

A Hierarchical System for Recognition, Tracking and Pose Estimation

P. Zehnder¹, E. Koller-Meier¹, R. Fransens², and L. Van Gool^{1,2}

¹ Computer Vision Laboratory, ETH Zurich, Switzerland

² ESAT/PSI, KU Leuven, Belgium

Draft Version: November 15, 2004 – 10:26 A.M.

Abstract. This chapter presents a system for the recognition, tracking and pose estimation of people in video sequences. It is based on a careful selection of Haar wavelet features and uses Support Vector Machines (SVM) in spaces of reduced dimensionality for classification. Recognition is carried out hierarchically by using a set of detectors and discriminators for people and poses. The characteristic features used in the individual nodes are learned automatically. Tracking is solved via a particle filter that utilizes the SVM output and a first order kinematic model to obtain a robust scheme that successfully handles occlusion, different poses and camera zooms.

Table of Contents

A Hierarchical System for Recognition, Tracking and Pose Estimation . . .	1
<i>P. Zehnder¹, E. Koller-Meier¹, R. Fransens², L. Van Gool^{1,2}</i>	

1 Introduction

As part of our research towards the semantic interpretation of films and sitcoms, this chapter focuses on the recognition, tracking and pose estimation of people. A possible application is a virtual commentator to be used as an add-on to the sound-track of programs for visually impaired people. This commentator would produce a spoken description of the scenes and events. Clearly, people recognition and tracking has applications in many other areas as well, like visual surveillance, human-computer interfaces, or video indexing.

People detection is a difficult task because the significant pattern variations are hard to parameterize analytically. Such variations comprise different lighting conditions and the variety of appearance, face expression and structural characteristics such as clothes, glasses or mustaches. Researchers have approached the detection problem with different techniques, including Support Vector Machines [14,2,9,3], neural networks [12] and geometric constraints between different face features [1,16] like eyes or lips.

Mikolajczyk *et al.* present a method for detecting humans in single images, see chapter ???. Their approach is based on a probabilistic assembly of robust part detectors and is able to detect full bodies and close-up views in the presence of clutter and occlusion.

Some approaches rely on specific assumptions that support the detection for example. A fixed camera and a static background facilitate the detection as a moving object can be segmented by background subtraction [8]. Unfortunately, this is not possible in our case.

For our application we use the combination of a wavelet transform for the feature selection and SVMs for classification. These choices render the approach quite generic, so that it can be used for other object categories as well. The resulting system can easily be trained for other categories without any alteration. Simultaneously, we have tried to make it time-efficient through careful feature selection, dimensionality reductions, and cascaded classification [2]. The approach is akin to that proposed by Evgeniou *et al.* [3], but faster and with a slight increase in performance. The work has also been inspired by the work of Viola and Jones [15], which presents acceleration techniques based on a special form of image representation, on a feature selection method based on AdaBoost [15] and a cascaded search framework. Post-processing removes multiple detections in the target area. The work presented here and in [2] combines ideas from [15] and [3] and introduces several extensions and modifications, with a special focus on dimensionality reduction at various levels of the algorithm.

Detection methods tend to scan complete images in a bottom-up fashion and are therefore not suitable for the processing of entire videos. processing. Hence, we have combined our detectors with particle filter tracking, thereby adding an efficient top-down component. As a result, the detection function is only evaluated at the likely new object positions. In comparison, most of the state-of-the-art approaches either recognize people at every frame or they detect them in an initial frame and then track them through the sequence. To handle cluttered backgrounds and partial occlusions, particle filters have proven effec-

tive as multiple hypotheses are handled simultaneously. Apart from edge-based image features [5] also color distributions [7] have been used in particle filter frameworks to track people. Up to now, only few systems have merged these two tasks [14,4]. Verma et al. [14] describe a face detection and tracking system that uses the detection method by Schneiderman et al. [13] and combine it with a particle filter. While our first segmentation step also applies wavelets, our further processing follows a different idea. Instead of using statistics of wavelet coefficients (histograms), we utilize the output of the SVM directly. Furthermore, features are combined hierarchically to model both the (upper part of a) body in general and specific characteristics of an individual person. Giebel et al. [4] present also a combination between a detector and a tracker. Although they use a particle filtering framework for tracking, they detect pedestrians using a hierarchical template matching technique where the matching involves traversing a tree structure of templates. Template matching provides good detection performance, but the creation of templates is generally a complex task.

