ELSEVIER

Pattern Recognition Letters 17 (1996) 335-343

Pattern Recognition Letters

Artificial neural networks in real-time car detection and tracking applications ★

Christian Goerick *, Detlev Noll, Martin Werner

Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany

Received 20 December 1995

Abstract

In this paper we present three applications representing different approaches to car detection and tracking problems. In conjunction with problem-adequate preprocessing, artificial neural networks are utilized to solve the essential subproblems. The approaches draw their sensory input from vision and times series based cues. All three applications satisfy real-time requirements on German *autobahnen*.

Keywords: Real-time computer vision; Object detection; Classification; Deterministic annealing; Feature matching

1. Introduction

In recent applications, the need for robust nonlinear signal processing and classification frequently arises. Artificial neural networks (ANNs), in conjunction with an appropriate preprocessing, are well suited to contribute to the solution of these problems. In this paper we present three different applications in the field of automotive related problems.

Various approaches to the problem of vehicle tracking have been published recently. In contrast to the approach in (Dickmanns et al., 1993; Dickmanns et al., 1994) and (Thomanek et al., 1994) we make no use of a dynamic system model, i.e. no explicit prediction for the tracking task is needed. Furthermore, our systems do not require any information about camera parameters, etc.

The systems described in (Koller et al., 1992; Koller et al., 1994) require a stationary camera and a dynamic motion model. No explicit classification instead an obstacle detection and tracking is performed in the approaches mentioned above as well as in two other recently published results (Bellon et al., 1994; Zielke et al., 1992).

We commence with a model-free approach to car detection and tracking in scenes taken from German *autobahnen*. The second approach deals with the same problem, but in this case a model based matching precedes the neural classification. In the final part of the contribution, a solution of a time series analysis problem to perform intelligent cruise control is sketched.

Throughout this paper we intend to stress the importance and usefulness of neural-like computations for problems involving natural environment and deterministic as well as random perturbations.

★ This work has been partially supported by the Daimler-Benz AG, the Adam Opel AG and the PROMETHEUS project.

* Corresponding author.

SSDI 0167-8655(95)00129-8

2. Feature based car detection and tracking

2.1. System overview

The feature based car detection and tracking system presented in this section is part of the CARTRACK system (Brauckman et al., 1994). It is a specialized monocular visual sensor system for detecting, tracking, and measuring rear or frontal views of automobiles in image sequences taken from the view point of a following car. The parts of the system described in this paper are the preprocessing and classification/detection modules. The classification and detection task is performed by means of ANNs. On a more abstract level, the approach resembles a fast hypothesis generation and testing method. The robustness and speed of the approach are gained by the preprocessing method as well as the integral treatment of image regions.

2.2. Preprocessing

The grey-scale images are preprocessed by a method we call local orientation coding (LOC) (Goerick and Brauckmann, 1994). The image features obtained are bit strings, each representing a binary code for the directional grey-level variation in the pixel's neighborhood. In a formal fashion the operator is defined as

$$b'(n,m) = \sum_{i,j \in I} k(i,j) \cdot u\big(\, b(n,m) - b(n+i, m+j) - t(i,j)\,\big) \qquad (1)$$

where $b(n,m)$ denotes the (grey-scale) input image, $b'(n,m)$ the output representation, $k(i,j)$ a coefficient matrix, $t(i,j)$ a threshold matrix, I a neighborhood (usually N_4 or N_8) and $u(\cdot)$ the unit step function. The matrices may have negative index values. All variables are integers. The output representation consists of labels, with each label corresponding to a specific orientation of the neighborhood. For a N_4 and a N_8 neighborhood on regular square grids, suitable choices for the coefficient matrices are

$$\begin{bmatrix} 0 & 1 & 0 \\ 2 & R & 4 \\ 0 & 8 & 0 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 2 & 4 \\ 8 & R & 16 \\ 32 & 64 & 128 \end{bmatrix}, \qquad \begin{array}{l} n \\ \lfloor\!\rightarrow m\,, \end{array}$$

where R is the reference position. This choice for N_4 leads to a set of labels $b'(n,m) \in [0,\ldots,15]$ corresponding to certain local structures (see Fig. 1).

The choice of the coefficients and the formulation of the operator gives rise to some properties:

- Due to the unique separability of the sum into its components, the information of the local orientation is preserved.
- The approach is invariant to absolute intensity values.
- The search for certain structures in the image reduces to working with different sets of labels. For horizontal structures mainly the labels 1, 8 and 9 have to be considered.

