Learning to Parse Pictures of People.

Paper no. 602 - Submission to ECCV 2002

No Institute Given

Abstract. The detection of people is one of the foremost problems for indexing, browsing and retrieval of video. The main difficulty is the large appearance variations caused by action, clothing, illumination, viewpoint and scale. Our work aims to find people in static video frames using learned models of the appearance of human body parts (head, limbs, hands) and the geometry of their assemblies. It is based on Forsyth & Fleck's general methodology of 'body plans' and Felzenszwalb & Huttenlocher's dynamic programming approach for efficiently assembling candidate parts into 'pictorial structures'. The simplistic part detectors used in these works made restrictive photometric assumptions that severely limited their practical applicability. Instead, we learn dedicated detectors for each body part using Support Vector Machines (SVMs) and the recently proposed Relevance Vector Machines (RVMs). In the past, SVMs have been successfully used to detect whole pedestrians in street scenes, allowing for a wide range of illumination, pose and clothing variations, and also combinations of rigidly positioned subimages (typically, the upper body, arms, and legs), but we are not aware of any previous work using SVMs to learn articulated body plans. RVMs are SVM-like classifiers that offer a well-founded probabilistic interpretation and sparse feature selection for reduced computation. We demonstrate the benefits of using RVMs experimentally in a series of results showing great promise for learning detectors in more general situations.

Keywords: object recognition, image and video indexing, grouping and segmentation, statistical pattern recognition, kernel methods.

1 Introduction

The detection of people in images is an important practical issue for content-based indexing of images and video. It is a difficult problem owing to the wide range of appearances that people can have. For instance, actors in typical Hollywood movies are shown in a great variety of activities, scales, viewpoints and lightings. We can not rely on frequently made simplifying assumptions such as non-occlusion, perfect background subtraction, *etc*: methods are clearly required for detecting people in general everyday situations.

To address this issue, Forsyth & Fleck introduced *body plans* [2] as a general methodology for finding people in images. However, they relied on simple body part detectors based on generalized cylinders. This is problematic, especially in the case of loose clothing. Felzenszwalb & Huttenlocher [1] showed how dynamic programming could be used to make the grouping of body plans (cast as pictorial structures) efficient, but relied on simplistic colour-based part detectors. Both of these works make

strong photometric assumptions about the body parts. We retain their ideas for composing parts into assemblies by building tree-structured models of people, but propose a more general approach to learning the body part detectors and the underlying geometric model, based on Support Vector Machines (SVM) and the related Relevance Vector Machines (RVM) [16,17]. In the past, SVM classifiers have been learned for humans as a whole [12] and also for rigidly connected assemblies of subimages (typically, upper body, arms, andlegs) [10], but not for flexibly articulated body models.

We present a series of experiments showing the promise of learning the articulated structure of people from training examples of body parts, using Support Vector or Relevance Vector Machines. Our contribution is three-fold. Firstly, we present a method for training the body part detectors from as few as 2 examples, selected from a pool of approximately 100 hand-labeled images. Secondly, we sketch a method for learning a body joint model, building on the recently proposed Adaptive Combination of Classifiers (ACC) model. Thirdly, we describe an efficient decoder for the learned models, schematically described as template matching + dynamic programming. Our initial experiments demonstrate that body part detectors learned with only 100 images from the MIT database can produce reliable part detections with as few as 4 false detections per true one, and that these can be robustly and efficiently assembled into correct body plans in 70% of cases. This is remarkable as even humans often find it difficult to classify isolated part subimages correctly.

The paper is structured as follows. We introduce our body plan model in §2, then discuss body part detectors learned by two competing algorithms, SVM and RVM, in §3. §4 presents our approach for learning and decoding body plans. Finally, §5 presents some results and discuss future work.

2 The Pictorial Structure of People

In the work of Marr & Nishihara [9] and many others [4,13], pictures of people are described geometrically as hierarchical assemblies of generalized cylinders and components. These early works represent the position of a part C relative to its parent P using *adjunct relationships* parameterized by C's position (p, r, θ) and angular orientation (ψ, ϕ, χ) in P's cylindrical coordinate system. Each joint is thus represented as a 6-vector $(p, r, \theta, \psi, \phi, \chi)$ with discrete toleranced values for each parameter. They note that perspective projection makes many parameters of the 3D model unobservable, and that the local image signature of a joint is a pair of axes, but emphasize the 3D structure, and attempt to recover that.

Recovering articulated 3D models from single images is a hard problem. Felzenswalb & Huttenlocher recently reconsidered Fischler & Elschlager's notion of *pictorial structure* and demonstrated its usefulness for detecting people in indoor scenes [1]. Pictorial structures are collections of image parts arranged in deformable configurations. They are directly adapted to monocular observations. Similarly, it was recently argued [11] that the singularities that characterize 3D tracking can be removed by using image based 'scaled prismatic models' — in essence, pictorial structure models. Other 2D part-based models use image edges [18] or motion models derived from dense optical flow [8] as features for detection and/or tracking.

