Proc. of the European Conference on Computer Vision, pp. 37-49, Dublin, 2000.

Pedestrian Detection from a Moving Vehicle

D.M. Gavrila

Image Understanding Systems, DaimlerChrysler Research,
Wilhelm Runge St. 11, 89081 Ulm, Germany,
email: dariu.gavrila@DaimlerChrysler.com, WWW: www.gavrila.net

Abstract. This paper presents a prototype system for pedestrian detection on-board a moving vehicle. The system uses a generic two-step approach for efficient object detection. In the first step, contour features are used in a hierarchical template matching approach to efficiently "lock" onto candidate solutions. Shape matching is based on Distance Transforms. By capturing the objects shape variability by means of a template hierarchy and using a combined coarse-to-fine approach in shape and parameter space, this method achieves very large speed-ups compared to a brute-force method. We have measured gains of several orders of magnitude. The second step utilizes the richer set of intensity features in a pattern classification approach to verify the candidate solutions (i.e. using Radial Basis Functions). We present experimental results on pedestrian detection off-line and on-board our Urban Traffic Assistant vehicle and discuss the challenges that lie ahead.

1 Introduction

We are developing vision-based systems for driver assistance on-board vehicles [7]. Safety and ease-of-use of vehicles are the two central themes in this line of work. This paper focusses on the safety aspect and presents a prototype system for the detection of the most vulnerable traffic participants: pedestrians. To illustrate the magnitude of the problem, consider the numbers for Germany: more than 40.000 pedestrians were injured in 1996 alone due to collisions with vehicles [6]. Of these, more than 1000 were fatal injuries. Our long-term goal is to develop systems which, if not avoid these accidents altogether, at least minimize their severity by employing protective measures in case of upcoming collisions.

An extensive amount of computer vision work exists in the area of "Looking-at-People", see [8] for a recent survey. The pedestrian application on-board vehicles is particulary difficult for a number of reasons. The objects of interest appear in highly cluttered backgrounds and have a wide range of appearances, due to body size and poses, clothing and outdoor lighting conditions. They stand typically relatively far away from the camera, and thus appears rather small in the image, at low resolution. A major complication is that because of the moving vehicle, one does not have the luxury to use simple background subtraction

methods to obtain a foreground region containing the human. Furthermore, there are hard real-time requirements for the vehicle application which rule out any brute-force approaches.

The outline of this paper is as follows. After reviewing past work on pedestrian detection, in Section 2, we present an efficient two-step approach to this problem. The Chamfer System, a system for shape-based object detection based on multi-feature hierarchical template matching, is described in Section 3. The following Section 4 deals with a Radial Basis Function (RBF)-based verification method employed to dismiss false-positives. Special measures are taken to obtain a "high-quality" training set. Section 5 lists the experiments on pedestrian detection; it is followed by a discussion of the challenges that lie ahead, in Section 6. We conclude in Section 7.

2 Previous Work

Most work on pedestrian detection [8] has taken a learning-based approach, bypassing a pose recovery step altogether and describing human appearance in terms of simple low-level features from a region of interest. One line of work has dealt specifically with scenes involving people walking laterally to the viewing direction. Periodicity has provided a quite powerful cue for this task, either derived from optical flow [17] or raw pixel data [5]. Heisele and Wöhler [10] describe ways to learn the characteristic gait pattern using a Time-Delay Neural Network with local receptive fields; their method is not based on periodicity detection and extends to arbitrary motion patterns.

A crucial factor determining the success of the previous learning methods is the availability of a good foreground region. Standard background subtraction techniques are of little avail because of a moving camera; here, independent motion detection techniques can help [17], although they are difficult to develop, themselves. Yet, given a correct initial foreground region, some of the burden can be shifted to tracking. For example, work by Baumberg and Hogg [2] applied Active Shape Models, based on B-splines, for tracking pedestrians. The interesting feature of this approach is that the Active Shape Models only deform in a way consistent with the training set; they can be combined with scale-space matching techniques to increase their coverage in image space [3]. In other work [10], color clusters are tracked over time; a pre-selection technique is used to identify the clusters that might correspond to the legs. Work by Curio et al. [4] uses a general- purpose tracker based on the Hausdorff distance to track the edges of the legs. Rigoll, Winterstein and Müller [18] perform Kalman filtering on a HMM-based representation of pedestrians.

