A Bayesian, Exemplar-Based Approach to Hierarchical Shape Matching

Dariu M. Gavrila

Abstract—This paper presents a novel probabilistic approach to hierarchical, exemplar-based shape matching. No feature correspondence is needed among exemplars, just a suitable pairwise similarity measure. The approach uses a template tree to efficiently represent and match the variety of shape exemplars. The tree is generated offline by a bottom-up clustering approach using stochastic optimization. Online matching involves a simultaneous coarse-to-fine approach over the template tree and over the transformation parameters. The main contribution of this paper is a Bayesian model to estimate the a posteriori probability of the object class, after a certain match at a node of the tree. This model takes into account object scale and saliency and allows for a principled setting of the matching thresholds such that unpromising paths in the tree traversal process are eliminated early on. The proposed approach was tested in a variety of application domains. Here, results are presented on one of the more challenging domains: real-time pedestrian detection from a moving vehicle. A significant speed-up is obtained when comparing the proposed probabilistic matching approach with a manually tuned nonprobabilistic variant, both utilizing the same template tree structure.

Index Terms—Hierarchical shape matching, chamfer distance, Bayesian models.

1 Introduction

Object detection is one of the central tasks in image understanding. Among the various visual cues that can be used to segment and compare objects, shape has the advantage that it provides a powerful object discrimination capability that is relatively stable to changes in lighting conditions.

This paper presents a novel Bayesian approach for hierarchical shape-based object representation and matching. It integrates a number of desirable features: generality, robustness, and efficiency. The generality refers to the ability to deal with arbitrary shapes, whether parameterized (e.g., polygons, ellipses) or not (e.g., outlines of pedestrians), whether involving closed contours or not. See Fig. 1. Objects are described in terms of a set of training shapes or exemplars, which cover the set of possible appearances due to geometrical transformations (e.g., rotation, scale) and intraclass variance (e.g., different pedestrians, different poses). Nontraining object samples are covered by a defined maximum allowable dissimilarity from a closest exemplar in the training set. Thus, no feature correspondence is required, only pairwise dissimilarities.

The proposed system is robust due to its use of template matching. It copes relatively well with the effects of suboptimal segmentation (e.g., "edge gaps") or partial occlusion by the use of correlation which integrates contributions at various image locations independently of each other.

• *The author is with the Machine Perception Department of DaimlerChrysler R&D, Wilhelm Runge St. 11, 89081 Ulm, Germany and with the Intelligent Systems Lab, Faculty of Science, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands.*
E-mail: dariu.gavrila@DaimlerChrysler.com.

Manuscript received 24 Jan. 2006; revised 4 Aug. 2006; accepted 19 Sept. 2006; published online 18 Jan. 2007.
Recommended for acceptance by L. Van Gool.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0041-0106.
Digital Object Identifier no. 10.1109/TPAMI.2007.1062.

Template-based systems are however notoriously computationally intensive and, therefore, it is especially with respect to efficiency that the proposed approach can make a difference. It employs a combined coarse-to-fine approach over a hierarchical shape representation and transformation parameters, which results in significant speed-ups compared to brute-force formulations; gains of several orders of magnitude are typical. Central is its ability to employ pruning techniques and to deal with object shape variations by means of distance transforms.

The proposed object representation and matching approach contains the following components:

- a set of exemplars capturing object appearance,
- a pairwise similarity measure between exemplars,
- recursive clustering and prototype selection: offline tree construction, and
- a (probabilistic) matching criterion: online tree traversal.

This formulation is very generic and applies to a large class of object detection systems. In this paper, we consider a particular instantiation based on shape-cues where the pairwise similarity measure between shape exemplars is the chamfer distance based on oriented edges [12], [23]. The tree construction process consists of a recursive clustering procedure where the objective function, the average intracluster similarity, is optimized stochastically by simulated annealing.

The main contribution of this paper is a Bayesian model for estimating the a posteriori probability of the object class, after a certain match at a node of the tree. This model takes into account object scale and saliency, and allows for a principled setting of the matching thresholds at the tree nodes such that unpromising paths are pruned early on during the tree traversal process. Fig. 2 illustrates the overall approach.

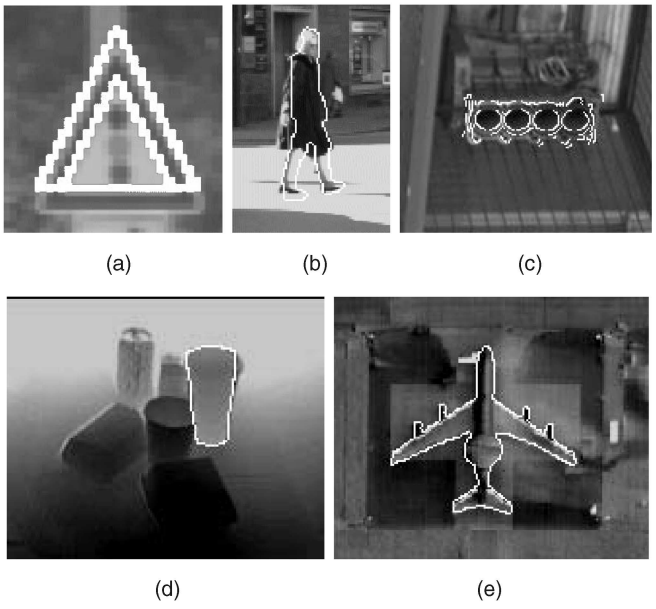


Fig. 1. Applications: detection of (a) traffic signs, (b) pedestrians, (c) engine parts, (d) objects in range images, and (e) planes.

The outline of the paper is as follows: Section 2 discusses previous work. Section 3 reviews the basic building blocks of hierarchical shape matching: the use of distance transforms for shape matching, the offline construction of the template tree, and, finally, the online matching. Section 4 discusses various quality measures of a shape-based exemplar representation such as specificity, coverage, and compactness. Furthermore, the effect of object scale is analyzed. This sets the stage for the probabilistic hierarchical matching model described in Section 5. The experiments are listed in Section 6, involving many thousands of images with ground truth data. Section 7 puts the proposed approach into context and identifies areas of improvement. Finally, Section 8 lists the conclusions.

2 Previous Work

There is a large body of literature on shape representation and matching, see, for example, a recent review by Zhang and Lu [30]. One line of research has dealt with learning shape models from a set of (closed-contour) training shapes. Shape registration [13], [26] plays herein a central role. It involves bringing the points across multiple shapes into correspondence, factoring out variations due to geometrical transformations between shapes (e.g., similarity) and maintaining only those changes related to inherent shape variation of the object class. The established point correspondences allow embedding the training shapes into a feature vector space, which in turn enables the computation of various compact parametric shape representations based on radial (mean-variance) [14] or modal (linear subspace, PCA) [6] decompositions or combinations there of [15].

Automatic shape registration methods only stand a reasonable chance of success if the respective shapes are sufficiently similar. For example, known methods will fail to correctly register a pedestrian shape viewed sideways with the feet apart to one with the feet together. This has negative implications in terms of the specificity of the derived shape model, as physically implausible, interpolated shapes are being represented. In order to cope with a larger set of shape variations in the training set, Duta et al. [8] and Gavrila et al. [10] combine shape registration and clustering and derive from the training samples a representation in terms of K shape clusters, where only the (similar) shapes within a cluster are embedded into the same vector space.

In this paper, we consider a shape representation and matching approach that makes even weaker assumptions, in the sense that it does not require closed contours and/or

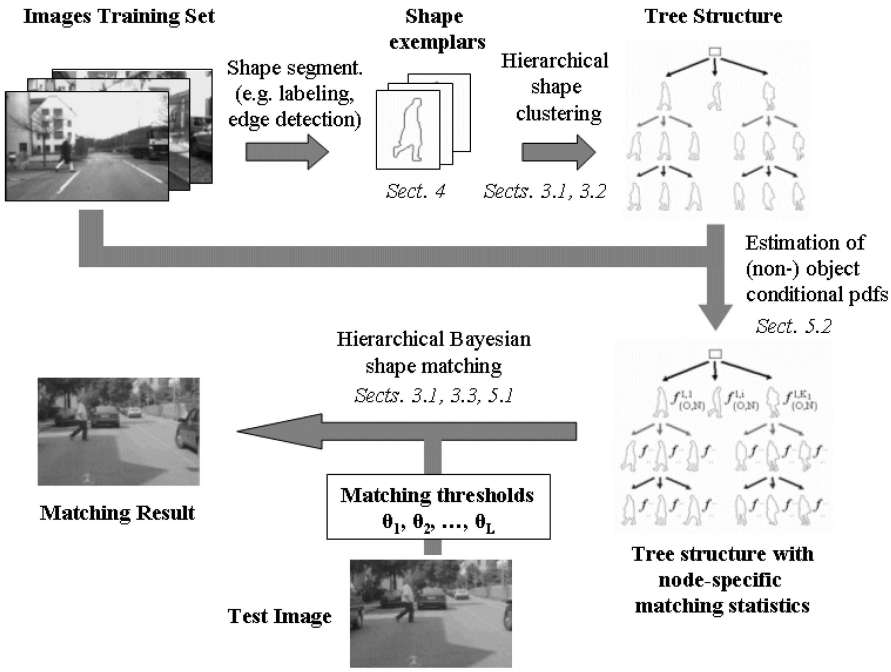


Fig. 2. Overview of the proposed Bayesian exemplar-based approach to hierarchical shape matching.

