due to the motion of the vehicle the subsequent images will differ from the previous ones in respect of perspective, specular reflections, occlusions, etc., and, accordingly provide an independent chance to recognize an actual obstacle if there is one. A final rejection thus occurs only if an obstacle candidate has been classified many times and has not been accepted even once.

If the time required for one cycle of detection and rejection is short, many independent images may be analyzed while approaching an obstacle candidate. The probability of rejecting an actual obstacle is then very small. On the other hand, it is possible that once in a while false alarms are accepted as an obstacle. They must be rejected during tracking.

4. GENERIC 2-D OBJECT MODELS

4.1 Effects of Distance on the Visual Appearance

As Figure 2 shows, the visual appearance of an object depends strongly on the distance from which it is seen. Moreover, for a given camera and object the area of the object's image is inversely proportional to the square of the distance.

Images of nearby objects are clear with large contrast and many visible details. Many features may, therefore, be detected in such an image. A lateral motion of the object causes a significant motion of its image.

Images of distant objects have a low contrast and do not display much internal structure. Their contours tend to be somewhat blurred. Because of their small size and low contrast hardly any internal features can be detected. Rotation of the camera is the dominating cause for motions of objects in the image.

In an intermediate range of distances the contours of objects tend to be clearly visible, but the object images do not have much internal structure. Motions of the camera and of the objects have similar effects on the motions of the image.

Because of these differences specific models are used to model the appearance of physical objects, depending on their distance. The models include a number of free parameters in order to let them match objects of different shapes. Therefore, they are called generic object models.

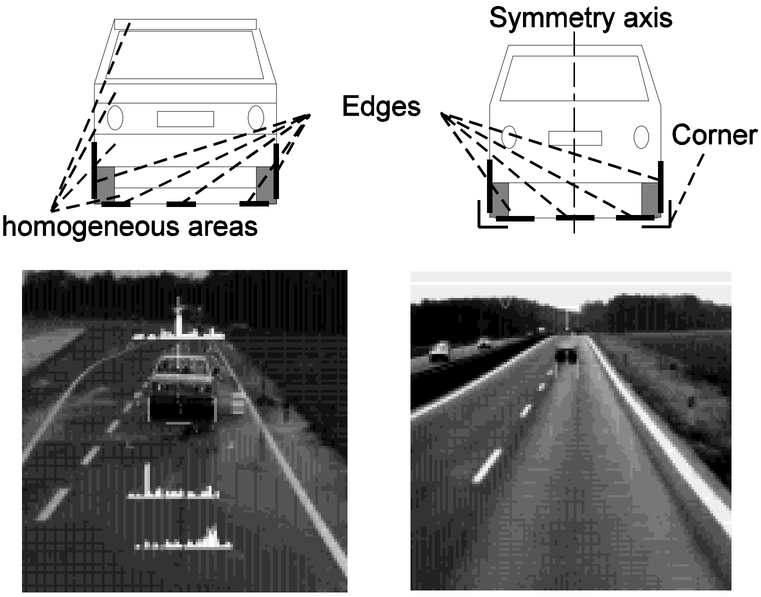

Figure 2
Different 2-D object models (top) take into account the significant differences in the appearance of nearby (left) and far away (right) objects.

4.2 The 2-D Model for the Nearby-range

Most of the obstacles that exist on freeways display a nearly rectangular contour with vertical and horizontal sides when seen from an approaching vehicle. Therefore, parts of rectangular contours are searched in the image.

Characteristic points on the contour that may be used for an object description are the corners or the centers of the sides of the rectangle. The upper part of an obstacle is often seen before a structured background, such as trees, mountains, or other vehicles; this makes it difficult to locate the upper part of the contour in the image. Usually numerous horizontal edges are visible, and it is not easy to determine which one is the upper edge of an obstacle. The lower part of the contour is usually seen before the surface of the road and may, therefore, be detected and tracked more easily. Recognizing only the left, right, and lower edge is sufficient for tracking the object in the image and for measuring its distance. Therefore, only these edges are looked for in the image.

Nevertheless, the large number of features that are often visible in the images of nearby objects makes it difficult to single out those belonging to the contour of the object. A statistical approach was chosen to solve this problem.

