IEEE Intelligent Systems, vol .13, nr. 6, 1999.

Autonomous Driving approaches Downtown

U. Franke, D. Gavrila
S. Görzig, F. Lindner, F. Paetzold, C. Wöhler
Daimler-Benz Research, T728
D-70546 Stuttgart, Germany
{franke, paetzold, goerzig}@dbag.stg.DaimlerBenz.com
{gavrila, lindner, woehler}@dbag.ulm.DaimlerBenz.com

Keywords: Machine Vision, Image Understanding, Autonomous Driving

Abstract

Most computer vision systems for vehicle guidance developed in the past were designed for the comparatively simple highway scenario.
Autonomous driving in the much more complex scenario of urban traffic or driver assistance systems like Intelligent Stop&Go are new challenges not only from the algorithmic but also from the system architecture point of view.
This contribution describes our current work on these topics. It includes the appropriate algorithms as well as approaches to control the various vision modules.

1. Introduction

Systems for vision-based navigation in the early 80's were based on the experience gained from static image processing. Since little computational power was available that time, it was common to take a picture, analyse it, and drive some distance in a blind fashion before the vehicle was stopped again for the next picture.

In 1986, Dickmanns [Dic86] demonstrated autonomous driving on highways at speeds up to 100 km/h using only a couple of 8086 processors. His idea was to use Kalman Filters to restrict the possible interpretations of the scene so that they are consistent with the dynamics of the considered systems as well as the interpretations derived in the past.

A large number of vision systems for lateral and longitudinal vehicle guidance, lane departure warning and collision avoidance has been developed during the last 10 years all over the world (e.g. [Tho88], [Web95], [Fra95], [Pom96]).

Most of the known autonomous demonstrators have been designed for highway traffic since this scenario is relatively simple: lanes are usually well marked and built with slowly changing curvature, traffic signs are large and clearly visible and other vehicles are the only potential obstacles that need to be considered. As shown in the final presentation of the European Prometheus project, the Daimler-Benz demonstrator vehicle VITA II is able to drive autonomously on highways and perform overtaking manoeuvres without any interaction [Ulm94].

A vision system would be even more attractive for future customers if its use were not limited to highway-like roads, but were also extended to support the driver in everyday traffic situations, including city traffic. Consequently, future computer vision research for traffic applications will have to consider a much wider range of situations than it does today.

Particularly attractive for driver assistance is the urban traffic environment. Imagine an *Intelligent Stop&Go* system that is able to behave like a human driver: it does not only keep the distance to its leader constant, as a radar based system would do, but also follows the leader laterally. Moreover, it stops at red traffic lights and stop signs, gives right of way to other ve-

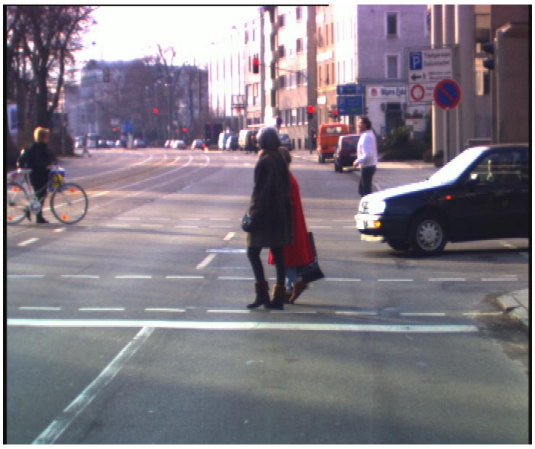


Fig.1: Image understanding in the urban environment is more challenging than on highways.

hicles if necessary and tries to avoid collisions with children running across the street.

Unfortunately, vision on urban roads turns out to be much more difficult than on highways due to the complexity of this environment. Figure 1 shows an everyday example, that we have to understand.

In this contribution we describe our approach to build an intelligent real-time vision system for this scenario. This includes stereo vision for depth-based obstacle detection and tracking, a framework for monocular detection and recognition of relevant objects and an attempt to realise such a system without the necessity of a super computer in the trunk. The computational power in our demonstrator car UTA (Urban Traffic Assistant) is currently limited to three 200 MHz PowerPCs.

2. Urban Applications and Vision Tasks

In order to limit the complexity of autonomous driving, we intend to realise the described *Intelligent Stop&Go* system first. The goal is to detect an appropriate leading vehicle and to signal the driver that the system is ready for autonomous following. It will be his responsibility to activate the system. If it cannot continue to follow the leader (e.g. since he is changing the lane), the system shall be allowed to drive a limited distance in autonomous mode searching for a new leader before the control of the vehicle is turned over to the driver again. Control will also be given back if the vehicle has to stop in front of a stop sign or if it is the first vehicle in front of a red traffic light.

Besides an *Intelligent Stop&Go* system as sketched above, driver assistant systems like rear-end collision avoidance or red traffic light recognition and warning are also of interest for urban traffic.

The most important perception tasks that have to be performed in order to build such systems are:

- The leading vehicle must be detected and its distance, speed and acceleration must be estimated in longitudinal and lateral direction.
- The course of the lane must be extracted even if it is not given by well painted markings and does not show clothoidal geometry.
- Small traffic signs and traffic lights have to be detected and recognised in a highly coloured environment.
- Different additional traffic participants like bicyclists or pedestrians must be detected and classified.
- Stationary obstacles that limit the available free space e.g. parking cars must be detected.

3. Stereo based Obstacle Detection and Tracking

For navigation in urban traffic it is necessary to build an internal 3D map of the environment in front of the car. This map must include position and motion estimates of relevant traffic participants and potential obstacles. In contrast to the highway scenario where one can concentrate on looking for rear sides of leading vehicles, our system has to deal with a large number of different objects.

Several schemes for object detection in traffic scenes have been investigated in the past. Besides the mentioned 2D

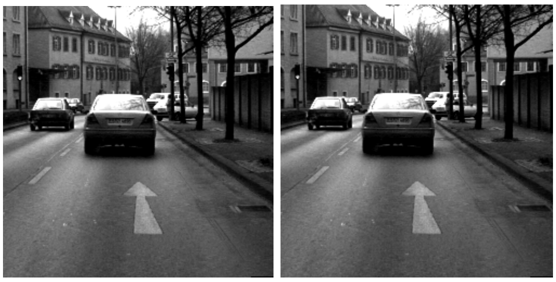


Fig.3.1: Stereo image pair.

model based approaches searching for rectangular, symmetric shapes, inverse perspective mapping based techniques [Bro97], optical flow based approaches [Enk97] and correlation based stereo systems [San96] have been tested.

