A Time Delay Neural Network Algorithm for Real-Time Pedestrian Recognition

Christian Wöhler, Joachim K. Anlauf, Till Pörtner, and Uwe Franke

Abstract— **In this paper we present an algorithm for recognizing walking pedestrians in sequences of grayscale stereo images taken from a moving camera pair. The method has been designed for use in a driver assistance system for the inner city environment warning the car driver of traffic participants that might cause dangerous situations. Our algorithm is divided into two parts: First, a preliminary detection and tracking stage consisting of a real-time stereo algorithm yields image regions possibly containing a pedestrian. During the subsequent classification stage, these temporal sequences of regions of interest are classified by a feed-forward time delay neural network (TDNN) with spatio-temporal receptive fields. It is possible to stabilize the recognition process by integrating feedback loops into the TDNN architecture. The complete detection and recognition algorithm runs at a speed of about 70 ms per cycle on a Power PC 604e.**

Keywords— **Image sequence, time delay neural network, pedestrian recognition, stereo vision**

I. Introduction

Our application domain is vision-based driver assistance in the inner city environment. In this scenario, the recognition and tracking of pedestrians is essential to avoid dangerous traffic situations. In the following, we will briefly describe some recent approaches for recognizing pedestrians on single images and image sequences.

Due to the fact that pedestrians are non-rigid objects and thus show a high variability especially in outdoor scenes, there are only few approaches that do not rely on motion information. Most methods use motion to extract moving pedestrians from a stationary background – a survey about human motion analysis can be found in [6]. In [11] Haar wavelet templates are extracted from single images which are then used for classifying frontal and rear views of pedestrians with a support vector machine. Segmentation and tracking of walking persons is discussed in [13], [14]. In [17] a real-time system for detecting and tracking a single person in front of a non-stationary background is presented. In this algorithm, a Gaussian model of the background and the person is generated based on color and location in the image. Other methods detect typical motion patterns generated by the legs of walking pedestrians in the xt plane [9] and in xyt space [10]. The algorithm presented in [12] does not require a stationary camera. This method searches for independently moving objects, for each of which a temporal sequence of image regions which are normalized in size is produced. The motion pattern appearing in such a sequence is classified based on optical flow.

Christian Wöhler, Till Pörtner, and Uwe Franke are with the Daimler-Benz Research, FT3/AB, D-89081 Ulm, Germany.

Joachim K. Anlauf is with the Rheinische Friedrich-Wilhelm-Universität, Institut für Informatik II, D-53117 Bonn, Germany.

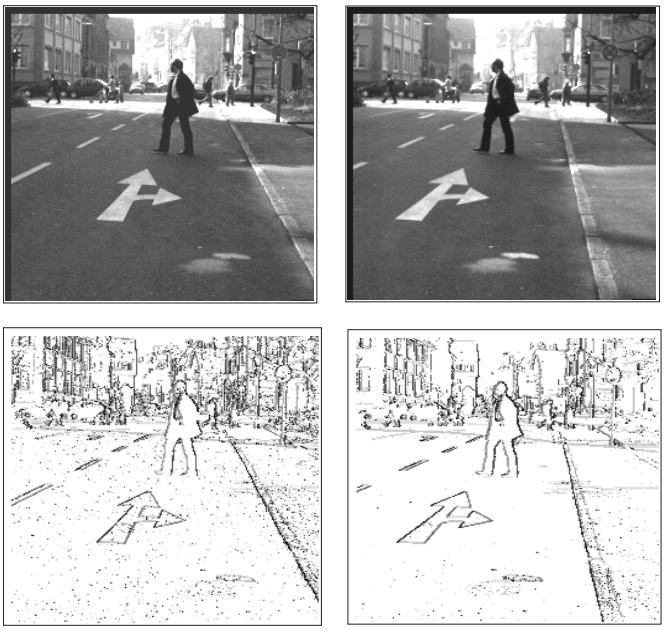

Fig. 1. Stereo image pair and corresponding feature images.

Our pedestrian recognition system is part of the Urban Traffic Assistant (UTA) project that aims at understanding urban scenes for future driver assistance [5]. In our algorithm, object detection is carried out by stereo vision. Each detected object is tracked such that a sequence of bounding boxes can be determined over time which is used to extract a sequence of image regions containing the lower half of the object, i. e. in the case of pedestrians, their legs. These regions are normalized in size and finally classified by a time delay neural network (TDNN) with spatio-temporal receptive fields.