The image of a typical object contains a number of homogeneous areas, delimited by approximately vertical edges. Many of these homogeneous areas begin, or end, near the left and right contours of the object. In an area of interest that encloses the image of the object candidate, the left and right ends of homogeneous areas, and their centers, are counted in each image row. Local maxima of the distribution of these features indicate a possible location of the object contour and its center line. Combining these yields strong expectations where the contours of the object should be found in the image. Vertical edge detectors based on controlled correlation [Kuhnert 1986] are then applied in the vicinity of the expected contours to find the exact locations of the contours.

Generally, the lower edge of an object is distinctly visible. In case of vehicles the lower edge of the shadow underneath the vehicle is taken as the lower edge of the object. It may usually be detected reliably and robustly because the shadow appears much darker than the pavement.

4.3 The 2-D Model for Distant objects

Images of distant objects are typically small and of low contrast (Figure 2). Only few features may be extracted; therefore, a statistical approach is not feasible for recognizing such objects. The left, right, and lower edges are detected directly by applying suitable edge detectors.

Due to the large image motions induced by unavoidable camera motions, the tracking of distant objects would not be robust if it were based only upon a few local edge elements. Using additional features, such as corners and symmetry, improves the robustness. In contrast to the near range where, due to the high spatial resolution of the image, object contours rarely contain any sharp corners, corners may be detected robustly in the far range. Specific corner masks are used for detecting the lower left and right corners of objects by the method of controlled correlation.

The images of many distant objects, including the rear-views of vehicles, have a vertical symmetry axis. Symmetry is, therefore, included as an additional feature in the object model. In the near range symmetry would be difficult to detect because of the many visible details, and finding a symmetry axis would be time-consuming due to the large areas in the image that must be processed. In the far range, where object images are small and do not contain many features, symmetry detection is fast and robust.

5. DETECTING VERTICAL SYMMETRY AXES Numerous methods for detecting and exploiting symmetry have been proposed. If object contours have been determined and converted into B-splines, symmetries may be extracted according to [Saint-Marc, Medioni 1990]. A method for detecting symmetries of polyhedrical objects has been described by [Jiang, Bunke 1990]. Shape recognition based on symmetry was introduced by [Yuen 1990]. [Friedberg 1986]

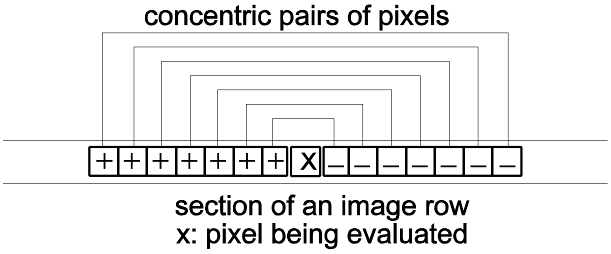

Figure 3
Pairs of pixels, centered around a pixel, x, are evaluated. For each pair the difference of the grey-levels is computed. The absolute values of the differences are added up. Their sum is the symmetry function at the center pixel x.

determined symmetry by computing a moment matrix. A known symmetrical object may be detected by convolving its image with a mirror image of a model; after detection it may be localized by convolving it with its own mirror image.

All these methods presuppose a good segmentation of the image (e.g. into object and background) or have been tested only on synthetic images or line drawings. Moreover, they require computing times of tens of seconds.

Methods that are suitable for real-time and real-world applications have also been proposed. [Zielke et al. 1992] determine for each pixel a symmetry measure, based on the even and the odd part of the symmetry function. A symmetry-based statistical approach has been used by [Kühnle 1991] to determine the center of the image of a car as seen from the rear. Histograms of symmetrical contour pixels, of symmetrical grey-level pixels, and of the lengths of horizontal edges are computed; local maxima in all three histograms indicate candidates for a vertical symmetry axis [Marola 1989].

In the sequel a method is described that detects individual symmetry points and then determines image columns containing a maximum number of symmetry points. For finding a symmetry point the even part of the symmetry function is determined. To accomplish this the similarity of grey-levels of pairs of pixels, located symmetrically to the left and to the right of a point in question, is studied (Figure 3). The absolute values of the differences of the grey-levels of each pair of pixels are added up and assigned to the pixel in question. A symmetry function is obtained by repeating this procedure for all pixels within an area of interest.