shape registration altogether. Instead, it relies only upon a pairwise similarity measure between the shape exemplars. Gavrila and Philomin [12] and Gdalyahu and Weinshall [13] first explored the use of a hierarchical shape representation built bottom-up from a set of shapes by dissimilarity-based clustering. Gavrila and Philomin [12] introduced this approach for the purpose of efficient exemplar-based object detection. No particular constraints on the shape exemplars were assumed. The tree was built by recursive partitional clustering based on distance transforms (DTs). Simulated annealing was used to obtain a good clustering solution at each level of the tree. Gdalyahu and Weinshall [13] considered closed contours and performed clustering based on the L_2 norm of automatically registered points. Their application context was fast retrieval of (already segmented) shapes. Later, work by Srivastava et al. [26] considered closed contours and shape retrieval as well, but involved geodesics for establishing correspondence. This allowed the computation of Karcher mean shapes. Olson and Huttenlocher [23] and Amit et al. [1] also construct a hierarchical representation; given their use of binary correlation in the later matching stage, the shape prototype is selected so as to capture overlapping pixels among the shape exemplars. Olson and Huttenlocher [23] cluster by the chamfer distance, whereas Amit et al. [1] use the Hamming distance. An interesting alternative measure for shape similarity is the use of shape contexts [3].

Given a hierarchical structure derived from a set of 2D shape exemplars by clustering, the next issue is how to use it for matching. Srivastava et al. [26] suggest binary hypothesis testing to distinguish between two probabilistic shape models. Their idea is to start at the top, compare the query with the shapes at each level, and proceed down the branch following the best match. Assuming K possible shapes at a particular level of the tree, this can be performed by $K-1$ binary tests. In experiments, this is simplified to selecting the best matching shape. Amit et al. [1] introduce successive approximations to likelihood tests arising from a naive Bayesian statistical model for the edge maps extracted from the original images.

Hierarchical approaches have also been used for speeding up matching with a single shape exemplar. Borgefors [5] uses multiple image resolutions for DT-based matching. Others use a pruning [16], [24] or a coarse-to-fine approach [25] in the parameter space of relevant template transformations. The latter approaches take advantage of the smooth similarity measure associated with DT-based matching; one need not match a template for each location, rotation, or other transformation.

Exemplar-based shape representations have furthermore been applied to tracking. Toyama and Blake [28] devise a probabilistic framework, termed Metric Mixture, which they use for tracking human bodies and mouths. Stenger et al. [27] extend the hierarchical representation of [12] by means of a Bayesian Filter.

Considering previous work, this paper is most related to [27] and [1] in the sense that no constraints are imposed on allowable shapes and in that a hierarchical exemplar representation is combined with a probabilistic matching model. However, the probabilistic model proposed here is considerably different. Stenger et al. [27] consider a tracking context, where the existence of the object class is implicitly assumed. Their aim is how to efficiently compute the density function over the state space and adapt this over time. In our detection context, the existence of the object class needs to be determined in the first place, thus, statistics regarding the background also need to be considered. Furthermore, we do not assume an underlying parameter space which we can sample in coarse-to-fine fashion, and which defines a hierarchical exemplar representation. Even when such parameter space would be available, this approach is likely to introduce redundancy in the representation, i.e., when distinct parameter settings generate similar shape exemplars. Here, the hierarchical structure is determined bottom-up, by shape clustering directly based on pairwise dissimilarity.

Compared to the detection approach of [1], we do not require 100 percent detection; instead, we define the associated matching thresholds based on a Bayesian a posteriori criterion. Hierarchy construction and matching is furthermore differentiated by our use of distance transforms as opposed to binary correlation with spread edges.

3 Basic Approach

This section reviews the basic components of hierarchical exemplar-based shape detection, as covered by our earlier work [12]. It involves the definition of a pairwise similarity measure between shape exemplars based on DT (Section 3.1), the offline construction of a hierarchical representation (Section 3.2), and the online hierarchical traversal for matching (Section 3.3).

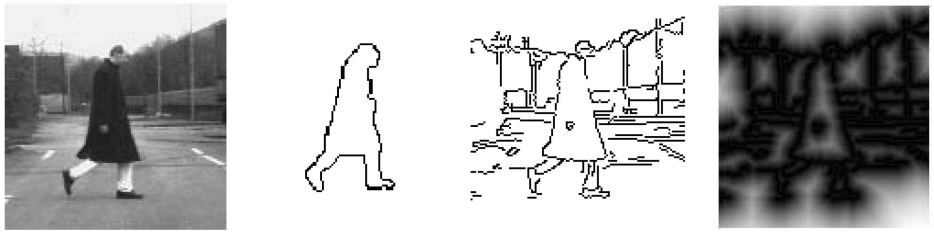
3.1 Similarity Measure: Distance Transforms

Image matching with distance transforms (DTs) involves two binary images, a segmented template T and a segmented image I, termed "feature template" and "feature image," respectively. The binary pixel values encode the presence/absence of a feature (e.g., edges) at a particular location. Matching T and I involves computing the DT of the feature image I. This transform converts a binary feature image into a nonbinary 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 distances are computed [4]. Of particular interest is the class of sequential DTs, or *chamfer* transforms, which approximate global distances in the image by propagating local distances in raster scan fashion. The *chamfer-2-3* variant [2], which we will use in the experiments, uses a 3×3 neighborhood with values 2 and 3 to denote distances between horizontal/vertical neighbors and diagonal neighbors, respectively.

After computing the DT, the template T is mapped onto the DT image of I by a transformation G (e.g., translation, rotation, scale); the matching measure $D_G(T, I)$ is determined by the pixel values of the DT image which lie "under" the feature 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. One possible matching measure is the average directed chamfer distance [2]

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

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. Other more robust and computationally



(a) (b) (c) (d)

Fig. 3. (a) Original image. (b) Template. (c) Edge image. (d) DT image.

intensive measures reduce the effect of missing features (i.e., due to occlusion or segmentation errors) by using the average truncated distance or the fth quantile value (the *Hausdorff* distance) [16]. Further work weighs individual pixel distance contributions based on probabilistic models for pixel-adjacency [22].

Once DTs and match measure have been defined, the task of a DT-based object detection system involves finding a geometrical transformation G for which the distance measure $D_G(T, I)$ lies below a user-supplied dissimilarity threshold θ

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

Fig. 3 illustrates the DT-based matching scheme for the typical case of edge features. The advantage of matching a template (Fig. 3b) with the DT image (Fig. 3d) rather than with the edge image (Fig. 3c) is that the resulting similarity measure will be smoother as a function of the template transformation parameters. This enables the use of various efficient search algorithms to lock onto the correct solution, as will be discussed shortly. It also allows more variability between a template and an object of interest in the image. Matching with the unsegmented (gradient) image, on the other hand, typically provides strong peak responses but rapidly declining off-peak responses.

3.2 Construction of Template Tree: Recursive Partitional Clustering

The basic idea for achieving an efficient shape representation is to group similar templates together and represent them by 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 significant speed-up can be achieved online. When applied recursively, this grouping leads to a template tree, see Fig. 4.

The starting point is a set of templates (or shape exemplars) which cover inherent object shape variations and allowable geometrical transformations other than translation (e.g., rotation, scale). The templates are assumed aligned with respect to translation. A template tree is subsequently constructed on top of these templates. The proposed algorithm involves a bottom-up approach and a partitional clustering step at each level of the tree. The input to the algorithm is a set of templates $\mathbf{t}_1, \ldots, \mathbf{t}_N$, a method to obtain a prototype template $\mathbf{p}_k$ from a subset of templates and the desired partition size K. The output is the K-partition and the prototype templates $\mathbf{p}_1, \ldots, \mathbf{p}_K$ for each of the K groups $S_1, \ldots, S_K$. At the leaf level, the input is provided by the shape examples, whereas, at the nonleaf level, it is the prototypes derived at the previous clustering step, one level lower.