The outline of the paper is as follows: In Section II we describe the stereo object detection algorithm. The TDNN architecture with spatio-temporal receptive fields is explained and performance results are given in Section III. Section IV contains a summary of the paper.

II. Detection and Tracking by Stereo Vision

The UTA system is equipped with a stereo-based object detection and tracking module that detects potential obstacles in real-time, including leading vehicles and pedestrians. This stereo system is based on a non-linear feature extraction which classifies each pixel according to the gray values of its 4 direct neighbors. We check whether each neighbor is much brighter, much darker or has similar brightness compared to the considered central pixel. This

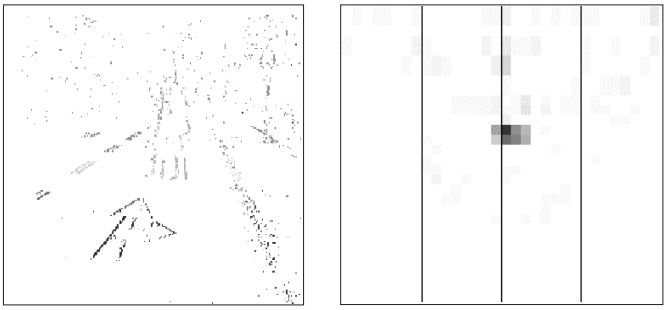

Fig. 2. Disparity image (left) and corresponding 2D depth map.

scheme leads to $3^4 = 81$ different classes encoding edges and corners at different orientations.

Fig. 1 shows a stereo image pair and the result of the sketched structure classification. Different gray values represent different structures, pixels in homogeneous areas are assigned to the "white" class and ignored in the sequel. The correspondence 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 parallel optical axes, pixels with identical classes must be searched on corresponding image rows only.

It is obvious that this approach cannot guarantee uniqueness of the correspondences. In case of ambiguities the solution giving the smallest disparity (i. e. the largest distance) is chosen to overcome this problem. This prevents wrong correspondences for example caused by periodic structures to generate phantom obstacles close to the camera. In addition to that, measurements that violate the ordering constraint for stereo features are ignored.

The outcome of the correspondence analysis is a disparity image, which is the basis for all subsequent steps. Fig. 2 tries to visualize such an image. Dark features are close to the observer (< 10 m), bright features far (> 30 m) away. If all features on the road plane are removed, a 2D depth map containing the remaining features can be generated by projection of the feature points to the ground. The map shown in Fig. 2 covers an area of 30 m in length and 8 m in width. In this "bird's eye view" the pedestrian crossing the road is clearly visible in the two-dimensional histogram and can easily be detected.

The detection step delivers a rough estimate of the object position and size. Subsequently, a 3D box is fitted to the cluster of feature points that contribute to the extracted peak area in the depth map. All obstacles found are tracked over time in the sequence of depth images and their motion states are estimated by means of Kalman filters.

Finally, objects that must be assumed to be pedestrians due to their width, height and/or motion are fed into the TDNN classifier presented in this contribution.

III. Classification of typical Motion Patterns of Pedestrians

A. Preprocessing of the detection results

The pedestrian recognition process is based on the characteristic criss-cross motion of the legs of a laterally walking pedestrian, i. e. walking at an angle smaller than about 45° with respect to the image plane. We therefore crop the lower half of the regions of interest delivered by the stereo algorithm and normalize them in size. We then combine them into image sequences covering a temporal range that approximately corresponds to one walking step. This leads to a sequence length of eight single images since each second stereo image pair is evaluated, i. e. 640 ms. After each stereo detection procedure, the batch of images is shifted backwards by one image, discarding the least recent image while placing the new image at the first position (first-in-first-out principle). The latency time of the algorithm thus corresponds to the temporal extension of the image sequences, after which a classification result is obtained every processing cycle, i. e. every 80 ms.