Local minima within a row of the symmetry function indicate symmetry points; in Figure 4 the symmetry points have been marked. Figure 5 shows the corresponding symmetry function, computed over a 40 pixels by 40 pixels area of

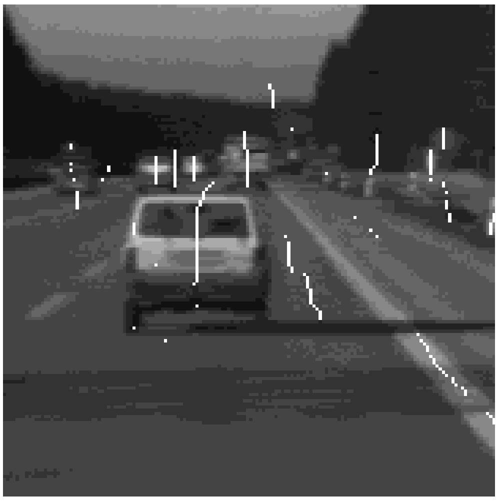

Figure 4
Symmetry points determined from 12-pixel wide environments

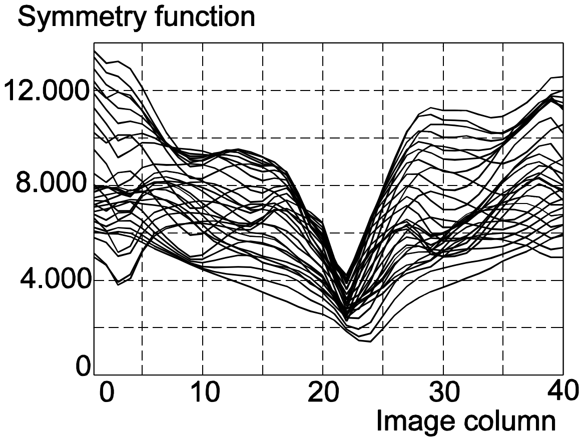

Figure 5
The symmetry function in an area of interest covering the car in Figure 4; each graph corresponds to one image row.

interest around the vehicle.

When symmetry points are searched, the width of the environment upon which the symmetry function is based should be somewhat larger than the width of the object. However, if the environment is chosen too large, many background pixels are included in the computation. A strongly asymmetrical background may then severely distort the symmetry function.

6. EXPERIMENTAL RESULTS

6.1 Test Environment

A real-time vision system BVV 3 [Graefe 1990] was used for testing the obstacle recognition system. Each of the 3 sub-processes, road recognition, obstacle detection, and obstacle tracking, was implemented on a separate parallel processor within the BVV 3.

The obstacle recognition system was designed to cooperate closely with other modules of a driver support system, for instance modules for modeling the dynamics of the recognized obstacles. Such a cooperation has been tested earlier and described by [Dickmanns, Christians 1989] and [Regensburger, Graefe 1990]. The tests reported in the sequel were, however, performed without any additional modules. The results may, therefore, be directly attributed to the modules described here. The situation assessment module (cf. Figure 1) was replaced by a message passing system that allowed communication between the recognition modules, but did not perform any additional processing.

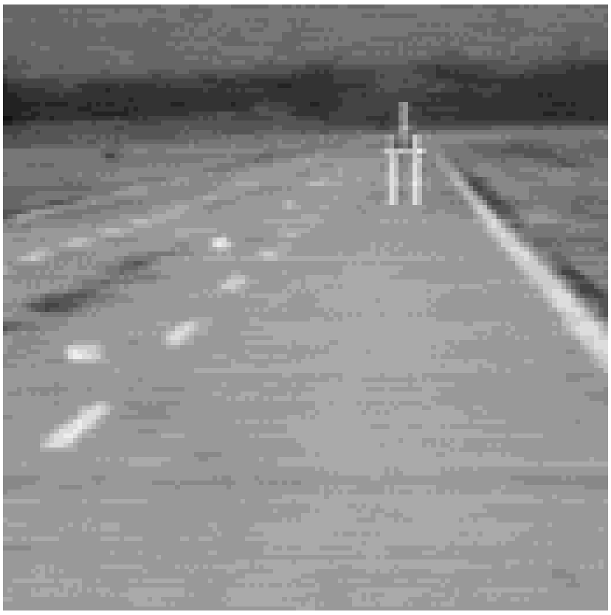

Figure 6
Passenger car seen from a distance of 300 m through the camera of the vision system, with overlaid markings showing the detected contour and symmetry axis

6.2 Recognition Distance

The ability to recognize obstacles from a great distance is of critical importance for driving at high speed. Experiments have shown that cars (moving and resting) can be tracked in distances of about 200 m; under cooperative conditions, e.g. high contrast and little camera motion, cars could be tracked in distances of over 300 m. As Figure 6 shows, a car in a distance of 300 m appears rather small in the camera image.