A complementary problem is to detect pedestrians whilst they stand still. A system that can detect pedestrians in static images is described in [15]. It basically shifts windows of various sizes over the image, extracts an overcomplete set of wavelet features from the current window, and applies a Support Vector Machine (SVM) classifier to determine whether a pedestrian is present or not.

The proposed system is, like [15], applied on pedestrian detection in static images. However, the brute-force window sliding technique used there is not feasible for real-time vision onboard vehicles, because of the large computational cost involved. We propose a shape-based system that does not require a region of interest, yet can very quickly "lock" onto desired objects, using an efficient coarse-to-fine technique based on distance transforms. The pattern classification approach is only applied at the second stage, for verification, allowing realtime performance. The resulting system is generic can been applied to other object recognition tasks as well.

3 Detection: The Chamfer System

We now discuss the basics and extensions of the Chamfer System, a system for realtime shape-based object detection.

3.1 Basics

At the core of the proposed system lies shape matching using distance transforms (DT) [11]. Consider the problem of detecting pedestrians in an image (Figure 1a). Various object appearances are modeled with templates such as in Figure 1b. Matching template T and image I involves computing the feature image of I, (Figure 1c) and applying a distance transform to obtain a DT-image (Figure 1d).

A distance transform converts a binary image, which consists of feature and non-feature pixels, into an image where each pixel value denotes the distance to the nearest feature pixel. A variety of DT algorithms exist, differing in their use of a particular distance metric and the way local distances are propagated. The *chamfer* transform, for example, computes an approximation of the Euclidean distance using integer arithmetic, typically in raster-scan fashion [1].

After computing the distance transform, the relevant template T is transformed (e.g. translated) and positioned over the resulting DT image of I; the matching measure $D(T, I)$ is determined by the pixel values of the DT image which lie under the "on" pixels of the transformed template. These pixel values form a distribution of distances of the template features to the nearest features in the image. The lower these distances are, the better the match between image and template at this location. There are a number of matching measures that can be defined on the distance distribution; one possibility is to use simple averaging. Other more robust (and costly) measures reduce the effect of missing features (i.e. due to occlusion or segmentation errors) by using the average truncated distance or the f-th quantile value (the *Hausdorff* distance), e.g. [11].

For efficiency purposes, we use in our work the average chamfer distance

$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t) \tag{1}$$

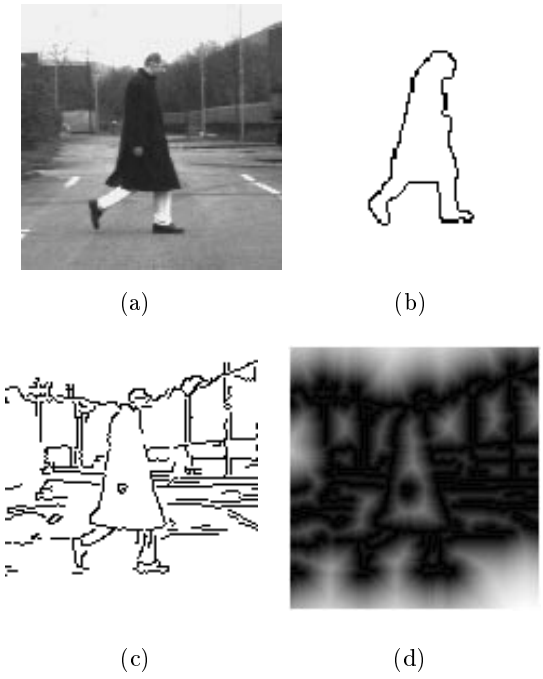

Fig. 1. (a) original image (b) template (c) edge image (d) DT image

where $|T|$ denotes the number of features in T and $d_I(t)$ denotes the chamfer distance between feature t in T and the closest feature in I.

In applications, a template is considered matched at locations where the distance measure $D(T, I)$ is below a user-supplied threshold θ

$$D(T, I) < \theta \tag{2}$$

The advantage of matching a template with the DT image rather than with the edge image is that the resulting similarity measure will be smoother as a function of the template transformation parameters. This enables the use of an efficient search algorithm to lock onto the correct solution, as will be discussed shortly. It also allows some degree of dissimilarity between a template and an object of interest in the image.