The most direct method to derive 3D-information is binocular stereo vision. The key problem is the correspondance analysis. Unfortunately, classical approaches like area based correlation techniques or edge based approaches are computationally very expensive. To overcome this problem, we have developed a feature based approach that is tailored to our specific needs and runs in real time on a 200 MHz PowerPC 604 [Fra96].

This scheme classifies each pixel according to the grey values of its four direct neighbours. It is checked whether each neighbour is much brighter, much darker or has similar brightness compared to the considered central pixel. This leads to 3^4=81 different classes encoding edges and corners at different orientations.

Fig.3.1 shows a stereo image pair taken from our camera system with a base width of 30 cm. The result of the structure classification is shown in Fig.3.2 for the left image. Different grey values represent different structures, pixel in homogeneous areas are assigned to the „white“ class and ignored in the sequel.

The correspondance analysis works on these feature images. The search for possibly corresponding pixels is reduced to a simple test whether two pixels belong to the same class. Thanks to the epipolar constraint and the fact that the cameras are mounted with parallel optical axis, pixels with identical classes must be searched on corresponding image rows only.

It is obvious that this classification scheme cannot guarantee uniqueness of the correspondances. If ambiguities occur, the solution giving the smallest disparity i.e. the largest distance is chosen to overcome this problem. This prevents wrong correspondances caused by for example periodic structures to generate phantom obstacles close to the camera. In addition, measurements that violate the ordering constraint are ignored.

The outcome of the correspondance analysis is a disparity image, which is the basis for all subsequent steps. Fig. 3.3 tries to visualise such an image.

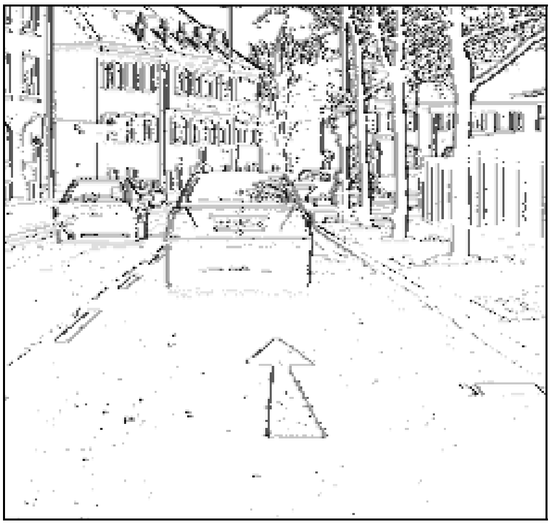


Fig. 3.2: Classified pixel.

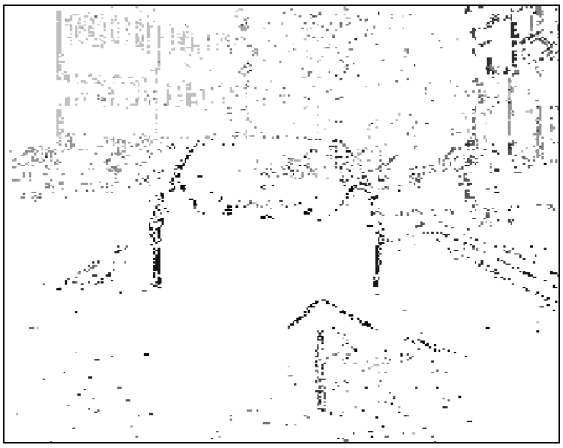


Fig. 3.3: Distance Image. The brightness of the feature pixel is proportional to their distance.

3.1 Structures on the Road

For the recognition of road boundaries it is helpful to know which structures in the image lie on the road surface. Such structures can easily be extracted from the disparity image if the road can be assumed to be (nearly) flat.

If the camera orientation is known, all points having a disparity in a certain interval around the expected value lie on the road, those with larger disparities belong to objects above the road. Looking for all points which lie within a height interval of [-15cm, 15cm] relative to the road yields the result displayed in Fig.3.4.

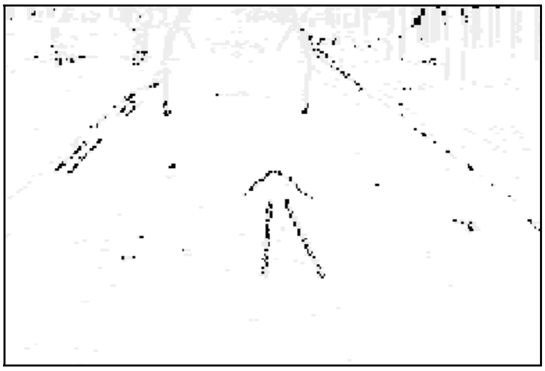


Fig. 3.4: Classified road pixel.

3.2 Estimation of Camera Height and Pitch Angle

In practice camera height and pitch angle are not constant during driving. Fortunately, the relevant camera parameters can be efficiently estimated themselves using the extracted road surface points. Least squares techniques or Kalman filtering can be used to minimise the sum of squared residuals between expected and found disparities.

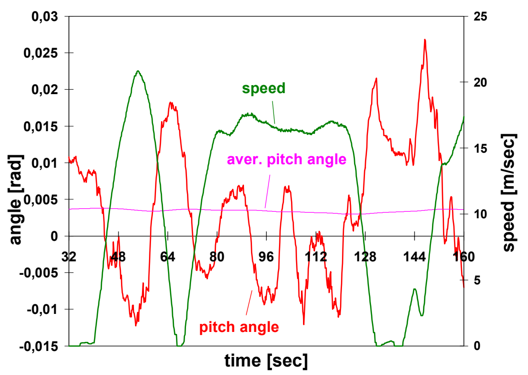


Fig. 3.5: Estimated camera pitch angle during acceleration and deceleration.

Results of a test drive are shown in Fig. 3.5. During acceleration, the pitch angle decreases, while it increases during breaking manoeuvres. In the time between 80 and 120 seconds the speed was nearly constant. The oscillations of the pitch angle are due to the uneven road surface.

