CARTRACK: Computer Vision–Based Car–Following

Thomas Zielke   Michael Brauckmann   Werner von Seelen

Institut für Neuroinformatik
Ruhr–Universität
4630 Bochum , Germany

Abstract

CARTRACK *is a computer vision system that can reliably detect, track, and measure vehicle rears in images from a video camera in a following car. The system exploits the symmetry property typical for the rear of most vehicles on normal roads. We present two novel methods for detecting mirror symmetry in images, one based directly on the intensity values and another one based on a discrete representation of local orientation.* CARTRACK *has been used for realtime experiments with test vehicles of Volkswagen and Daimler-Benz.* *

1 Introduction

The work described in this paper is part of a joint research effort, aimed at the development of an experimental vehicle which has some automatic driving abilities on normal roads. Our interest in symmetry detection originates from the problem of car-following by computer vision, i.e. the problem of how a vehicle equipped with a camera and control computers can be programmed to automatically keep a safe driving distance to a leading car. There are three major visual tasks the system has to cope with:

1. Detecting cars driving ahead of one on the road. This means repeated visual scanning of the road in front of the car until an object appears which can be identified as another vehicle.

2. Visual tracking of a car seen from behind while its image position and size may vary greatly.

3. Accurate measuring of the car's dynamic image size for the speed control.

*The work described in this paper has been funded by the German Federal Ministry of Research and Technology (BMFT) and the German automobile industry as part of the PROMETHEUS (PRO-ART) project.

Figure 1: A typical image in a car-following situation on an Autobahn. The overlay plot in the lower half shows the intensity distribution for the scan line marked in the upper half of the image.

The methods presented here exploit the symmetry property typical for the rear of most vehicles on normal roads. Mirror symmetry with respect to a vertical axis is one of the most striking generic shape features available for object recognition in a car-following situation. Initially, we use an intensity-based symmetry finder to detect image regions that are candidates for a leading car. The vertical axis of symmetry obtained from this step is also an excellent feature for measuring the leading car's relative lateral displacement in consecutive images because it is invariant under (vertical) *nodding* movements of the camera and under changes of object size. To exactly measure the image size of the car in front, a novel edge detector has been developed which enhances pairs of edge points if the local orientations at these points are mutually symmetric with respect to a known symmetry axis. Edge points that do not have a mirror symmetric counterpart are suppressed.

0-8186-2840-5/92 $03.00 © 1992 IEEE

In computer vision publications symmetry has received attention primarily as a perceptual grouping phenomenon, as a means of recovering object (surface) orientation, as a practical constraint for "shape from contour", and for shape description. For this paper, previous work on *symmetry detection* and *finding symmetry axes* is most relevant. There exist optimal solutions for this problem if the data can be mapped onto a set of points in a plane and only accurately symmetric point configurations are searched for. However, for real image data such methods cannot be used because they fail to detect imperfect symmetry. Another class of algorithms assumes that the figure, i.e. the image region, for which symmetry axes are sought, can be readily separated from the background. Friedberg [1], for example, shows how to derive axes of skewed symmetry of a figure from its matrix of moments. Marola [2] proposes a method for object recognition using the symmetry axis and a symmetry coefficient of a figure. The method requires the computation of central moments too but, in contrast to [1], it is based on intensity values and hence takes into account the internal structure of the object as well . However, these methods either assume that the segmentation problem has been solved or that there is no segmentation problem (e.g. uniform background intensity). Saint-Marc and Medioni [3] propose a B-spline contour representation which facilitates symmetry detection. They show how the elemental conic segments of the B-splines can be analytically checked for mutual symmetry. Their approach is particularly interesting for detecting axes of skew symmetry and parallel symmetry. Gross and Boult [4] compare local and global methods for finding symmetry axes. They describe a global moment-based algorithm which analytically derives axes of skew symmetry from segmented object contours. A second method proposed by Gross and Boult is a local one that uses a tangent-based constraint for pairs of contour points. There are some similarities between this approach and the work by Yeshurun et. al. [5]. They suggest an "interest detection operator" which assigns a "symmetry value" to every point in an image.