K-way clustering is implemented as an iterative function optimization process. Starting with an initial (random) partition, templates are moved back and forth between groups, while the following objective function E is minimized

$$E = \sum_{k=1}^{K} \sum_{i=1}^{n_k} D(\mathbf{t}_i, \mathbf{p}_k^*). \tag{3}$$

Here, $D(\mathbf{t}_i, \mathbf{p}_k^*)$ denotes the distance measure between the ith element of group k and the prototype $\mathbf{p}_k^*$ for that group at the current iteration. n_k denotes the current size of group k. Given the availability of shape correspondence, $\mathbf{p}_k^*$ involves the analytically computed mean shape [6], [26]. In the more general case of no shape correspondence, $\mathbf{p}_k^*$ can be taken as the template with the smallest mean dissimilarity to the other templates in a group. This allows the offline computation of D to be stored as a dissimilarity matrix.

A low E-value is desirable since it implies a tight grouping; this lowers the distance threshold that will be required during matching (see (6)), which, in turn, likely decreases the number of locations which one needs to consider during matching. Simulated Annealing (SA) [18] is used to perform the minimization of E. SA is a well-known stochastic optimization technique where, during the initial stages of the search procedure, moves can be accepted which increase the objective function. The aim is to do enough exploration of the search space, before resorting to greedy moves, in order to avoid local minima. Candidate moves are accepted according to probability p:

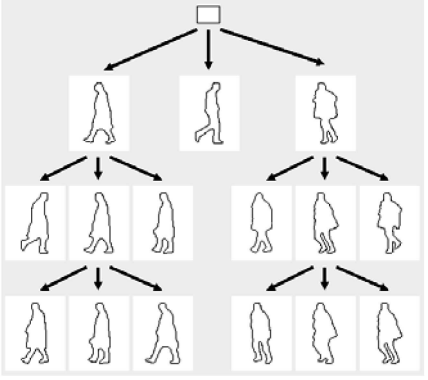


Fig. 4. A hierarchical structure for pedestrian shapes (partial view).

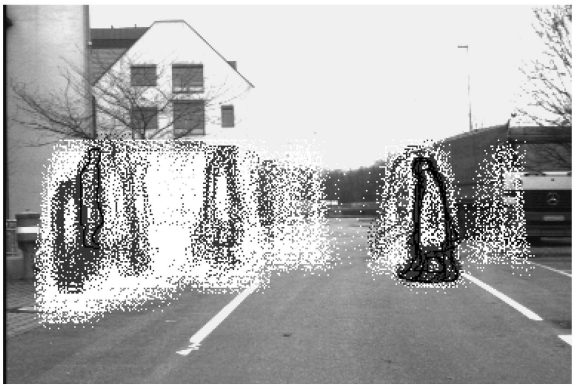


Fig. 5. Intermediate matching results for a three-level template tree: Templates matched successfully at levels 1, 2, 3 (leaf) are shown in white, gray, and black, respectively.

$$p = \frac{1}{1 + e^{\frac{\Delta E}{T}}}, \tag{4}$$

where T is the temperature parameter which is adjusted according to a certain "cooling" schedule (we use an exponential schedule [18]).

Other algorithms could have been used for partitional clustering based on similarity values, e.g., [9], [29]. SA has some appealing theoretical properties, such as convergence to the global minimum in the limiting case (e.g., sufficient high initial temperature, infinitesimal small temperature decrements). Although the underlying optimality conditions cannot be met in practice, we selected SA because it nevertheless tends to outperform deterministic approaches at still manageable large iteration counts. The drawback of its computational cost is not a major issue, considering the template tree is constructed offline. It is worth allocating substantial resources to devise an efficient representation offline (in the sense of minimizing E) because this translates in online computational gains. See Fig. 4 for a typical partial view. Observe how the shape similarity increases toward the leaf level.

3.3 Hierarchical Matching

Online matching can be seen as traversing the tree structure of templates. Processing a node involves matching the corresponding (prototype) template $\mathbf{p}$ with the image at some interest locations. For the locations where the distance measure between template and image is below a user-supplied threshold θ_p, the child nodes are added to the list of nodes to be processed. For locations where the distance measure is above-threshold, search does not propagate to the subtree; it is this pruning capability that brings large efficiency gains.

The above coarse-to-fine approach is combined with a coarse-to-fine approach over the transformation parameters (i.e., translation). Image locations where matching is successful for a particular nonleaf node give rise to a new set of interest locations for the child nodes on a finer grid in the vicinity of the original locations. See Figs. 5 and 6. At the root, the interest locations lie on a uniform grid over the image. By following a path in the tree toward the leaf node, both template suitability and template localization increase. Final detections are the successful matches at the leaf level of the tree.

Let $\mathbf{p}$ be the template corresponding to the node currently processed during traversal at level l and let $C = \{\mathbf{t}_1, \ldots, \mathbf{t}_c\}$ be the set of templates corresponding to its child nodes. Let δ_p be the maximum distance between $\mathbf{p}$ and the elements of C.

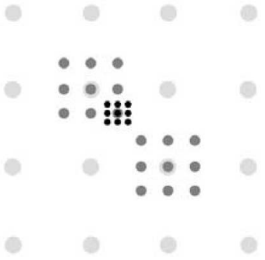


Fig. 6. Illustration of expanded interest locations on a coarse-to-fine grid as search goes from top (large light gray dots) to intermediate (medium, dark gray dots) and leaf level (small black dots) in a three level template tree.

$$\delta_p = \max_{t_i \in C} D(\mathbf{p}, \mathbf{t}_i). \tag{5}$$

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

$$\theta_p = \tau_{tol} + \delta_p + \frac{1}{2}\mu\sigma_l, \tag{6}$$

one has the desirable property that, using untruncated distance measures such as the chamfer distance, one can guarantee that the above coarse-to-fine approach using the template tree will not miss a solution.

Comparing the above hierarchical matching approach with an equivalent brute-force method, one observes that, given image width W, image height H, and K templates, the brute-force version would require $W \times H \times K$ correlations. In the presented hierarchical version, both factors $W \times H$ and K are pruned (by a coarse-to-fine approach in transformation space and in template space). It is not possible to provide an analytical expression for the speed-up, since it depends on the actual image data and template distribution. We measured gains of several orders of magnitude in the applications we considered.

4 Shape Exemplar Representation

The hierarchical structure discussed previously is motivated by efficiency considerations. The best achievable detection performance, i.e., correct versus false detections, is however capped by the matching results obtained at the leaf level. This, in turn, depends on how well the shape exemplars and associated dissimilarity thresholds represent object variation.

Here, we consider some qualitative aspects of a non-hierarchical, "flat" exemplar representation. We refer to the *coverage* and *specificity* of a particular exemplar-based representation as the degree that possible object and nonobject instantiations lie within a given dissimilarity interval from a nearby shape exemplar, respectively. We refer to *compactness* as to the degree that possible object instantiations lie within the dissimilarity interval from multiple shape exemplars. See Fig. 7 for a visualization, simplified in the sense that, in reality, the exemplars are not necessarily embedded in a common feature vector space. Increasing the number of exemplars is generally favorable

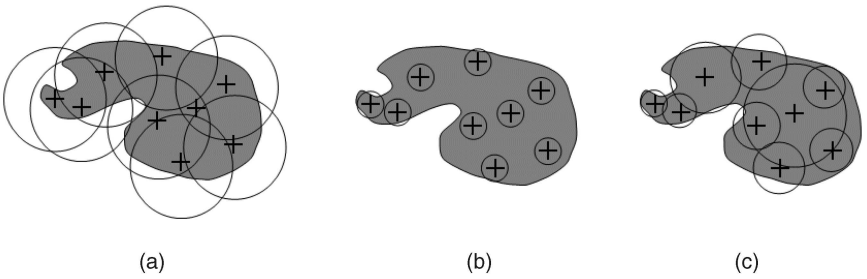


Fig. 7. Representation of object manifold by exemplars. (a) Good coverage, bad specificity, bad compactness. (b) Bad coverage, good specificity, good compactness. (c) Reasonable trade-off between coverage, specificity, and compactness.

from the detection performance point of view. An enlarged (representative) exemplar set allows decreasing the dissimilarity thresholds, increasing specificity without decreasing coverage. These improvements in detection performance need, however, to be balanced with increased memory and computational cost, especially when the compactness of a representation degrades.

An important issue is how to match at multiple object scales, given that the exemplar-based representation is not scale-invariant. One possibility is to maintain the shape exemplars at a single scale and resize the image accordingly. This approach avoids the memory cost of storing exemplars at multiple scales. However, this comes at the expense of possible lower matching performance, when, due to lower image resolution, segmentation is degraded (e.g., edge segmentation in Section 3.1).