3.3 Obstacle Detection and Tracking

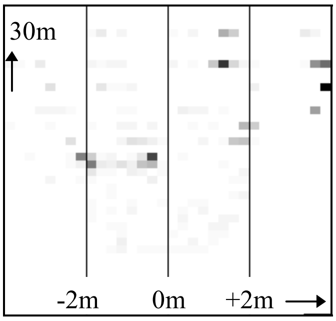


Fig. 3.6: Depth map, peaks are caused by the car and the trees.

If all features on the road plane are removed, a 2D depth map containing the remaining features can be generated. The map shown in Fig. 3.6 covers an area of 30m in length and 8m in width. The leading vehicle causes the two peaks at 15m distance and lateral positions of 0m and –2m. This map is used to detect objects that are tracked subsequently. In each loop, already tracked objects are deleted in this depth map prior to the detection.

The detection step delivers a rough estimate of the object width. A rectangular box is fitted to the cluster of feature points that contributed to the extracted area in the depth map. This cluster is tracked from frame to frame. For the estimation of the obstacle distance, a disparity histogram of the object's feature points is computed. In order to obtain a disparity estimate with subpixel accuracy, a parabola is fitted to the maximum of this histogram.

In the current version, an arbitrary number of objects can be considered. Sometimes the right and left part of a vehicle are initially tracked as two distinct objects. These objects are merged on a higher „object-level“ if their relative position and motion fulfil reasonable conditions.

From the position of the objects relative to the camera system their motion states i.e. speed and acceleration in longitudinal as well as lateral direction are estimated by means of Kalman filters.

The longitudinal motion parameters are the inputs for a distance controller. Fig. 3.7 shows the results of a test drive in the city of Esslingen, Germany. The desired distance is composed of a safety distance of 10 meters and a time headway of 1 second. Notice the small distance error when the leader stops.

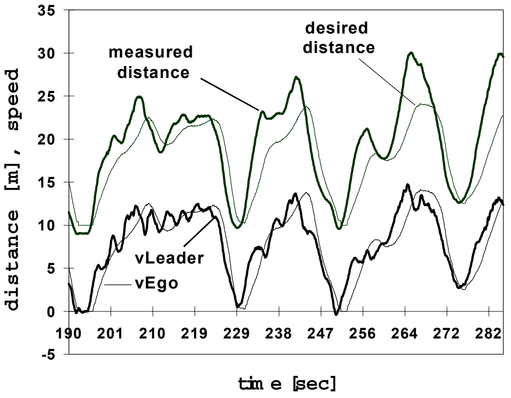


Fig. 3.7: Autonomous vehicle following in city traffic.

4. Object Recognition

The previous section dealt with the use of stereo vision to detect and track obstacles in front of the vehicle. One essential thing that makes Stop&Go *intelligent* is the ability to recognise objects. Two classes of objects are relevant for this application:

- elements of the infrastructure (road boundaries, road marks, traffic signs and traffic lights) and
- traffic participants (pedestrians, (motor) bicycles and vehicles).

How do we recognise various objects? Although devising a general framework is difficult, we often find ourselves applying two steps, detection and classification. The purpose of the detection step is to efficiently obtain a region of interest (ROI), i.e. a region in image space or parameter space that could be associated with a potential object. Besides the described stereo approach, we have also developed detection methods based on shape, colour and motion. For example, shape and colour cues are used to find potential traffic signs and arrows on the road (see Subsections 4.1 and 4.2), motion is used to find potential pedestrians (see Subsection 4.4).

Once a ROI has been obtained, more computation-intensive algorithms are applied to „recognise" the object, i.e. to estimate model parameters. In our application to natural scenes, objects have a wide variety of appearances because of shape variability, different viewing angles and illumination changes. Because explicit models are seldom available, we derive models implicitly by learning from examples. Recognition is seen as a classification process. We have implemented a large set of classifiers for this purpose, i.e. Polynomial Classifiers, Principal Component Analysis, Radial Basis Functions, (Time Delay) Neural Networks and Support Vector Machines (the latter in collaboration with the MIT CBCL Laboratory [Pap98]). Our emphasis on learning approaches to object recognition is backed up by many hours of data recorded on a digital video recorder while driving in urban traffic. Interesting data segments are labelled off-line.

4.1 Road Boundaries and Markings

4.1.1 Road Recognition

Road recognition has two major objectives. As a stand alone application, it enables automatic road following. In the context of a Stop&Go system, lane changes of the leading vehicle have to be registered in order to return the control of

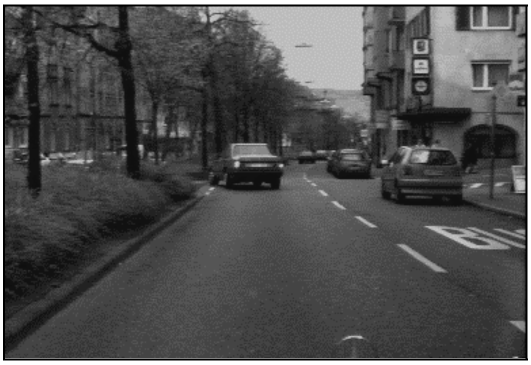


Fig. 4.1 Urban road scenario.

the vehicle back to the driver, as stated in chapter 1.

Furthermore, the lateral control behaviour of the Stop&Go system can be improved. As the standard solution, lateral guidance is accomplished by a so called lateral tow bar controller. It requires distance and angle with respect to the leading vehicle as measured by the stereo vision system described above. A tow bar controlled vehicle drives approximately along the trajectory of the leading vehicle but tends to cut the corner in narrow turns. This undesirable behaviour can be controlled if the position of the ego vehicle relative to the lane is known.

Lanes of urban roads are not as well defined as those of highways. Lane boundaries often show poor visibility. Various objects, as traffic participants or background infrastructure, clutter the image. The road topology is comprised of a variety of different lane elements as stop lines, zebra crossings and forbidden zones. A comprehensive geometrical road model can not be readily defined. All these characteristics suggest a data driven, global image analysis that robustly separates lane structures from background.