For our application we need methods that do not require computationally expensive image preprocessing or segmentation. More importantly, we need algorithms which can be applied locally, since this is the only way real-time performance can be achieved without dedicated hardware. For this reason, our methods for detecting mirror symmetry in images are truly "low-level", i.e. they do not require some kind of object-related preprocessing or a transformation of the whole object region into a higher-level representation.

Throughout the paper we will assume that only vertical or near vertical symmetry axes in the image are of interest. Moreover, we only look at real mirror symmetry which may be slightly distorted. These assumptions have proven valid for the car-following application. However, the basic methods for measuring intensity symmetry and detecting orientation symmetry can be readily adapted to other situations. The intensity symmetry finder and the symmetry-enhancing edge detector are described as generally applicable algorithms in Section 2 and Section 3, respectively. In Section 4 we discuss the aptness of our approach to the problem of car-following and automatic visual headway control. The same section also describes the CARTRACK system, accompanied by a presentation of results.

2 The Intensity Symmetry Finder

In this section we look at the intensity distribution of an image profile as a continuous one-dimensional function. Any function $G(u)$ can be broken into a sum of its *even part* $G_e(u)$ and its *odd part* $G_o(u)$. We use d to denote the width of the interval on which $G(u)$ is defined. In an image, d is given by the distance between the intersection points of the scan line with the image borders. Intuitively, the degree of symmetry of a function with respect to the origin should be reflected by the relative significance of its even part compared with its odd part. This is essentially the idea underlying our definition of a symmetry measure.

Usually the primary objective is not to measure the degree of global symmetry of a scan line with respect to its center point. We are rather interested in local symmetry intervals that may be centered anywhere within the definition interval. Both the size and the center position of a local symmetry interval has to be found. Let $G(x)$, $0 \leq x < d$, be the intensity distribution along a straight scan line in an image (see the example in Fig. 1). Using the substitution $u = x - x_s$ we can shift the origin of $G(u)$ to any Position x_s on the scan line. A second parameter, w, will be used for varying the size of the interval being evaluated. The parameter x_s ($w/2 \leq x_s \leq d - w/2$) may also be thought of as denoting the location of a potential symmetry axis with w being the width of the symmetric interval about x_s . For a given interval of width w about x_s, we define the even function of $G(x) = G(x_s + u)$ and its odd function such that

$$E(u, x_s, w) := \begin{cases} \frac{G(x_s+u)+G(x_s-u)}{2} & \text{if } -w/2 \leq u \leq w/2 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$O(u, x_s, w) \; := \begin{cases} \dfrac{G(x_s+u) - G(x_s-u)}{2} & \text{if } -w/2 \le u \le w/2 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

For any fixed pair of values x_s and w, the significance of either $E(u, x_s, w)$ or $O(u, x_s, w)$ may appropriately be expressed by their respective energy contents, the integral over their squared values ($Energy[f(x)] := \int f(x)^2 dx$). However, there is the problem that the mean value of the odd function always is zero whereas the even function in general has some positive mean value. Therefore we introduce a normalized even function which has a zero mean value:

$$E_n(u, x_s, w) \; := \; E(u, x_s, w) - \frac{1}{w} \int_{-w/2}^{+w/2} E(v, x_s, w)\, dv \quad (3)$$

With E_n and O we construct a normalized measure for the degree of symmetry by means of the *contrast function*: $C(a, b) = (a-b)/(a+b)$. The contrast function has the property of decreasing sensitivity to absolute differences of increasing magnitudes. The symmetry measure $S(x_s, w)$ is a function of two variables, i.e. it is a number that can be computed for any potential symmetry axis at x_s with respect to an observation interval of width w :

$$S(x_s, w) = \frac{\int E_n(u, x_s, w)^2 du - \int O(u, x_s, w)^2 du}{\int E_n(u, x_s, w)^2 du + \int O(u, x_s, w)^2 du} \quad (4)$$

$$-1 \le S(x_s, w) \le 1$$

We get $S = 1$ for ideal symmetry, $S = 0$ for asymmetry, and $S = -1$ for the ideally antisymmetric case.