An adaption mechanism for the parameters $t(i,j)$ of the coding algorithm yields a high level of flexibility with respect to illumination conditions (Goerick and Brauckmann, 1994).

The local orientation coding is combined with a histogram technique. A lateral (resp. vertical) histogram h_{lat} of the output representation for row n and feature f (label) is computed according to

$$h_{\mathrm{lat}}(n,f) = \sum_{m} \delta\big(b'(n,m) - f\big),$$

$$h_{\mathrm{vert}}(m,f) = \sum_{n} \delta\big(b'(n,m) - f\big), \qquad (2)$$

where δ denotes the Kronecker delta function. Out of these orientation code histograms we obtain estimates for image regions likely to show a vehicle on the road based on the increased occurrence of certain sets of labels (Goerick and Brauckmann, 1994). For example, horizontal lower boundaries of objects can be detected by investigating the lateral histograms of the feature set $\{1,8,9\}$. The feature vector finally presented to the neural network comprises concatenated lateral and vertical histograms of the region under consideration. The histograms are scaled to a prior determined size to gain scale invariance.

The set of lateral and horizontal histograms can be considered as orthogonal projections of the labels. Therefore, they expose a certain resemblance to a modified Hough or Radon transform (Jain, 1989). However, as opposed to a complete transform, due to the coding two projections are sufficient for our application. Another advantage of the use of histograms

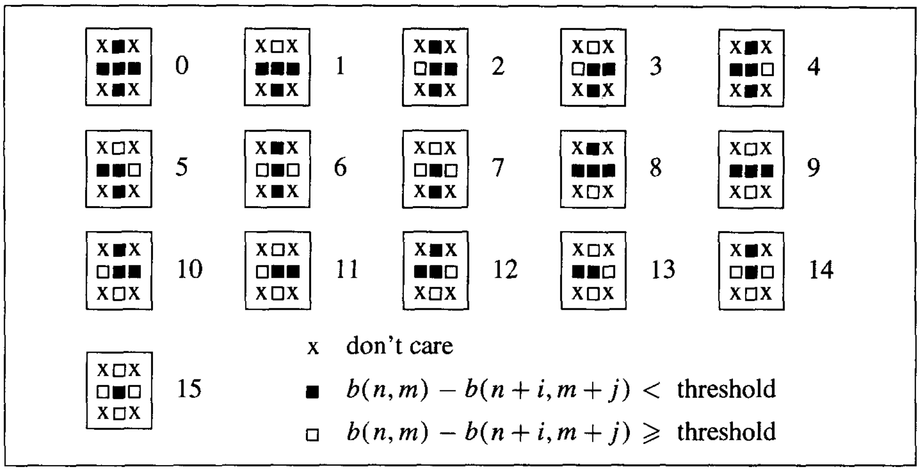

Fig. 1. Coding results for N_4.

is that object boundaries need not to be closed contours. Finally, the work with the projections, being one-dimensional signals instead of two-dimensional images or high-dimensional transformations, gives rise to a significant gain in processing speed.

2.3. Classification

The previously sketched preprocessing removes the dependence on the absolute grey-values from the data. Furthermore, the shift and scale variance are removed to some extent. The classifier still has to cope with partial occlusion, varying illumination conditions, tilt of an object, differently resolved structures depending on the distance of the object under consideration, noise and perturbations induced by the recording and processing equipment, different viewpoints and different kind of cars with different shapes and colors. These possible factors influence the classification of the object. Additionally, the classifier should be able to generalize from relative few training examples to the necessary features characterizing a car. Therefore, a neural network has been chosen for solving the classification task. It is a feed-forward neural network with one hidden layer trained by the error back-propagation algorithm (Hertz et al., 1991; Rumelhart et al., 1986).

A set of neural networks with slightly varying dimensions is currently in use. The number of neurons in the first layer is typically between 350 and 450, and the number of neurons in the hidden layer is 10 to 40. The database comprises 2000 examples for the learning phase.