The global detection analyses the polygonal contour images. Fig. 4.1 shows an image of the considered road, Fig. 4.2 the extracted edges. A database organises the polygons with respect to their length, orientation, position and mutual spatial relations such as parallelism and collinearity. It provides fast filters for these attributes. Arbitrary combinations of properties can be specified to detect possible road structures. Detected lane boundaries are subsequently classified as curbs, markings and clutter by a Polynomial Classifier. Results of the lane boundary recognition are shown in figure 4.2 with dark lines, a recognised pedestrian crossing is shown in figure 4.3. Regions where obstacles have been detected by the stereo vision system of chapter 3 are ruled out as possible positions of road structures.

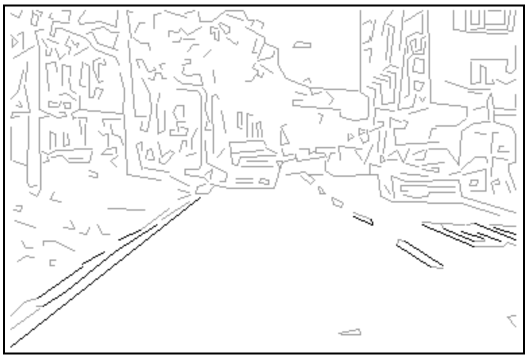


Fig. 4.2: Polygonal edge image (light) and detected lane structures (dark).

Even though the entire global detection needs only 80-100ms on a PowerPC 604, data driven methods are computationally quite expensive. As learned from the work on highway lane following, model based estimation of the road geometry is very efficient in that only small portions of the image have to be considered when lane markings are locally tracked in image sequences.

The urban lane tracker combines global analysis with this local principle. For as long as possible, boundaries detected by the global analysis are tracked locally in order to estimate the vehicle's lateral po-

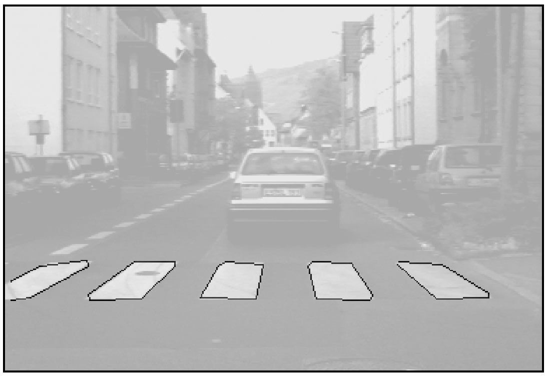


Fig. 4.3: Detected zebra crossing.

sition and yaw angle in a few milliseconds. Tracking and global detection are supervised to collaborate efficiently [Pae98].

4.1.2 Arrow Recognition

The recognition of arrows on the road follows the two-step procedure, detection and classification, mentioned at the beginning of this section. It uses shape and colour cues in a region-based approach. The detection step consists of a colour segmentation step and a filtering step.

The colour (i.e. greyscale) segmentation step involves reducing the number of colours in the original image to a handful. In this application, this reduction is based on the minima and plateaus of the greyscale histogram. Following this greyscale segmentation a colour connected components (CCC) analysis is applied to the segmented image. The algorithm produces a database containing information about all regions in the segmented image. Among the computed attributes are the area, bounding box, aspect ratio, length and smoothness of contour as well as additional features which are determined on a need-basis, due to computational cost. We have developed a query language for this region database, dubbed „Meta-CCC", which allows queries based on attributes of single regions as well as on relations between regions (e.g. adjacency, proximity, enclosure and collinearity). The filtering step thus involves formulating a query to select candidate regions from the database. The resulting set is normalised for size and given as input to a radial basis function (RBF) classifier. Fig. 4.4 shows the original and the obtained result.

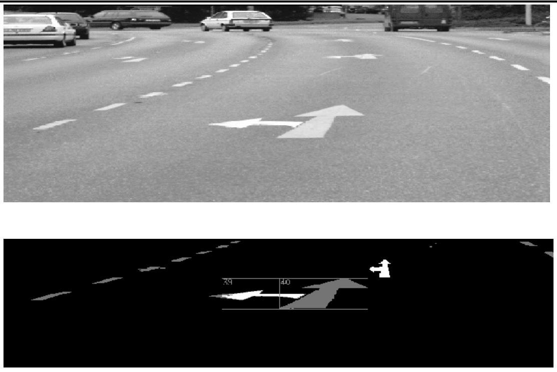


Fig. 4.4: Above: Street scene displaying a direction arrow. Below: the segmented and classified arrow.

4.2 Traffic Signs

4.2.1 Colour

Our colour traffic sign recognition system [Jan93] goes back to the Prometheus project and was originally developed with the highway scenario in mind. Overall, it follows the same steps as described in the previous subsection on road arrow detection, that is, colour segmentation, filtering and classification. The colour segmentation step involves pixel classification using a look-up table. The look-up table was generated off-line in a training process using a polynomial classifier. The outcome of the segmentation process are pixels labeled "red", "blue", "yellow" and "uncoloured". As before, we apply the CCC algorithm and formulate queries on the resulting regions using the MetaCCC procedure. Typical queries would include searching for red regions of particular shape which enclose an uncoloured region. The next step, classification, is done with a RBF classifier in a multi-stage process. The input is a colour-normalised pictograph, extracted from the original image at the candidate locations provided by the MetaCCC procedure. The classification stages involve colour, shape and pixel values. The results are stabilised by integration over time [Jan93].

There are a number of challenges to traf-

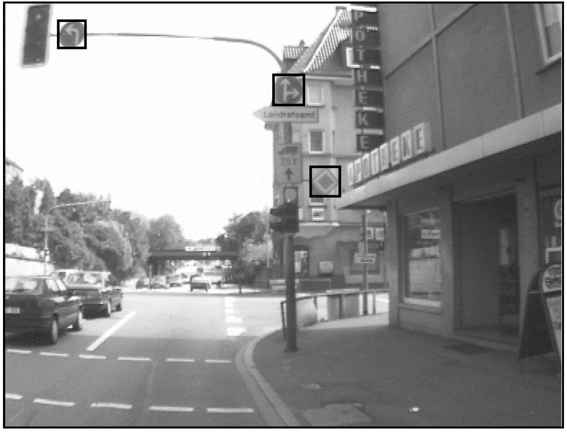


Fig. 4.5: Recognised traffic signs in a wide-angle view.

fic sign recognition in the urban scenario. First, the field of view needs to be extended, which results in a lower effective image resolution. At the same time, the use of wide-angle lenses introduces significant lens distortion. Traffic signs are also not viewed head-on, as most of the time on highways. In the city, they must be recognised when they are lateral to the camera and appear skewed in the image. The sketched system has been extended in order to cope with these problems and the urban specific signs that are relevant for our goals have been added. Figure 4.5 shows some results.