The function $S(x_s, w)$ gives a normalized measure for symmetry independent, of the width of the interval being evaluated. For detecting axes of symmetry, however, we need a measure for the significance of an axis. This is a question of scale and depends both on the degree of symmetry within a given interval and on the interval's relative size with respect to the overall extent of the signal. We define a confidence measure [0...1] for the hypothesis that there is a significant symmetry axis originating from an interval of width w about the position x_s as

$$S_A(x_s, w) = \frac{w}{2\, w_{max}} \left(S(x_s, w) + 1 \right), \quad w \le w_{max} \quad (5)$$

w_{max} is the maximal size of the symmetric interval. Considering a global symmetry comprising the entire scan line to be the most significant case would imply $w_{max} = d$. However, w_{max} is better thought of as a limitation of the search interval, meaning that any

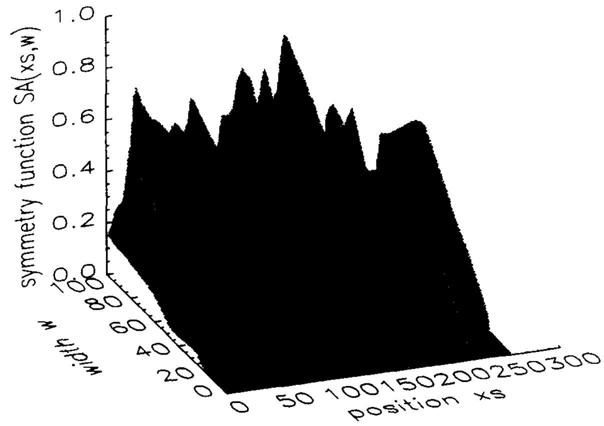

Figure 2: A plot of $S_A(x_s, w)$ for the image row marked in Fig. 1.

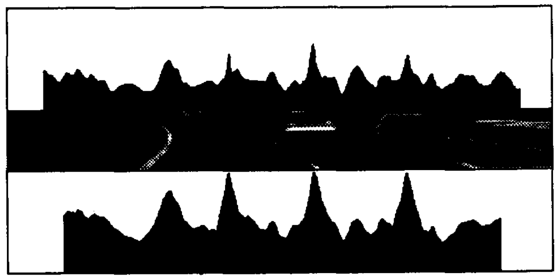

Figure 3: A section of the image in Fig. 1 and two combined one-dimensional symmetry histograms for $w = 65$ (top) and 100 (bottom) resp. ($w_{max} = 100$).

perfectly symmetric interval of width w_{max} and wider corresponds to an indisputable symmetry axis.

Two-dimensional symmetry is formed by a systematic coincidence of one-dimensional symmetries. Elementary mirror symmetries are searched for in image profiles (i.e., along straight scan lines in an image). We find local symmetry axes of the two-dimensional grey level distribution by combining symmetry histograms from several parallel image profiles. This is easily done by summation of the confidence values for each axis position and subsequent maxima detection.

Fig. 1 shows the image for which Fig. 2 and Fig. 3 illustrate the process of intensity-based symmetry detection. Fig. 2 is a plot of the confidence function for symmetry axes in a horizontal range of about half the image width. Large symmetry intervals give rise to steep peaks. Fig. 3 shows the part of the original image for which two combined one-dimensional symmetry histograms ($\sum S_A(x_s, w)$ over 54 image rows) has been computed. The combined histograms contain clearly distinguishable peaks indicating the symmetry axes of the three cars.