These networks are known to be universal approximators for any continuous valued function (Hornik et al., 1989). Furthermore, it is shown that these structures can, with some small modifications, approximate *a posteriori* probabilities in the sense of a Bayesian classifier (Finke and Müller, 1993). A lower bound for the number of hidden neurons is determined by the approach taken from (Mirchandani et al., 1989). A near optimal number of hidden neurons is determined by the statistics of experiments. The inputs for the classifier are certain subsets of the histograms. The output is the class of the region under consideration. The analysis of the classifier and alternative solutions, which might improve the performance, are subject to current research. Of special interest is the introduction of *a priori* knowledge into the neural network and the predetermination of the optimal structure for a specific task.

The complete system has been implemented and extensively tested on the Mercedes Benz VITA II test vehicle (Brauckman et al., 1994). The system can handle three objects concurrently. The time required for an initial search of an object is 120 ms, the cycle time for object tracking is 60 ms. The operating range for detection is 5 m to 80 m, whereas once the objects have been detected, they can be tracked up to a distance of 100 m. For a further evaluation of the system see (Bohrer et al., 1995).

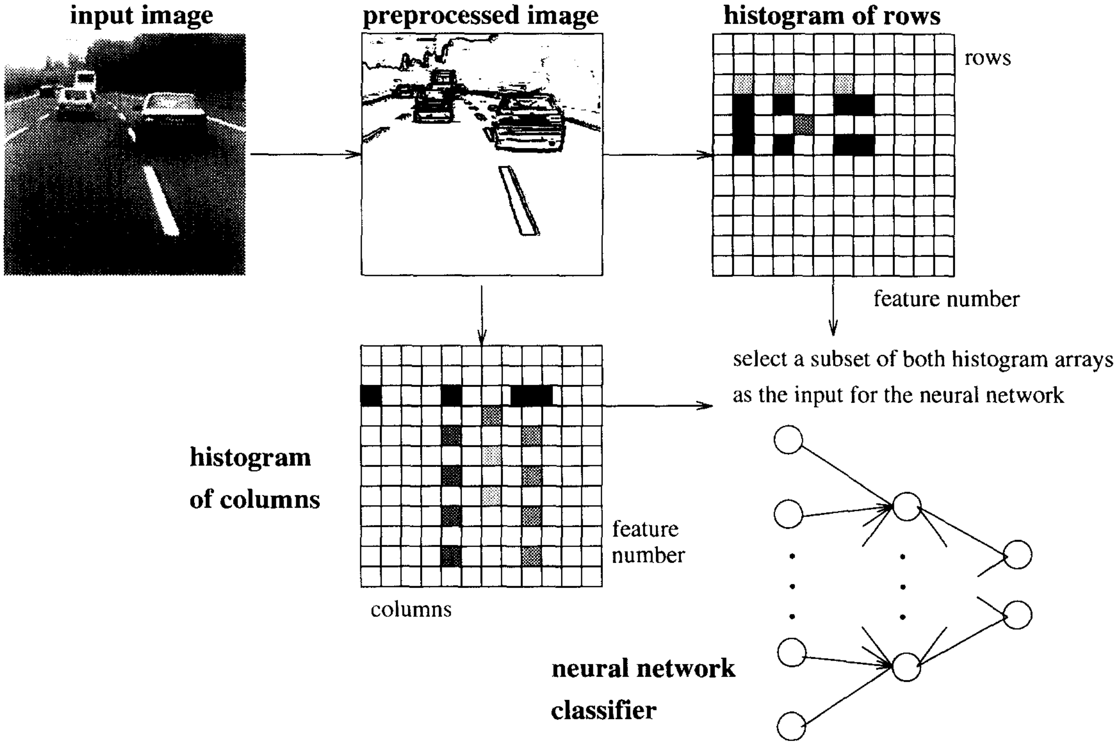

Fig. 2. Schematics of the detection process.

3. Model based object classification and tracking

3.1. System overview

In this section a feature and model based approach to real-time vehicle tracking and classification is described (Noll and Werner, 1995; Noll et al., 1995). After an optimal correspondence between model and image features has been established by means of a deterministic annealing algorithm, a matching vector is derived. This vector is classified by a multi-layer perceptron. The use of preprocessing hardware for the feature extraction in conjunction with a standard DSP module enables the system to run at a rate of 8–12 frames per second.