In the remainder of this paper, we consider the case where, due to efficiency reasons, shape exemplars are pregenerated at multiple scales. Matching such multiscale representation could simply involve scaling-up a dissimilarity threshold accordingly. However, when scaling up, increasing the distance thresholds in many cases results in a degradation of specificity of the representation. This is because of the presence of spurious (edge) features in the background, whose density is independent of the scale of the object, and which are increasingly mismatched. In order to maintain a particular detection performance, it will be necessary to counteract this effect by increasing the number of exemplars in the training set.

We experimentally determine how detection performance is influenced by the number of shape exemplars and how this depends on increasing object scale. This allows an appropriate choice for the shape exemplars at the leaf level, i.e., on which the tree is built, see Section 6.1.

5 Probabilistic Matching

5.1 Probabilistic Model

One important question is how to set the matching thresholds associated with each node in the template tree in a principled manner. Manual parameter setting is not practical for trees that can contain hundreds if not thousands of nodes. One possibility is to use (6), but the resulting thresholds are, in practice, very conservative. In many applications, one can lower the thresholds to speed up matching at the cost of possibly missing a solution. In this section, we are interested to derive an a posteriori probability criterion on which to base our decision rule (thresholds).

First, we introduce some notation. Binary state random variable $X \in \{O, N\}$ denotes the presence of an object O or background N at a particular node and image location. At the leaf level of the tree, the object class O occurs for the best matching template at the best location. Furthermore, the object class O_l occurs at level l for the ("optimal") path from the root to the best matching leaf level node, together with the associated locations on the coarse-to-fine image grid, e.g., Fig. 6 (for notational simplicity, we do not include in the remainder the subscripts regarding image location). The dissimilarity measurement obtained at the $l\text{th}$ level of the tree, associated with random variable D_l, is denoted by $d_l \in \Re$. Define $d_{1:l} = \{d_i\}_{i=1}^{l}$ to be the measurements from the top level up to level l, along a particular path in the tree.

Desired is a Bayesian framework for modeling the a posteriori probability of the object class at a particular node of the tree, given (dissimilarity) measurements along the path to that node. Given the Bayes rule

$$p(O_l|d_{1:l}) = \frac{p(O_l)\ p(d_{1:l}|O_l)}{p(d_{1:l})} \tag{7}$$

and

$$p(d_{1:l}) = p(O_l)\ p(d_{1:l}|O_l) + p(N_l)\ p(d_{1:l}|N_l), \tag{8}$$

one obtains

$$\begin{aligned} p(O_l|d_{1:l}) &= \frac{p(O_l)\ p(d_{1:l}|O_l)}{p(O_l)\ p(d_{1:l}|O_l) + p(N_l)\ p(d_{1:l}|N_l)} \\ &= \frac{1}{1 + \frac{p(N_l)}{p(O_l)}\ \frac{p(d_{1:l}|N_l)}{p(d_{1:l}|O_l)}}. \end{aligned} \tag{9}$$

Assuming the following Markov property along the path from the root to the current node,

$$p(d_l|d_{1:l-1}X_l) = p(d_l|d_{l-1}X_l), \tag{10}$$

and considering three possible transitions from a parent node at level $l-1$ to a current node at level l,

1. $O_{l-1}O_l$: both parent and current node lie on the optimal path,
2. $O_{l-1}N_l$: parent lies on the optimal path but current node does not, and
3. $N_{l-1}N_l$: parent does not lie on the optimal path (and, consequently, neither does current node).

We arrive to the following recursive form of the posterior (see the Appendix for derivation):

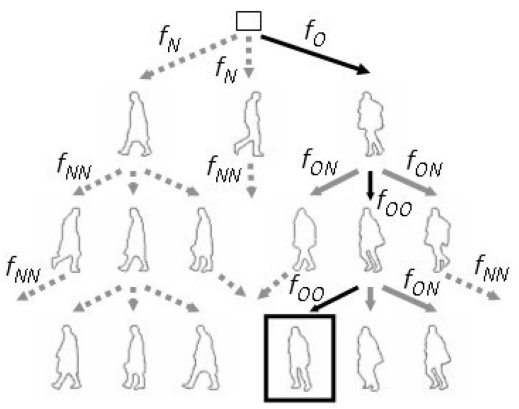


Fig. 8. Collecting distance measurements during training for the purpose of estimating $f_O(d_1)$ and $f_{OO}(d_l|d_{l-1})$ (solid black), $f_{ON}(d_l)$ (solid gray) and, $f_N(d_1)$ and $f_{NN}(d_l|d_{l-1})$ (dotted gray). The best matching solution at the leaf level is marked by a rectangle. Figure does not capture multiple image locations.

$$p(O_l|d_{1:l}) = \frac{1}{1+\alpha_l} \tag{11}$$

with for $l > 1$

$$\alpha_l = \frac{p(O_{l-1}|d_{1:l-1})\, p(d_l|d_{l-1}N_lO_{l-1})\, p(N_l|O_{l-1}) + p(N_{l-1}|d_{1:l-1})\, p(d_l|d_{l-1}N_lN_{l-1})}{p(O_{l-1}|d_{1:l-1})\, p(d_l|d_{l-1}O_lO_{l-1})\, p(O_l|O_{l-1})}$$

and

$$\alpha_1 = \frac{p(N_1)}{p(O_1)} \frac{p(d_1|N_1)}{p(d_1|O_1)}$$

Let $f_{XY}(d_l|d_{l-1})$ and $f_X(d_1)$ denote the conditional probability functions associated with $p(d_l|d_{l-1}XY)$ and $p(d_1|X)$, respectively. Approximations for the various f values are derived from histogramming dissimilarity measurements at the nodes of the template tree. For example, $f_{OO}(d_l|d_{l-1})$ is derived by collecting dissimilarity measurements in training images at nonleaf nodes along the path from the top to the best matching template at the leaf level. $f_{ON}(d_l|d_{l-1})$ is derived by collecting the dissimilarity measurements at the nodes and locations which, at the current level, deviate from this optimal path. $f_{NN}(d_l|d_{l-1})$ is derived by collecting dissimilarity measurements not on the optimal path at the current or previous level. See Fig. 8.

5.2 Model Instantiation

In practice, it is possible to collect sufficient data for a good approximation of f_{XY} at the higher levels of the tree, where the nodes are frequently accessed. When examples are scarce (e.g., typically pertaining to the object class), the aggregation of dissimilarity measurements at various nodes of the tree and/or the use of parametric models becomes necessary. Denote a particular node by its level l in the tree and by shape t and scale s of underlying template. We model

$$\begin{cases} f_O^{l,t,s}(d_l) = f_O^{l,s}(d_l), & l = 1 \\ f_{OO}^{l,t,s}(d_l|d_{l-1}) = f_{OO}^{l,s}(d_l|d_{l-1}), & l > 1. \end{cases} \tag{12}$$

Thus, given the presence of the object class, dissimilarity measurements observed at a node are assumed dependent of the level (accounting for the varying search grid size and number of prototypes at a level) and object scale (as discussed in Section 4); they are assumed to be independent of the particular template shape. Similarly, we model

$$\left\{ f_{ON}^{l,t,s}(d_l|d_{l-1}) = f_{ON}^{l,s}(d_l|d_{l-1}), \quad l > 1. \right. \tag{13}$$

For a transition within the nonobject class, we make no such assumptions and maintain

$$\begin{cases} f_N^{l,t,s}(d_l) & l = 1 \\ f_{NN}^{l,t,s}(d_l|d_{l-1}) & l > 1. \end{cases} \tag{14}$$

Thus, in addition to level and object scale, dissimilarities are also assumed dependent on template shape (introducing the aspect of template saliency).

The chi-square and exponential distribution were used earlier to model $f_O(d)$ in the nonhierarchical context [28]. Our experiments indicated an appreciable imprecision in modeling the tail of various distributions. We therefore chose to incorporate additional degrees of freedom by means of the gamma distribution. The gamma probability density function, parameterized by a and b, is given by

$$y = f(d|a,b) = \frac{1}{b^a\Gamma(a)}\, d^{a-1}e^{-\frac{d}{b}} \quad a > 0,\ b > 0, \tag{15}$$

where $d \in [0, \infty)$. The gamma function Γ is defined by the integral

$$\Gamma(a) = \int_0^\infty x^{a-1}e^{-x}dx,\ a > 0. \tag{16}$$

The chi-square and exponential distribution are special cases of the gamma distribution, namely, for $b = 2$ and $a = b = 1$, respectively. The experiments show that distributions $f_O^{l,s}(d_l)$, $l \geq 1$ are very well fitted by the gamma distribution, see Section 4. The same applies for $f_{XX}^{l,t,s}(d_l|d_{l-1})$, $l > 1$, given a discretization of d_{l-1}.

The sole distribution not fitted well by the gamma distribution (or by other well-known parametric distributions) in our preliminary experiments was $f_N^{1,t,s}(d_1)$. We chose to fit a nonparametric model using normal kernel smoothing [20].