Following this line of research, we represent people using a 2D articulated appearance model composed of 15 part-aligned image rectangles surrounding the projections of body parts: the complete body, the head, the torso, and the left and right upper arms, forearms, hands, thighs, calves and feet, numbered from 1 to 15 as in fig.1. Each body part P_i is a rectangle parameterized in image coordinates by its center $[x_i, y_i]$, its length or size s_i and its orientation θ_i. A coarse resolution whole-body image is included in case 'the whole is greater than the sum of the parts'. During training and detection, we discretize the admissible range of sizes and orientations. As discussed later, we use a range of 8 scales between 12 and 96 pixels, and 36 orientations equally spaced every 10 degrees. There are also 14 body joints between the parts - the plexus between body and torso, the neck between head and torso, the hips between torso and thighs, the knees between thighs and calves, the ankles between calves and feet, the shoulders between torso and upper arms, the elbows between upper arms and forearms and the wrists between forearms and hands. Fig.1 shows the body model in average position, using a single aspect ratio of 16/9 for all body parts.

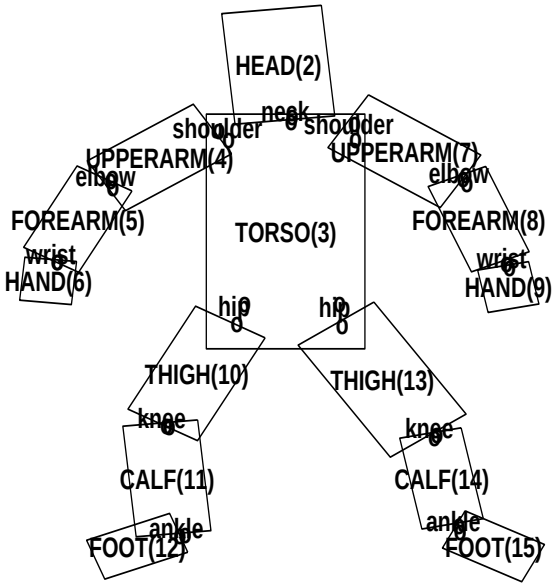

Fig. 1. Our articulated body model with its 14 joints and 15 body parts.

Expressed in terms of the probabilistic formulation of pictorial structure, the posterior likelihood of there being a body with parts P_i at image locations l_i ($i \in \{1...15\}$) is the product of the *data likelihoods for the 15 parts* (*i.e.* the classification probabilities for the observed subimages at the given part locations to be images of the required parts) times the product of the *prior likelihoods for the 14 'joints'* (*i.e.* for a coherent body to generate an image with the given relative geometric positionings between each part and its parent in the body tree). The negative log likelihood for the total body as-

sembly $A = \{l_1, \ldots, l_{15}\}$ can thus be written as follows, where E is the list of body joints ('edges' of the body tree):

$$L(A) \;=\; -\sum_i \log p_i(l_i) - \sum_{(ij)\in E} d_{ij}(l_i, l_j)$$

Felzenswalb & Huttenlocher model body parts as constant color regions of known shapes and body joints as rotational joints. In this paper, we machine-learn the 29 distributions $p_i(l_i)$ and $d_{ij}(l_i, l_j)$ from sets of positive and negative examples. We model the part and articulation likelihoods using linear Support Vector or Relevance Vector Machines. Our work can be viewed as an extension of Mohan's recent work on *combined classifiers* [10], where 'component' classifiers are trained separately for the limbs, torso and head based on image pixel values, and 'combination' classifiers are trained for the assemblies based on the scores of the component classifiers in fixed image regions. However, we learn part-aligned, rather than image-aligned, classifiers for each body part, and we extend the 'combination' classifier to include deformable, articulated structures rather than rigid assemblies.

3 Detecting Body Parts

In our model, learning each body part amounts to estimating its probability given the observed image distribution at its location. Detecting and labelling body parts is a central problem in all component-based approaches. Clearly the image must be scanned at all relevant locations and scales, but there is a question of how to handle different part orientations, especially for small, mobile highly articulated parts such as arms and hands. One can either work in the image frame, trying to build a general detector that is capable of finding the part whatever its orientation, or work in a part-aligned frame, building a detector that works for just one orientation and scanning this over all relevant orientations. The part-aligned approach has the potential to produce simpler detectors from less (but better labelled) training data, and the advantage (in the articulated body-plan framework) that it also recovers the part orientation. Which approach is faster or better must depend on the relative complexity and reliability of all-orientation and one-orientation detectors, but in general it is difficult to build good transformation invariance into general-purpose detectors. The image-frame approach is well-adapted to pedestrian detection applications such as Mohan's [10], where one wants a relatively coarse whole-person detector for distant people with similar poses (mainly standing or walking). But our ultimate goal is to detect people labelled with detailed part locations in applications where the person may be in any pose and partly occluded, and for this we believe that the part-based body plan approach is preferable.