4.2.2 Greyscale

Clearly, colour information is beneficial for traffic sign recognition. Yet there are still some issues open regarding colour segmentation. The difficulty is that there is quite a range in which the same „true" traffic sign colour can appear in the image, depending on illumination condition, i.e. whether camera looks towards or away from the sun, day or night, rain or sunshine. In order to answer the question of how successful we can be without colour cues we are also working on a shape-based traffic sign detection system in greyscale, described here. This system might also be useful for the case one cannot use colour because of cost considerations.

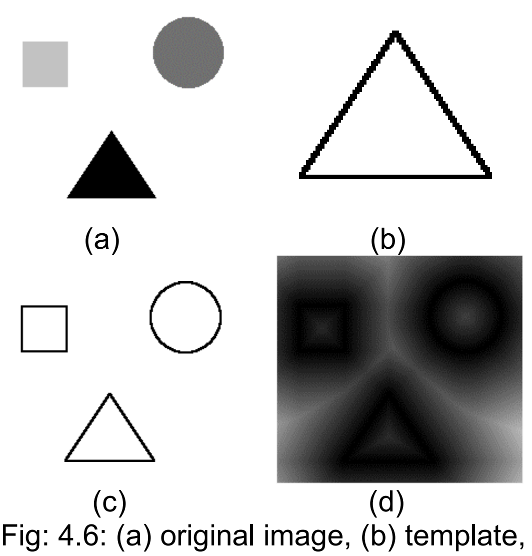


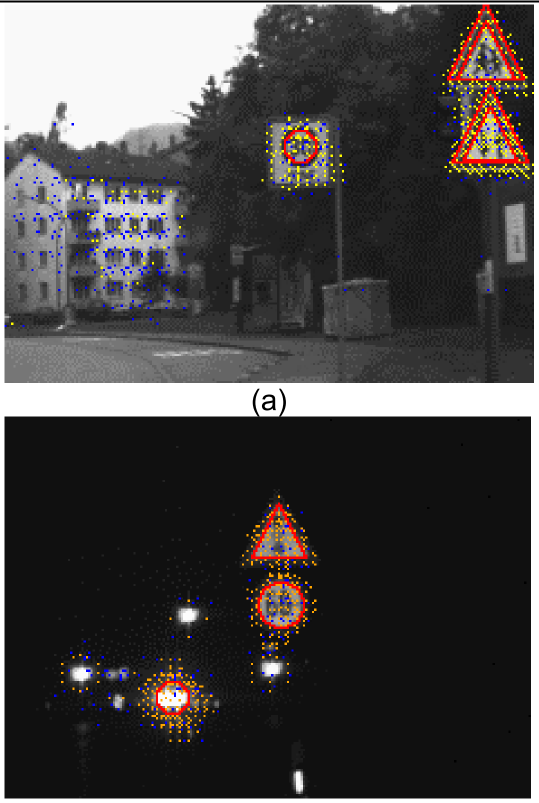
Fig: 4.6: (a) original image, (b) template, (c) edge image and (d) Distance-Transform image (see text).

The greyscale detection system works on edge features, rather than region features. The approach uses a hierarchical template matching technique based on distance transforms (DTs) [Gav98]. DT-based matching allows the matching of arbitrary (binary) patterns in an image. These could be circles, ellipses, triangles but also non-parameterised patterns, for example outlines of pedestrians (see also subsection 4.4.1)

The pre-processing step involves reading a test image (Figure 4.6a), computing thresholded edge image (Figure 4.6c), and computing its distance image ("chamfer image", Figure 4.6d). The distance image has the same size as the binary edge image; at each pixel it contains the image distance to the nearest edge pixel of the corresponding binary edge image.

The matching step involves correlating a binary shape pattern (e.g. Figure 4.6b) with the distance image; at each template location a correlation measure gives the sum of nearest-distance of template points to image edge points. A low value denotes a good match (low dissimilarity), a zero value denotes a perfect match. If the measure is below a user-defined threshold one considers a pattern detected.

The advantage of matching a template with the DT image is that the resulting similarity measure will be smoother as a function of the template parameters (transformation, shape). This enables the use of various efficient search algorithms to lock onto the correct solution. It also allows more variability between a template and an object of interest in the image. Matching with the edge image (or the unsegmented gradient image, not shown here), on the other hand, typically provides strong peak responses but rapidly declining off-peak responses, which do not facilitate efficient search strategies or allow object variability.



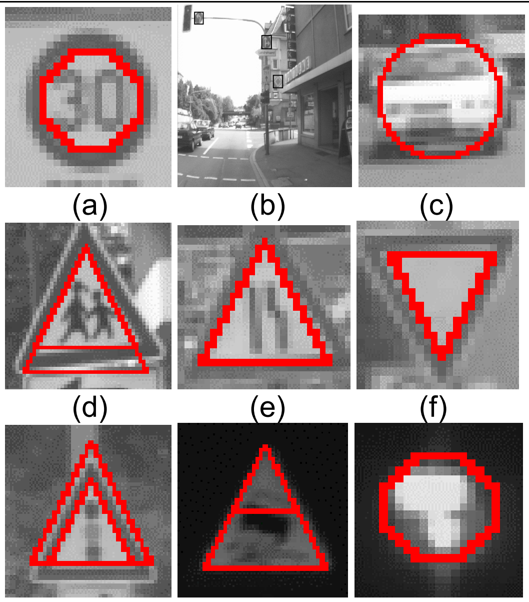
(a)

(b)

Fig. 4.7: Traffic sign detection: (a) day and (b) night (white dots denote intermediate results; the locations matched during hierarchical search).

We have extended the basic DT-matching scheme with a hierarchical approach, where in addition to a coarse-to-fine search over the translation parameters, templates are grouped off-line into a template hierarchy based on their similarity. This way, multiple templates can be matched simultaneously at the coarse levels of the search, resulting in various speed-up factors. The template hierarchy can be constructed manually or can be generated automatically from available examples (i.e. learning). Furthermore, in matching, features are distinguished by type and separate DT's are computed for each type (e.g. based on edge orientations). For details, refer to [Gav98].