The recognition task within this work is solved using a hierarchical approach using a series of detectors for people and discriminators for specific characters and poses. Besides the aspect of time savings, the hierarchical decision tree also reduces the rate of false positives in comparison to several independent detectors. Nakajima et al. [8] also follow the concept of a hierarchical technique. Their system for people recognition and pose estimation uses color histograms and local features. SVMs are learned for all combinations of classes and combined hierarchically to form a decision tree. In comparison to our approach, besides different features and using a different SVM scheme, the authors in [8] employ a static background and no tracking is involved. Furthermore, it is assumed that people do not change clothing.

The novelty of the proposed system lies in the original mixture of a hierarchical feature selection based on wavelet transforms which feeds into a SVM hierarchy to classify specific characters and poses. In combination with a particle filtering framework, we achieve a reliable and fast system for recognition, tracking and pose estimation of people.

The following sections describe in more detail the approach to person detection, recognition, pose estimation, and integrated tracking, in that order.

2 Pedestrian Detection

The following sections describe the key components of the pedestrian – or general object – detection system as presented by Depoortere *et al.* [2], which provides the foundation of our work on recognition and tracking. It gives an overview of the combination of the wavelet transform features and SVM classification. It also discusses how the approach can be accelerated in various ways. Combining those techniques it is possible to build a very efficient single object class detector. As already mentioned, this work builds further on ideas first propounded in [3,15].

The pedestrian detector classifies a rectangular image window as pedestrian or non-pedestrian (we refer to ‘pedestrians’ here to emphasize the detection of

full-body partners; the sequel where we focus on sitcoms actually employs torso detection). Therefore, to detect all pedestrians in an image, an exhaustive search over the whole image and at multiple scales is required.

The actual detection is split into two parts: *feature extraction* and *classification*. The former consists of a 2D Haar wavelet transformation of the patch, where the resulting feature vector is then classified by a SVM.

This approach has been shown to deliver very good detection performance. Nevertheless, it is computationally very expensive. Therefore, various techniques have been investigated to cut down processing time without sacrificing detection quality. They include the following aspects:

- integral image representation / acceleration of the Haar feature extraction
- dimension reduction via feature selection
- dimension reduction by projecting the feature vector in a low dimensional subspace
- reduction of the number of support vectors
- building a cascade of classifiers of increasing complexity

Indeed, it is possible to make the algorithm several magnitudes faster, so that it works almost in real-time and as such is also suitable for use in video sequences instead of static images only.

2.1 Wavelet Transform

The 2D wavelet transform is used to extract a collection of features from a rectangular patch of interest in an image. There exist various other ways of transforming an image into a feature vector, but wavelets have some very attractive properties for our application as they can be interpreted as a decomposition of the shape of an object. Yet, no specific kind of model is assumed which allows the system to be used in varying applications. Additionally, it is reasonably fast to compute a wavelet transform, especially when Haar-type wavelets are used.

Other feature extraction techniques typically have drawbacks in at least one of these two aspects. Either they do not provide enough shape information, or it is very expensive to extract the features. Color histograms for example are a purely global feature of the object region while raw pixel values do not include edge information. Complex shape models, on the other hand, may provide quite accurate and valuable information, but the calculation of the best matching parameters is generally expensive.

In figure 1 the 2D wavelet transform using Haar wavelets is depicted. Basically, it is a decomposition of an image using the three characteristic patterns as shown. One can think of them as simple edge elements, oriented horizontally, vertically and diagonally. The image is represented as a combination of wavelets at various positions and scales.