3.2 Extensions

The main contribution of the Chamfer System is the use of a template hierarchy to efficiently match whole sets of templates. These templates can be geometrical transformations of a reference template, or, more general, be examples capturing the set of appearances of an object of interest (e.g. pedestrian). The underlying idea is to derive a representation off-line which exploits any structure in this

template distribution, so that, on-line, matching can proceed optimized. More specifically, the aim is to group similar templates together and represent them two entities: a "prototype" template and a distance parameter. The latter needs to capture the dissimilarity between the prototype template and the templates it represents. By matching the prototype with the images, rather than the individual templates, a typically significant speed-up can be achieved on-line. When applied recursively, this grouping leads to template hierarchy, see Figure 2.

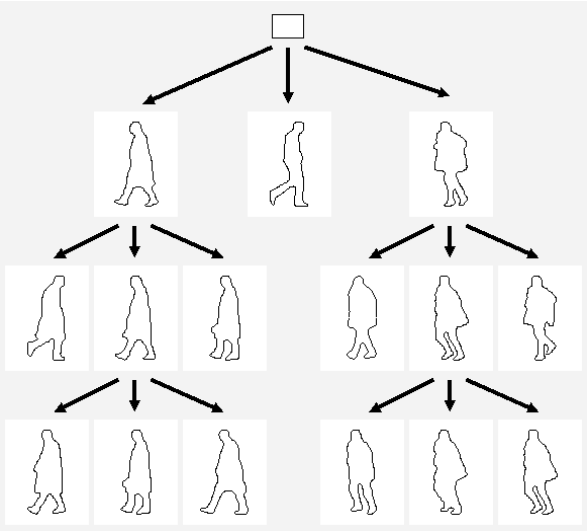

Fig. 2. A hierarchy for pedestrian shapes (partial view)

The above ideas are put into practice as follows. Offline, a template hierarchy is generated automatically from available example templates. The proposed algorithm uses a bottom-up approach and applies a partitional clustering algorithm at each level of the hierarchy. The input to the algorithm is a set of templates $\mathbf{t_1}, ..., \mathbf{t_N}$, their dissimilarity matrix (see below) and the desired partition size K. The output is the K-partition and the prototype templates $\mathbf{p_1}, ..., \mathbf{p_K}$ for each of the K groups $S_1, ..., S_K$. The K-way clustering is achieved by iterative optimization. Starting with an initial (random) partition, templates are moved back and forth between groups while the following objective function E is minimized

$$E = \sum_{k=1}^{K} \max_{\mathbf{t_i} \in \mathbf{S_k}} D(\mathbf{t_i}, \mathbf{p_k^*}) \tag{3}$$

Here, $D(\mathbf{t_i}, \mathbf{p_k^*})$ denotes the distance measure between the i-th element of group k and the prototype for that group at the current iteration, $\mathbf{p_k^*}$. The distance measure is the same as the one used for matching (e.g. chamfer or Hausdorff

distance). Entry $D(i,j)$ is the ijth member of the dissimilarity matrix, which can be computed fully before grouping or only on demand.

One way of choosing the prototype $\mathbf{p_k^*}$ is to select the template with the smallest maximum distance to the other templates. A low E-value is desirable since it implies a tight grouping; this lowers the distance threshold that will be required during matching (see also Equation 5) which in turn likely decreases the number of locations which one needs to consider during matching. Simulated annealing [13] is used to perform the minimization of E.

Online, matching can be seen as traversing the tree structure of templates. Each node corresponds to matching a (prototype) template $\mathbf{p}$ with the image at some particular locations. For the locations where the distance measure between template and image is below a user-supplied threshold θ_p, one computes new interest locations for the children nodes (generated by sampling the local neighborhood with a finer grid) and adds the children nodes to the list of nodes to be processed. For locations where the distance measure is above the threshold, search does not propagate to the sub-tree; it is this pruning capability that brings large efficiency gains. Initially, the matching process starts at the root and the interest locations lie on a uniform grid over relevant regions in the image. The tree can be traversed in breadth-first or depth-first fashion. In the experiments, we use depth-first traversal, which has the advantage that one needs to maintain only $L-1$ sets of interest locations, with L the number of levels of the tree.

