Motion-Based Recognition of Pedestrians

B. Heisele C. Wöhler
Daimler-Benz AG, Research and Technology
Image Understanding, FT3/AB
P.O. Box 2360, 89013 Ulm, Germany
{Heisele,Woehler}@DBAG.Ulm.DaimlerBenz.Com

Abstract

In this paper we present an algorithm for recognizing walking pedestrians in sequences of color images taken from a moving camera. The recognition is based on the characteristic motion of the legs of a pedestrian walking parallel to the image plane. Each image is segmented into region-like image parts by clustering pixels in a combined color/position feature space. The proposed clustering technique implies matching of corresponding clusters in consecutive frames and therefore allows clusters to be tracked over a sequence of images. Based on the observation of clusters over time a two-stage classifier extracts those clusters which most likely represent the legs of pedestrians. A fast polynomial classifier performs a rough preselection of clusters by evaluating temporal changes of a shape-dependent cluster feature. The final classification is done by a time delay neural network (TDNN) with spatio-temporal receptive fields.

1. Introduction

Our application domain is vision-based driver assistance in the inner city. In this application early recognition and tracking of pedestrians is essential for collision avoidance. In the following we briefly review some approaches for recognizing pedestrians.

Due to the high variability in the appearance of pedestrians in outdoor scenes, there are only few approaches which do not rely on motion information. In [8] wavelet templates are used to extract frontal and rear views of pedestrians in single images. Most approaches use motion to extract moving pedestrians from a stationary background and perform a classification based on the human gait – a survey about human motion analysis can be found in [3]. Segmentation and tracking of walking persons is discussed in [10, 11]. A real-time system for detecting and tracking a single person in an arbitrary scene is presented in [14]. The system generates Gaussian models of the background and the person based on color and image location. In [6, 7] patterns generated by walking persons are detected in the XT plane [6] and XYT space [7]. A method which does not require a fixed camera is proposed in [9]. For each independently moving object they extract a temporal sequence of image regions which are normalized in size. The motion pattern in such a sequence is classified based on optical flow.

In contrast to [9] our method does not require the detection of independently moving objects. Each image is segmented into a set of clusters by grouping pixels with similar color and position in the image plane. Tracking of clusters over time is implicitly done by the proposed clustering technique. A polynomial classifier preselects clusters which most likely correspond to the legs of pedestrians. For each selected cluster we extract a sequence of regions containing the cluster. The regions are normalized in size and finally classified by a time delay neural network (TDNN).

The outline of the paper is as follows: In Section 2 we briefly describe the segmentation by clustering and the tracking of clusters. The polynomial classifier is explained in Section 3, the TDNN in Section 4. We then present our results in Section 5 and give a summary of the paper in Section 6.

2. Segmentation and Tracking

The proposed segmentation method assumes that pixels of similar color which are close to each other in the image belong to the same physical object and thus can be grouped together. Each pixel n in the image is described by a feature vector $\mathbf{f}_n$, containing its color in the RGB space and its position in the image plane: $\mathbf{f}_n = (R_n, G_n, B_n, x_n, y_n)$, where x_n is the horizontal and y_n the vertical position of pixel n in the image plane. The task of clustering is to find a given number R of prototypes $\mathbf{p}_r$, which minimize the sum of quantization errors: $\sum_n \|\mathbf{f}_n - \mathbf{p}_{r(n)}\|^2$, where $\mathbf{p}_{r(n)}$ is the prototype closest to the feature vector $\mathbf{f}_n$ in the color/position feature space.

For the first image of a sequence we use a divisive clustering technique [5] which partitions the image into a preselected number of clusters. In the consecutive frames the prototypes of the previous frame are shifted in the feature space by parallel k-means clustering [2] to fit the new image data. Thus we obtain consistent segmentation results over time. Moreover, no explicit matching of corresponding clusters is required. Assuming a continuous motion behavior of clusters, prediction techniques are used to increase the robustness of tracking. A detailed description of the applied clustering techniques can be found in [4].

Fig. 1 shows the results of clustering in a scene with a crossing pedestrian. In all sequences investigated in this paper the number of clusters was set to 128 which leads to reasonable segmentation results. However, the requirement of a fixed number of clusters may lead to problems when dealing with sequences where the complexity of the observed scene is changing rapidly over time.