The obstacle tracking module does not include any distance measurement. The distance from the obstacle was, therefore, measured by external means. Visible markers were placed on the pavement in known intervals. As this was impossible on a public road, the runway of a closed airfield was used for the experiments to measure the maximum recognition and tracking range.

6.3 Robustness Although the obstacle recognition system was designed for freeways, due to the model-based tracking module it proved to be so robust that it was also tested on ordinary highways and furthermore in heavy city traffic. There, other cars, as well as the own car, changed lanes frequently causing time-varying occlusions, and together with structured backgrounds and markings on the pavement generated a high degree of visual complexity. Even in such difficult scenes (Figure 7) vehicles could be tracked reliably over extended periods of time.

The 2-D object models employed by the recognition system assume a nearly rectangular object contour and they have been optimized especially for cars and trucks. Nevertheless, it could be shown that various other objects that may occasionally be seen on roads, such as boxes, barrels and, in some cases, persons, may also be classified and tracked. However, tracking such objects that do not match the generic models well is less reliable, especially under conditions of poor lighting or when they are far away and their images are small. Increasing the repertoire of different object models in the system should further improve the system performance in this respect.

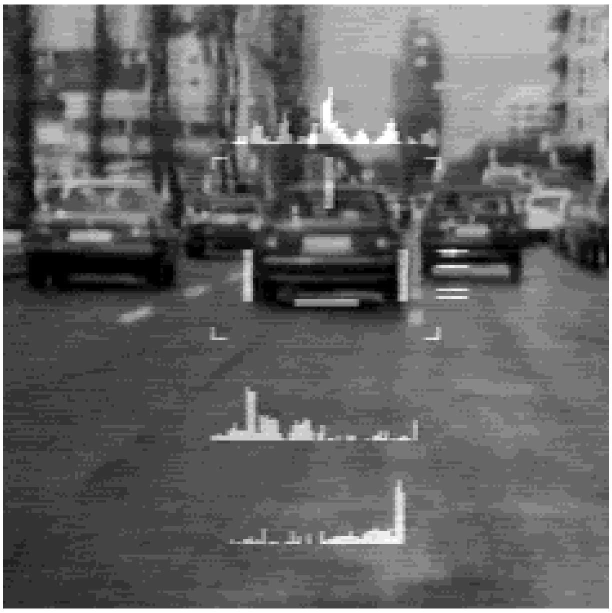

Figure 7
Tracking a vehicle in heavy city traffic

6.3 Speed

The computing time required by a module of a real-time vision system is always of prime importance. The cycle time of the object classifier and tracker was, therefore, measured in a number of experiments. One parallel processor of the robot vision system BVV 3 using an Intel 386/20-MHz CPU, augmented by a special co-processor for fast feature extraction [Graefe, Fleder 1991], was used as a hardware platform in the experiments. The basic cycle time of the vision system is the image field period of the camera’s video signal, i.e. 20 ms. Thus, all cycle times are integer multiples of 20 ms.

Tracking a distant object requires always less than 20 ms of computing time; the cycle time is in this case 20 ms.

The time required for tracking a nearby object depends on the size of its image (see Table 2). On freeways the cycle time is usually between 40 and 60 ms. Only when the distance to the preceding vehicle is exceptionally short, e.g. in stop-and-go traffic, a cycle time of 80 ms may be observed. In such situations it would be preferable anyway to evaluate the image of a wide-angle camera instead of the telescopic image used, because the lower edge of the vehicle may not even be visible if a vehicle in such a short distance is imaged by the telescopic camera.

Table 2

Cycle times for tracking nearby-objects

Object width	Cycle time
< 45 pixels	40 ms
45 – 90 pixels	60 ms
90 – 120 pixels	80 ms

6.4 Reliability of Classification

Parameters such as scene complexity or image quality that are essential for judging the performance of the tracking module are difficult to define quantitatively. Therefore, long image sequences covering a wide variety of situations were evaluated. A 20-min run on a freeway may be used as an example. It included 60,000 images processed in real time without any pre-selection.

Goal of the classification of obstacle candidates is the discrimination between false alarms and actual obstacles. It should be remembered that primarily it must be avoided to ever mistakenly reject any actual obstacle as a false alarm. (The other type of misclassification, accepting a shadow as an obstacle, is much less dangerous.) Table 3 shows the results obtained during the test.