Let $\mathbf{p}$ be the template corresponding to the node currently processed during traversal at level l and let $C = \{\mathbf{t_1}, ..., \mathbf{t_c}\}$ be the set of templates corresponding to its children nodes. Let δ_p be the maximum distance between $\mathbf{p}$ and the elements of C.

$$\delta_p = \max_{t_i \in C} D(\mathbf{p}, \mathbf{t_i}) \tag{4}$$

Let σ_l be the size of the underlying uniform grid at level l in grid units, and let μ denote the distance along the diagonal of a single unit grid element. Furthermore, let τ_{tol} denote the allowed shape dissimilarity value between template and image at a "correct" location. Then by having

$$\theta_p = \tau_{tol} + \delta_p + \frac{1}{2}\mu\sigma_l \tag{5}$$

one has the desirable property that, using untruncated distance measures such as the chamfer distance, one can assure that the coarse-to-fine approach using the template hierarchy will not miss a solution. The thresholds one obtains by Equation (5) are very conservative, in practice one can use lower thresholds to speed up matching, at the cost of possibly missing a solution (see Experiments).

4 Verification: RBF-based pattern classification

As result of the initial detection step, we obtain a (possibly empty) set of candidate solutions. The latter are described by a template id and the particular image location where the match was found. The verification step consists of revisiting

the original image, extracting a rectangular window region corresponding to the bounding box of the template matched, normalizing the window for scale, and employing a local approximator based on Radial Basis Functions (RBFs) [16] to classify the resulting $M \times N$ pixel values.

While training the RBF classifier, RBF centers are set in feature space by an agglomerative clustering procedure applied on the available training data. Linear ramps, rather than Gaussians, are used as radial functions, for efficiency purposes. Two radius parameters specify each such ramp, the radius where the ramp initiates (descending from the maximum probability value) and the radius where the ramp is cut off (after which probability value is set 0). These parameters are set based on the distance to the nearest reference vector of the same class and to that of the nearest reference vector of one of the other classes, in a manner described in [14]. The recall stage of the RBF classifier consists of summing probabilities that an unknown feature vector corresponds to a particular class, based on the contributions made by the various RBF centers.

One quickly realizes that the two classes involved (i.e. pedestrian and non-pedestrian) have quite different properties. The pedestrian class is comparably well localized in feature space, while the non-pedestrian class is wide spread-out. Our aim is to accurately model the target class, the pedestrians, while mapping the vast region of non-pedestrian is both impractical and unnecessary. The only instances of the non-pedestrian class really needed are those which lie close to the imaginary border with the target class. In order to find these, an incremental bootstrapping procedure is used, similar to [15]. This procedure adapts at each iteration the RBF classifier based on its performance of a new batch of no-target data. It only adds the non-target class examples which were classified incorrectly to the training set; then, it retrains the RBF classifier.

We take incremental bootstrapping a step further and integrate the detection system into the loop, reflecting the actual system coupling between detection and verification. Each batch of new non-target data is thus prefiltered by the detection unit, which will introduce a useful additional bias towards samples close to the imaginary target vs. non-target border in feature space.

5 Experiments

Experiments with pedestrian detection were performed off-line as well as on-board the Urban Traffic Assistant (UTA) demo vehicle.

We compiled a database of about 1250 distinct pedestrian shapes at a given scale; this number doubled when mirroring the templates across the y-axis. On this set of templates, an initial four-level pedestrian hierarchy was built, following the method described in the previous Section. In order to obtain a more compact representation of the shape distribution and provide some means for generalization, the leaf level was discarded, resulting in the three-level hierarchy used for matching (e.g. Figure 2) with about 900 templates at the new leaf level, per scale. Five scales were used, with range 70-102 pixels.

A number of implementation choices improved the performance and robustness of the Chamfer System, e.g. the use oriented edge features, template subsampling, multi-stage edge segmentation thresholds and ground plane constraints. Applying SIMD processing (MMX) to the main bottlenecks of the system, distance transform computation and correlation, resulted in a speed-up of factor 3-4. See [9].

