Intensity and Edge-Based Symmetry Detection Applied to Car-Following [*]

Thomas Zielke, Michael Brauckmann, and Werner von Seelen

Institut für Neuroinformatik, Ruhr-Universität, 4630 Bochum, Germany.

Abstract. We present two methods for detecting symmetry in images, one based directly on the intensity values and another one based on a discrete representation of local orientation. A *symmetry finder* has been developed which uses the intensity-based method to search an image for compact regions which display some degree of mirror symmetry due to intensity similarities across a straight axis. In a different approach, we look at symmetry as a bilateral relationship between local orientations. A *symmetry-enhancing edge detector* is presented which indicates edges dependent on the orientations at two different image positions. SEED, as we call it, is a detector element implemented by a feedforward network that holds the symmetry conditions. We use SEED to find the contours of symmetric objects of which we know the axis of symmetry from the intensity-based symmetry finder. The methods presented have been applied to the problem of visually guided car-following. Real-time experiments with a system for automatic headway control on motorways have been successful.

1 Introduction

Our interest in symmetry detection originates from the problem of car-following by Computer Vision, i.e. the problem of how an automobile equipped with a camera and control computers can be programmed to automatically keep a safe driving distance to a car in front. There are three major visual tasks the system has to cope with:

1. Detecting leading cars. 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's rear while its image position and size may vary greatly.
3. Accurate measuring of the car's dynamic image size needed for the speed control.

The methods presented here exploit the symmetry property of the rear view 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 edges that have a symmetric counterpart with respect to a given axis.

[*] This work has been supported by the German Federal Ministry of Research and Technology (BMFT) and by the Volkswagen AG (VW). PROMETHEUS PRO-ART Project: ITM8900/2

Lecture Notes in Computer Science, Vol. 588
G. Sandini (Ed.)
Computer Vision – ECCV '92
© Springer-Verlag Berlin Heidelberg 1992

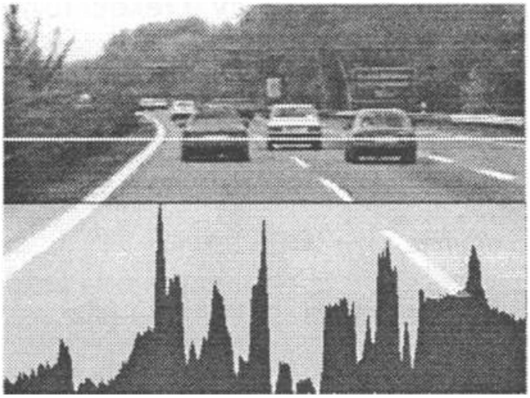

Fig. 1. A typical image in a car-following situation. The overlay plot in the lower half shows the intensity distribution along the scan line marked in the upper half of the image.

In the following, we briefly review previous work on *symmetry detection* and *finding symmetry axes*. 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. Other algorithms assume 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 is based on central moments too, but, in contrast to [1], it directly uses 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.

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. To our knowledge no "low-level" methods for symmetry detection in images have been reported upon in the literature to date. That is, all the methods we found require some kind of object-related preprocessing or a transformation of the whole object region into a higher-level representation.

2 Finding Axes of Intensity Symmetry

Two-dimensional symmetry is formed by a systematic coincidence of one-dimensional symmetries. Elementary mirror symmetries are searched for along straight scan lines in an image. In this section we look at the intensity distribution along a scan line as a continuous function.

2.1 A Measure for Local Symmetry within a 1D Intensity Function

Any function $G(x)$ can be written as the sum of its even component $G_e(x)$ and its odd component $G_o(x)$, provided the origin is at the center of the definition interval. We use d to denote the width of the interval on which $G(x)$ 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 component compared with its odd component. 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. Rather, in the context of object recognition, we are interested in local symmetry intervals. Both the size and the center position of such intervals have to be found. For this, two additional parameters are introduced. One (x_s) that permits shifting the origin of the 1D function, and another (w) is for varying the size of the interval being evaluated. The values x_s are restricted by d and by the actual choice of w:

$$x_s = x_0 - \frac{d-w}{2} \ldots x_0 \ldots \frac{d-w}{2} + x_0 \ , \ w \leq d \tag{1}$$

x_s 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 . We define the even function of $G(x - x_s)$ and its odd function for a given interval of width w such that

$$E\left(x, x_s, w\right) := \begin{cases} \frac{1}{2}\left(G(x - x_s) + G(x_s - x)\right) \text{ if } |x - x_s| \leq w/2 \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{2}$$

$$O\left(x, x_s, w\right) := \begin{cases} \frac{1}{2}\left(G(x - x_s) - G(x_s - x)\right) \text{ if } |x - x_s| \leq w/2 \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{3}$$