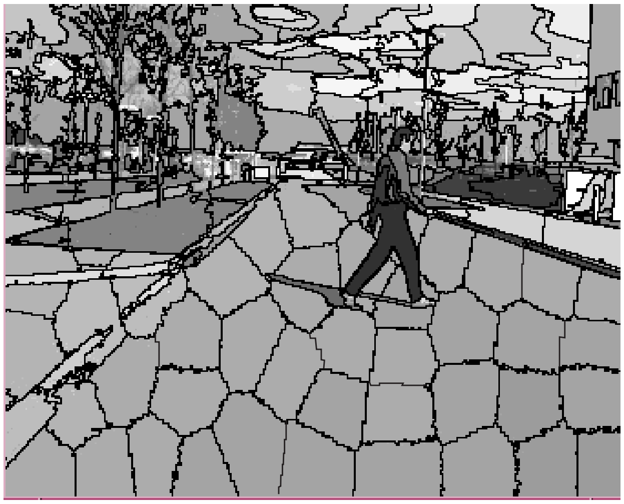

Figure 1. Clustering in the combined color/position feature space. Dark lines represent the boundaries of the clusters

3. Classification of Temporal Shape Variations

Usually both legs or at least parts of both legs of a pedestrian are combined into the same cluster due to their similarity in color. In the case of a pedestrian walking parallel to the image plane, temporal changes in the shape of such a cluster can be used to distinguish it from clusters belonging to other parts in the scene. To approximate the shape of a cluster we use a rectangular box which bounds the eigenvectors of the covariance matrix of the corresponding cluster in the image plane (see Fig. 2).

Only the box width is taken as an input feature for the classifier. As shown in Fig. 3 the periodicity in the human gait is reflected in the time signal of the box width. To extract the dominant frequency a Fast Fourier Transform with a time window of fixed size is applied to the signal. The width of the time window was set to 16 frames (= 0.64 sec at 25 frames/sec) which is roughly the duration of one step of a pedestrian walking at medium speed. Only the first two lines of the spectrum at 1.56 Hz and 3.125 Hz are used for classification. We adapted a complete quadratic polynomial classifier with a sample set generated from 24 clusters representing moving legs and 40 clusters representing backgound. The spectra have been calculated by shifting the time window continuously over the time signal. This lead to 668 spectra of pedestrians and 777 spectra representing background.

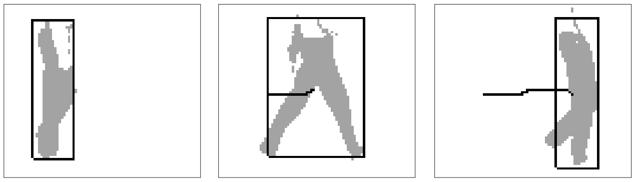

Figure 2. Temporal changes in a cluster belonging to the legs of a pedestrian. The box used for approximating the shape of the cluster and the trajectory of the centroid of the cluster are drawn as dark lines.

About 16 % of the spectra belonging to legs of pedestrians could not be classified correctly. The main reason is an inaccurate segmentation due to low color contrast between the legs and the surrounding image parts. Problems also occur when pedestrians are walking at sharp angles ($\leq 50\,\text{deg}$) to the optical axis which significantly reduces the change in amplitude in the time signal. However, since a pedestrian occurs about 30 to 40 times in an image sequence, the probability of its detection is still sufficiently high. The necessity of a more sophisticated second stage classifier follows from the high number of false alarms: about 7 % of the background clusters have been classified as belonging to pedestrians. Since each image contains 128 clusters there are about 9 false alarms per image. They are caused by the motion of the camera which leads to changes in the shapes of clusters belonging to the background.

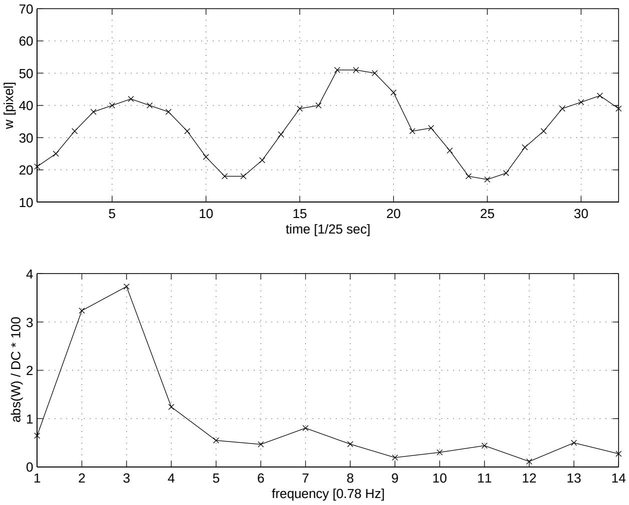

Figure 3. The upper diagram shows the box width of a cluster belonging to the legs of a pedestrian. The lower diagram shows the spectrum calculated by a FFT (window size = 32) and normalized by the direct component.