Finally, given that a parent node at level $l-1$ has C children and each candidate location is expanded into P new locations (on a finer grid, see Fig. 6), we model:

$$p(O_l|O_{l-1}) = \frac{1}{C\,P}. \tag{17}$$

Trivially, $p(N_l|O_{l-1}) = 1 - p(O_l|O_{l-1})$.

We are now in the position to derive node-specific dissimilarity thresholds based on three different criteria. The first two are based on dissimilarity values directly, the third on a probability criterion. As a first option, one can specify a desired object throughput rate θ_l at each level of the tree. The associated dissimilarity thresholds δ are selected such that

$$\theta_l = F_O^{l,s}(\delta_{l,s}), \tag{18}$$

where $F_O^{l,s}$ is the cumulative distribution function associated with $f_O^{l,s}$. Similarly, when specifying a nonobject throughput rate θ_l for a certain tree level, one obtains

$$\theta_l = F_N^{l,t,s}(\delta_{l,t,s}). \tag{19}$$

Alternatively, one can specify a threshold θ_l on the minimum a posteriori probability by (11). Obviously, the three criteria cannot be set independently. The last criterion

Fig. 9. Examples of the object and nonobject class in the test set, shown in the top and bottom row, respectively.

has the advantage that it allows direct control of the efficiency of the hierarchical matching process, avoiding the exploration of unpromising paths in the tree. It is the criterion we will use at the experiments in next section.

6 Experiments

We tested the basic version of the hierarchical shape detector (Section 3) in a wide range of applications, from the detection of traffic signs and pedestrians from a moving vehicle, plane detection in aerial images, engine detection for visual inspection, to 3D object localization in depth images for robot vision. See Fig. 1.

Given the large shape variation, the lack of an explicit model and the difficult segmentation problem, the pedestrian application is certainly the most challenging among these. For example, we had to use more than 100 times as many shape exemplars for the pedestrian application as for the traffic sign application in order to obtain a decent performance; the traffic sign application involves rigid objects of few standardized dimensions and shapes. Furthermore, segmenting the edges of pedestrians is more challenging than those of traffic signs because of less pronounced contrast. Considering finally the relevance of pedestrian detection for a number of important application settings (e.g., driver assistance systems, visual surveillance), we selected this task to illustrate the concepts discussed in Sections 4 and 5.

The data sets used in this section involved a wide variety of pedestrian appearances with different poses (standing versus running), clothes, and ages of pedestrians and various (day time) lighting conditions. The pedestrians were not significantly occluded. Ground truth data was obtained by manually labeling the pedestrian contours. The training and test sets were separated.

In all experiments, we used the average directed *chamfer-2-3* distance (1) as the dissimilarity measure because of efficiency considerations. To alleviate the effects of missing data, distance contributions of individual pixels were truncated before being averaged. Chamfer images and distances were separately computed for eight different edge orientation-intervals following [12], [23]; matching results for the individual edge orientation intervals were summed to an overall match measure.

6.1 Nonhierarchical Exemplar Pedestrian Representation

In order to obtain an indication of the appropriate number of exemplars needed at the leaf level of the tree, we first conducted tests on a nonhierarchical representation. ROC detection performance was related to the number of exemplars and object scale (see Section 4). The training set consisted of a set of 5,749 binary images representing manually labeled pedestrian shapes. Partitional clustering was applied on this data set for various values of K, i.e., $K = 50$, 150, 500, 1,500, and 5,749, see Section 3.2. The resulting K shape prototypes were subsequently selected as the exemplars of a nonhierarchical, "flat" pedestrian representation. The test set consisted of 4,070 and 9,770 rectangular image regions containing object and nonobject class, respectively. See Fig. 9. Both training and test sets were scaled such that the patterns were of the same size; this was done separately for sizes of 20, 50, 80, and 140 pixels.

To obtain detection performance at a particular object scale, we iterated over the samples in the test set (object and nonobject class separately) and histogrammed the minimum distance to the elements of the training set. From the two cumulative histograms, we derived the corresponding ROC curve by considering a particular distance threshold (x-axis of histogram) and identifying the fraction of object and nonobject samples which have lower distance values. The resulting ROC curves are shown in Fig. 10.

Several observations can be made from Fig. 10. At small object sizes (Fig. 10a), performance is comparatively low; the exemplar-based shape representation is not able to capture sufficient object detail to allow an effective discrimination between object and background. As the object size increases, performance increases (Figs. 10b and 10c) up to a point, after which, performance decreases again (Fig. 10d). Here, the number of exemplars is no longer sufficient to cover the possible shape variation.

A second observation is that, at large false alarm rates, there is little difference in the performance obtained with the various values of K. Evidently, the coverage and sensitivity obtained with few exemplars is similar to that reached by using many exemplars, given one uses a large (relaxed) distance threshold. This is the case of Fig. 7a. Given that one uses smaller distance thresholds, gaps in coverage of the pedestrian manifold start appearing, and

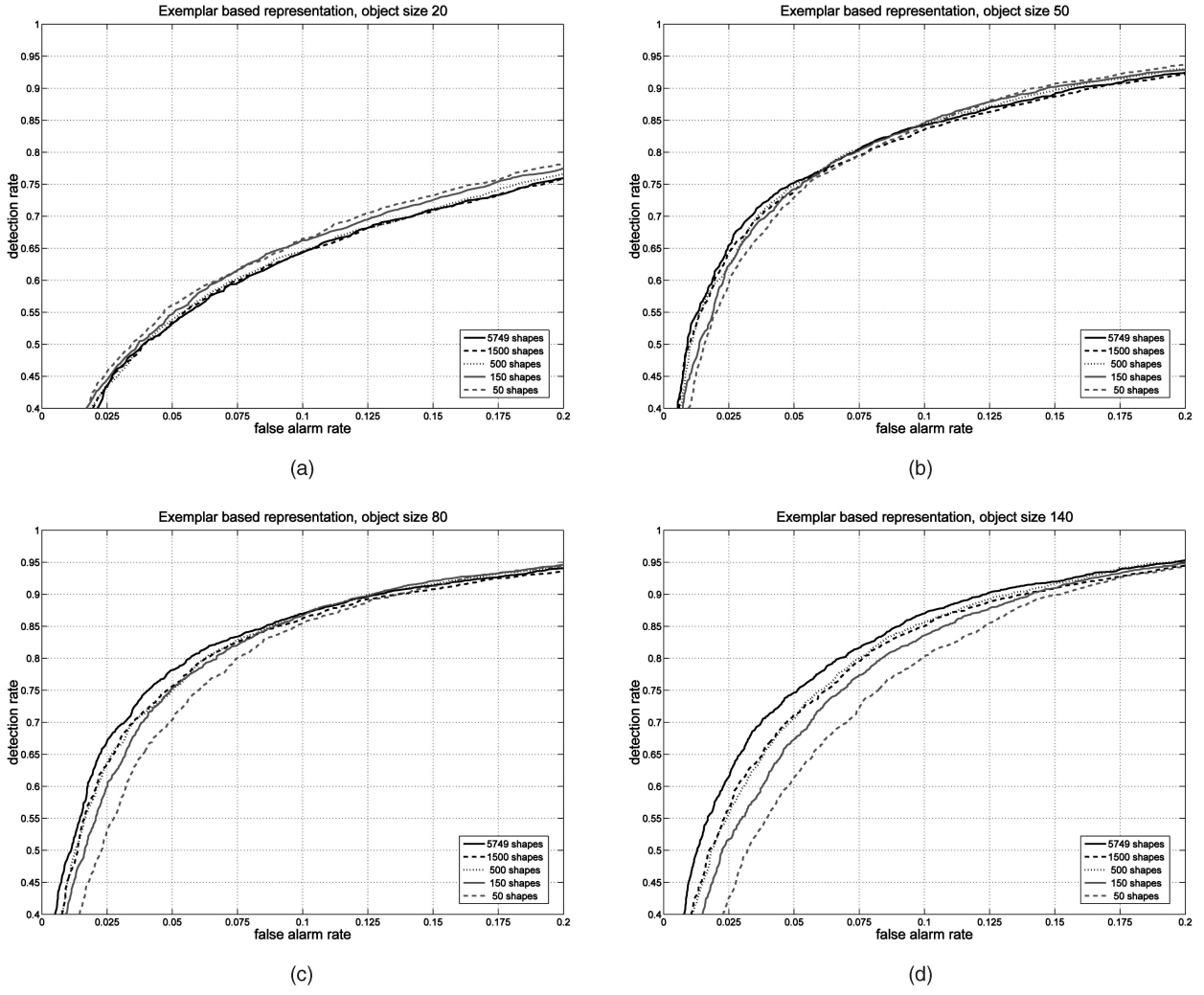


Fig. 10. ROC performance of nonhierarchical exemplar representation as a function of number of exemplars for different object sizes. (a) Size 20. (b) Size 50. (c) Size 80. (d) Size 140.

representations with larger numbers of exemplars start to outperform the ones with fewer exemplars. This is the case of Fig. 7b. Furthermore, the divergence in performance at lower false alarms increases for larger object scale (i.e., compare Figs. 10a and 10d).