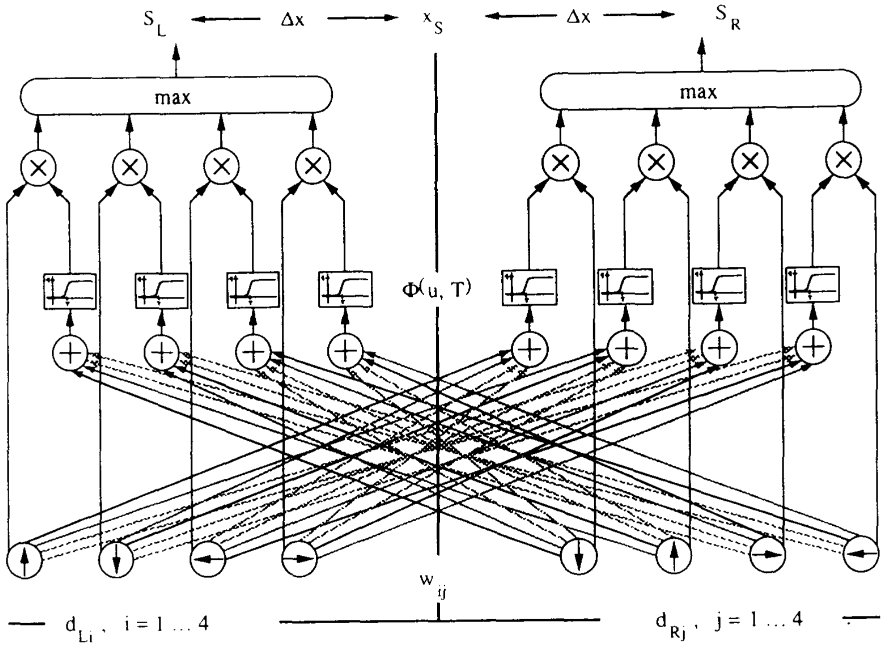

Figure 4: Architecture of the Symmetry-Enhancing Edge Detector element (SEED).

3 The Symmetry-Enhancing Edge Detector (SEED)

When contour curves are used as basic features for symmetry detection the relationship between mutually symmetric points along two sides of an object can be defined as a function of the directions of the two tangents [4]. However, we may want to do without a segmentation process which has to produce differentiable curve segments, or the image may be such that recognizing symmetric objects becomes much easier when symmetry is detected first. Starting from a lower feature level we use local orientation instead of tangent direction and define a set of directional relationships for mirror symmetry. For a given axis of symmetry we try to find pairs of image loci that may contribute to the perception of symmetry by their orientational relationship. In the context of the car-following application, mirror symmetry with respect to a vertical axis is our primary concern and the method for detecting symmetry of local orientation will therefore be described for the case.

This section describes a detector element which receives input from two image loci through a discrete representation of local orientation. The two outputs of the detector element indicate edge points dependent on the strength of the corresponding local orientation features *and* on the degree of compatibility of the two orientation inputs according to a criterion for discrete orientation symmetry. Figure 4 shows the architecture of the detector element. "SEED", as we call it, is a detector element implemented by a feedforward network whose connection weights represent the symmetry conditions. The inputs d_{Li} and d_{Rj} respectively correspond to the outputs of two sets of n directional filters that sample the spatial orientation at two different image loci. For clearness of the illustration, only four directional inputs are drawn on either side.

Let d_{Li} and d_{Rj} $(i, j = 1...n)$ denote the output of the orientation-tuned filters for n discrete directions and let w_{ij} denote the factors specifying a direction's degree of compatibility with a direction on the opposite side of the detector element. Then the left and right output respectively of the detector element is defined as

$$S_L = \max_i \left[\, d_{Li} \, \Phi(\sum_j d_{Rj} w_{ij}, T) \, \right] \quad (6)$$

$$S_R = \max_j \left[\, d_{Rj} \, \Phi(\sum_i d_{Li} w_{ji}, T) \, \right] \quad (7)$$

$$\Phi(u, T) = \frac{1}{1 + \exp(T - u/k)} \quad (8)$$

$\Phi(u, T)$ is a sigmoid threshold function with upper limit 1 and lower limit 0 . (8) contains two parameters that need to be chosen appropriately. k is a scaling

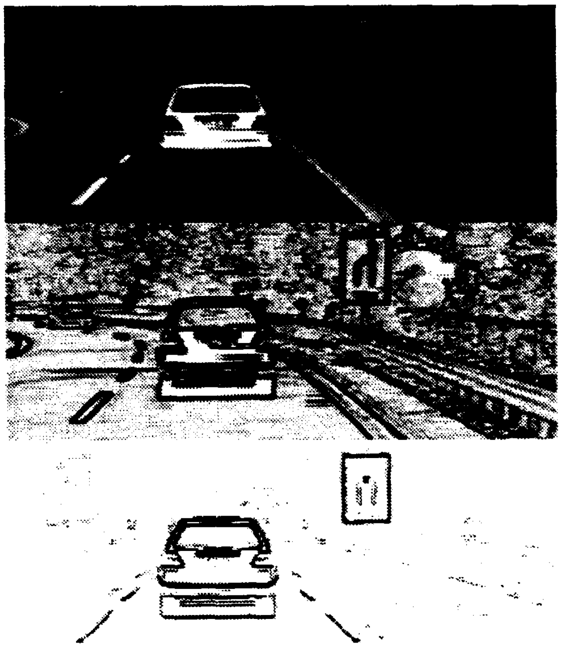

Figure 5: Edge-based object detection. SEED has been applied to the upper image at two positions obtained from the intensity-based symmetry finder. The edges of the car and the road sign clearly stand out (lower image). For comparison, the image in the middle is the result of conventional Sobel-filtering.

factor for normalizing u, such that T can be chosen independently of the weighting factors w_{ij}. The factor k has to be determined once for a given weight matrix. The parameter T should be adapted to the average edge contrast in the image.