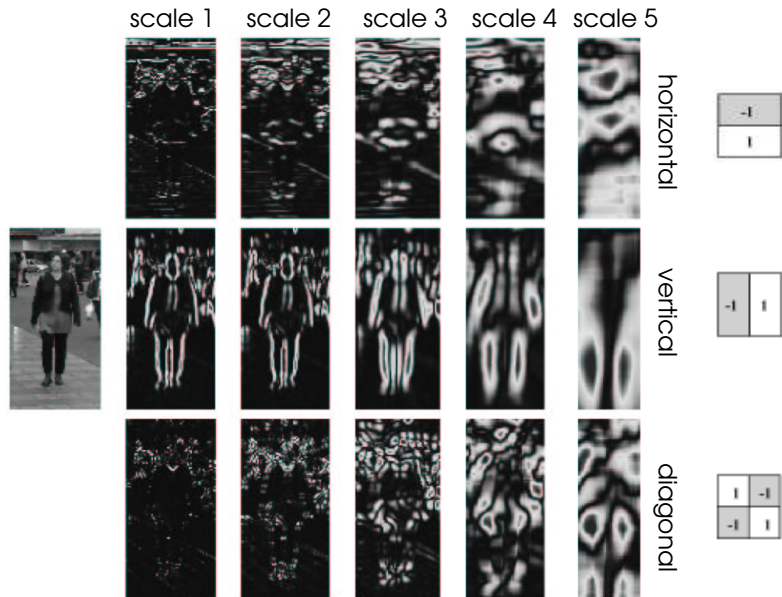


Fig. 1. Schematic representation of the 2D wavelet transform using Haar wavelets.

2.2 Integral Image

To speed up the calculation of the Haar Wavelets, Viola and Jones [15] have proposed to represent the image as an “integral image”. In this form, each pixel in the image is replaced by the integral over all the pixels contained in upper-left quadrant with respect to that pixel. Given an integral image, a Haar feature at any scale can be computed by only a few lookups and basic operations. Although the conversion of the image is an additional processing step for each image in a sequence, the accumulated speedup is more dominant when evaluating a large number of frames inside the same image.

2.3 Support Vector Machine Classification

Support Vector Machines (SVMs) learn pattern classification from positive and negative exemplars. The basic idea is to map the training data into a higher dimensional feature space where the two classes can be separated by a linear hyperplane. This is done in an optimal way, namely by maximizing the margin between the hyperplane and the closest patterns of both classes. An important point to note is that the mapping is done implicitly by using a kernel function K . Such a kernel has the property that while being some – usually nonlinear – function of two vectors in the input space it corresponds to a dot product of the arguments in the mapped feature space.

After the training stage we get an evaluation function which is based on a number of kernel evaluations on support vectors $\{\mathbf{s}_i\}_{i=1}^{N_s}$ – the patterns lying closest to the separating plane:

$$f(\mathbf{x}) = \sum_{i=1}^{N_s} \alpha_i y_i K(\mathbf{s}_i, \mathbf{x}) + b \quad (1)$$

where \mathbf{x} represents the feature vector based on wavelets. $y_i \in \{-1, 1\}$ denotes the class membership of the respective support vector \mathbf{s}_i . The parameters α_i and b are coefficients respectively an offset determined in the training phase while searching the optimal separating plane. The class membership of \mathbf{x} is then given by taking the sign of the evaluation function $\text{sgn}(f(\mathbf{x}))$.

There exist various kernels that can be used for SVM classification. A concrete example is a polynomial of degree 2:

$$f(\mathbf{x}) = \sum_{i=1}^{N_s} \alpha_i y_i (\mathbf{s}_i \cdot \mathbf{x} + 1)^2 + b \quad (2)$$

which is used extensively in our work. This kernel is moderate in computational complexity but still delivers good classification performance.

SVMs have been used for the recognition of all kinds of objects with a good robustness against different lighting conditions, appearances and backgrounds. Also, the underlying mathematical foundation (statistical learning theory) is well developed.