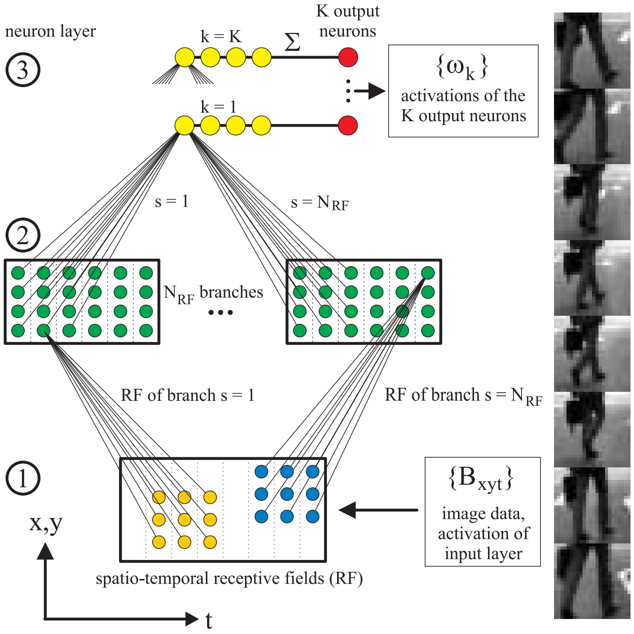

Figure 4. Left: Architecture of the time delay network with adaptable spatio-temporal receptive fields. Right: A typical training example displaying the motion pattern of a pedestrian's legs.

4. Final Classification with a TDNN

The typical motion pattern of the pedestrian's legs is a very stable and characteristic feature. We perform the final classification of the regions possibly containing a pedestrian's legs delivered by the polynomial classifier by means of a time delay neural network (TDNN) that has specially been designed for processing grayscale image sequences, simultaneously performing an object recognition and motion analysis on them. The classical application domain of TDNNs is speech recognition (see e. g. [12]); a TDNN specially adapted to motion detection is described in [1]. A detailed description of the TDNN architecture we use for pedestrian recognition can be found in [13]; here, only the basic concept will be outlined (see Fig. 4):

The regions of interest around the color clusters selected by the polynomial classifier as possible pedestrians are cropped and scaled to the size $S_x^{(1)} \times S_y^{(1)}$ pixels with $S_x^{(1)} = S_y^{(1)} = 24$ (the index (1) refers to the first network layer). The image sequence to be classified was chosen to contain $S_t^{(1)} = 8$ frames. The raw grayscale values B_{xyt} of the $S_x^{(1)} \times S_y^{(1)} \times S_t^{(1)}$ image sequence serve as an input to the three-dimensional input layer. To take into account the three-dimensionality of the input and the locality of the object features, we make use of the concept of spatio-temporal receptive fields. This means that each neuron in the second layer is not connected to the complete input layer like e. g. in the multi layer perceptron architecture but only to a limited region of it, sized $R_x \times R_y \times R_t$ pixels. This region is called the *receptive field* of the neuron. Neighboring layer 2 neurons "see" neighboring regions of the input layer; the distance of the centers of two neighboring receptive fields in the three different dimensions is given by D_x, D_y, and D_t. The network consists of N_{RF} different branches as shown in Fig. 4. Each branch possesses one set of receptive fields and therefore extracts one spatio-temporal feature from the input sequence. We use shared weights within each network branch, i. e., each layer 2 neuron of a certain branch s receives input from its respective receptive field by means of the same configuration of receptive field weights r^s_{mnp}, $1 \leq m \leq R_x$, $1 \leq n \leq R_y$, $1 \leq p \leq R_t$, and $1 \leq s \leq N_{RF}$. The parameter s denotes the network branch. The neurons are of the McCullough-Pitts type such that the output ξ^s_{ijt} of the neuron at position (i, j, t) in branch number s in the second network layer amounts to

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

with $g_2(x) = \tanh(x)$ as a sigmoidal activation function and θ^s as the respective threshold value.