3.2. Establishing feature correspondence

The correspondence between model and image features is established by a constrained elastic net (CENET) algorithm iteratively minimizing the energy function

$$E = -\alpha K \sum_j \ln \sum_i s_{ij} \exp\left(- \frac{|x_i - y_j|^2}{2K^2} \right) + \beta \sum_j |\hat{y}_j - y_j|^2 . \tag{3}$$

Here, x_i and $y_j \in [0,1]^2$ are image and model features, respectively, $s_{ij} \in [0,1]$ is a similarity measure of these features, α and β are positive constants, K is an annealing parameter, and $\hat{y}_j$ is a reference model feature. This reference model feature is calculated by

$$\hat{y}_j = T_{\hat{a}}\left(y_j^0\right),$$

where T_a is a transformation with transformation parameter a, and y_j^0 is the model feature as stored in the database. The optimal transformation parameters $\hat{a}$ are obtained by a LSE (least square estimation) such that

$$\sum_j (\hat{y}_j - y_j)^2$$

is minimal. For affine transformations T_a the LSE can be efficiently calculated.

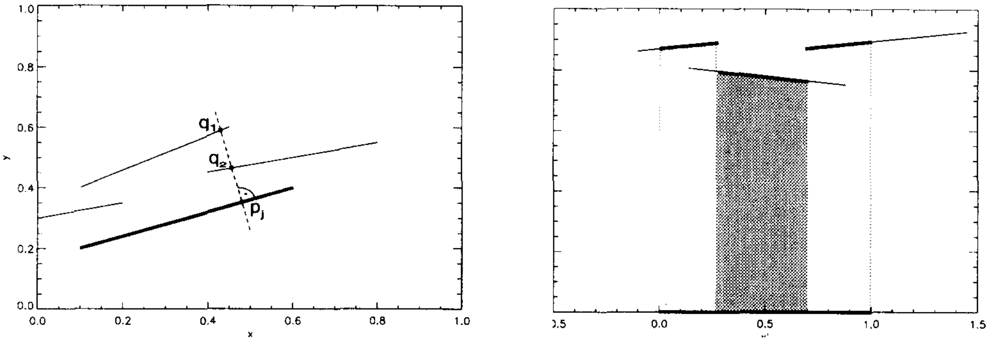

Fig. 3. A model feature (bold line) and three image features in the original coordinate system (left) and after the transformation (right).

Eq. (3) can be derived by the maximum entropy method (Kapur and Kesavan, 1992). A similar approach has been used by (Rose et al., 1993) yielding energy functions to solve the travelling salesman problem and clustering problems by deterministic annealing. The terms weighted by the constants α and β measure the correspondence of image and model features and the deformation of the model, respectively. Minimization of (3) is performed by deterministic annealing (Rose et al., 1993) according to

$$\Delta y_j \sim -K\frac{\partial E}{\partial y_j} \tag{4}$$

$$= c_1 \sum_i w_{ij}\left(x_i - y_j(n)\right) + c_2\left(\hat{y}_j(n) - y_j(n)\right), \tag{5}$$

$$w_{ij} = \frac{s_{ij}\exp\left(-\frac{|x_i - y_j(n)|^2}{2K^2(n)}\right)}{\sum_k s_{kj}\exp\left(-\frac{|x_k - y_j(n)|^2}{2K^2(n)}\right)}. \tag{6}$$

The annealing parameter K can be interpreted as a range of influence. It is decreased in the course of the iteration process at a logarithmic rate.

If this algorithm is applied to consecutive frames of a sequence, object tracking is achieved by using the detected position in the previous frame as the initial position for the current. In the application described in Section 3.4 line segment features extracted by hardware are used to achieve real-time performance. The adaption of the algorithm to this kind of features is described in (Noll and Werner, 1995) in detail.

3.3. Classification

Having established an optimal feature correspondence between the image and one or more models, the next step is to decide whether or not the object belongs to one of the classes represented by the models. If only a single model is used (as is usually the case in tracking problems) the classifier performs as a hypotheses tester. The input to the classifier is a matching vector $\boldsymbol{m}$. Each component m_j of $\boldsymbol{m}$ is related to a model feature and describes the match of that feature to the set of all image features. This takes into account that a model feature can match several small image features and is contrary to e.g. (Deriche and Faugeras, 1990) proposing the *Mahalanobis* distance as a matching measure for line segments. Furthermore, the algorithm proposed in the following can be extended to features other than line segments.