Figure 4.7 illustrates the followed hierarchical approach. The white dots indicate locations where the match between image and a (prototype) template of the template tree was good enough to consider matching with more specific templates (e.g. the children) on a finer grid. The final detection result is also shown. More detection results are given in Figure 4.8, including some false positives (c and i). We achieve typical detection rates of 90% on single frames, with 4-6% false positives. More than 95% of the false positives were rejected in a subsequent pictograph classification stage using a RBF network.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4.8: More detection results.

4.3 Traffic Lights

The recognition of traffic lights follows the same three steps used before: colour segmentation, filtering and classification. Colour segmentation uses a simple look-up-table in order to determine image parts in the traffic light colours red, yellow, and green. By applying the before-mentioned colour connected components (CCC) algorithm to the segmented image only the blob-like shaped regions the area of which lies within a certain range are selected as possible traffic light candidates using the MetaCCC procedure.

A region of interest (ROI) of a size adapted to the blob diameter is then cropped such that it contains not only the luminous part of the traffic light but also its dark box. The ROI is normalised to a uniform size. Eventually, a local contrast normalisation by means of a simulated

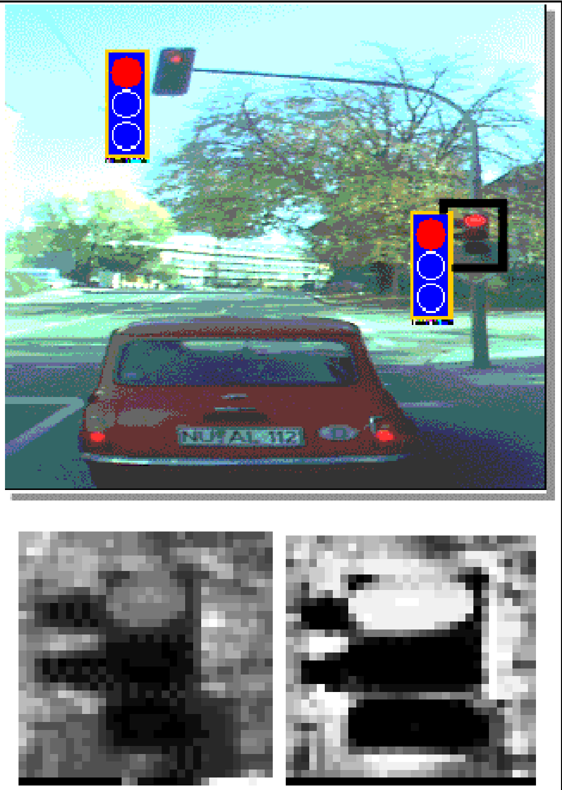


Fig. 4.9: Example of the recognition of a red traffic light. Additionally shown are the cropped and normalised ROI before (left) and after (right) pre-processing it by means of a simulated Mahowald retina.

Mahowald retina is carried out additionally (see Fig. 4.10).

This pre-processed ROI serves as an input to a three-layer feed-forward neural network that performs the actual object recognition task. It is constructed such that a neuron of the second network layer does not „see" the complete underlying image but only a small region of it, i.e. its receptive field. These receptive fields extract local features from the input image that have been learned during the training process. The actual classification takes place in the higher network layers. The network has K output neurons for the K different object classes to be distinguished; the output neuron with the highest activation denotes the class to which the object is assigned. In the case of traffic lights, we have K=2 classes, the class „traffic light" and the class „garbage".

The appearance of red, red-yellow, yellow, and green traffic lights is trained separately, respectively. On a 133 MHz Power PC, the algorithm runs at a rate of about 4 images per second, depending on the number of traffic light candidates. In experiments, we found recognition rates of above 90%, with false positive rates below 2%.

4.4 Pedestrians

This subsection deals with work in progress in recognising the most vulnerable traffic participants, the pedestrians. We can either recognise pedestrians by their shape (subsection 4.4.1) or by their characteristic walking pattern (subsection 4.4.2).

4.4.1 Towards Pedestrian Recognition from Shape

We are currently compiling a large set of pedestrian outlines to account for the wide variability of pedestrian shapes in the real world. Our first approach to pedestrian recognition by shape has involved blurring the pedestrian outlines in the database (Fig. 4.10) and performing a principal component analysis on the resulting data set. This results in a compact representation of the original image data in terms of the eigenvectors. The first eigenvectors (and the mean) represent characteristic features of the pedestrian distribution, the last eigenvectors mostly represent noise (see Fig. 4.11). In order to find pedestrians in a test image, we apply a gradient

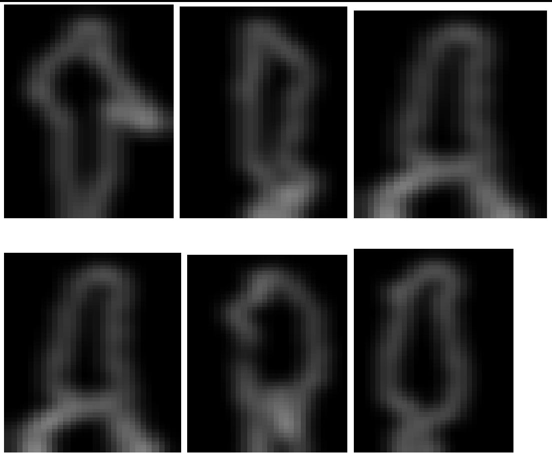


Fig. 4.10:Examples of the pedestrian database.

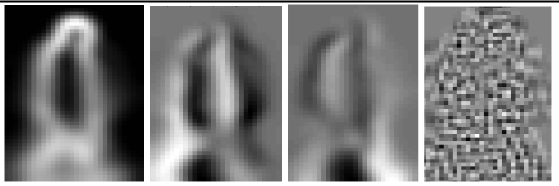


Fig. 4.11: Principal component analysis: eigenvectors 0 (mean), 1, 2, 25.

operator to the image and normalise for image energy. The resulting normalised gradient image is then correlated with the first few eigenvectors of the pedestrian data set. Preliminary findings show that we need no more than the first 10 eigenvectors. A threshold on the correlation value determines whether a pedestrian is recognised or not, see Fig. 4.12 for a matching result. Besides the correct solution, we also obtain a false positive at the windows. The latter could be accounted for if camera positioning with respect to the road is considered. For a different learning-based approach to pedestrian recognition, see [Pap98].