For any fixed pair of values x_s and w, the significance of either $E(x, x_s, w)$ or $O(x, x_s, w)$ may appropriately be expressed by their respective energy contents, the integral over their squared values ($Energy[f(x)] := \int f(x)^2 dx$). However, the problem arises that the mean value of the odd function always is zero whereas the even function in general has some positive mean value. This bias has to be subtracted from the even function in order to render the two energy quantities comparable in the sense that their dissimilarity indicates symmetry or antisymmetry respectively. We introduce a normalized even function which has a zero mean value:

$$E_n\left(x, x_s, w\right) := E\left(x, x_s, w\right) - \frac{1}{w} \int E\left(x, x_s, w\right) dx \tag{4}$$

With E_n and O we construct a normalized measure for the degree of symmetry by means of the *contrast function* : $C\left(a, b\right) = (a - b)/(a + b)$. 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 x_s with observation interval w.

$$S\left(x_s, w\right) = \frac{\int E_n\left(x, x_s, w\right)^2 dx \ - \ \int O\left(x, x_s, w\right)^2 dx}{\int E_n\left(x, x_s, w\right)^2 dx \ + \ \int O\left(x, x_s, w\right)^2 dx} \ ; \ -1 \leq S\left(x_s, w\right) \leq 1 \tag{5}$$