Fig. 5 demonstrates the effect of applying SEED to symmetric objects in an image. After the vertical symmetry axis has been found both for the car and the road sign, SEED is applied twice. The directional filters used are implemented by 3×3 Sobel masks. The edges which do not confirm the assumption of mirror symmetry about the given axis are suppressed. Edges arising from the symmetric structures of the two objects clearly stand out.

4 Visually Controlled Car-Following

When studying traffic scenes imaged by a video camera from a driver's position, it becomes clear that symmetry in various forms of appearance plays a major role in this visual world. Many common objects on or beside roads display a distinguished degree of mirror symmetry. This applies in particular to other vehicles and road signs. Of course, restricting the environment model to symmetric structures alone would be all but realistic, but the analysis of our application problem once again suggests that symmetry is an important perceptual concept. In the CARTRACK system symmetry is used for three different purposes:

1. Initial candidates for vehicle objects on the road are detected by means of the assumption that compact image regions with a high degree of (horizontal) intensity symmetry are likely to be nonincidental.
2. Visual tracking of vehicle objects seen from behind is greatly facilitated by the invariance of the symmetry axis under changes of size and vertical position of the object in an image sequence.
3. When using the symmetry constraint, separating the lateral vehicle edges from the image background becomes feasible, even on a low processing level. From the lateral contours of a vehicle, its width in image coordinates can be determined accurately.

4.1 What Visual Information is Needed ?

The visual information that we want to extract from the camera images is the size (in image coordinates) of a leading vehicle. In this section, we briefly address the question to what extent the image size (image width resp.) of an object in front of a moving car is an appropriate visual parameter for the control of driving maneuvers in a car-following situation. An excellent analysis on what kind of visual information a human driver might use has been published by Lee [6]. He emphasizes the role of a simple optic variable, the so called *time-to-collision* (TTC), rather than information about distance, speed, or acceleration/deceleration. Lee's theory of visual control based on information about time-to-collision encourages to carefully study visually controlled behavior in terms of the kind of sensor information really needed for performing a certain task. The TTC at time t is the expected time elapsing until a certain object point would hit the plane which includes the optic center and to which the optical axis is normal. The TTC is important as a dynamic variable, rather than for its absolute values.

Consider the car-following situation as depicted in Fig. 6. The distance $z(t)$ between the two cars decreases when the relative velocity $v(t) = v_A(t) - v_B(t)$ of car A becomes greater than zero. The TTC in world coordinates is defined as

$$\tau(t) = \frac{z(t)}{v(t)} \tag{9}$$

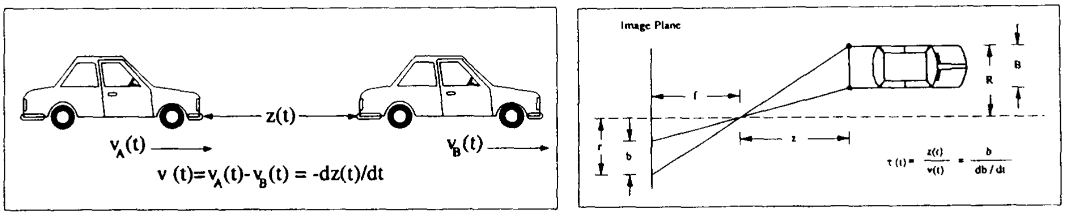

Figure 6: The car-following situation and the model of the imaging geometry.