Our detector works with a generalized feature pyramid spanning 8 scales and 36 orientations $0°, \ldots, 350°$. During training, the articular structure of each training image is clicked, and for each designated part a 14×24 subimage aligned with its axes and scaled to its size is extracted as shown in fig.2. We learn 15 Support Vector or Relevance Vector Machines for the 14 individual parts and the whole body, and during detection run each of them over the scale-orientation-position feature pyramid, then assemble the results as discussed in the next section.

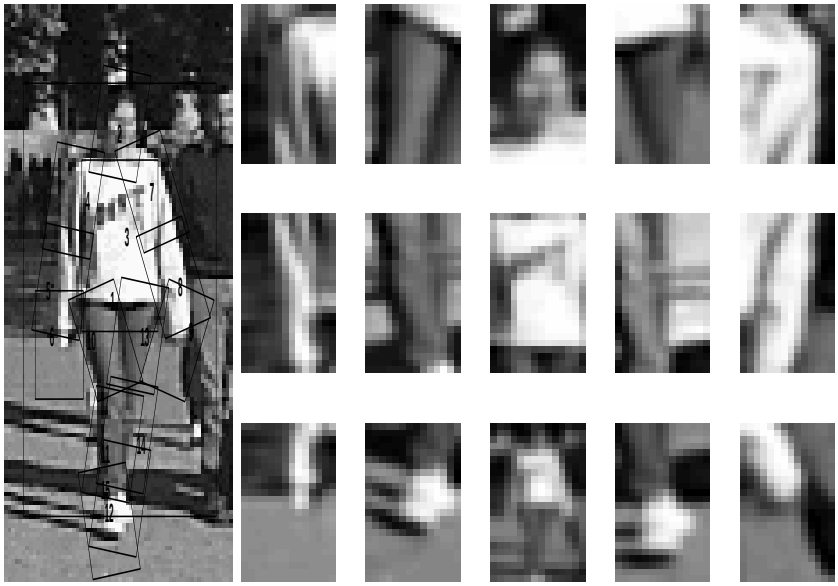

Fig. 2. A hand-labelled training image from the MIT database and its extracted body part subimages. Reading vertically from left to right: left upper arm, forearm, hand; left thigh, calf and foot; head, torso and whole body; right thigh, calf, foot; right upper arm, forearm and hand.

3.1 Feature Selection

The problem of feature selection for object recognition has received a lot of interest in recent years and numerous feature sets have been suggested, including image pixel values, wavelet coefficients and Gaussian derivatives. Wavelets are currently popular, but as a general representation for human body parts it is unclear whether standard (rectangular) or non-standard (square) wavelet constructions are most suitable. Heisele *et al* recently obtained better results for their SVM face detector using gray levels rather than Haar wavelets[3]. Some authors also feel that wavelets are unsuitable as a general image representation because they represent point events rather than line or curve ones. As an alternative, they propose ridgelets and curvelets, which could prove useful for detecting human limbs.

Here we leave such issues for future work and use a more traditional feature set consisting of the Gaussian filtered (rescaled and reoriented) image and its first and second derivatives. In detail, our feature vector for an image rectangle at location-scale-orientation $[x_i, y_i, s_i, \theta_i]$ contains the absolute values of the responses of the six Gaussian $\sigma = 1$ filters $\{G, \nabla_x G, \nabla_y G, \nabla_{xx} G, \nabla_{xy} G, \nabla_{yy} G\}$ in the rectangle's 14×24 window (*i.e.* 14×24 after rescaling by s and rotation by θ). There are thus 2016 features per location. For Y-Cr-Cb color images we use only the luminance values Y. These features seem appropriate for representing the appearance of body part detail over a range of scales and orientations. The absolute values of the filter responses are normalized to a scle of $[0 - 255]$ for each image. Note that we do not require the extracted features

to be scale- or orientation-invariant. On the contrary, we seek features that are tuned to the characteristic scales and orientations of the detail in the aligned body-part images. Some examples of the feature vectors are shown in fig.3

To implement this, the Gaussian filters are computed using 9 rotated images from 0 to 80 degrees and 7 standard deviations $\sigma = \frac{8}{(9-s)} = \{1, 1.14, 1.33, 1.6, 2, 4, 8\}$. For each standard deviation we also downsample by σ, so the standard deviation of the filters in their resampled 14×24 windows is always 1. For any given size and orientation, we select the feature vector that best approximates the part-aligned region as an axis-aligned rectangle of height 24, thus quantizing the possible sizes to $\frac{192}{(9-s)} = \{24, 27, 32, 38, 48, 64, 96, 192\}$. This restriction is important for the detection process, where it is necessary to sample all possible locations, orientations and scales. This choice of primitives makes reasonably few assumptions about the nature of the features to be learned, which can be arbitrary combinations of shape, luminance and texture.