B. The feed-forward TDNN architecture with spatio-temporal receptive fields

The classification of image sequences possibly containing the legs of a pedestrian delivered by the stereo algorithm is performed by means of a feed-forward TDNN that has specially been designed for processing sequences of grayscale images by simultaneously performing the task of object recognition and motion analysis on them. One of the classical application domains of TDNNs is speech recognition (see e. g. [15]); a TDNN that has specially been adapted to perform motion detection on image sequences is described in [1]. A detailed description of our feed-forward TDNN concept can be found in [16]; here, we will outline only the basic concept. Our TDNN architecture is shown in Fig. 3. The lower half of each region of interest delivered by the stereo algorithm is cropped and scaled to the size $S_x^{(1)} \times S_y^{(1)}$ pixels with $S_x^{(1)} = S_y^{(1)} = 24$ where the index (1) refers to the first network layer. As explained above, the length of the sequence is $S_t^{(1)} = 8$ frames. The three-dimensional input of the TDNN consists of the raw greyscale pixel values of the $S_x^{(1)} \times S_y^{(1)} \times S_t^{(1)}$ image sequence. Already the classical TDNN architecture (see e. g. [8]) is characterized by the fact that it is not fully connected; a neuron of a higher network layer does not receive input from all neurons of the underlying layer but only from a limited region of it. While classical TDNNs only possess temporal receptive fields, we extend this concept towards spatio-temporal receptive fields. This means that each neuron in the second network layer is only connected to a small three-dimensional region of the input layer, sized $R_x \times R_y \times R_t$ pixels, that is called the *receptive field* of the neuron. The distance of the centres of two neighboring receptive fields in the three different dimensions is given by D_x, D_y, and D_t. This takes into account the spatial and temporal locality of the object and motion features, which is in strong

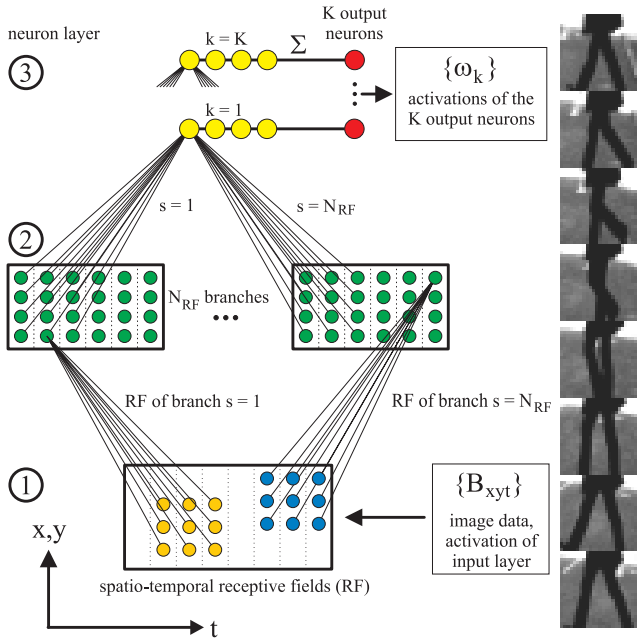

Fig. 3. Left: Architecture of the feed-forward TDNN with spatio-temporal receptive fields. Right: A typical image sequence displaying the motion pattern of a pedestrian's legs.

contrast to the standard multi layer perceptron (MLP) architecture the structure of which is invariant with respect to permutations of the input neurons. Another more technical advantage of the concept of spatio-temporal receptive fields is the fact that the number of free weight parameters of this architecture is usually less than the number of the input pixels, which is impossible for an MLP architecture even for the case of a simple linear perceptron. This leads to an increased generalization capability and reduces overtraining effects.

The TDNN is composed of N_{RF} different branches – as shown in Fig. 3, a "branch" consists of a three-dimensional layer of neurons (in Fig. 3, this is denoted by network layer 2) "looking at" the underlying input sequence by means of their respective receptive fields. As we follow the *shared weights* principle inside each branch a set of weight factors $\{r^s_{mnp}\}$ is assigned to each layer 2 neuron of the branch numbered by s, with m and n as the spatial and p as the temporal coordinate inside the weight configuration of the receptive field. It is $1 \le s \le N_{RF}$, $1 \le m \le R_x$, $1 \le n \le R_y$, and $1 \le p \le R_t$. The weight configurations of the spatio-temporal receptive fields act as spatio-temporal filters, which means that in each network branch one distinct spatio-temporal feature is extracted from the input sequence. The activation patterns in network layer 2 then display N_{RF} "filtered" versions of the original input sequence; the neurons are of the McCulloch-Pitts type such that the output ξ^s_{ijt} of the layer 2 neuron at position (i, j, t) in branch number s is given by