It can be shown that $d\tau(t)/dt > -0.5$ is, in principle, a sufficient control condition for collision avoidance [6]. Lee elaborates on how various driving maneuvers could be understood in terms of the TTC and its time derivative. He also points out that the TTC is a purely relative and easily accessible optical variable. Consider the imaging model shown in Fig. 6. On the assumption that the observer is moving in the direction of gaze toward a plane to which the motion vector is normal, the TTC can be obtained simply from the distance between two image points (b) of the surface and the rate of change of this distance:

$$b = \frac{fB}{z}\,, \quad \frac{db}{dt} = \frac{-dz\, fB}{dt\, z^2} = \frac{-dz}{dt\, z}\, b \qquad (10)$$

where f is the focal length ($f \ll z$) and B is the distance between the surface points in world coordinates. Since $v(t) = -dz(t)/dt$, substituting (9) into (10) we get

$$\tau(t) = \frac{b}{db\,/\,dt} \qquad (11)$$

In practice, the simplifying assumptions for which (11) holds apply only approximately and the estimated TTC is related to the real TTC by a function depending on the true position of the *focus of expansion* and the surface orientation. This function will vary smoothly over time in most practical cases and therefore will not affect the dynamic of the control system. In a closed-loop system based on computer vision the main problem is not that the estimated TTC may not correspond accurately to the actual physical interpretation of the TTC, but it is rather the signal-to-noise ratio (SNR) of the TTC measurement in the image. If image segmentation could reliably produce object regions, the TTC would therefore be better derived from the change of apparent area. This has recently been demonstrated by Cipolla and Blake [7]. However, for their experiments they pick images that facilitate fitting a closed contour to the border of the looming object.

The purpose of this section is to emphasize the potential of the dynamic image size (width) of leading cars as a visual variable for controlling driving maneuvers like collision avoidance, maintaining a safe distance, and closing up. However, if the relation between the image size of a vehicle and its distance to the observer is known, the absolute image size is sufficient as a magnitude for headway control. The advantage of using the time-to-collision is that it is independent of the actual size of an approaching object. In practice, this is in fact a serious safety advantage since the detection and tracking process may not always lock on to the whole object region in an image sequence but to a conspicuous part of the object, like the number plate or the rear windscreen for example.

4.2 The CARTRACK System

CARTRACK is a real-time implementation, in the sense that we can use it for conducting experiments in test cars on normal roads. This performance is achieved by means of an adaptable processing window whose position and size are controlled by a predictive filter and a number of rules. While the size of the processing window varies depending on the object size, the amount of image data processed by the symmetry finder is kept approximately constant by means of a scan line oriented image compression technique. The result of the symmetry finder is a symmetry histogram. Its highest peak is tentatively taken as the horizontal object position. Then SEED is used to verify and correct the object position by trying to find symmetric edges in addition to the intensity symmetry. Given that the object's symmetry axis has been found with high accuracy, it is relatively easy to correct the vertical object position. Finally, a boundary point is located on either side of the object, providing the start points for a boundary tracking algorithm which uses SEED to "see" only symmetric edge pairs. On both sides of a leading car, a sufficiently long contour segment in the image is extracted and the maximal lateral distance between the two contours is taken as its image width.

In Fig. 7 the image width of a leading car is plotted versus time in units of video frames (1 frame = 40 msec). The data have been measured by CARTRACK in real-time from a video of a car-following experiment. We picked some car on an Autobahn and

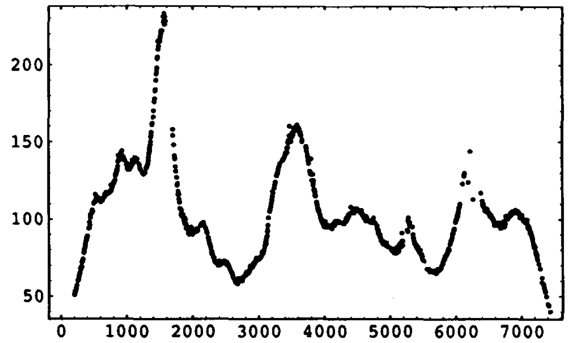

Figure 7: Image width of a leading car vs. time in units of video frames. The plot shows 951 measurements during a 5 minutes car-following experiment.

manually followed it for about 5 minutes at about 90–110 km/h. While we were following the leading car it changed lane five times (around frame times 3710, 4630, 5150, 6160, and 7200). At around frame time 1800 we underran a motorway bridge whose shadow caused a temporary camera blackout. In order to test the robustness of the tracking algorithm we varied the distance to the leading car. The width data plotted in Fig. 7 are in the range from about 40 pixels to 230 pixels. Accordingly, the distance to the leading car varies between 15 and 70 meters approximately (we use a camera lens with 25 mm focal length). During the 5 minutes of car-following covered by Fig. 7 the CARTRACK system ran a total of 1594 processing cycles. 1509 cycles (94.7%) were classified as valid detections and their results were used for the tracking process. Sections of the lateral contours of the leading car were found in 951 cycles (63% of 1509). The width measurements plotted in Fig. 7 are only those that are based on lateral contour segments.