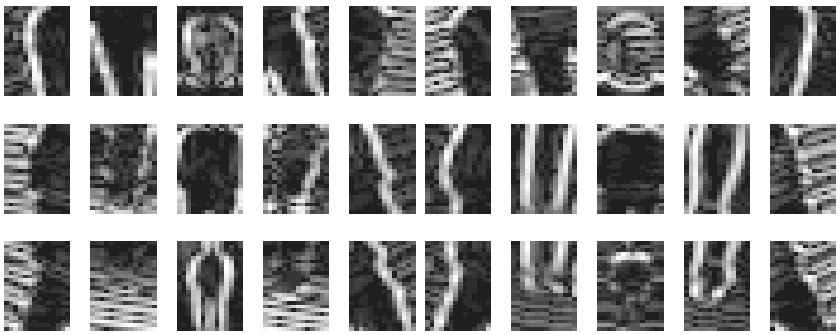

Fig. 3. The $\nabla_x G$ and $\nabla_y G$ feature images for the example in fig.2.

3.2 Training

Using the 2016-dimensional feature vectors for all body part in the training set, we trained two linear classifiers for each part, one using a Support Vector Machine and the other using a Relevance Vector Machine. SVMs and RVMs are grounded on results from statistical learning theory, which suggest that they should give good classification performance even when there are relatively few training examples. Here we decided to put that claim to a severe test by training on the minimum number of examples that give reasonable results — in our case, about 100.

In both cases, we trained all 15 part classifiers separately against a common 'background' model. As background examples we simply take other pieces of the training images that do not contain people. Note that we do not attempt to learn multi-class classifiers for every part against every other one. We are trying to learn *reliable filters* for rejecting non-parts within an articulated body plan framework, not isolated part detectors or part-type classifiers *per se*. We expect the overlap in appearance between

different parts to be significant, and we do not want this to cause missed detections in ambiguous cases.

Support Vector Machines: SVMs are discriminant classifiers that give a yes/no decision, not a probability. However in our experiments we treat the SVM scores (scalar products in feature space) as if they were log likelihoods for the body parts given the image values[1]

Relevance Vector Machines: RVMs [16,17]are Bayesian kernel methods that choose sparse basis sets using an 'automatic relevance determination' style prior that pushes non-essential weights to zero. The functional form of the final classifier is the same as that of an SVM - only the fitted weights are different. RVMs do not usually give significantly better error rates than the corresponding SVMs, but they often give similar results with many fewer kernels. Here we use logistic linear discriminant RVMs, whose output directly models the log-odds for a part versus a non-part at the given point. In this paper, we used the RVM for reducing the number of examples or 'relevance vectors'. In practice, our classifiers use between 2 and 3 positive and negative 'relevance vectors' each (to be compared with 100-200 support vectors used by the SVM classifiers).

In the linear case, the sparsity of the RVM basis translates into sparse use of the input features, which potentially means that less features will have to be extracted from the images and hence that the method will be faster to run. This is a very exciting area which we plan to investigate further in future work.

3.3 Detection

We detect all the body parts at once, in a single scan of the orientation-scale pyramid. The detection scores for each part reduce to a simple convolution product against a mask containing the discriminant sum of weighted support vectors or relevance vectors. Conceptually, this is nothing more than template matching, where each of the support or relevance vectors is used as a template, and the resulting score is a weighted sum of the correlation with the search region. Based on our expectation of the false detection rates of our detectors, we retain the best 50 candidates for each part. Ultimately, we would like to use a fixed threshold for selecting candidates, but this was not the case in the experiements reported here.

4 Parsing the body tree

In a non-articulated, image-aligned method such as that of Mohan [10], assembling the part detections is relatively straightforward - decompose the search window into subwindows, keep the highest score for the appropriate part in each subwindow, and compose the scores into a single, low-dimensional feature vector. A linear SVM can be learned for the overall body detection, given this second-stage feature vector.

[1] For a more principled approach, the scores of any discriminant can be converted to probabilities as follows: run the detector over a validation set and fit density models to its positive-example and negative-example output scores. At any given score, the ratio of the positive-example density to the negative-example one is an estimate of the positive-to-negative odds ratio for detections at that score.

In our articulated, part-aligned method, the composition of part-models is only slightly more difficult, and can be cast as a combinatorial search - from all detected parts, search for the assemblies looking most like people. Since assemblies are naturally described as trees, efficient dynamic programming algorithms can be used to build the second-stage classifier, as we now describe.

4.1 Parsing/decoding algorithm

Given N candidate body part locations $l_{k,n}$ detected by each body part classifier C_k, we are looking for a 'parse' of the scene into one or more 'body trees'. One important subproblem is to assign a 'valid detection' or 'false alarm' label to each candidate, based not only on the candidate's scores, but on the local configuration between the candidates and its neighbors. Our approach relies on algorithms previously described by Ioffe and Forsyth as 'Viterbi decoding' [7]and by Felzensbaum and Huttenlocher as dynamic programing [1], so we only briefly sketch the solution. Given the detection scores $D_k(l_{kn})$ for all candidates $n = 1...N$, we search for the best candidate as a function of its direct parent, in the hypothesis of a given body plan. For the leaves of the body tree (hans, feet and head), this is computed by algorithm 1.