2.4 Feature Selection

The task of feature selection is important as it determines the speed of the system. A complete wavelet transform even on a small region can easily lead to very high dimensional data vectors. Sizes of hundreds or thousands of dimensions are not uncommon. Furthermore, the problem is aggravated if the density of wavelets at larger scales is artificially increased in order to gain higher spatial resolution.

Four methods of feature selection are presented in the following. The question of the choice of the best one cannot be answered in a general way. Experiments have shown that different settings – e.g. number of features – cause different methods to stand out. Consequently, the choice has to be taken by empirical evaluation.

Mean based This method was proposed by Papageorgiou *et al.* [11]. It separates the features into classes according to scale and orientation. For each class the mean value is calculated over all coefficients of a class and over all training images. The coefficients are then normalized by dividing by their respective class mean. Finally, for each coefficient, the mean is calculated of the normalized value over all training data. The effect of this procedure is that the attractive

coefficients can be identified as those with values much lower or much higher than 1 – corresponding to constant uniform regions or consistent intensity changes. Unspecific features on the other hand result in a mean of approximately 1.

Bhattacharyya based The Bhattacharyya based method is based on class conditional densities $p(x_i|\Omega)$ and $p(x_i|\bar{\Omega})$ for all features $i = 1, \dots, n$ and with Ω the pedestrian class and $\bar{\Omega}$ the non-pedestrian class. These densities can be estimated from the training data. To determine interesting features, the Bhattacharyya measure is determined:

$$b(p(x_i|\Omega), p(x_i|\bar{\Omega})) = \int \sqrt{p(x_i|\Omega)p(x_i|\bar{\Omega})} dx \quad . \quad (3)$$

A small value of this measure indicates a strong feature.

AdaBoost based AdaBoost itself does not originate in the feature selection field. As the name suggests it is a boosting technique which has the goal to improve the accuracy of a given learning algorithm. This is done by iteratively combining weak classifiers in order to build a strong classifier. An important point is that in each iteration the focus is put on the examples in the training set that have not been classified correctly in the previous stages.

The adaptation of boosting to feature selection is based on the principle of using a separate classifier for each single feature. As a consequence – by successively selecting classifiers – a collection of features is set up respectively. In our case a feature corresponds to one component of the wavelet transform. The classification rule is based on the histograms of that feature over the positive as well as the negative training set. The selection of the classifiers is carried out iteratively and in each iteration the best classifier at that stage is determined according to the minimum weighted error on the training set. Weights are assigned to each example of the training set individually and updated after every iteration. Correctly classified examples are reduced in weight whereas erroneous classifications result in higher weights. The algorithm finishes when the desired number of features is selected.

SVM gradient based This method is based on an initially trained SVM classifier that uses all available features. For this classifier the gradient of the evaluation function is calculated at the locations of the support vectors. After setting each entry of each gradient vector to its absolute value, the vectors are summed up. In the resulting sum we can identify interesting features by higher values.

2.5 Projecting Onto a Low Dimensional Subspace

In addition to feature selection, the dimensionality of the problem can be further reduced by projecting the feature vector onto a lower dimensional subspace. This does not reduce the number of features to be calculated but the calculation of

the SVM evaluation function becomes cheaper. In order not to lose accuracy we have to identify the directions of maximal discriminative power. These are found by putting the gradients in the support vectors in a matrix \mathbf{N} as columns and calculating the eigenvectors of the matrix

$$\mathbf{G} = \mathbf{N}\mathbf{N}^T \quad . \quad (4)$$

2.6 SV Reduction

The amount of computations of the SVM evaluation function depends directly on the number of support vectors. Therefore it is desirable to keep the number of support vectors as low as possible. Osuna and Girosi [10] propose a method that replaces or approximates the evaluation function by another function of the same family, but which has a considerably lower number of support vectors than the original.