All objects that were reported by the detector were correctly classified, and only a very small fraction of the system resources were wasted tracking false alarms.

Table 3

Results of object classification during a 20-min run on a freeway

Number of actual objects misclassified as false alarms		0
Percentage of false alarms that were correctly rejected	immediately	67%
	within a few seconds of tracking	33%
Number of final misclassifications		0
Fraction of time spent tracking false alarms		1.3%

6.5 Quality of Tracking

Because the true motions of the objects tracked in natural scenes were not measured, it is difficult to determine the quality of the tracking, or the accuracy of the localization of object features, quantitatively. Figures 8 and 9 are intended to give at least a qualitative impression of the quality of tracking.

The ranges of applicability of the two models, for the near range and for the far range, overlap to a large extent. This allows a direct comparison between the two models by using an object in the overlap range. In each of the Figures, 8 and 9, the left graphs show the image coordinates of the two sides of a car driving in front of the vehicle with the camera. The right graphs show the width of the car's image in pixels. The scene was recorded on video tape while driving in a car, and then evaluated in real time in the laboratory, using the methods for the near range (Figure 8) and for the far range (Figure 9) on the same analog data.

Both methods yield nearly identical results. The method for the far range yields slightly smoother curves (in both cases raw, unfiltered data are shown). This is probably due to the robust localization of the lower corners. Altogether both curves indicate that the unfiltered measured data are consistent and free from significant outliers, in spite of the image deterioration caused by the video tape recorder. This may be taken as evidence for a robust tracking.

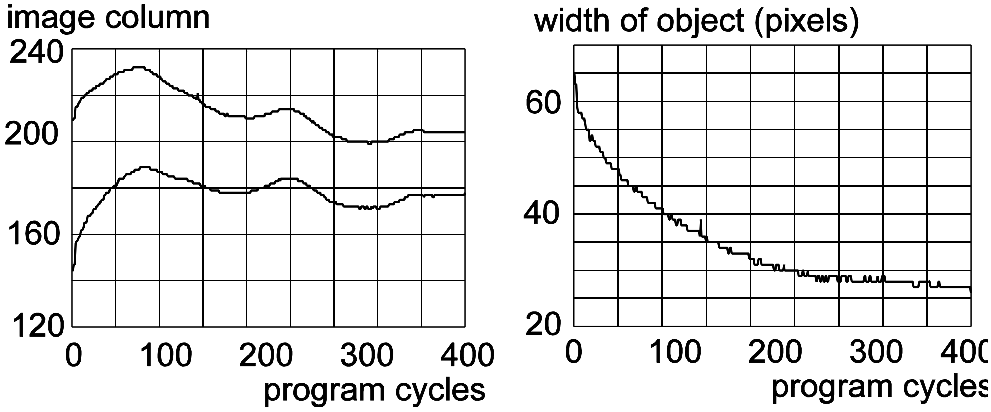

Figure 8
Results of tracking a car in an intermediate distance range using the **model for the far range** (1 program cycle is equivalent to 20 ms)

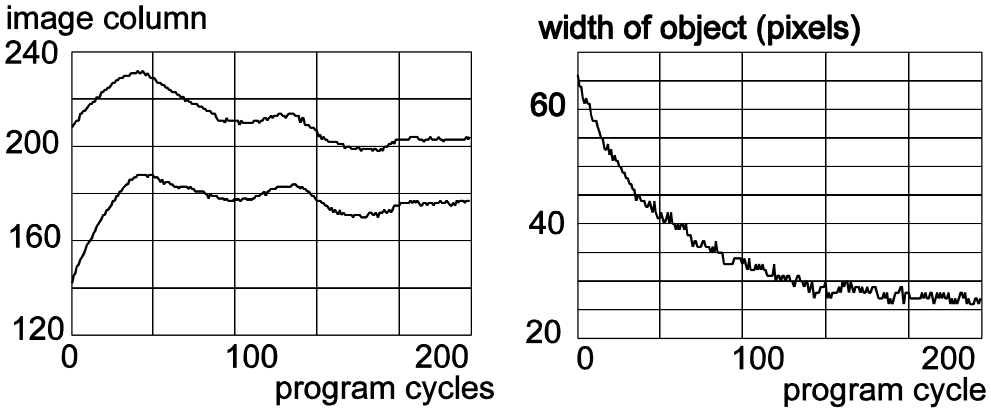

Figure 9
Results of tracking a car in an intermediate distance range using the **model for the near range** (1 program cycle is, on average, approximately equivalent to 40 ms)

7. SUMMARY

As an essential part of a driver support system for road vehicles a module for the robust classification and tracking of objects on roads has been realized. Obstacles are recognized in real time and from great distances (200 to 300 m) as required when driving at high speed on freeways. The obstacle recognition module was originally designed for the relatively simple scenes on freeways, but it has proven sufficiently robust to allow the tracking of moving vehicles even in heavy city traffic.