Algorithm 1 leaf location

$B_k(l_{jm}) = \min_{\{n=1...N\}} -D_k(l_{kn}) + d_{kj}(l_{kn}, l_{jm})$
$l_k^*(l_{jm}) = \arg\min_{\{n=1...N\}} -D_k(l_{kn}) + d_{kj}(l_{kn}, l_{jm})$

Based on this computation, we can score candidates from the bottom up, using the recursion formula2.

Algorithm 2 bottom up

$B_k(l_{jm}) = \min_{\{n=1...N\}} -D_k(l_{kn}) + d_{kj}(l_{kn}, l_{jm}) + \Sigma_{\{c|k=pa(c)\}} B_c(l_{kn})$
$l_k^*(l_{jm}) = \arg\min_{\{n=1...N\}} -D_k(l_{kn}) + d_{kj}(l_{kn}, l_{jm}) + \Sigma_{\{c|k=pa(c)\}} B_c(l_{kn})$

At the root node (sentence level) we obtain the simple formula 3for scoring the high level hypotheses.

Algorithm 3 root location

$B_r = \min_{\{n=1...N\}} -D_r(l_{rn}) + \Sigma_{\{c|r=pa(c)\}} B_c(l_{rn})$
$L_r^* = \arg\min_{\{n=1...N\}} -D_r(l_{rn}) + \Sigma_{\{c|r=pa(c)\}} B_c(l_{rn})$

Choosing the most probable root node, we are now in a position to move back to the other nodes in a top down fashion by choosing $L_k^* = l_k^*(L_{pa(k)})$ for all nodes, given their parents. Note that this algorithm has a complexity $O(MN^2)$ with M the

number body parts and N the number of candidates per body part. As an example of the detection results obtained with this method, Fig. 5 shows the first three prefered parses for three test image, ranked by order of increasing energies (negative likelihoods).

4.2 Learning the body tree

The cost functions used in our body tree model are based on geometric constraints on the relative positions of parts at a body articulation, as in Felzenswald and Huttenlocher[1]. Essentially, the articulation model is a linear combination of the differences between two joint locations, as predicted separately by the two body parts meeting at the articulation.

Algorithm 4 joint distance(l_i, l_j)

Compute joint location x_{ij}, y_{ij} given first body part location l_i
Compute joint location x_{ji}, y_{ji} given second body part location l_j
Return distance $d_{ij} = w^x_{ij}|x_{ij} - x_{ji}| + w^y_{ij}|y_{ij} - y_{ji}| + w^\theta_{ij}|\theta_i - \theta_j - \theta_{ij}| + w^s_{ij}|\log\frac{s_i}{s_j} - \log s_{ij}|$

Each body joint is parameterized by the relative sizes $s_{ij}$and angles θ_{ij}between its parts, and the four rigidity parameters $w^x_{ij}, w^y_{ij}, w^\theta_{ij}, w^s_{ij}$ governing the admissible range of apparent deformations of the articulation in position, size and orientation. We learned the relative sizes $s_{ij}$and angles θ_{ij} of each articulation by simply taking the average relative positions of all pairs of body parts over the training set.

For learning the rigidity parameters, we again used a Support Vector Machine. For each articulation A_{ij} between parts P_i and P_j ,we learned a 'combination classifier' based on a five-dimensional feature vector $F^0_i = D_i + D_j, F_{i^x} = |x_{ij} - x_{ji}|, F^y_i = |y_{ij} - y_{ji}|, F^\theta_i = |\theta_i - \theta_j - \theta_{ij}|, F^s_i = |\log\frac{s_i}{s_j} - logs_{ij}|$.

Using positive and negative examples from our training set, we used a linear SVM classifier to learn a set of weights $w^0_{ij}, w^x_{ij}, w^y_{ij}, w^\theta_{ij}, w^s_{ij}$ such that the likelihood is positive for all positive example, and negative for all negative examples. We experimentally verified that the learned weights have correct signs, i.e. $w^0_{ij} > 0$ and $wij^x < 0, w^y_{ij} < 0, w^\theta_{ij} < 0, w^s_{ij} < 0$, so that the learned model can indeed be related to the log-likelihood of the articulation

$$L(A_{ij}) = F^0_i - \frac{|w^x_{ij}|}{w^0_{ij}}F^x_i - \frac{|w^y_{ij}|}{w^0_{ij}}F^y_i - \frac{|w^\theta_{ij}|}{w^0_{ij}}F^\theta_i - \frac{|w^s_{ij}|}{w^0_{ij}}F^s_i$$

In our experiments with the MIT pedestrian database, the learned models performed slightly better than the naive approach assigning equal weights to all parameters and all articulations, and we expect the method to be of even greater benefit for dealing with the more complicated cases of people in action such as running or jumping.