$$\xi^s_{ijt} = g_2\left(\sum_{p=1}^{R_t}\sum_{n=1}^{R_y}\sum_{m=1}^{R_x} r^s_{mnp} \times B_{D_x(i-1)+m, D_y(j-1)+n, D_t(t-1)+p} - \theta^s\right) \quad (1)$$

with $g_2(x) = \tanh(x)$ as a sigmoidal activation function and θ^s as the respective threshold value. The receptive field weights, i. e. filter coefficients, are adapted during the training process; the features to be extracted are thus learned from the training examples instead of being imposed a priori.

The second neuron layer is then connected to the third one by purely temporal receptive fields. Their weight configuration is denoted by $\{v^s_{ijqk}\}$; i. e. a neuron in network layer 3 is not connected to the complete filtered sequence in layer 2 but only to R_h subsequent frames of it (see Fig. 3) such that it can detect specific motion patterns independent of the time at which they occur in the sequence. To each branch s and each output class k one such temporal receptive field is assigned, which produces the activations

$$\sigma_{kt} = g_3\left(\sum_{s=1}^{N_{RF}}\sum_{q=1}^{R_h}\sum_{j=1}^{S_y^{(2)}}\sum_{i=1}^{S_x^{(2)}} v^s_{ijqk}\,\xi^s_{i,j,t+q-1}\right), \quad (2)$$

$g_3(x) = \tanh(x)$, in neuron layer 3. In the case of pedestrian recognition, we only have $K = 2$ output classes, i. e. the "pedestrian" ($k = 1$) and the "garbage" ($k = 2$) class. The actual output neurons of the TDNN then perform a classwise temporal integration of the activations of neuron layer 3, resulting in the output values

$$\omega_k = \sum_{t=1}^{S_t^{(3)}} \sigma_{kt} \quad (3)$$

of the network.

The network is trained by a simple gradient descent rule which corresponds to the well-known backpropagation algorithm (see e. g. [8]). For each training step one image sequence is chosen at random from the training set. With a quadratic error measure defined by $\epsilon = \frac{1}{2}\sum_{k=1}^{K}(\omega_k - \tau_k)^2$ and, for a pattern of class c, $\tau_c = 1$, $\tau_k = 0$ for $k \neq c$, the variation of a weight parameter is proportional to the derivative of the error measure ϵ with respect to it, i. e.

$$\Delta v^s_{abck} = -\eta_v \frac{\partial \epsilon}{\partial v^s_{abck}}, \qquad \Delta r^s_{abp} = -\eta_r \frac{\partial \epsilon}{\partial r^s_{abp}},$$
$$\Delta \theta^s = -\eta_t \frac{\partial \epsilon}{\partial \theta^s}. \quad (4)$$

The weight parameters and threshold values are initialized by small positive and negative random numbers of the order $\mathcal{O}\left(10^{-6}\right)$.

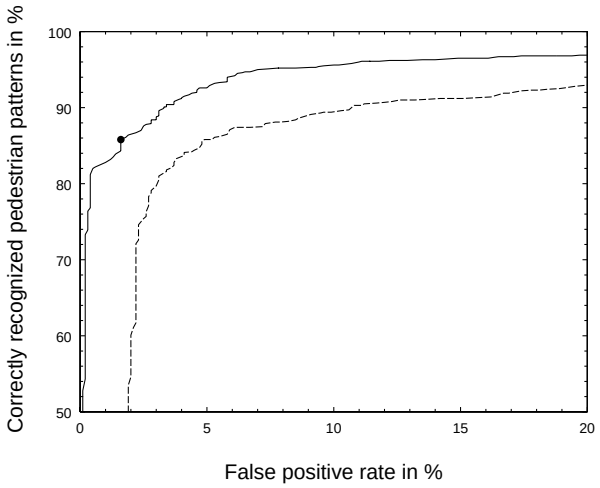

Fig. 4. ROC curve of the TDNN after the first training cycle (dashed curve) and after the bootstrapping step (solid curve). The test set contains 1000 pedestrian and 1337 garbage patterns. The circular dot marks the point with $q = 0.73$ at which our system is actually running in the test vehicle.