6.2 Hierarchical Pedestrian Detection

The detection experiments involved a training set of 2,666 pedestrian instances and a test set of 2,254 pedestrian instances (1,306 images). The number of the templates in the original training set was doubled by mirroring the template shapes across the y-axis. On the resulting set, a four-level pedestrian tree was built, following Section 3.2.

The tree construction process was performed separately for each of the nine template scales (height range 36-84 pixels, increments of six pixels) that were used. At the leaf level of the scale-specific trees, all available shape exemplars were used from the training set, appropriately scaled. At a nonleaf level l, we select the number of template nodes to increase quadratically with template height h as

$$N_{l,h} = \left(\frac{h}{h_{min}}\right)^2 N_{l,h_{min}}, \tag{20}$$

where $N_{l,h_{min}}$ is the number of templates at level l for the smallest height h_{min} (36 pixels). We set

$$N_{3,h_{min}} = 100,\ N_{2,h_{min}} = \left\lceil \frac{1}{10}\ N_{3,h_{min}} \right\rceil,\ N_{1,h_{min}} = \left\lceil \frac{1}{10}\ N_{2,h_{min}} \right\rceil. \tag{21}$$

Fig. 11 illustrates the Simulated Annealing optimization approach corresponding to shape clustering at the leaf levels of these nine trees. The figure plots the objective function as a function of the iteration count. All plots show the same typical behavior: With an increasing number of iterations, both short-term variance and mean of the objective function decrease as the temperature parameter tends toward zero following the exponential annealing schedule.

In order to improve the compactness of the representation, the leaf level of the original tree was discarded, resulting in a three-level tree used for matching. Following the above choices for $N_{l,h}$, the new leaf levels of the scale-specific trees

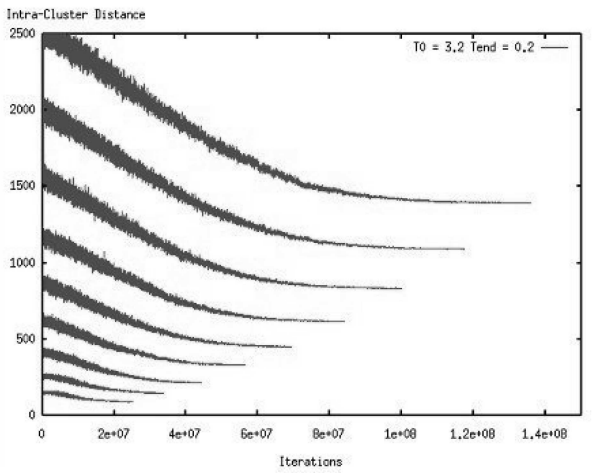


Fig. 11. Shape clustering by simulated annealing, objective function (average intracluster distance) as a function of iteration. Curves correspond to clustering shape exemplars of various scales.

contain between 100 and 544 exemplars, for object scales 36 to 84 pixels, respectively. This corresponds approximately to the gray ROCs (50-150 shapes) in Fig. 10 a and Fig. 10b and the dotted black ROC (500 shapes) in Fig. 10c. It thus indeed represents a reasonable trade-off between ROC performance and computational/memory cost, considering the experiments from last section.

The resulting scale-specific trees were subsequently merged into one overall template tree, which contained 27, 267, and 2,666 templates at the first, second, and leaf level, respectively. An increase in computational efficiency was obtained by subsampling the template points, based on the level of the corresponding node in the tree. We used a point sampling rate of 6, 3, 1 for the three levels from top to bottom, respectively. The spatial grid sizes on which templates were matched with the image were $\sigma = 9, 3, 1$ pixels, respectively (see Fig. 6).

Independently of the particular DT-based dissimilarity measure used, we found that having essentially only one edge segmentation threshold was not always appropriate. A restrictive value would result in sufficient edges to guide the search at the coarser level of the tree, but matching at the finer level would suffer. Setting the edge threshold to include all edges needed for a fine-level match would be computationally intensive and degrade the underlying coarse-to-fine concept. In the experiments, we set multiple edge thresholds and compute the associated distance images based on the level of the tree where matching was conducted.

With the representational structure and matching logic in place, we now turn our attention toward selecting the appropriate dissimilarity thresholds, following the probabilistic approach described in Section 5. The nine different template scales were aggregated to four scale intervals 36-48, 54-60, 66-72, 78-84 (index $s = 1, \ldots, 4$) for the purpose of computing the various distribution functions.

Fig. 12a shows the cumulative distribution function of the distance values at the top level of the tree for the pedestrian and nonpedestrian class. Recall that, for the object class, distance distributions were aggregated by object scale (12), whereas, for the nonobject class, separate distributions were maintained for each node (14). The four curves associated with $F_O^{1,s}(d_1)$ $(s = 1, \ldots, 4)$ are those in Fig. 12a which have the strongest slope upwards. The other 27 curves represent $F_N^{1,t,s}(d_1)$ for the nodes at the top level. The curves in Fig. 12 are furthermore gray-coded; those corresponding to smallest and largest object scale ($s = 1$ and $s = 4$) are shown in light and dark gray, respectively; the intermediate object scales ($s = 2$ and $s = 3$) are plotted in black.

Fig. 12b shows the computed posterior $p(O|d_1)$ at the top level. We would like to visually verify that the proposed probabilistic model indeed captures a measure of object saliency. We indicated in the figure two objects at the same object scale, which correspond to the maximum and minimum a posteriori probability for a given distance value. One observes that the most salient object is one which involves a pedestrian with feet apart, whereas the least salient is one which has the feet closed. Indeed, with

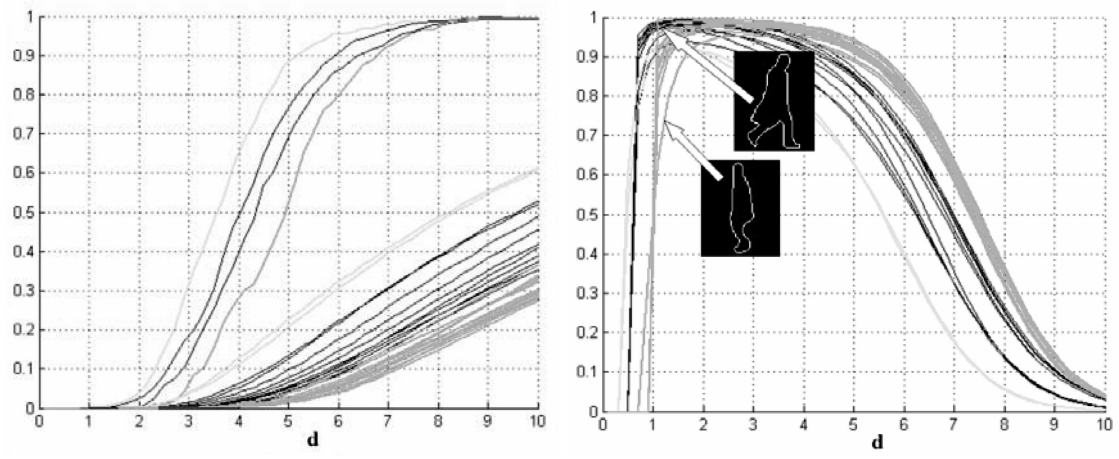


Fig. 12. At the top level of the template tree. (a) Cumulative distributions $F_O^{1,s}(d_1)$ and $F_N^{1,t,s}(d_1)$. (b) Posterior p(O | d_1) (scaling factor p(O) set to 1), for scales $s = 1, \ldots, 4$ and nodes $n = 1, \ldots, 27$. Plots corresponding to $s = 1$ and $s = 4$ are shown in light and dark gray, respectively.

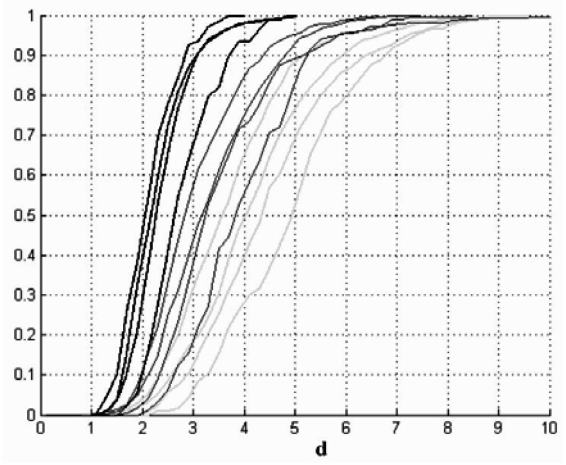


Fig. 13. Distributions $F_O^{l,s}(d_l)$ for level $l = 1$ (light gray), $l = 2$ (dark gray), and $l = 3$ (black), each plotted for object scale $s = 1, \ldots, 4$.

quite a few diagonally oriented edges, the former pattern arises less likely by accident in the data set depicting urban traffic scenes than the pattern which essentially consists of two major vertical lines. The latter is more likely to match upon man-made structures in the image.