5 Implementation and results

We implemented and tested our method using Matlab. The system consists in several function. An interactive application was designed for hand-labelling the examples and

storing the location of body joint and body parts. This data served both for training and testing our implementation. Another function was implemented for computing the image pyramid and extracting the image signatures at all locations x, y, s, θ . This was used both for generating the training data used by SVM and RVM, and to perform detection against the learned models. Finally, a parser was implemented based on the dynamic programming approach described above, which reads candidate locations from the 15 body part detectors, and produces a ranked list of candidate assemblies.

For learning SVM, we used a public domain implementation done at MIT (SvmFu, version 3.0). We implemented the learning algorrithm for RVM in Matlab following the same data format as SvmFu.

5.1 Experimental setup

We selected 100 frontal images from the MIT pedestrian database and labelled them into 15 parts each, as shown in Fig. 2. For each example, the labelling is carried out very effectively by clicking 14 body joints. Visible parts were clicked with the left mouse button, while occluded parts were clicked at their most likely (hidden) location, using the right mouse button. Only visible parts were used for training the part models. Hidden parts were useful for training the geometric model only. For every image, we additionnally picked 5 background regions to be used as negative examples.

In some of our experiments, we generated artificially larger data sets by varying the location of the hand-labelled data, and picking random samples of background. But this appeared to significantly decrease the precision and recall of the trained detectors, so we abandonned the approach. As a result, each body part classifier was trained with slightly less than 100 positive examples, and 500 negative examples.

We selected another 100 frontal images from the MIT pedestrian database, and labelled them exactly as before, to serve as a test set. The test set was used to evaluate each of the body part detectors as well as the detector of assemblies.

5.2 Detection of body parts.

It is traditional to compare detectors by tracing their ROC curves, i.e. tracing the true detection rate (recall) as a function of the false detection rate (1 - precision). In our case, the single most important parameter is the false detection rate necessary for achieving the maximum recall. We therefore compared the two detectors on our test set by measuring the false detection rates necessary for detecting all (known) body parts. We worked separately with each part detector, with the following results (in percentage).

As can be seen, RVM classifiers perform only slighly worse than their SVM couterparts, with mean false detection rates of 80.1 % and 78.5 % respectively. This is remarkable, given the extremely small number of relevance vectors used by the RVM detectors. For the purpose of detection, the advantages of the RVM clearly outweigh their inconvenience. Note that those false detection rates are high in absolute value, especially since we did not count confusions between parts as false detections. In practice, this means the actual number of false detections, including confusions between parts, may become 15 times as high. Even in this more conservative hypothesis, the number of candidates to be examined remains quite tractable, on the order of 75 candidates per

Part	1	2	3	4	5	6	7
SVM	84.9	81.6	87.2	76.6	77.4	78.2	73.8
RVM	84.3	86.4	87.0	76.4	75.8	82.2	79.0

8	9	10	11	12	13	14	15
75.3	78.6	82.4	76.6	77.4	77.1	73.3	77.5
81.2	84.8	81.2	77.4	80.0	78.4	76.8	80.1

Table 1. False detection rates for SVM and RVM body part detectors.

part. The worse-case number of body joint hypotheses is therefore $14 \times 75^2 = 78750$. In practice, we observed an average number closer to $14 \times 20^2 = 5600$ and used 50 candidates as a safe bet in all our experiments

We also note that the worst results are obtained for the torso (3) and face (2) models. The torso is probably the hardest body part to detect, since it is almost entirely shapeless, and is best detected indirectly, from geometric clues. In the case of the face model, the learned detectors are clearly not state-of-the-art, and we anticipate that they could be improved considerably by increasing the size of the training set and bootstrapping our classifier.

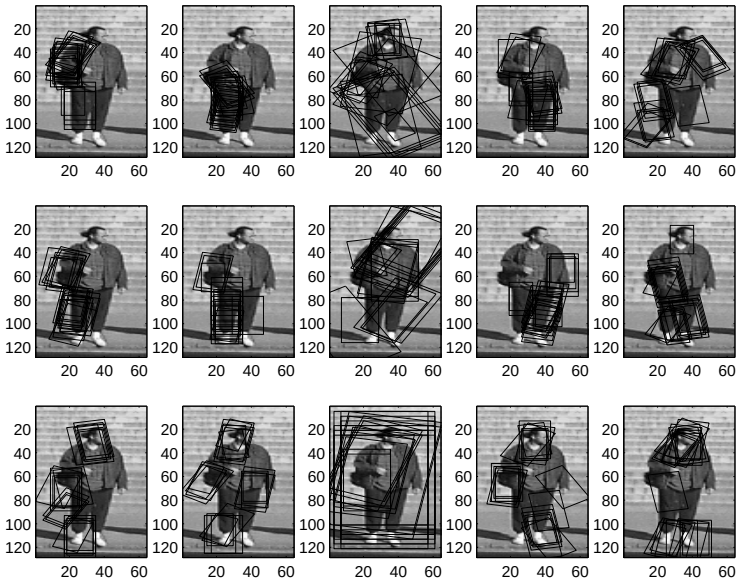

Fig. 4. Part detection results from test collection.