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

2.2 The Symmetry Finder

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

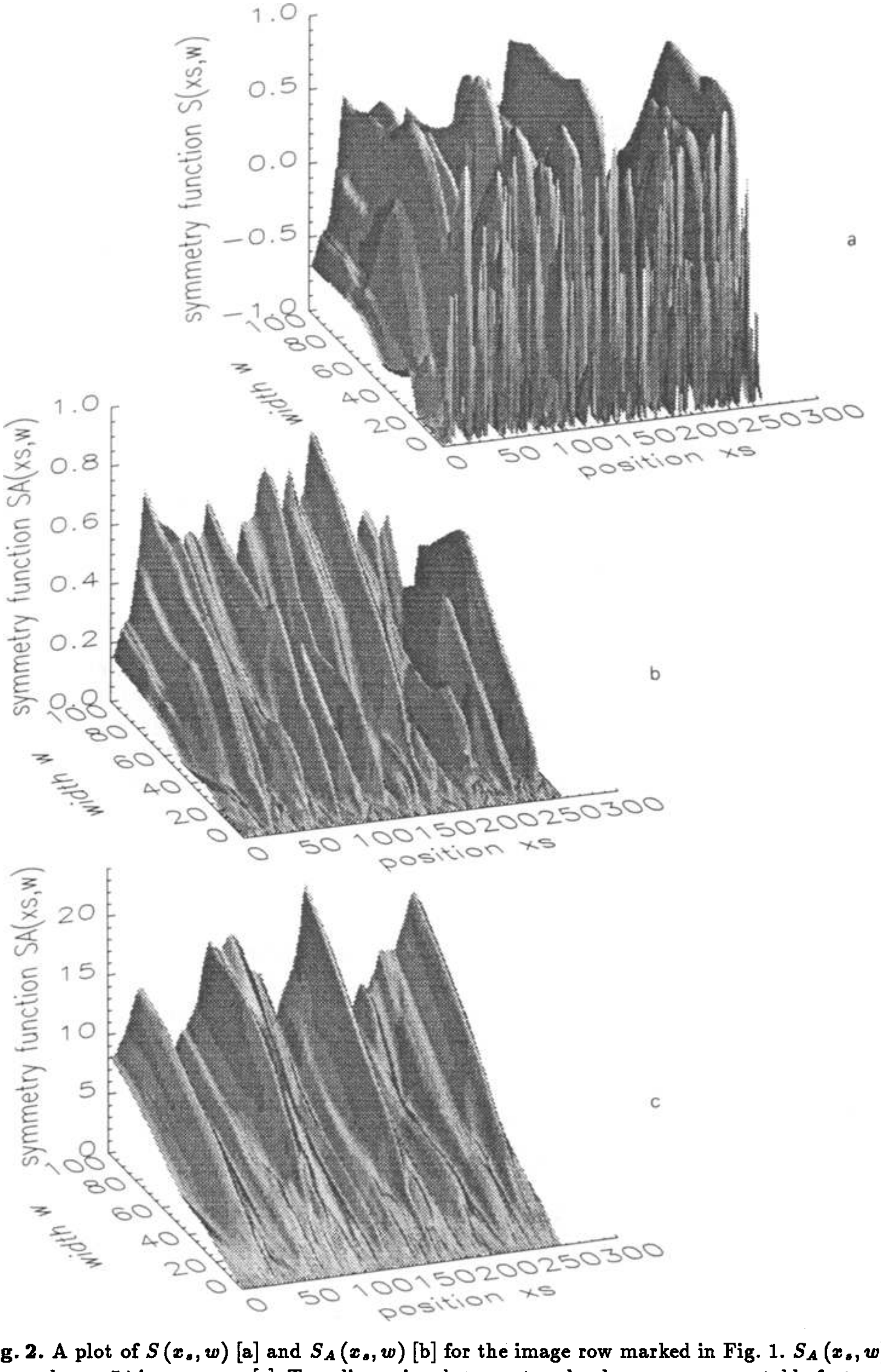

Fig. 2. A plot of $S(x_s, w)$ [a] and $S_A(x_s, w)$ [b] for the image row marked in Fig. 1. $S_A(x_s, w)$ summed over 54 image rows [c]. Two-dimensional symmetry clearly emerges as a stable feature.

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 a symmetry axis originating from an interval of width w about the position x_s as

$$S_A(x_s, w) = \frac{w}{2\,w_{max}} \left(S(x_s, w) + 1 \right), \quad w \leq w_{max} \tag{6}$$

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

Two-dimensional symmetry detection requires the combination of many symmetry histograms (accumulated confidence for symmetry axis vs. position of axis). This is easily done by summation of the confidence values for each axis position, provided the symmetry axis (axes) is (are) straight. Figure 2 illustrates the process of intensity-based symmetry detection. The image for which the simulation results are presented here is shown in Fig. 1 . The symmetry position x_s is varied in a horizontal range of about half the image width. Figure 2a is a plot of the function $S(x_s, w)$ for one image row across the three cars. Many local symmetry peaks exist for small w's. Figure 2b is a plot of the confidence function for symmetry axes. Large symmetry intervals give rise to steep peaks. After summation of $S_A(x_s, w)$ over a number of image rows, clearly distinguishable peaks emerge, indicating the symmetry axes of the three cars, Fig. 2c .

3 Detecting Symmetry of Local Orientations

3.1 Analysis of Orientation Symmetry

When curves (e.g. contour segments) are used as basic features for symmetry detection [3] the relationship between mutually symmetric points along two curves can be defined as a function of the directions of the two tangents. 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 symmetry relationship for it.

Figure 3 depicts two banks of directional filters and all possible pairs of directions between them. Having a vector of eight direction-specific orientation values at each image position is actually common in practice (e.g. when Sobel filter masks are being used for preprocessing). The problem now is to define a symmetry relationship between the two local orientations each represented by a vector of n directional values. There are $n^2/2$ different pairs of directions for which it has to be decided whether this combination agrees with our notion of symmetry. It is not always obvious which pairs of directions are to be regarded symmetric. In Fig. 3 the direction pairs are grouped into six categories.

Ideal symmetry means that the two directions can be mapped onto each other by a reflection about the symmetry axis and a subsequent reversal of direction (the edge polarities at the two points have opposite sign). In case of *inverse symmetry*, the two directions are the reflection of each other. With *antisymmetry* we denote cases which are the opposite of what we consider ideal symmetry. *Non-symmetry* covers all cases where the pair of directions is considered neither symmetric nor antisymmetric, i.e. it is the neutral relation. The category *near (inverse) symmetry* expresses approximate symmetry.

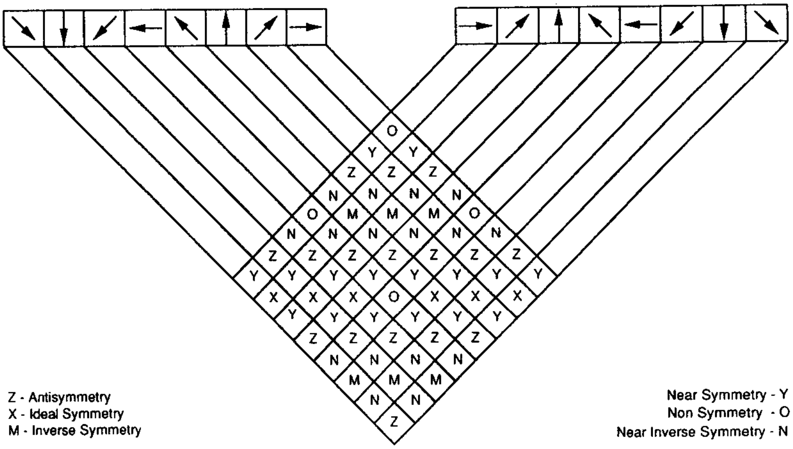

Fig. 3. Symmetry as a relationship between local orientations. The pairs of directions are categorized according to their degree and kind of symmetry respectively.