Our preliminary experiments on a dataset of 900 images with no significant occlusion (distinct from the sequences used for training) showed detection rates in the 60-90 % range using the Chamfer System alone. With this setting, we obtained a handful of false detections solutions per image, of which approximately 90 % were rejected by the RBF classifier, at a cost of falsely rejecting 15 % of the pedestrians correctly detected by the Chamfer System.

Figure 3 illustrates some candidate solutions generated by the Chamfer System. Figure 4 shows intermediate results; matches at various levels of the template hierarchy are illustrated in white, grey and black for the first, second and leaf level, respectively. We undertook various statistics on our dataset, one of which is shown in Figure 5. It shows the cumulative distribution of average chamfer distance values on the path from the root to the "correct" leaf template. The correct leaf template was chosen as the one among the training examples to be most similar with the shape labeled by the human for a particular image. It was Figure 5, rather than Equation (5), that was used to determine the distance thresholds at the nodes of the template hierarchy. For example, from Figure 5 it follows that by having distance thresholds of 5.5, 4.1 and 3.1 for nodes at the first, second and leaf level of the hierarchy, each level passes through about 80% of the correct solutions. Figure 5 provides in essence an indication of the quality of the hierarchical template representation (i.e. how well the templates at the leaf level represent the shape distribution and good the clustering process is).

In general, given image width W, image height H, and K templates, a brute-force matching algorithm would require $W \times H \times K$ correlations between template and image. In the presented hierarchical approach both factors $W \times H$ and K are pruned (by a coarse-to-fine approach in image space and in template space). It is not possible to provide an analytical expression for the speed-up, because it depends on the actual image data and template distribution. Nevertheless, for this pedestrian application, we measured speed-ups of three orders of magnitude.

The Urban Traffic Assistant (UTA) vehicle (Figure 7) is the DaimlerChrysler testbed for driver assistance in the urban environment [7]. It showcases the broader Intelligent Stop & Go function, i.e. the capability to visually "lock" onto a leading vehicle and autonomously follow it, while detecting relevant elements of the traffic infrastructure (e.g. lane boundaries, traffic signs, traffic lights). Detected objects are visualized in a 3-D graphical world in a way that mimicks the configuration in the real world. See Figure 7a. The pedestrian module is a recent addition to UTA. It is being tested on traffic situations such as shown in Figure 8, where, suddenly, a pedestrian crosses the street. If the pedestrian module is used in isolation, the system runs at approximately 1 Hz on a dual-Pentium 450 MHz with MMX; 3-D information can be derived from the flat-

world assumption. In the alternate mode of operation the stereo-module in UTA is used to provide a region of interest for the Chamfer System; this enables a processing speed of about 3 Hz.

For updated results (including video clips) the reader is referred to the author's WWW site **www.gavrila.net**.

6 Discussion

Though we have been quite successfull with the current prototype pedestrian system, evidently, we only stand at the beginning of solving the problem with the degree of reliability necessary to actually deploy such a system. A number of issues remain open in the current system. Starting with the Chamfer System, even though it uses a multi-stage edge segmentation technique, matching is still dependend on a reasonable contour segmentation. Furthermore, the proposed template-based technique will not be very suitable for detecting pedestrians very close to the camera. Currently, a multi-modal shape tracker is being developed (i.e. [12]) to integrate results over time and improve overall detection performance; single-image detection rates of 50% might not be problematic after all. Regarding the verification stage, the choice for a RBF classifier is probably not a determining factor; it would be indeed interesting to compare its performance with that of a Support Vector Machine [15].

The experiments indicated that detection performance varied considerably over parts of our database, according to the degree of contrast. Once the database is extended to include partially occluded pedestrians, or pedestrians at night, this variability is only going to increase, increasing the challenge how to report the detection performance in a representative manner. Also, larger test sets will be needed; we will have an enlarged pedestrian database of 5000 images with ground truth (i.e. labeled pedestrian shapes) in the near future.

7 Conclusions

This paper presented a working prototype system for pedestrian detection onboard a moving vehicle. The system used a generic two-step approach for efficient object detection. The first step involved contour features and a hierarchical template matching approach to efficiently "lock" onto candidate solutions. The second step utilized the richer set of intensity features in a pattern classification approach to verify the candidate solutions (i.e. using Radial Basis Functions). We found that this combined approach was able to deliver quite promising results for the difficult problem of pedestrian detection. With further work on (e.g. temporal integration of results, integration with stereo/IR) we hope to come closer to the demanding performance rates that might be required for actual deployment of such a system.