5.3 Detection of body trees

After selecting the best 50 candidates for each body part, in every image of the test collection, we evaluated our method by comparing the first best three detections obtained by our method with the correct interpretation. Thus, the task was purely that of detecting humans in the test set. In a first experiment, we used a naive geometric model with uniform rigidity parameters for all body joints, and obtained a correct detection in 70 cases using the SVM scores and 62 cases using the RVM scores. We missed the person completely in 30 cases and obtained incorrect interpretations caused by false positives in the background in 34 cases.

In a second experiment, we learned the geometric model using body joints from all training images. We used the correct assemblies as positive examples, and circular permutations of the body parts as negative examples. Using the learned model, we were able to improve the correct detection rates to 68 % using RVM scores and 78 % using SVM scores. We plan to better evaluate the method in our future work. In particular, we would like to better understand how much each body part detector contributes to the overall score, and how reliable the final labeling is for each body part. At this point, we can only emphasize that the detection of hands and feet is often important and reliable, despite the small size of the corresponding image regions, and that, in general, the geometric constraints significantly outweigh the image terms (by a factor of 10 to 20), which confirms the intuition behind the body plan approach.

6 Future work

The good detection rates achieved by the method make a convincing case that the body-plan strategy is applicable to real problems in image and video indexing. We now plan to extend this work to video, where we hope to improve the detection rates even further by making use of temporal and kinematic constraints. We are also encouraged to better evaluate the quality of the method beyond the pure detection problem - towards part-by part verification and 3D interpretation of the parse trees.

The main limitation of our model is that we cannot deal with missing or occluded parts. One possible way around this important problem is to consider all possible locations as was done by Felzensbaum and Huttenlocher. But we fear that this may limit the geometric precision of the method even further. All our experiments show that discretizing the space of locations too coarsely results in poor detection. Therefore, we are looking for alternative ways to handle incomplete assemblies, while increasing the precision of the locations, especially in sizes. Recently, Ioffe and Forsyth proposed to use mixtures of trees for learning and parsing incomplete and multiple body plans [5][6] which promises to offer a useful framework for solving the missing part problem..

Another drawback of our approach is the conflict between spatial resolution and localization. This is caused by the constraint that we set, to learn all detectors in the same geometric framework. One parameter of our implementation proved critical - the range of scales at which training and detection was performed. As of this writing, we allow a discrete range of boxes with fixed aspect ratio (16/9) and length 24, 27, 32, 38, 48, 64 and 96. As is visible in some of the figures, this results in poor localization of the