Fig. 12b furthermore nicely illustrates the need for a nontrivial adjustment of the distance thresholds with increasing object scale (see previous discussion in Section 4). Consider the light gray plots at a posterior value of 0.6, for example. The associated distance threshold is about 5. A linear scaling of the distance threshold to obtain an object scale comparatively to the dark gray plots would result in a distance threshold of roughly 5×2 (factor 2 because the dark gray plot stands for an average template height of 81 pixels while the light gray plot stands for an average template height of 42 pixels). But, as can be seen from Fig. 12b, this setting would result in an a posterior value below 0.1; thus, it is significantly lower than the 0.6 value obtained at the smaller object scale.

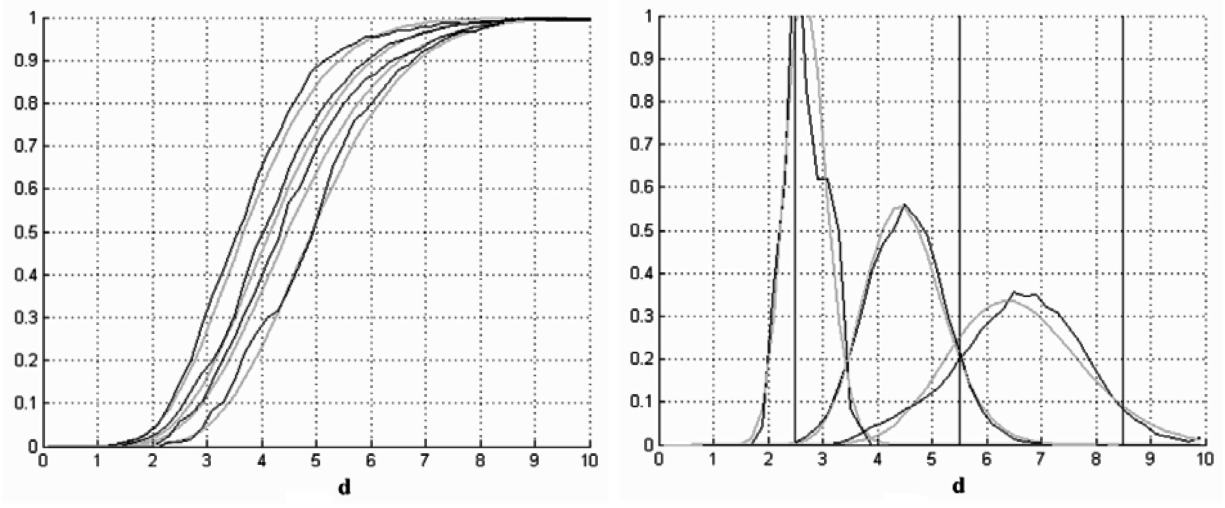
Fig. 12b also illustrates what may appear to be a counterintuitive result, namely, that for decreasing distances starting from $d_l = 1$, the posterior decreases back to zero. This might be considered an aberration given the scarcity of data in that range, leading to a manual specification of the posterior as 1 in that range. However, a plausible explanation for this result is that, if a template matches very well ($d_l < 1$), this is much more likely to be the effect of strong edge clutter in the background than of a very good matching template. In the experiments, we choose the latter interpretation, not reverting to some ad hoc logic.

Fig. 13 shows the cumulative distributions for the object class at various scales and levels of the tree. It captures the decrease of the distance values along the path from the top of the tree toward a correct solution on the leaf level.

Fig. 14 illustrates the application of parametric models using the gamma distribution, as discussed in Section 5.2. Fig. 14a involves various approximations of $F_O^{1,s}(d_1)$ at the top level. Fig. 14b shows a typical result for $f_{NN}^{3,t,s}(d_3|d_2)$ at a leaf level node.

Fig. 15 illustrates some final detection results. Considering the difficulty of the problem at hand, performance is quite favorable, with correct detections in a wide range of scenes. The system is far from flawless, however, with its main shortcomings being the production of false positives in heavy textured image regions (e.g., see fourth row, first and second columns) and nondetections in image areas of low contrast and occlusion (e.g., see fourth row, third and fourth columns). The last row shows detection results for a single image sequence.

We compared the performance of probabilistic hierarchical shape detection, where per-level thresholds involved the a posteriori criterion, to an earlier, nonprobabilistic version [12] where the per-level thresholds involved distance values, properly tuned. Detections were considered correct if the four corners of the bounding boxes associated with the found shape template were all within 20 pixels of the manually labeled location. The outcome of this comparison is summarized in Table 1. As can be seen, at approximately equal detection and false positive rates,



(a) (b)

Fig. 14. Distributions and parametric fits. (a) $F_O^{1,s}(d_1)$ for object scale $s = 1, \ldots, 4$ shown in black, gamma fit in gray. (b) $f_{NN}^{3,t,s}(d_3|d_2)$ for particular t and s and three values of d_2 (vertical lines) shown in black, gamma fit in gray.

Fig. 15. Hierarchical probabilistic shape-based pedestrian detection.

TABLE 1
Detection Performance and Computational Cost of State-of-the-Art Hierarchical Shape Detector versus Proposed Probabilistic Extension

Nr Pedestrians = 2254 Nr Images = 1306	distance thresholds [12]	probability thresholds (a-posteriori criterion)
Correct Detections	1789 (79%)	1799 (80%)
False Positives	5096	5036
Pixel Correlations (10^9)	44	16

the proposed approach manages to reduce computational cost (determined by the number of pixel correlations) by a significant factor. The hierarchical shape detector runs image at 7-15 Hz on a 2.4 GHz Pentium IV processor.

7 Discussion

The previous section has, among other things, shown that a surprising large variation in object shape can be captured by a discrete set of shape exemplars when represented in a hierarchical fashion. This beneficial effect has its limits, undoubtedly. As was seen in Section 6.1, a strongly increasing number of exemplars are needed to maintain a certain ROC performance as the pedestrian comes closer, up to the point where the approach is not practical anymore due to storage and processing requirements.

One possible solution is to utilize a hybrid discrete-continuous shape representation. This could involve matching first with the discrete hierarchical exemplar representation and, at the leaf level, switching to a more compact continuous representation, such as the linear subspace shape model (PDM) employed by Cootes et al. [6]. The premise of obtaining a sound PDM model, namely, very similar training

Fig. 16. Pedestrian classification—detections shown in white, solutions classified as pedestrians marked by STOP sign.

shapes to allow successful automatic shape registration, would be met at the leaf node of the tree.

Another solution for counteracting the unfavorable complexity of exemplar-based approaches is the use of component-based approaches (e.g., [19], [21]). In our case, separate hierarchical representations could be built for object parts, and the detection results be merged, taking into account spatial relationships.

So far, we considered the hierarchical shape detector in isolation. In a typical application, the shape detector is combined with other modules for additional robustness and efficiency. A particular worthwhile combination is the use of the shape-based detector and a texture-based pattern classifier for object recognition. Pattern classifiers [17] that work on pixel values (or derived filter coefficients) tend to be sensitive to spatial misalignment of a ROI. Applying them exhaustively over the image, is on the other hand, typically not an option due to large computational cost. The idea is to use a shape detector to efficiently localize candidate object instances, which are subsequently verified with a more powerful pattern classifier, based on richer texture cues. We in fact employ this combined approach for the applications depicted in Fig. 1, e.g., see Fig. 16. In the pedestrian application, the use of the proposed shape detector furthermore has the advantage that it can index onto a set of specialized (body pose-specific) texture classifiers. The resulting mixture-of-experts classifier scheme manages to reduce the false positives by an order of magnitude, without appreciably reducing the correct detection rate [11].

Another worthwhile possibility is to precede the shape detector with an additional attention focusing mechanism. For example, in the pedestrian system by Gavrila and Munder [11], stereo vision is used to quickly identify obstacle regions in front of a vehicle before initiating shape-based pedestrian detection. The use of the additional depth cue furthermore manages to reduce the number of false detections by a further order of magnitude. The performance shown in Table 1 is thus in practice significantly enhanced by the use of preceeding/following modules based on complementary visual cues. A comparison of state-of-the-art pedestrian systems [7], [11], [19], [21], using the same data set and performance metrics, is worthwhile for future work.