Each point $\boldsymbol{p}_j$ of the model feature j is assigned a value

$$\text{match}(\boldsymbol{p}_j, \boldsymbol{q}_i) = \max\left(\cos(\Delta\phi_{ij})\,(1 - \kappa|\boldsymbol{p}_j - \boldsymbol{q}_i|), 0\right) \tag{7}$$

in the interval $[0, 1]$ measuring its match to image feature i. Here, $\boldsymbol{q}_i$ is the point of intersection of image feature i with the line perpendicular to model feature j passing through $\boldsymbol{p}_j$ (as indicated in Fig. 3), $\Delta\phi_{ij}$ is the angle enclosed by the features, and κ is some positive constant. Thus, match$(\boldsymbol{p}_j, \boldsymbol{q}_i)$ takes the distance and the angle difference of the features into account.

Using (7), the component m_j of the match vector is calculated by integration over the best matches of all model feature points:

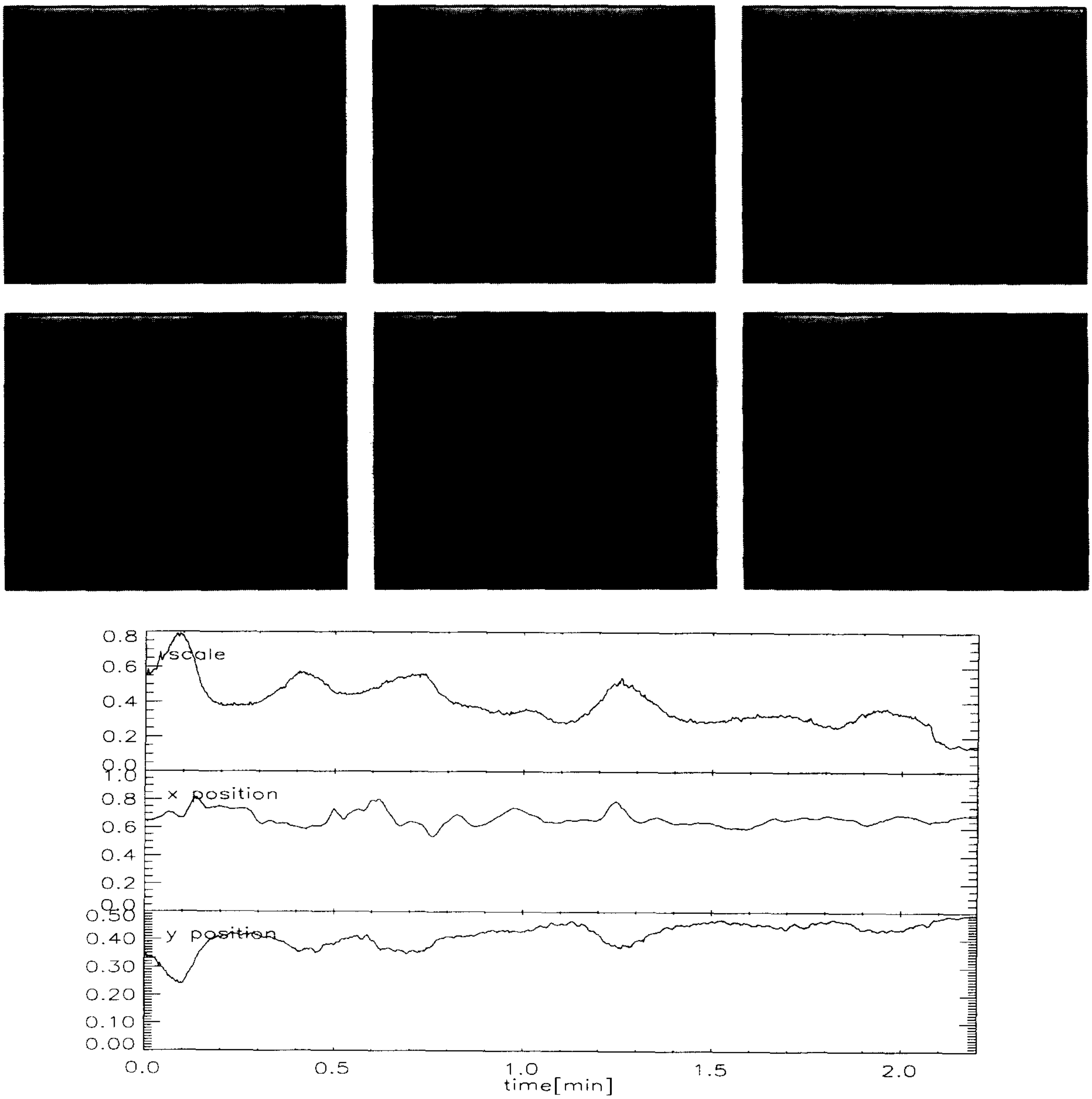

Fig. 4. Top: Six frames of a tracking sequence. Bottom: Plot of the model parameters in the course of the tracking sequence.