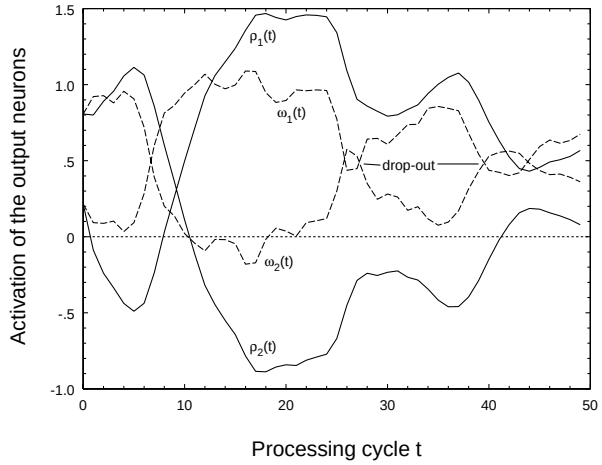

Fig. 5. Output signal of the TDNN on an image sequence displaying a walking pedestrian. Without feedback loops, there are two drop-outs visible in the signals $\omega_1(t)$ and $\omega_2(t)$ (dashed curves) which are removed when applying the feedback loops ($\rho_1(t)$ and $\rho_2(t)$, solid curves). The additional weights are $a_1 = a_2 = 0.5$, $f_{\text{diag}} = f_{11} = f_{22} = 0.5$, and $f_{\text{offd}} = f_{12} = f_{21} = -0.3$. A short recognition delay of 200 ms, however, is introduced with respect to the first appearance of the pedestrian in the scene. The regarded image sequence corresponds to sequence 4 in Fig. 7.

C. Performance of the TDNN classifier

In principle, there are eight network parameters that can be chosen freely. The procedure of determining the network parameters optimally suited to perform a specific recognition task is described in detail in [16]. The network output is interpreted such that if $q\omega_1 > \omega_2$, the input sequence is classified as a pedestrian pattern, where $q = 1$ corresponds to the situation in which the pattern class is simply given by the output neuron with the highest activation. By varying the parameter q over a range of, say, 0 to 2, one obtains the rate of correctly recognized pedestrian patterns versus the rate of false positives, i. e. garbage patterns incorrectly classified as pedestrians. This curve is called the rate of classification (ROC) curve (see e. g. [11]) of the system and mirrors its performance.

As a first result, it came out that the performance is always much better for training sequences of length $S_t^{(1)} = 8$ than for ones of length $S_t^{(1)} = 6$ or 4. On our first training set consisting of 3926 pedestrian and 4426 garbage examples the configuration $R_x = R_y = 9$, $R_t = 5$, $N_{RF} = 2$, $R_h = 3$, $D_x = D_y = 5$, $D_t = 1$ then turned out to yield by far the best performance. By means of a bootstrapping step, i. e. by adding 522 false positives delivered by this first system to the training set, we could significantly improve the performance of the system (Fig. 4). For use in our test vehicle we adjusted the parameter q such that the system achieves a recognition rate of 85.8% of the pedestrian patterns versus 1.6% false positives. This means in practice that in scenes of a length of some seconds, pedestrians are recognized with a high stability. There may be a drop-out on every sixth image while tracking and classifying a pedestrian; the recognition result, however, is strongly confirmed if a certain object detected by the stereo algorithm has been determined to represent a pedestrian at several subsequent time steps. Example image sequences are shown in Fig. 7.

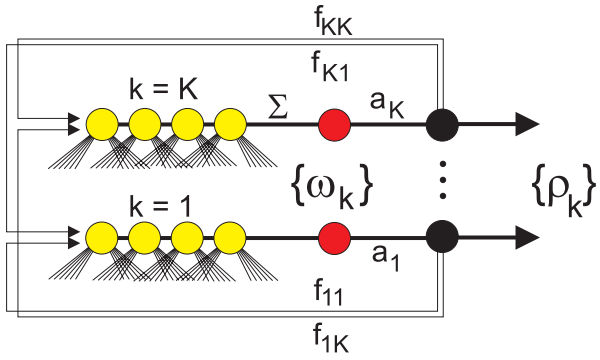

Fig. 6. Neuron layer 3 of the TDNN with additional feedback weights.