The current implementation had to be trimmed for fast computation. The weight factors representing the symmetry criterion within SEED, for example, had to be chosen as powers of two or zero respectively. The directional filter kernels are simple Sobel masks. The threshold function Φ is implemented as a "hard" threshold. The symmetry finder cannot take advantage of the full vertical image resolution for shortage of processing time. Nonetheless, the CARTRACK system has performed well during tests in the experimental vans of Volkswagen and Daimler-Benz. With the current computer system (single 68040 CPU, 25MHz) we achieve cycle times between 0.2 and 0.4 seconds for the output data to the van's control system.

Fig. 8 shows a short sequence of successive processing windows during a car-following experiment. It also demonstrates what happens when a tracking phase gets interrupted by a change of the leading car. The white vertical line for the measured symmetry axis and the black horizontal line marking the bottom of the object region are only displayed for processing cycles that have been classified as a successful detection attempt. The lateral contour segments used for measuring the object width have been marked white.

5 Conclusions

Symmetry in an image may exist on various levels of abstraction. We deal with the two lowest levels of image features, intensity values and local orientation. A *symmetry finder* is presented based on a normalized measure for the degree of intensity symmetry within scan line intervals. It can detect axes of intensity symmetry without any prior image segmentation step. For detecting symmetry which is formed by a relationship between local orientations, we present a *symmetry-enhancing edge detector*. The edge detector extracts pairs of edge points if the local orientations at these points are mutually symmetric with respect to a certain axis. Other edge points are suppressed.

We believe that symmetry is a powerful concept that facilitates object detection and recognition in many situations. Many objects display some degree of overall symmetry, offering to exploit symmetry constraints as early as in the low-level processing stages already. In the context of stereo and motion processing, we have shown earlier how visual processing can benefit when higher level knowledge about the geometry of the expected environment is used as a "hard constraint" in early image processing [8]. The symmetry enhancing edge detector in conjunction with the symmetry finder is another example of a kind of algorithm we call *minimal*, in the sense that no irrelevant information is extracted from the scene when an expected "normal" situation is analyzed.

Computer vision-based car-following is the application for which the symmetry detection methods have been developed. For this application one needs computationally efficient methods, which also have the robustness required for processing outdoor image data. We found that compact image regions with a high degree of (horizontal) intensity symmetry are likely candidates for the initial detection of vehicles on the road, even when the viewing conditions are unfavorable. Visual tracking of automobiles from behind can be done fast and reliably using the symmetry axis as the guiding object feature. Finally, when the edge detector only "sees" symmetry edges, extracting the car's lateral boundaries becomes possible even in situations where background and object are hard to distinguish.