The module is based on generic 2-D models optimized for the rear views of cars and trucks. However, the generic models are sufficiently general to also allow the tracking and classification of various other objects that may occasionally be encountered on roads. For a truly reliable recognition of arbitrary objects the repertoire of internal object models should be expanded.

ACKNOWLEDGEMENT

Part of the work reported has been performed with support from the Ministry of Research and Technology (BMFT) and from the Daimler-Benz AG within the EUREKA project PROMETHEUS.

REFERENCES

Alizon, J.; Gallice, J.; Trassoudaine, L.; Treuillet, S. (1990): Multi-sensory data fusion for obstacle detection and tracking on motorway. In Monique Thonnat (Ed.): Proceedings of the Workshop on Vision (Prometheus, Pro-Art). Sophia Antipolis, pp 179-187.

Carlsson, S.; Eklundh, J.-O. (1990): Object detection using model based prediction and motion parallax. In Monique Thonnat (Ed.): Proc. of the Workshop on Vision (Prometheus, Pro-Art). Sophia Antipolis, pp 139-145.

Davies, E.R. (1990): Machine Vision: Theory, Algorithms, Practicalities. London: Academic Press, 1990.

Dickmanns, E.D.; Christians, Th. (1989): Relative 3-D State Estimation for Autonomous Visual Guidance of Road Vehicles. Proceedings, Intelligent Autonomous Systems. Amsterdam.

Efenberger, W.; Ta, Q.-H.; Tsinas L.; Graefe V. (1992): Automatic Recognition of Vehicles Approaching from Behind. Proceedings of the IEEE Intelligent Vehicles '92 Symposium. Detroit, pp 57-62.

Enkelmann, W. (1990): Obstacle detection by evaluation of optical flow fields. In Monique Thonnat (Ed.): Proceedings of the Workshop on Vision (Prometheus, Pro-Art), Sophia Antipolis, pp 146-166.

Friedberg, S.A. (1986): Finding Axes of Skewed Symmetry. Computer Vision, Graphics, and Image Processing, **34**, Nr. 2. San Diego: Academic Press, pp 138-155.

Graefe, V. (1989): Dynamic Vision Systems for Autonomous Mobile Robots. Proceedings, IEEE/RSJ International Workshop on Intelligent Robots and Systems, IROS '89. Tsukuba, pp 12-23.

Graefe, V. (1990): The BVV-Family of Robot Vision Systems. In O. Kaynak (ed.): Proceedings of the IEEE Workshop on Intelligent Motion Control, Istanbul, pp IP55-IP65.

Graefe, V. (1992): Vision for Autonomous Mobile Robots. Proceedings, IEEE Workshop on Advanced Motion Control. Nagoya, pp 57-64.

Graefe, V. (1993): Vision for Intelligent Road Vehicles. Proceedings, IEEE Symposium on Intelligent Vehicles. Tokyo, pp 135-140.

Graefe, V.; Fleder, K. (1991): A Powerful and Flexible Co-processor for Feature Extraction in a Robot Vision System. International Conference on Industrial Electronics, Control, Instrumentation and Automation (IECON '91). Kobe, pp 2019-2024.

Graefe, V.; Jacobs, U. (1991): Detection of Passing Vehicles by a Robot Car Driver. IEEE/RSJ International Workshop on Intelligent Robots and Systems, IROS '91. Osaka, pp 391-396.

Hartmann, G.; Mertsching, B. (1992): A Hierarchical Vision System. Proceedings, IEEE Intelligent Vehicles '92 Symposium. Detroit, pp 18-23.

Jiang, X.Y.; Bunke, H. (1990): Detektion von Symmetrien polyedrischer Objekte. In Großkopf (Hrsg.): Mustererkennung 1990, 12. DAGM-Symposium, Oberkochen-Aalen, Informatik-Fachberichte 254. Berlin: Springer, pp 225-231.