3.2 The Symmetry-Enhancing Edge Detector (SEED)

If orientation is measured by a small number of directional filters which may have overlapping angular sensitivity, all the above types of symmetry relations can be present concurrently between two image loci, varying in degree of significance. We have developed a method which combines evidence for the different categories of orientation symmetry. It serves as a mechanism for detecting edges that are related to other edges by a certain degree of symmetry. In the following we describe 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. For clearness of the illustration, only four directional inputs are drawn on either side. The basic idea is to view the table of symmetry relations in Fig. 3 as a matrix of weight factors (w_{ij}) and compute weighted sums of the directional inputs (d_{Li},d_{Rj}), which then serve as inhibitory signals for the corresponding directional inputs (d_{Rj},d_{Li}) on the respective other side of the detector. SEED is a detector element implemented by a feedforward network whose connection weights represent the symmetry conditions.

The weighted sums represent the degrees of accumulated evidence for the compatibility of a given edge direction on either side of the detector element with the local orientation present at the respective other side's input. The degree of evidence is thresholded by a sigmoid function (Φ). The threshold function prevents a strong directional signal from causing an edge signal when there is only weak symmetry. By taking the products of the directional inputs and the normalized degrees of symmetry a second layer of discrete orientation representation is constructed, now with an enhanced sensitivity to orientation patterns that have a symmetric counterpart across a given axis.

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 resp. of the detector element is defined as

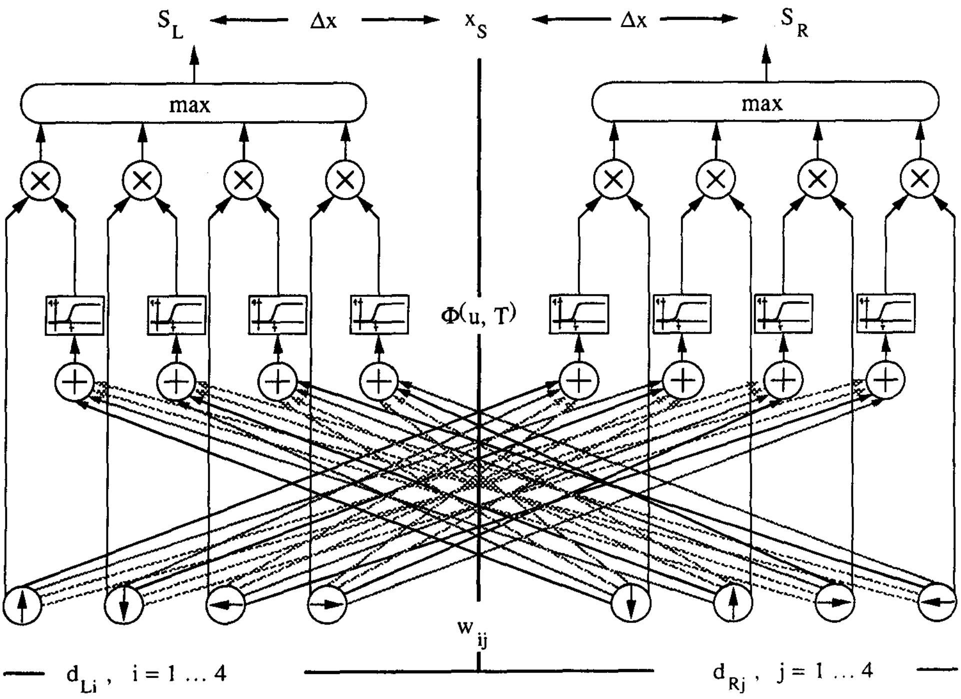

Fig. 4. **Architecture of the Symmetry-Enhancing Edge Detector element (SEED).**

$$S_L = \max_i \left[d_{Li}\ \Phi(\sum_j d_{Rj} w_{ij}, T) \right] \; ; \; S_R = \max_j \left[d_{Rj}\ \Phi(\sum_i d_{Li} w_{ji}, T) \right] \tag{7}$$

$\Phi(u,T)$ is a sigmoid threshold function with upper limit 1 and lower limit 0 :

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

Equation (8) contains two parameters that need to be chosen appropriately. k is a scaling 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 $\mathbf{W}$. We compute k from the product of $\mathbf{W}$ and an angular influence matrix $\mathbf{A}$, with $a_{ij} = \| \cos(2\pi(i-j)/n) \|$, $(i, j = 1...n)$. The maximum of the elements of $\mathbf{W} \cdot \mathbf{A}$ is a good approximation for the "gain" caused by the weighted summation of the directional inputs. The parameter T has to be determined experimentally. For best results, T should be adapted to the average edge contrast in the image. We usually set T to half the value of the average response of the directional filter for the strongest direction at edges.