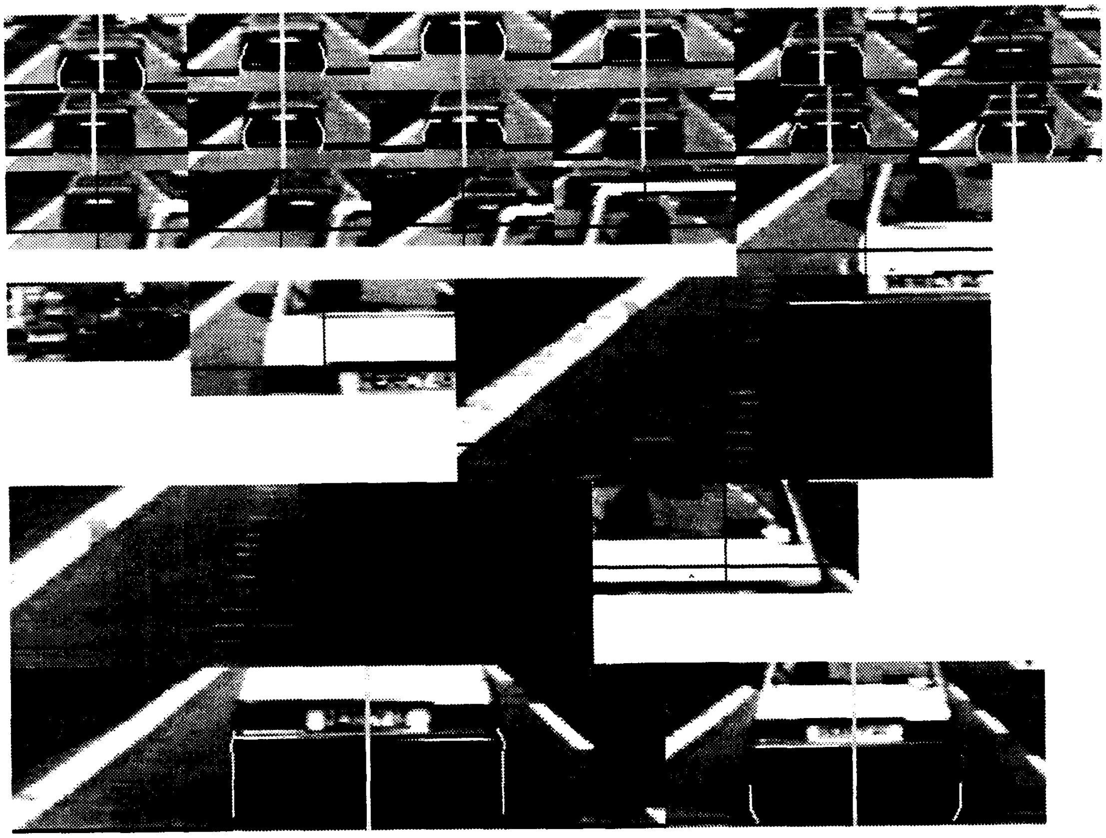

Figure 8: The CARTRACK system in operation. A sequence of 24 successive processing windows shown in the order from left to right and from top to bottom.

References

[1] S. A. Friedberg, "Finding Axes of Skewed Symmetry", *Computer Vision, Graphics, and Image Processing* **34**, pp. 138–155, 1986.

[2] G. Marola, "Using Symmetry for Detecting and Locating Objects in a Picture", *Comp. Vision, Graph., and Image Proc.* **46**, pp. 179–195, 1989.

[3] P. Saint-Marc and G. Medioni, "B-Spline Contour Representation and Symmetry Detection", in *Computer Vision – ECCV '90* (Lecture Notes in Comp. Science 427, O. Faugeras, Ed.), Springer, Berlin, pp. 604–606, 1990.

[4] A. Gross and T. Boult, "SYMAN: a Symmetry Analyser", *in Proc., Computer Vision and Pattern Recognition, Hawaii*, pp. 744–746, 1991.

[5] Y. Yeshurun, D. Reisfeld, and H. Wolfson, "Symmetry: a context free Cue for Foveated Vision", in *Neural Networks for Perception, Vol. 1, Human and Machine Perception* (H. Wechsler, Ed.), Academic Press, San Diego, pp. 477–491, 1992.

[6] D. Lee, "A Theory of Visual Control of Braking Based on Information about Time-To-Collision", *Perception* **5**, pp. 437–459, 1976.

[7] R. Cipolla and A. Blake, "Surface Orientation and Time to Contact from Image Divergence and Deformation", in *Computer Vision – ECCV '92* (Lecture Notes in Comp. Science 588, G. Sandini, Ed.), Springer, Berlin, pp.187–202, 1992.

[8] H.A. Mallot, Th. Zielke, K. Storjohann, and W. von Seelen, "Topographic Mapping for Stereo and Motion Processing", in *Proc., SPIE* **1382**, *Intell. Robots and Comp. Vision IX: Neural, Biological, and 3-D Methods, Boston*, pp. 397–408, 1990.

[9] B. Ulmer, "VITA – An Autonomous Road Vehicle (ARV) for Collision Avoidance in Traffic", in *Proc., Intelligent Vehicles '92, Detroit, USA*, pp. 36–41, 1992.