The receptive fields thus act as spatio-temporal filters and produce one "filtered" version of the input sequence per branch s in neuron layer 2. The receptive field weights, i. e., filter coefficients, are adapted during the training process; the features to be extracted are therefore 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 the weights of which are denoted by v^s_{ijqk}, i. e., a neuron in layer 3 is not connected to the complete underlying filtered sequence but only to R_h subsequent frames of it. In the spatial dimensions, however, neuron layers 2 and 3 are fully connected. To each branch s and each output class k of the network there belongs one such temporal receptive field; they produce 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 pedestrians, we only have to deal with $K = 2$ output classes, i. e., "pedestrian" and "garbage". The output neurons 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. In each training step, one image sequence is chosen at random from the training set. Defining a quadratic error measure $\epsilon = \frac{1}{2}\sum_{k=1}^{K}(\omega_k - \tau_k)^2$ with, for a pattern of class c, $\tau_c = A$, $\tau_k = 0$ for $k \neq c$, A close to 1, the variations of the network weights in this training step are proportional to the derivative of the error measure ϵ with respect to them, 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. A typical image sequence used as a training example is shown in Fig. 4. Large overlapping receptive fields between the first two network layers yield a high size and displacement tolerance; we therefore use the network parameter configuration $R_x = R_y = 11$, $R_t = 5$, $N_{RF} = 2$, $R_h = 3$, $D_x = D_y = 6$, and $D_t = 1$.

Like the polynomial pre-classifier, the TDNN was trained on a set of 628 example sequences containing a pedestrian's legs and 777 garbage patterns. As in this first step the rate of false positive recognitions was still quite high (about 10 % on several independent test sets), the network trained in this manner was applied to a number of further image sequences containing no pedestrians at all to produce a set of 1091 incorrectly classified garbage patterns to be added to the training set. After this bootstrapping step, at least 90 % of the pedestrian examples are correctly classified on several independent test sets, while the false positive rate now lies well below 2 %.

5. Results

Experiments were carried out on traffic scenes. A 3-chip CCD camera, connected to a digital video recorder (Y:U:V, 720 × 576 pixels, 25 frames/sec.), was used for taking images. The resolution of the images was reduced to half the number of rows and columns (360 × 288 pixels). All images have been partitioned into 128 clusters which took approximately 4 seconds on a Sun SPARC-20. The polynomial pre-classification is performed within a few milliseconds for all clusters of an image, while the verification carried out by the TDNN then takes about 50 milliseconds per pedestrian candidate.

The effective probability that a pedestrian is correctly recognized by the two combined classifiers is about 75 % at a certain time step. This means in practice that in scenes of a length of some seconds, the pedestrians are recognized with a high stability – there may be a drop-out on every fourth or fifth image – while in the average on each eighth image of the sequence one false positive recognition occurs. The false positives can easily be distinguished from the correctly recognized pedestrians as they randomly occur at arbitrary positions, while the pedestrian is always represented by the same cluster such that the recognition result is confirmed if a certain cluster has been determined to represent a pedestrian at several time steps.

In Fig. 5 a-d, we present some example image sequences, all of them taken by a non-stationary camera. The first two examples (a-b) show a single pedestrian crossing the street. The classification also works in the presence of strong disturbances (c). It is even possible to recognize single pedestrians in a crowd crossing the street (d).

6. Summary

In this paper, we propose a method to perform the detection and recognition of pedestrians in color image sequences taken by a moving camera. Each image is segmented into clusters by grouping pixels in a combined color/position feature space. The clustering implies a tracking of clusters over time. In order to determine if a cluster belongs to the legs of a pedestrian a quadratic polynomial classifier checks for periodicity in the temporal shape variation of a cluster. Due to background motion the false positive rate of this preclassification is quite high. Therefore a second stage classification is carried out by a TDNN with adaptable spatio-temporal receptive fields. The input data of the TDNN are temporal sequences of gray-valued image regions selected by the polynomial classifier.
As we have shown in experiments, the proposed method yields a stable and robust recognition and tracking of pedestrians even under difficult conditions (i. e. strong egomotion, partial occlusions, sudden changes in illumination). A future extension is the adaption of the initial clustering algorithm to grayscale images. This will significantly accelerate the segmentation process such that real-time tracking and recognition will be possible on standard hardware.

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

[3] D. M. Gavrila. The visual analysis of human movement: a survey. Accepted for publication in *Computer Vision Image Understanding*.

[4] 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.