4.4.2 Towards Pedestrian Recognition from Motion

The neural network used for traffic light recognition can be extended into the temporal dimension; the input then consists of greyscale image sequences. This results in a time delay neural network (TDNN) architecture with spatio-temporal receptive fields.

The receptive fields act as filters to extract spatio-temporal features from the input image sequence. Thus, features are not hard-coded but learned during a training process. Subsequent classification relies on the „filtered" image sequences rather than on the raw image data.

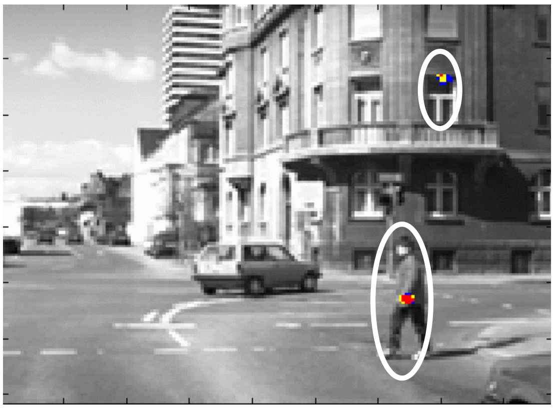


Fig. 4.12: Matching results: The pedestrian is correctly detected, a false alarm occurs in the window above.

The TDNN is currently applied to the recognition of pedestrian gait patterns. A detection step determines the approximate position of possible pedestrians; we have two methods at our disposal:

- Colour clustering on monocular images in a combined colour/position feature space [Hei98]. A fast polynomial classifier selects the clusters possibly containing a pedestrian's legs by evaluating temporal changes of a shape-dependent cluster feature.
- 3D segmentation by stereo vision: the stereo algorithm described in section 3 determines bounding boxes circumscribing obstacles in the scene.

Final verification on the candidate region sequence is then performed by the TDNN, compare Fig. 4.13. We are currently examining whether the TDNN approach can be used for segmentation-free object and motion detection and recognition.

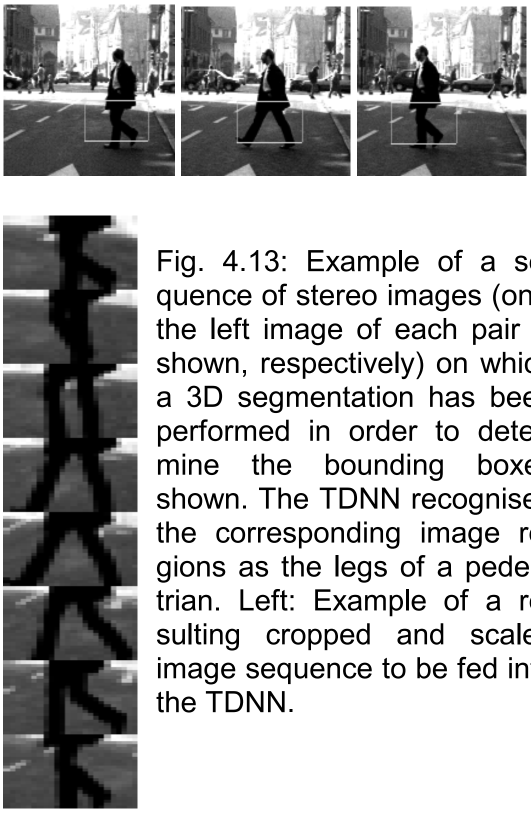


Fig. 4.13: Example of a sequence of stereo images (only the left image of each pair is shown, respectively) on which a 3D segmentation has been performed in order to determine the bounding boxes shown. The TDNN recognised the corresponding image regions as the legs of a pedestrian. Left: Example of a resulting cropped and scaled image sequence to be fed into the TDNN.

5. Putting things together

In our well known Prometheus demonstrator VITA II for driving autonomously on highways each vision module was assigned to a subset of processors in a static configuration. The modules were running permanently, even though some results were of no interest for vehicle guidance at that moment.

To use this architecture for computer vision in complex inner city traffic would require even more resources. But why looking for new traffic signs or continuously determine the lane width while the car is stopping at a red traffic light? While following a leading vehicle, why should not only tasks be processed which are useful in that situation?

Although we were successful with the mentioned brutal force approach, such questions reveal some disadvantages:

- There is no concept for controlling the modules, e.g. to focus the resources on relevant tasks for specific situations.
- It is not scaleable for a larger number of modules.
- There is no uniform concept for the interconnection and Cupertino of modules.
- The development of new applications usually requires extensive reimplementations.
- Reuse of old modules can be difficult due to missing interfaces.

The growing complexity of autonomous systems hence requires an architecture which can cope with these problems. The pursued approach is to make an explicit distinction between the modules (obstacle detection, vehicle control etc.) on one hand and the connection of modules to perform a specific application (e.g. autonomous Stop&Go driving) on the other hand.

Each module is connected to the system by a module interface. This allows developers to concentrate on solving the computer vision problem instead of having to worry about how to connect their work to the application environment.

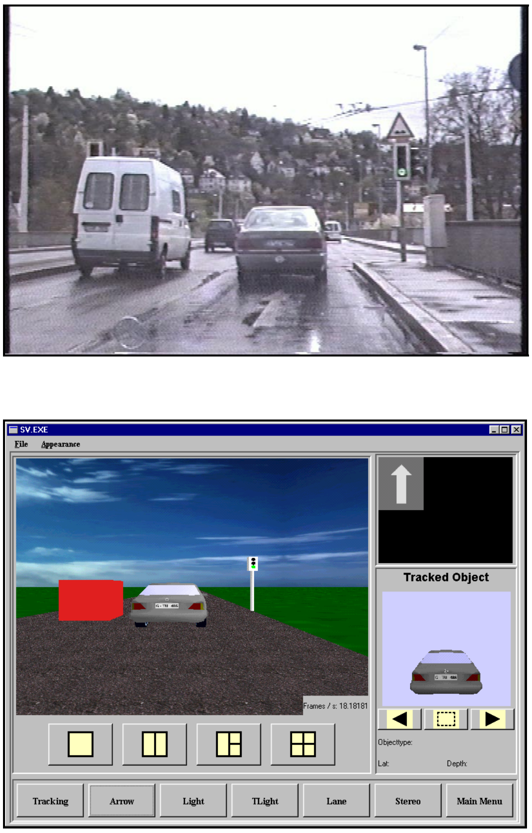


Fig. 5.1: Visualization of the objects seen by the system. The touch screen acts as the interface between developer and vision modules.