D. Stabilizing the recognition result by feedback loops

The stereo detection algorithm yields a bounding box that is usually exactly circumscribing a detected object especially when the object is tracked by applying Kalman filtering. In the case of a sudden strong camera movement – the camera motion is currently not known to the filter algorithm –, however, the bounding box position for some cycles slightly differs from the object position such that only part of the object is contained in it, which leads to short drop-outs in the recognition of a pedestrian over time (Fig. 5). A possibility to prevent these drop-outs which are usually not longer than 3 processing cycles or 240 ms is to add the feedback weights $\{f_{ij}\}$ to the feed-forward TDNN architecture as shown in Fig. 6. The network output is not fed back into neuron layer 1 or 2 as additional neurons in these layers do not fit with the blockwise processing mechanism of receptive fields. The construction shown in Fig. 6, however, corresponds to applying a recursive filter to the

network output. It was not possible to train the feedback weights from examples as thousands of disappearing/reappearing pedestrian examples would be needed which are difficult to obtain. Nevertheless, important properties of the feedback mechanism can be described analytically, i. e. under certain symmetry assumptions, the weights $\{f_{ij}\}$ can be chosen manually according to the recognition problem. Following Fig. 6, the recurrent network output is defined as

$$\vec{\rho}(t) = G\vec{\rho}(t-1) + A\vec{\omega}(t) \tag{5}$$

with $A_{ij} = a_i\delta_{ij}$, $G_{ij} = f_{ij}$, $\vec{\omega}(t) = (\omega_1(t), \ldots, \omega_K(t))$, and $\vec{\rho}(t) = (\rho_1(t), \ldots, \rho_K(t))$. The output $\vec{\rho}(t)$ is evaluated in the same way as the pure feed-forward output $\vec{\omega}(t)$, as described in Section III-C.

Eq. (5) is a first order inhomogeneous difference equation the general solution $\vec{\rho}(t)$ of which is the sum of the general solution $\vec{\rho}^{\,(H)}(t)$ of the corresponding homogeneous difference equation and a special solution $\vec{\rho}^{\,(S)}(t)$ of the inhomogeneous equation (5). The solution of the homogeneous equation $\vec{\rho}(t) = G\vec{\rho}(t-1)$ is

$$\vec{\rho}^{\,(H)}(t) = G^t\vec{\rho}^{\,(H)}(0) = \sum_{k=1}^{K} b_k\lambda_k^t\vec{e}_k \tag{6}$$

with $\{\lambda_k\}$ as the eigenvalues and $\{\vec{e}_k\}$ as the eigenvectors of the matrix G of the feedback weights. The coefficients $\{b_k\}$ are determined by the initial conditions. The homogeneous solution $\vec{\rho}^{\,(H)}(t)$ is stable for

$$\gamma = \max_k \{\mid \lambda_k \mid\} \leq 1. \tag{7}$$

If a solution obeys condition (7) and the eigenvalue with the largest absolute value γ is less than zero it is stable but shows a damped oscillatory behavior. For time-independent $\vec{\omega}(t) = \vec{\omega}(0)$, the general solution of (5) is $\vec{\rho}(t) = G^t\vec{\rho}(0) + (1-G)^{-1}(A\vec{\omega}(0))$; the system relaxes exponentially with the relaxation time $\nu = 1/\ln\gamma$ towards the fixed point $(1-G)^{-1}(A\vec{\omega}(0))$. The stability criterion (7), however, is valid independent of the form of the inhomogenity $A\vec{\omega}(t)$, i. e. our model has the important property that *always* the feedback weights $\{f_{ij}\}$ have to be chosen according to (7). In the pedestrian recognition scenario with $K = 2$, we defined for symmetry reasons $f_{\text{diag}} = f_{11} = f_{22}$ and $f_{\text{offd}} = f_{12} = f_{21}$. We then have $\gamma = \mid f_{\text{diag}} \pm f_{\text{offd}} \mid$ depending on the sign of f_{offd}; the corresponding relaxation time ν should be of the same order of magnitude as the duration of the drop-outs to be removed. In Figs. 5 and 7 it is shown how the feedback loops achieve to stabilize the recognition result.