$$m_j = \int \max_{q_i} \left(\text{match}(p_j, q_i)\right) \mathrm{d}p_j. \qquad (8)$$

For line segment features, this integration is efficiently performed in a transformed coordinate system. The transformation ensures that the model feature is mapped on the interval $[0, 1]$ of the abscissa and the ordinate corresponds to match(p_j, q_i) (see Fig. 3).

The boundaries of the integration intervals (compare Fig. 3, right-hand side) can be efficiently computed in the transformed coordinate system by the sweep line algorithm (Sedgewick, 1988). As already stated, this method can be generalized to the matching of parameterized curves of higher order (e.g. splines).

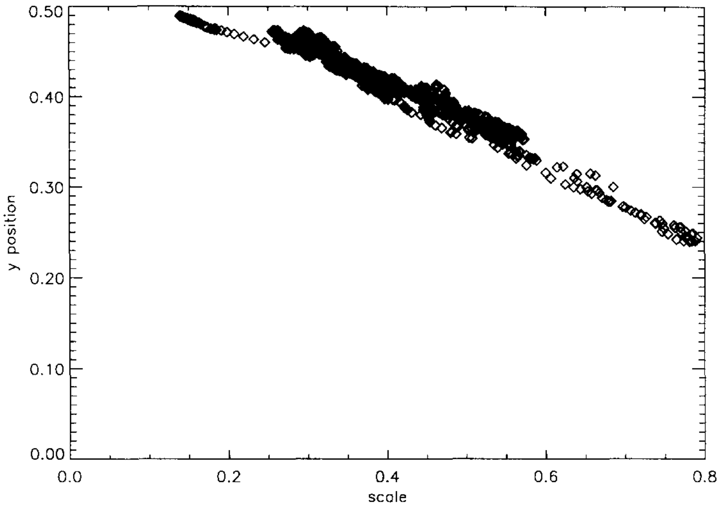

Fig. 5. Estimated y-position versus estimated scale s (compare Fig. 4).

However, the incorporated transformation will be nonlinear in this case.

The resulting matching vectors $\boldsymbol{m}$ are classified by a standard multi-layer perceptron trained by the backpropagation algorithm (compare Section 2.3). Further details on the classification can be found in (Noll and Werner, 1995).

3.4. Results

The CENET algorithm and the hypothesis tester (neural network) for a passenger car model have been implemented to run on a single TI TMS320C40 processor. A series of experiments have been taken out to test the algorithms in terms of their tracking and classification performance.

In Fig. 4 six frames from a tracking sequence with changing environment (urban and non-urban, approx. 2 minutes) are shown. Within this sequence, the vehicle being tracked largely changes its position and scale within the image (see parameter plots in Fig. 4, bottom). Note that the tracking process is robust with regard to non-statistical background features (e.g. shadows, other vehicles etc.) and that no (e.g. Kalman-) filtering is used to smooth and/or predict the model parameters. Assuming the road to be planar, a linear functional dependency between scale s and y-position of the object in the image is expected. This dependency can be used to test the consistency of the tracking data. As can be seen in Fig. 5 the linearity assumption holds to a large extent (the correlation coefficient is $\rho = -0.98$). The remaining scatter is due to small inaccuracies of the parameter estimation and the fact that the assumption of planarity is not always applicable.

From the change in scale (parameter s) of the model an optic variable, the so-called time-to-collision (TTC), can be derived. Due to Lee (1976), this measure is well suited for visual controlled behavior, especially for collision avoidance.

Depending on the number of image features, the system performs at a rate of 8-12 frames per second. The model used for the tracking process has been obtained from the image of a different car.