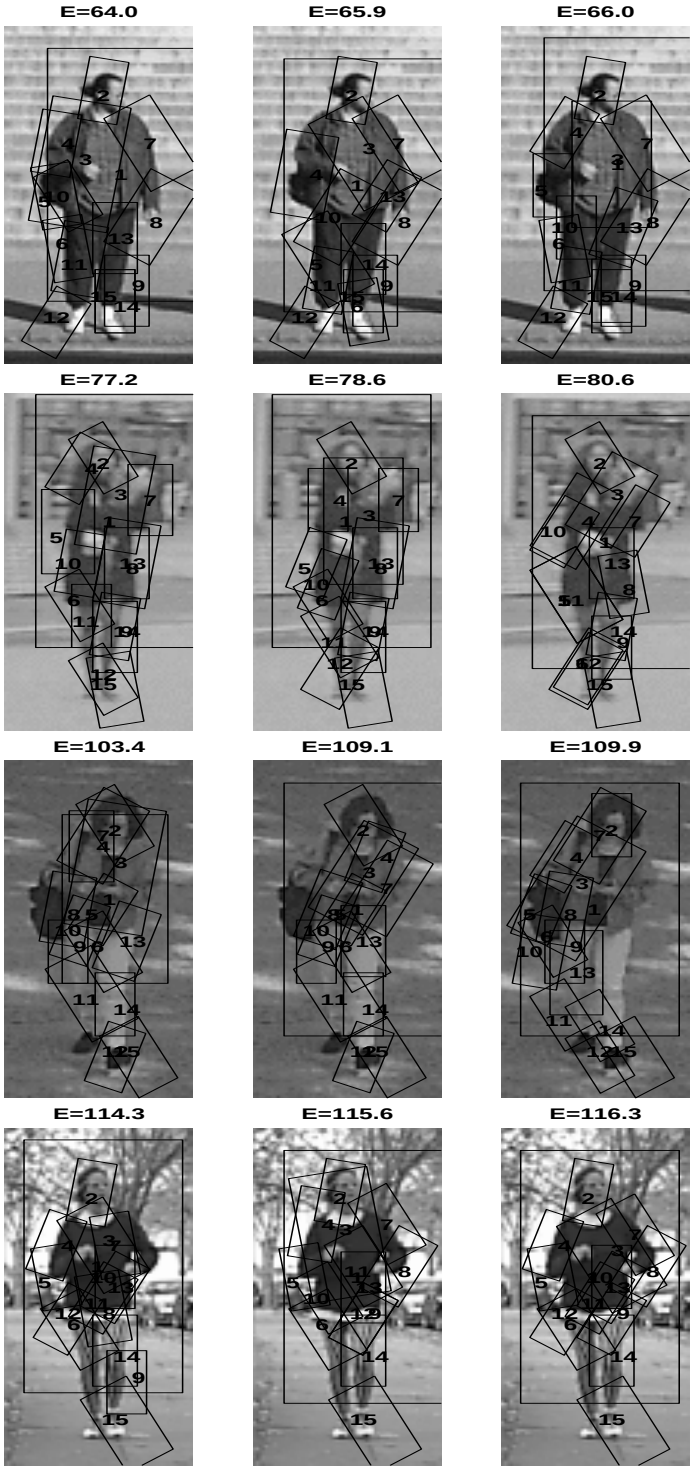

Fig. 5. Ranked detection results using learned model and RVM scores.

smaller features (such as the hands and feet). On the other hand, we experimented with segments of lengths 12, 14, 16, 18, 24, 36, 48 and 96 and a spatial resolution of 12, but found that the precision was deteriorated for the larger features (as in the head, torso and body). Clearly, a compromise has to be found, and more experiments are needed.

Finally, we have chosen to use gaussian image derivatives in all our experiments because they are a popular and consensual choice. But the construction of the image pyramid is computationnally expensive, and we plan to move to a more efficient implementation, which could rely on a more thorough selection of the feature vectors. This could be based on using RVM classifiers to learn 'relevant features' rather than relevant vectors. Alternatively, Sidenblah and Black [15][14] have recently presented an approach for learning the image statistics of people vs. background from video, which could prove very useful for learning better models by selecting better features.

7 Conclusion

By combining several recent approaches, we have come to some positive conclusions. Firstly, it is possible to learn appearance models for human body parts from examples and to use them as input to a body plan parser, at least for a modest-size problem such as pedestrian detection. Secondly, we have been able to learn geometric models for the combination of the detected parts, allowing us to robustly estimate the likelihood of a body part assembly, without recourse to sampling or HMM distributions, which require thousands of examples to be learned efficiently. Thirdly, the learned model leads to a very efficient decoding algorithm, amouting to no more than the combination of very well known template matching and dynamic programming techniques.

Those positive conclusions encourage us to investigate two main directions of research. We expect that boostrapping (adding examples on which our current model fails, and retraining) should offer even better detection rates, and increased discrimation between body parts. And we plan to learn appearance models for people in motion pictures and video, in which case we will need to extend our approach to the more difficult problems of multiple resolutions, activities and viewpoints.

References

1. Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient matching of pictorial structures. In *CVPR*, 2000.
2. D. Forsyth and M. Fleck. Body plans. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
3. B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. Technical report, AI Memo 1687, Massachusetts Institute of Technology, 2000., 2000.
4. D. Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.
5. Sergey Ioffe and David Forsyth. Human tracking with mixtures of trees. In *ICCV*, 2001.
6. Sergey Ioffe and David Forsyth. Mixtures of trees for object recognition. In *CVPR*, 2001.
7. Sergey Ioffe and David Forsyth. Probabilistic methods for finding people. *International Journal of Computer Vision*, 43(1), 2001.

8. S. Ju, M. Black, and Y. Yacoob. Cardboard people: a parameterized model of articulated image motion. 1996.
9. D. Marr and H.K. Nishihara. Representation and recognition of the spatial organization of three dimensional structure. *Proceedings of the Royal Society of London B*, 200:269–294.
10. Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example-based object detection in images by components. *IEEE trans. PAMI*, 23(4), 2001.
11. D. Morris and J. Rehg. Singularity analysis for articulated object tracking. In *Int. Conference on Computer Vision and Pattern Recognition (CVPR98)*, 1998.
12. C. Papageorgiou. Object and pattern detection in video sequences. Technical report, Master's thesis, Massachusetts Institute of Technology, 1997.
13. K. Rohr. Incremental recognition of pedestrians from image sequences. In *CVPR93*, pages 8–13, 1993.
14. H. Sidenbladh and M. Black. Learning the statistics of people in images and video. *International Journal of Computer Vision*, 2001.
15. H. Sidenbladh, F. Torre, and M. Black. A framework for modeling the appearance of 3d articulated figures. In *Fourth International Conference on Automatic Face and Gesture Recognition, Grenoble, France*, 2000.
16. M. Tipping. *The relevance vector machine*. Morgan Kaufmann, 2000.
17. M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.
18. Liang Zhao and Chuck Thorpe. Recursive context reasoning for human detection and parts identification. In *IEEE Workshop on Human Modeling, Analysis and Synthesis*, 2000.