IV. Summary and Conclusion

In this paper we propose a method for pedestrian recognition and tracking on grayscale stereo images taken by a moving camera pair. The stereo vision algorithm achieves a detection and tracking of objects in the scene and calculates bounding boxes around them. The lower parts of these bounding boxes are grouped together into gray-valued image sequences which serve as an input to a feed-forward TDNN with spatio-temporal receptive fields. The TDNN determines if the detected object is a pedestrian by performing a classification based on the typical criss-cross motion pattern of a pedestrian's legs. The recognition is stabilized by feedback loops added to the feed-forward TDNN architecture. One complete detection, tracking, and classification cycle takes about 70 ms on a Power PC 604e.

As we have shown in experiments, the proposed method yields a stable and robust real-time recognition and tracking of pedestrians even under difficult conditions like strong egomotion or sudden changes in illumination.

References

[1] S. Ambellouis and F. Cabestaing. Motion analysis with a time delayed neural network. In *Symposium on Robotics and Cybernetics, CESA '96 IMACS Multiconference, Computational Engineering in Systems Applications*, pages 328–332, Lille, France, 1996.

[2] A. Broggi. Obstacle and Lane Detection on the ARGO. *IEEE Conference on Intelligent Transportation Systems*, Boston, 1997.

[3] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis.* Wiley-Interscience, 1973.

[4] U. Franke, I. Kutzbach. Fast Stereo based Object Detection for Stop&Go Traffic. *IEEE International Conference on Intelligent Vehicles*, pages 339-344, Tokyo, 1996.

[5] U. Franke et al. Steps towards an Intelligent Vision System for Driver Assistance in Urban Traffic. *IEEE International Conference on Intelligent Vehicles*, Boston, 1997.

[6] D. M. Gavrila. The visual analysis of human movement: a survey. To appear in *Computer Vision Image Understanding.*

[7] B. Heisele, U. Kressel, and W. Ritter. Tracking non-rigid, moving objects based on color cluster flow. In *Proc. Computer Vision and Pattern Recognition*, pages 253–257, San Juan, 1997.

[8] J. A. Hertz, A. Krogh, R. G. Palmer. Introduction to the Theory of Neural Computation. Addison-Wesley, Redwood City, CA, 1991.

[9] S. A. Niyogi and E. H. Adelson. Analyzing and recognizing walking figures in xyt. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 469–474, 1994.

[10] S. A. Niyogi and E. H. Adelson. Analyzing gait with spatiotemporal surfaces. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 64–69, Austin, 1994.

[11] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 193–199, San Juan, 1997.

[12] R. Polana and R. Nelson. Low level recognition of human motion. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 77–82, Austin, 1994.

[13] J. Segen and S. Pingali. A camera-based system for tracking people in real time. In *International Conference on Pattern Recognition*, pages 63–67, Vienna, 1996.

[14] S. Shio and J. Sklansky. Segmentation of people in motion. In *IEEE Workshop on Visual Motion*, pages 325–332, 1991.

[15] A. Waibel, T. Hanazawa, G. Hinton, K. J. Lang. Phoneme recognition using time delay neural networks. *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pages 328-339, 1989.

[16] C. Wöhler and J. K. Anlauf. A Time Delay Neural Network Algorithm for Estimating Image-pattern Shape and Motion. To appear in *Image and Vision Computing Journal.*