4. Time series based attention control

In the third application a test vehicle is equipped with five laser range finders measuring the distances to objects in their field of view yielding five time series $d_1(t), \dots, d_5(t)$. The task is to determine the center of attention, i.e. the most important beam for automated cruise control based on the distance information and the velocity of the test vehicle. The time series are heavily disturbed by noise, drop outs and measuring

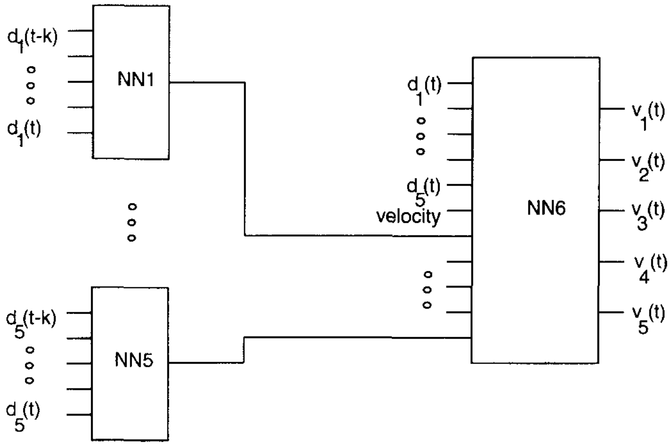

Fig. 6. Processing architecture

inaccuracies. In order to solve the problem at hand, it was decomposed into two parts. The solution of the subproblems is achieved by a modular neural network based approach (see Fig. 6).

At first, for each beam a confidence measure $k_i(t) \in [0,1]$ of the current distance value is determined based on the current and k preceding distance values $d_i(t),\dots,d_i(t-k)$. The confidence values vary in the range of zero (unreliable distance information) to one (reliable distance information). As opposed to Kalman filtering (Gelb, 1986), a non-model based approach is pursued. The confidence measures are computed by single hidden layer neural networks implementing the mapping

$$f_i : (d_i(t),\dots,d_i(t-k)) \mapsto k_i(t). \tag{9}$$

The network is trained to cope with outliers, invalid data and noise, i.e. the manually constructed training set comprises all necessary examples for proper functionality of the resulting network. The nonlinearity of the network permits a fast computation of valid confidence values, because the forward computation does not expose convergence or settling time problems during the recall phase.

The second part of the processing is the fusion of the confidence information $k_i(t)$, the distance information $d_i(t)$ and the velocity information $s(t)$ to determine the focus of attention. This task is solved by a single hidden layer neural network as well. The mapping g performed in this second stage is

$$g : (k_1(t),\dots,k_5(t),d_1(t),\dots,d_5(t),s(t)) \mapsto (v_1(t),\dots,v_5(t)). \tag{10}$$

The outputs $v_1(t),\dots,v_5(t)$ vote for the significance of the corresponding distance information for the control task. A straight forward approach to determine the most significant channel i^* is to apply a winner-takes-all policy, i.e.

$$i^*(t) = \{i \mid v_{i^*} \geqslant v_i \,\forall i\}. \tag{11}$$

More elaborated policies can be found in (Battiti and Colla, 1994). The most significant beam is used to control the distance to the object represented by this beam. The control task is performed by the built-in control system of the test vehicle. The execution time for the complete process is 1.5 ms on a standard 486 PC.

The two stages can be characterized by the kind of data being processed in the respective stage, namely temporal and spatial information in the first and second stage. We consider this subdivision as a major reduction of the process complexity, leading to simple and easy to maintain implementations of neural networks. In general, modular approaches expose some advantages over monolithic ones (Jacobs et al., 1995).

A first version of the system has been tested on a Daimler-Benz test vehicle. The results are promising and all leading cars have correctly been localized by the system. The false alarm rate in the case of the absence of a leading car currently exceeds an acceptable level. This problem should be solved by an improved training set for the second stage. However, the detection performance of the system is superior to an existing fuzzy logic based approach to the problem (personal communications).

5. Summary

In this contribution different approaches to solve a real-world/real-time problem have been presented. The essential key subproblems have been solved with artificial neural networks. We have demonstrated the applicability of one network module to solve different tasks. This procedure reduces design cost and complexity significantly and marks one step towards the constructive design of nonlinear systems.

References

Battiti, R. and A.M. Colla (1994). Democracy in neural nets: voting schemes for classification. *Neural Networks* 4, 691-704.

Bellon, A., J.-P. Dérutin, F. Heitz and Y. Ricquebourg (1994). Real-time collision avoidance at road-crossings on board the prometheus-prolab 2 vehicle. In: *Proc. Intelligent Vehicles '94 Symposium*, Paris, France, 56-61.