[5] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, 1980.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] A. Waibel, T. Hamazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition: Neural networks versus hidden markov models. In *Proc. of the IEEE Int. Conf. on Acoust., Speech, Signal Processing*, pages 107–110, 1988.

[13] C. Wöhler and J. K. Anlauf. A Time Delay Neural Network Algorithm for Estimating Image-pattern Shape and Motion. Submitted to *Image and Vision Computing Journal.*

[14] 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.

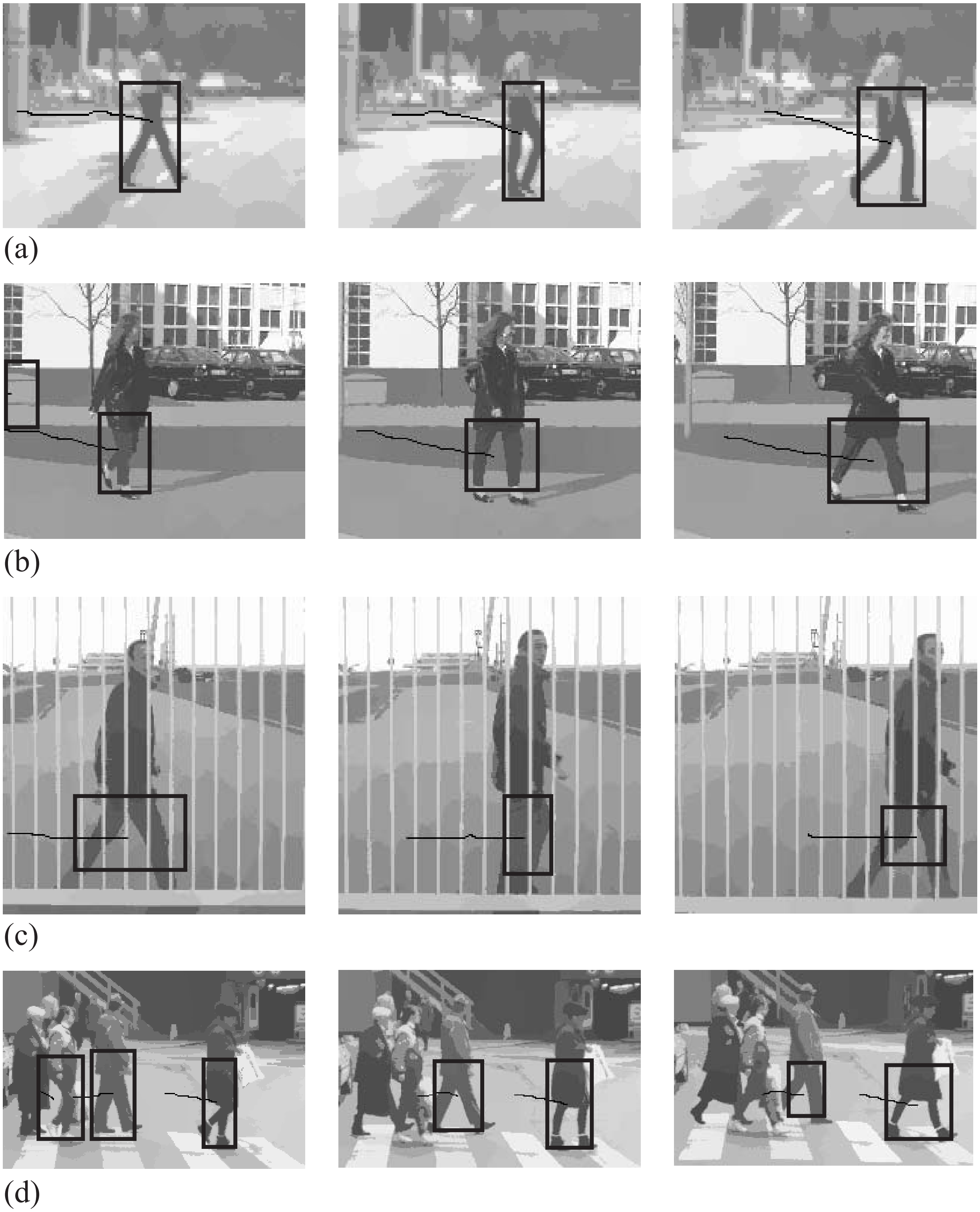

Figure 5. Example image sequences displaying pedestrians. The images shown have already been processed by the clustering algorithm. The clusters recognized as pedestrians are marked by their respective circumscribing bounding boxes.