Korn, A. (1989): Zur Erkennung von Bildstrukturen durch Analyse der Richtungen des Grauwertgradienten. In Burkhardt et al. (Hrsg.): Mustererkennung 1989, 11. DAGM-Symposium, Hamburg, Informatik-Fachberichte 219. Berlin: Springer, pp 507-511.

Kühnle, A. (1991): Symmetry-based recognition of vehicle rears. Pattern Recognition Letters, Volume 12, Number 4. North-Holland, pp 249-258.

Kuhnert, K.-D. (1986): A Model-driven Image Analysis System for Vehicle Guidance in Real Time. Proceedings of the Second International Electronic Image Week. CESTA, Nice, pp 216-221.

Marola, G. (1989): Using Symmetry for Detecting and Locating Objects in a Picture. Computer Vision, Graphics, and Image Processing, Vol. 46, Nr. 2, San Diego: Academic Press, pp 179-195.

Masaki, I. (1992): Vision-based Vehicle Guidance. I. Masaki, editor. New York: Springer 1992.

Mori, H.; Nakai, M.; Chen, H. (1989): A Mobile Robot Strategy: Stereotyped Motion by Sign Pattern. Preprint of the 5th International Symposium of Robotics Research. Tokyo, pp 225-236.

Raboisson, S.; Even, G. (1990): Road Extraction and Obstacle Detection in Highway Environment. 2nd Prometheus Workshop on Collision Avoidance (CED 3). Coventry, pp 74-83.

Regensburger, U.; Graefe, V. (1990): Object Classification for Obstacle Avoidance. Proceedings, SPIE Symposium on Advances in Intelligent Systems. Boston, pp 112-119.

Saint-Marc, PH.; Medioni, G. (1990): B-Spline Contour Representation and Symmetry Detection. In G. Goos, J. Hartmanis (Ed.): Computer Vision – ECCV '90, Lecture Notes in Computer Science, Vol. 427. Berlin: Springer, pp 604-606.

Seitz, P.; Lang, G.K.; Gilliard, B.; Pandazis, J.C. (1991): The robust recognition of traffic signs from a moving car. In Radig (Ed.): Mustererkennung 1991, 13. DAGM-Symposium, Informatik-Fachberichte 290, München, Springer, pp 287-294.

Schmid, M.; Thomanek, F. (1992): Real-time Detection and Recognition of Vehicles for an Autonomous Guidance and Control System. Pattern Recognition and Image Analysis, Vol. 3, pp 337-380.

Solder, U. (1992): Echtzeitfähige Entdeckung von Objekten in der weiten Vorausschau eines Straßenfahrzeugs. Dissertation, Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München.

Solder, U.; Graefe, V. (1993): Visual Detection of Distant Objects. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS '93. Yokohama, pp 1042-1049.

Thomanek, F.; Dickmanns, D. (1992): Obstacle Detection, Tracking and State Estimation for Autonomous Road Vehicle Guidance. Proc. 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS '92. Raleigh, pp 1399-1406.

Tsinas, L., Graefe, V. (1992): Automatic Recognition of Lanes for Highway Driving. IFAC Conference on Motion Control for Intelligent Automation. Perugia, pp 295-300.

Wershofen, K. P. (1992): Real-time Road Scene Classification Based on a Multiple-lane Tracker. Proceedings, International Conference on Industrial Electronics, Control, Instrumentation and Automation (IECON '93). San Diego, pp 746-751.

Young, E.; Tribe, R.; Conlong, R. (1992): Improved Obstacle Detection by Sensor Fusion. IEE Colloquium on "Prometheus and Drive", Savoy Place.

Young, G.-S.; Hong T.-H.; Herman, M.; Yang, J.C.S. (1992): Obstacle Detection for a Vehicle Using Optical Flow. Proc. IEEE Intelligent Vehicles '92 Symposium. Detroit, pp 24-29.

Yuen, Sh.-Y.K. (1990): Shape from Contour Using Symmetries. In G. Goos und J. Hartmanis (Ed.): Computer Vision – ECCV '90, Lecture Notes in Computer Science, Vol. 427. Berlin: Springer. pp 437-453.

Zielke, T.; Brauckmann, M.; v. Seelen, W. (1992): Intensity and Edge-Based Symmetry Detection Applied to Car-Following. In G. Sandini (Ed.): Computer Vision – ECCV '92, Lecture Notes in Computer Science, Vol. 558. Berlin: Springer, pp 865-873.