[17] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

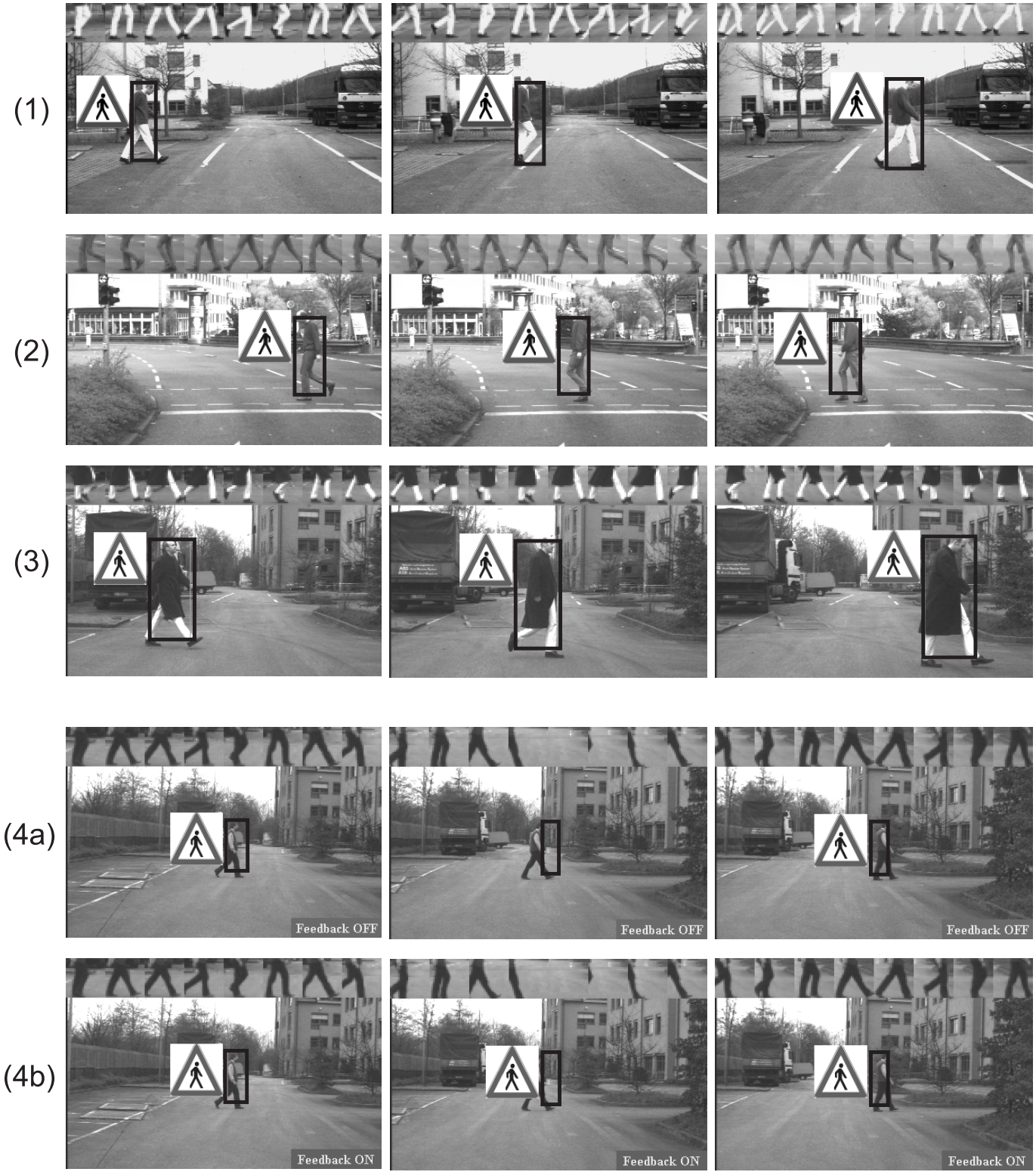

Fig. 7. Example image sequences displaying recognized pedestrians, all of them taken by a non-stationary stereo camera pair in an urban traffic environment. Only the left stereo image of each pair is shown, respectively. All three example sequences show a pedestrian crossing the street. A recognized pedestrian is marked by a triangular "pedestrians" traffic sign that also indicates the direction of motion. The black bounding boxes have been determined by the stereo algorithm. In the upper part of the image, the input of the TDNN, i. e. the pedestrian's legs on the current and the 7 preceeding images is shown, respectively. The interval between two subsequent images is 8 cycles in each example. On sequences 1, 2, 3, and 4a, pedestrian recognition is carried out without applying feedback loops. On the second image of sequence 4a, however, the pedestrian is lost by the Kalman tracker assuming constant motion as a consequence of a sudden camera movement. This behavior is stabilized as shown in sequence 4b by applying feedback loops of the type as explained in Section III-D with the same feedback weight values $a_1 = a_2 = 0.5$, $f_{11} = f_{22} = 0.5$, and $f_{12} = f_{21} = -0.3$ as in Fig. 5. For the network output on sequence 4, see Fig. 5.