8 Conclusions

This paper presented a novel probabilistic hierarchical approach for shape-based object detection. A Bayesian model was developed to estimate the a posteriori probability of the object class at the various node of a tree structure, built automatically from examples. The model took into account several object characteristics such as scale and saliency.

In the context of pedestrian detection, this paper provided an experimental answer to the question of how many pedestrian exemplars one needs to obtain a certain detection performance and how this depends on object scale. The paper furthermore demonstrated the appeal of utilizing the a posteriori probability criterion at each tree node in order to directly control the efficiency of hierarchical shape matching. It showed a significant speed-up versus a nonprobabilistic matching variant, where dissimilarity thresholds were manually tuned, one per tree level.

Appendix

For the object class,

$$\begin{aligned}
p(d_{1:l}|O_l) &= p(d_{1:l}|O_lO_{l-1})\; p(O_{l-1}|O_l) \\
&\quad + p(d_{1:l}|O_lN_{l-1})\; p(N_{l-1}|O_l) \\
&= p(d_{1:l}|O_lO_{l-1}) = p(d_{1:l-1}|O_{l-1})\; p(d_l|d_{l-1}O_lO_{l-1}) \\
&= p(O_{l-1}|d_{1:l-1})\; \frac{p(d_{1:l-1})}{p(O_{l-1})}\; p(d_l|d_{l-1}O_lO_{l-1}) \\
&= p(O_{l-1}|d_{1:l-1})\; \frac{p(d_{1:l-1})}{p(O_l)}\; p(d_l|d_{l-1}O_lO_{l-1})\, p(O_l|O_{l-1})
\end{aligned} \tag{22}$$

given $p(O_{l-1}|O_l)=1$, $p(N_{l-1}|O_l)=0$, and

$$p(O_l)/p(O_{l-1}) = p(O_l|O_{l-1}).$$

For the nonobject class,

$$\begin{aligned}
p(d_{1:l}|N_l) &= p(d_{1:l}|N_lO_{l-1})\; p(O_{l-1}|N_l) \\
&\quad + p(d_{1:l}|N_lN_{l-1})\; p(N_{l-1}|N_l) \\
&= p(d_{1:l-1}|O_{l-1})\; p(d_l|d_{l-1}N_lO_{l-1})\; p(O_{l-1}|N_l) \\
&\quad + p(d_{1:l-1}|N_{l-1})\; p(d_l|d_{l-1}N_lN_{l-1})\; p(N_{l-1}|N_l) \\
&= p(O_{l-1}|d_{1:l-1})\; \frac{p(d_{1:l-1})}{p(O_{l-1})}\; p(d_l|d_{l-1}N_lO_{l-1}) \\
&\quad p(N_l|O_{l-1}) \frac{p(O_{l-1})}{p(N_l)} + p(N_{l-1}|d_{1:l-1})\; \frac{p(d_{1:l-1})}{p(N_{l-1})} \\
&\quad p(d_l|d_{l-1}N_lN_{l-1})\; p(N_l|N_{l-1}) \frac{p(N_{l-1})}{p(N_l)} \\
&= p(O_{l-1}|d_{1:l-1}) \frac{p(d_{1:l-1})}{p(N_l)} p(d_l|d_{l-1}N_lO_{l-1})\; p(N_l|O_{l-1}) \\
&\quad + p(N_{l-1}|d_{1:l-1})\; \frac{p(d_{1:l-1})}{p(N_l)}\; p(d_l|d_{l-1}N_lN_{l-1})
\end{aligned} \tag{23}$$

given $p(N_l|N_{l-1}) = 1$.

Substituting (22) and (23) in (9), we obtain the recursive form of the Bayes rule of (11).

Acknowledgments

The author would like to thank M. Hofmann for his assistance at the experiments of Section 6.1. He also appreciates the many interesting discussions with S. Munder.

REFERENCES

[1] Y. Amit, D. Geman, and X. Fan, "A Coarse-to-Fine Strategy for Multiclass Shape Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 12, pp. 1606-1621, Dec. 2004.

[2] H. Barrow et al., "Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 659-663, 1977.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 509-522, May 2002.

[4] G. Borgefors, "Distance Transformations in Digital Images," *J. Computer Graphics, Vision, Image Processing,* vol. 34, no. 3, pp. 344-371, June 1986.

[5] G. Borgefors, "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, no. 6, pp. 849-865, Nov. 1988.

[6] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active Shape Models—Their Training and Applications," *Computer Vision and Image Understanding,* vol. 61, no. 1, pp. 38-59, 1995.

[7] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. pp. 886-893, 2005.

[8] N. Duta, A.K. Jain, and M.-P. Dubuisson-Jolly, "Automatic Construction of 2D Shape Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 5, pp. 433-446, May 2001.

[9] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral Grouping Using the Nystrom Method," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 2, pp. 214-225, Feb. 2004.

[10] D.M. Gavrila, J. Giebel, and H. Neumann, "Learning Shape Models from Examples," *Proc. German Assoc. Pattern Recognition Conf.,* pp. 369-376, 2001.

[11] D.M. Gavrila and S. Munder, "Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle," *Int'l J. Computer Vision,* vol. 73, no. 1, pp.41-59, June 2007.

[12] D.M. Gavrila and V. Philomin, "Real-Time Object Detection for 'Smart' Vehicles," *Proc. Int'l Conf. Computer Vision,* pp. 87-93, 1999.

[13] Y. Gdalyahu and D. Weinshall, "Flexible Syntatic Matching of Curves and Its Application to Automatic Hierarchical Classification of Silhouettes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 12, pp. 1312-1328, Dec. 1999.

[14] C. Goodall, "Procrustes Methods in the Statistical Analysis of Shape," *J. Royal Statistical Soc. B,* vol. 53, no. 2, pp. 285-339, 1991.

[15] T. Heap and D. Hogg, "Improving the Specificity in PDMs Using a Hierarchical Approach," *Proc. British Machine Vision Conf.,* 1997.

[16] D. Huttenlocher, G. Klanderman, and W.J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 850-863, Sept. 1993.

[17] A. Jain, R. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-37, Jan. 2000.

[18] S. Kirkpatrick Jr., C.D. Gelatt, and M.P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, pp. 671-680, 1993.

[19] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian Detection in Crowded Scenes," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 878-885, 2005.

[20] MathWorks Matlab, Function *ksdensity*, 2005.

[21] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-Based Object Detection in Images by Components," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 4, pp. 349-361, Apr. 2001.

[22] C.F. Olson, "A Probabilistic Formulation for Hausdorff Matching," *Proc. Conf. Computer Vision and Pattern Recognition,* 1998.

[23] C.F. Olson and D.P. Huttenlocher, "Automatic Target Recognition by Matching Oriented Edge Pixels," *IEEE Trans. Image Processing,* vol. 6, no. 1, pp. 103-113, Jan. 1997.

[24] D.W. Paglieroni, G.E. Ford, and E.M. Tsujimoto, "The Position-Orientation Masking Approach to Parametric Search for Template Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 7, pp. 740-747, July 1994.

[25] W. Rucklidge, "Locating Objects Using the Hausdorff Distance," *Proc. Int'l Conf. Computer Vision,* pp. 457-464, 1995.

[26] A. Srivastava, S.H. Joshi, W. Mio, and X. Liu, "Statistical Shape Analysis: Clustering, Learning, and Testing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 4, pp. 590-602, Apr. 2005.

[27] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla, "Model-Based Hand Tracking Using a Hierarchical Bayesian Filter," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 9, pp. pp.1372-1385, Sept. 2006.

[28] K. Toyama and A. Blake, "Probabilistic Tracking with Exemplars in a Metric Space," *Int'l J. Computer Vision,* vol. 48, no. 1, pp. 9-19, 2002.

[29] M. Yang and K. Wu, "A Similarity-Based Robust Clustering Method," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 4, pp. 434-448, Apr. 2004.

[30] D. Zhang and G. Lu, "Review of Shape Representation and Description Techniques," *Pattern Recognition,* vol. 37, pp. 1-19, 2004.

Dariu M. Gavrila received the MSc degree in computer science from the Free University in Amsterdam in 1990. He received the PhD degree in computer science from the University of Maryland at College Park in 1996. He was a visiting researcher at the MIT Media Laboratory in 1996. Since 1997, he has been a senior research scientist at DaimlerChrysler Research in Ulm, Germany. In 2003, he was named professor in the Faculty of Science at the University of Amsterdam, chairing the area of Intelligent Perception Systems (part time). Over the last decade, Professor Gavrila has specialized in visual systems for detecting human presence and recognizing activity, with application to intelligent vehicles and surveillance. He has published more than 20 papers in this area in leading vision conferences and journals. His personal Web site is www.gavrila.net.