8 Acknowledgements

The author would like to thank Frank Lindner for making the RBF classifier software available.

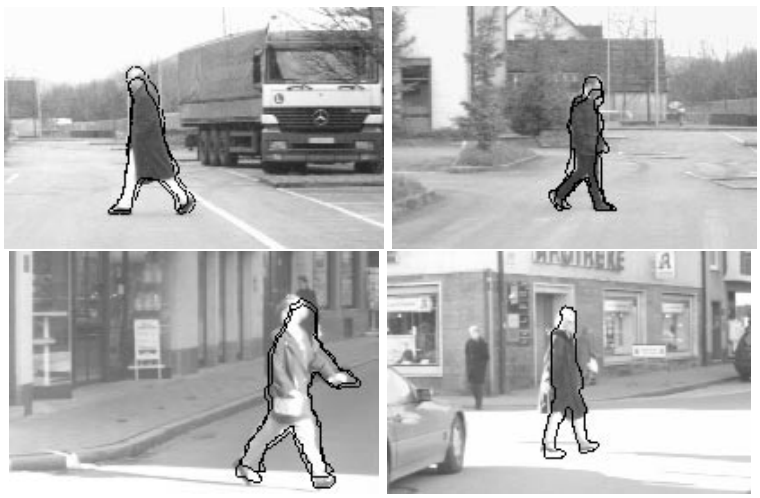

Fig. 3. Pedestrian detection results obtained by the Chamfer System

References

1. H. Barrow et al. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. of the International Joint Conference on Artificial Intelligence*, pages 659–663, 1977.
2. A. Baumberg and D. Hogg. Learning flexible models from image sequences. In *Proc. of the European Conference on Computer Vision*, pages 299–308, 1994.
3. T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and applications. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
4. C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, C. Bruckhoff, T. Bergener, and W. von Seelen. Walking pedestrian detection and classification. In *Proc. of the Deutsche Arbeitsgemeinschaft fr Mustererkennung*, pages 78–85, Bonn, Germany, 1999.
5. R. Cutler and L. Davis. Real-time periodic motion detection, analysis and applications. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–331, Fort Collins, U.S.A., 1999.
6. Infosystem der Deutschen Verkehrssicherheitsrates. Unfallstatistik fussgänger. In *www.bg-dvr.de*, 1996.
7. U. Franke, D. M. Gavrila, S. Görzig, F. Lindner, F. Paetzold, and C. Wöhler. Autonomous driving goes downtown. *IEEE Intelligent Systems*, 13(6):40–48, 1998.

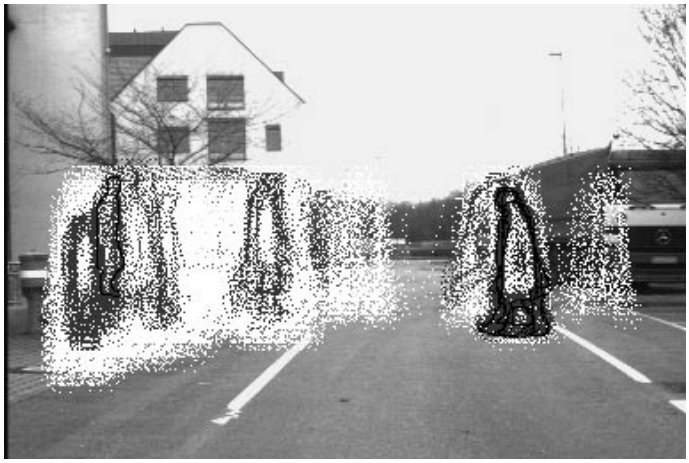

Fig. 4. Intermediate matching results for a 3-level template hierarchy: templates matched succesfully at levels 1, 2, 3 (leaf) are shown in white, grey, and black, respectively.