Figures 5 and 6 show results. With the intensity-based symmetry finder we determine the position of the symmetry axis. At the axis position found, SEED is applied. The weight vector [Z,X,M,O,Y,N] (see Fig. 3) used to process the images in this section is [-2,2,2,0,1,0]. The edges which do not confirm the assumption of mirror symmetry about the given axis are suppressed. Edges arising from the symmetric structures of an object clearly stand out.

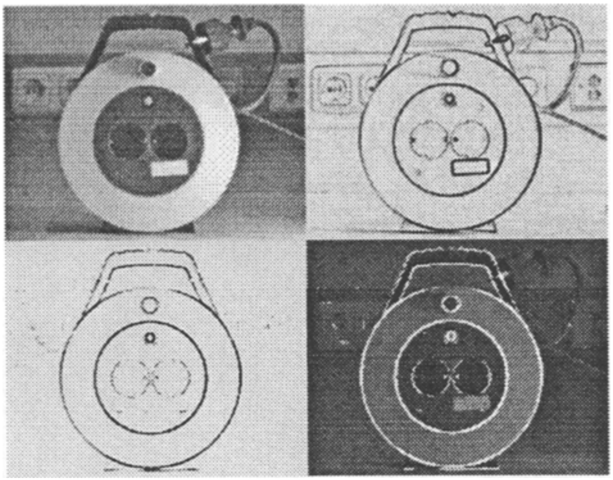

Fig. 5. A picture of a household cable drum. Viewed from the front, it is a mirror-symmetric object with some asymmetric internal structures. Applying a Sobel edge filter to this image results in the upper right image. l.l.: The result of applying SEED at the position of the symmetry axis . l.r.: The symmetric edges (binarized) superimposed onto the original image.

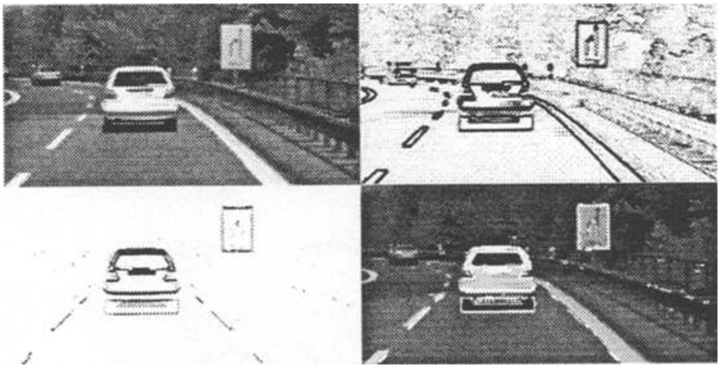

Fig. 6. Multiple object detection. In addition to the processing steps demonstrated in Fig. 5, SEED is applied twice, picking out the edges of the road sign as well.

4 The CARTRACK System

Our methods for symmetry detection and symmetry filtering have been used to build a vision system (dubbed CARTRACK) for detecting and tracking cars and other vehicles having an approximately mirror-symmetric rear view. 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 a car's rear 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 image width can be determined accurately.

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.

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. Nonetheless, the CARTRACK system has performed well during tests for an intelligent cruise control system in an experimental van of Volkswagen (VW). With the current computer system, based on a single MC68040 microprocessor (25MHz), we achieve a cycle time of about 300 milliseconds for the output data to the van's control system.

5 Conclusion

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.

The application for which the symmetry methods have been developed is vision-based car-following. 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.

References

1. Friedberg, S. A. : Finding Axes of Skewed Symmetry. Computer Vision, Graphics, and Image Processing 34 (1986) 138–155
2. Marola, G. : Using Symmetry for Detecting and Locating Objects in a Picture. Computer Vision, Graphics, and Image Processing 46 (1989) 179–195
3. Saint-Marc, P., Medioni, G. : B-Spline Contour Representation and Symmetry Detection, Proceedings, First European Conf. on Computer Vision, Antibes, France (1990) 604–606