Bohrer, S., T. Zielke and V. Freiburg (1995). An integrated obstacle detection framework for intelligent cruise control on motorways. In: *Proc. Intelligent Vehicles Symposium*, Detroit, MI. IEEE Press, New York.

Brauckmann, M.E., C. Goerick, J. Groß and T. Zielke (1994). Towards all around automatic visual obstacle sensing for cars. In: *Proc. Intelligent Vehicles '94 Symposium*, Paris, France, 79-84.

Deriche, R. and O. Faugeras (1990). Tracking line segments. In: O. Faugeras, ed., *Proc. ECCV '90*, Antibes, France. Springer, Berlin, 259-268.

Dickmanns, E.D. et al. (1993). An all-transputer visual autobahn-autopilot/copilot. In: *Proc. ICCV '93*, Berlin, 608-615.

Dickmanns, E.D. et al. (1994). The seeing passenger car 'vamors-p'. In: *Proc. Intelligent Vehicles '94 Symposium*, Paris, France, 68-73.

Finke, M. and K.-R. Müller (1993). Estimating a-posteriori probabilities using stochastic network models. In: *Proc. Summer School on Neural Networks*, Bolder, CO.

Gelb, A. (1986). *Applied Optimal Estimation*. MIT Press, Cambridge, MA.

Goerick, C. and M. Brauckmann (1994). Local orientation coding and neural network classifiers with an application to real time car detection and tracking. In: W.G. Kropatsch and H. Bischof, eds., *Mustererkennung 1994, Proc. 16th Symposium DAGM and 18th Workshop ÖAGM*, Technische Universität Wien.

Hertz, J.A., R.G. Palmer and A.S. Krogh (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley, Reading, MA.

Hornik, K., M. Stinchcombe and H. White (1989). Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 359-366.

Jacobs, R.A., M.I. Jordan and A.G. Barto (1995). Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive Sci.* 15, 219-250.

Jain, A.K. (1989). *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ.

Kapur, J.N. and H.K. Kesavan (1992). *Entropy Optimization Principles with Applications*. Academic Press, San Diego, CA.

Koller, D., K. Daniilidis, T. Thórhallson and H.-H. Nagel (1992). Model-based object tracking in traffic scenes. In: G. Sandini, ed., *Proc. ECCV '92*, Santa Margherita Ligure, Italy. Springer, Berlin, 437-452.

D. Koller, J. Weber and J. Malik (1994). Towards real time visual based tracking in cluttered traffic scenes. In: *Proc. Intelligent Vehicles '94 Symposium*, Paris, France, 201-206.

Lee, D. (1976). A theory of visual control of braking based on information about time-to-collision. *Perception* 5, 437-459.

Mirchandani, G., W. Cao and B. Bosworth (1989). Efficient implementation of neural nets using an optimal relationship between number of patterns, input dimension and hidden nodes. *Proc. ICASSP*, 2521-2523.

Noll, D. and M. Werner (1995). Real-time vehicle tracking and classification. Technical Report IR-INI 95-04, Institut für Neuroinformatik, Ruhr-Universität Bochum.

Noll, D., M. Werner and W. von Seelen (1995). Real-time vehicle tracking and classification. In: *Proc. Intelligent Vehicles '95 Symposium*, Detroit, MI, 101-106.

Rose, K., E. Gurewitz and G.C. Fox (1993). Constrained clustering as an optimization method. *IEEE Trans. Pattern Anal. Mach. Intell.* 15 (8), 785-793.

Rumelhart, D., J. L. McClelland and the PDP Research Group (1986). *Parallel Distributed Processing*. MIT Press, Cambridge, MA.

Sedgewick, R. (1988). *Algorithms*, 2nd edition. Addison-Wesley, Reading, MA.

Thomanek, F., E.D. Dickmanns and D. Dickmanns (1994). Multiple object recognition and scene interpretation for autonomous road vehicle guidance. In: *Proc. Intelligent Vehicles '94 Symposium*, Paris, France, 231-236.

Zielke, T., M. Brauckmann and W. von Seelen (1992). Intensity and edge-based symmetry detection applied to car-following. In: G. Sandini, ed., *Proc. ECCV '92*, Santa Margherita Ligure, Italy. Springer, Berlin, 865-873.