8. D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.
9. D. M. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *Proc. of the International Conference on Computer Vision*, pages 87–93, Kerkyra, Greece, 1999.
10. B. Heisele and C. Wöhler. Motion-based recognition of pedestrians. In *Proc. of the International Conference on Pattern Recognition*, 1998.
11. D. Huttenlocher, G. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.
12. M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 1998.
13. S. Kirkpatrick, Jr. C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1993.
14. U. Kressel, F. Lindner, C. Wöhler, and A. Linz. Hypothesis verification based on classification at unequal error rates. In *Proc. of ICANN*, 1999.
15. C. Papageorgiou and T. Poggio. A pattern classification approach to dynamical object detection. In *Proc. of the International Conference on Computer Vision*, pages 1223–1228, Kerkyra, Greece, 1999.
16. T. Poggio and F. Girosi. Networks for approximation and learning. *Proc. of the IEEE*, 78(9):1481–1497, 1990.
17. R. Polana and R. Nelson. Low level recognition of human motion. In *Proc. of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 77–82, Austin, 1994.
18. G. Rigoll, B. Winterstein, and S. Mller. Robust person tracking in real scenarios with non-stationary background using a statistical computer vision approach. In *Proc. of Second IEEE Int. Workshop on Visual Surveillance*, pages 41–47, Fort Collins, USA, 1999.

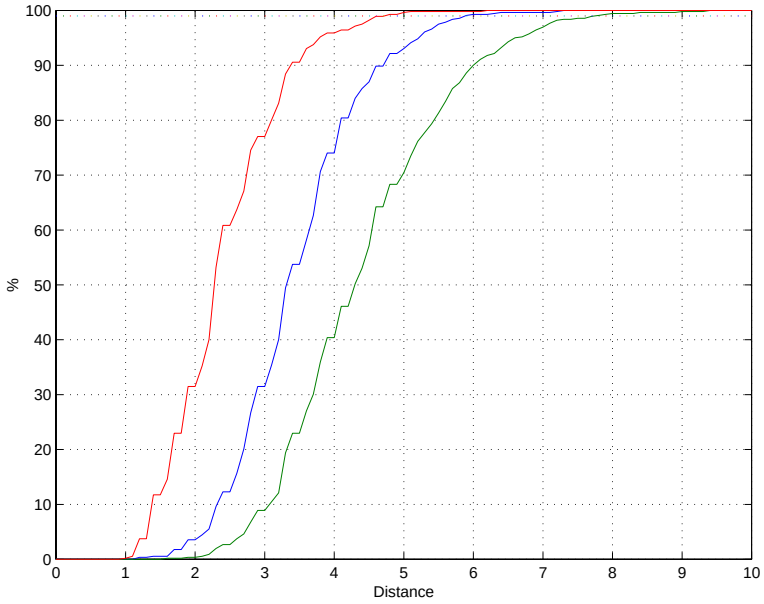

Fig. 5. Cumulative distribution of average chamfer distance values on the path from the root to the "correct" leaf template: first (right curve), second (middle curve) and leaf level (left curve).

This article was processed using the LaTeX macro package with LLNCS style

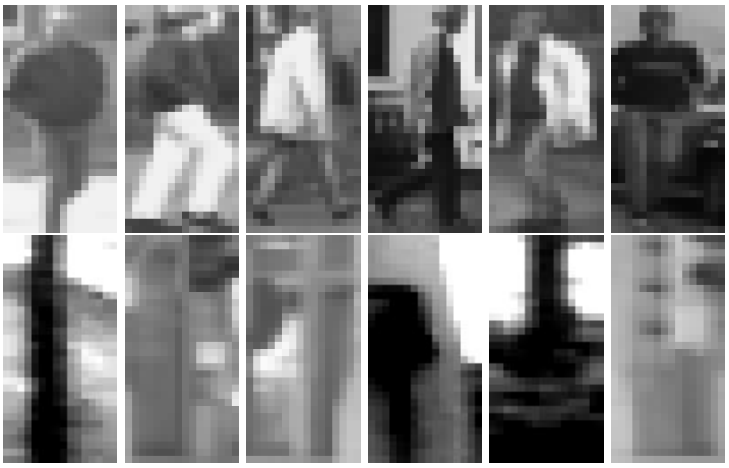

Fig. 6. RBF-based verification: accepted (top row) and rejected (bottom row) candidate solutions

(a)

(b)

Fig. 7. The Urban Traffic Assistant (UTA) demonstration vehicle: (a) inside and (b) outside view

Fig. 8. A potentially dangerous traffic situation: a pedestrian suddenly crossing the street