An application is formed by connecting together a group of these modules. The lane keeping application for example can be implemented by connecting the lane detection module and the lateral vehicle control module.

This flexible architecture allows the development of various applications without modifying the modules. Moreover, the system allows to change the connections during runtime. This enables the economical use of computational resources and adaptation of the system to the current situation. It is possible, for example, to accomplish lane keeping on highways and switch dynamically to the Stop&Go application in the city. For a more detailed description of the system see [Goe98].

So far, most of the modules described in the paper have been integrated into the system. Currently, our demonstrator car UTA is able to recognise traffic lights, traffic signs, overtaking vehicles, direction arrows, pedestrian crossings and to follow a leading vehicle (Stop&Go) autonomously. The system is running on three 200 MHz AIX/PowerPCs for the computer vision modules and a Windows/Pentium II for the visualisation of the results (see image 5.1). The next steps are to include the missing modules, and to adapt the system to multiple environments and applications:

- autonomous driving on highways
- speed limit assistant: the driver is warned when driving faster than allowed on the current road
- lane departure warning
- enhanced cruise control: the vehicle slows down, when a car in front falls below the desired speed. It raises the speed again, if there is no obstacle ahead.

6. The Road Ahead

The experience gained from tests on urban roads strengthens our confidence that computer vision will go downtown and vision-based driver assistance and autonomous driving will appear in the inner city environment in the not so distant future.

In this paper we have focused on computer vision issues, but there are also related issues which are of great interest for this kind of applications.

Angle of View: A serious problem in urban traffic is the necessary angle of view. On one hand we need a wide angle lens to see traffic lights and signs if we are close to them. On the other hand, the precision of stereo based distance measurement and the performance of object detection and recognition, in particular traffic sign recognition, grows with the focal length. We expect that these problems could be rather solved with high resolution chips of 1000^2 or 2000^2 pixels rather than with rotatable cameras, vario-lenses or multi-camera systems.

Camera Dynamics: A second sensor problem is the insufficient dynamic of the common CCD. The new logarithmic CMOS-Chips (High Dynamic Range Chip) promise a way out of this dilemma. These chips will help us on sunny days to see structures in the shadowed areas as well as at night-time, when bright lights glare into the camera and confuse the automatic camera control.

Application Specific Hardware: Another important problem we have to solve in order to bring intelligent vision systems to the market is the price of the appropriate hardware. Analogue chips for early vision steps and modern FPGA technology are possible solutions for future mobile vision systems. We are currently investigating the possibility of using these programmable arrays in order to speed up edge detection and the sketched distance transform.

Besides these technical issues, there are important legal (i.e. liability) and acceptance issues which need to be resolved before vehicles with an *Intelligent Stop&Go* system can be driven downtown by our customers.

References

[Bro97] A.Broggi: „Obstacle and Lane Detection on the ARGO“, IEEE Conf. on Intelligent Transportation Systems ITSC ’97, Boston, 9.-12.Nov.1997

[Dic86] E.D.Dickmanns, A.Zapp: "A curvature-based scheme for improving road vehicle guidance by computer vision", Proc. SPIE Conference on Mobile Robots, Vol. 727, 1986, S. 161-16

[Enk97] W.Enkelmann: „Robust Obstacle Detection and Tracking by Motion Analysis“, IEEE Conf. on Intelligent Transportation Systems ITSC ’97, Boston, 9.-12.Nov.1997

[Fra95] U.Franke, S.Mehring, A.Suissa, S.Hahn: „The Daimler-Benz Steering Assistant - a spin-off from autonomous driving“, Intelligent Vehicles ‘94, Paris, Oktober 1994, pp.120-126

[Fra96] U.Franke, I.Kutzbach: “Fast Stereo based Object Detection for Stop&Go Traffic“, Intelligent Vehicles ‘96, Tokyo, pp.339-344

[Gav98] D. Gavrila: „Multi-feature Hierarchical Template Matching Using Distance Transforms", ICPR'98

[Goe98] S.Görzig, U.Franke: „ANTS – Intelligent Vision in Urban Traffic”, in IEEE Conference on Intelligent Transportation Systems, October 1998, Stuttgart

[Hei98] B.Heisele and C.Woehler: „Motion-Based Recognition of Pedestrians", ICPR'98

[Jan93] R.Janssen, W.Ritter, F.Stein and S.Ott: „Hybrid Approach for Traffic Sign Recognition", Proc. of Intelligent Vehicles Conference, 1993.

[Pae98] F.Paetzold, U.Franke; “Road Recognition in Urban Environment”, IEEE Conference on Intelligent Transportation Systems, October 1998, Stuttgart

[Pap98] C.Papageorgiou, M.Oren and T.Poggio: „A General Framework for Object Detection", Int. Conf. on Computer Vision, 1998

[Pom96] D.Pommerleau, T.Jochem: ”Rapidly Adapting Machine Vision for Automated Vehicle Steering”, IEEE Expert, Vol.11, No.2, pp19-27

[San96] K.Saneyoshi: „3-D image recognition system by means of stereoscopy combined with ordinary image processing“, Intelligent Vehicles ‘94, 24.-26. Oct. 1994, Paris, pp.13-18

[Tho88] C.Thorpe, M.Herbert, T.Kanade and S.Shafer: „Vision and Navigation for the Carnegie-Mellon Navlab“, IEEE-PAMI, vol.10, no.3, 1988

[Ulm94] B.Ulmer: „VITA II - Active collision avoidance in real traffic“, Intelligent Vehicles ‘94, Paris, 24.-26. Oct.. 1994, S.1-6

[Web95] J.Weber et al.: „New results in stereo-based automatic vehicle guidance“, Intelligent Vehicles ‘95, 25./26. Sept. 1995, Detroit, pp.530-535