2.7 Cascade of Classifiers

Doing a brute force search on an image has the drawback that we need to evaluate an overwhelming number of candidate windows just to find a few pedestrians. A means to counter this problem is to use a cascade of classifiers instead of a single classifier. The cascade is built as a series of increasingly complex classifiers. The task of the early stages is to throw away most windows with a minimum amount of calculations. The evaluation of the powerful but slow classifiers further down the cascade is limited to only a fraction of the original window set.

2.8 The Pedestrian Detector in Figures

The methods described within this section have been implemented in a combined system in order to investigate the achievable speedup and the impact on the precision. Also it has been compared to the detector presented by Papageorgiou *et al.* in [11]. To allow for this, the same basic setup with 1326 features has been used. The proposed modification were then applied as follows. As a first step, additional features have been added to account for the possible decrease of accuracy due to the applied acceleration methods. In detail, a total number of 4737 has been used. The system was split up into a cascade of 4 steps using different numbers of features and kernels:

1. a SVM on the basis of 4 features, with a linear kernel
2. a SVM on the basis of 29 features, with a linear kernel
3. a SVM on the basis of 29 features, with a kernel of degree 2
4. a SVM on the basis of 1326 features, with a kernel of degree 2

The feature selection methods used are SVM gradient based for step 4 and AdaBoost based for the first three steps respectively. The most complex system using 1326 was accelerated by subspace projection using 13 dimensions. For the

last two steps, the number of support vectors could be reduced from 1594 and 2118 to 106 and 299 respectively (see section 2.6).

In the end, the achieved speedup was of a factor of 905, that is roughly 3 magnitudes. On an AMD Athlon 1.4GHz, the detector runs at 6 frames/sec for images of 640x480 pixels. Alternative accelerations of this system have been proposed by Papageorgiou [11], but with a substantial loss of precision as a result. Our approach on the other hand has an even improved detection performance compared to the original system.

3 People Recognition

In this section, the goal is not only to detect people, but also to recognize certain individuals. In particular, we apply these techniques to the characters in the sitcom ‘Fawlty Towers’, used as a common test case in the CogViSys project. We have systematically adapted our detectors from full-body to upper-body detectors, as the legs are too often occluded in the Fawlty Tower images. The way to build these detectors was identical, however. Next, we describe different strategies towards the recognition task, i.e. a ‘direct’, purely sequential search with completely independent detectors for different characters, vs. a ‘hierarchical’ search, where object classes share features.

3.1 Direct Approach

Based on the pedestrian detector described above, a system for detecting and recognizing people – in our sitcom application actors – is constructed. The goal is to use several person-specific detectors instead of having just one single detector for any person. The methodology that underlies the pedestrian detector is flexible enough to produce these specialized detectors as well. All that is required are separate training sets for each of the individuals that shall be recognized. For each individual the most important features are determined first by the method described in section 2.4.

In this manner a set of detectors is created, one for each of the individuals. So, a single character is detected and recognized in one step. In order to find all characters on an image the detectors are applied separately on the image. Thus, the characters are located sequentially.

3.2 Hierarchical Approach

In this section we present a more elaborate technique for the detection of particular actors. Instead of applying a separate detector for each actor, which starts from scratch, the detection works in stages, where each accomplishes a task at a certain “level of detail”. The reason is the observation (section 7.1) that a detector built specifically for one person also tends to respond more to other people than to other object classes. This suggests that a particular detector for some

person basically behaves like a general person detector with a specialization on a particular individual.

As a consequence, a hierarchical approach is proposed for the task of detection and recognition of people. The principle is demonstrated for the case of a two stage hierarchy and two different persons. In a first step, a general people detector is used to locate people independently from their identity. In a second step, the system tries to discriminate the different characters. The advantage of this approach is that we avoid repeating the general people detection part as it would be the case when using the direct approach. So, our system can be described in the following way:

- **Stage 1: General People Detector.** To construct the general people detector a training set is acquired containing exemplars of *all* the characters to be recognized. This serves as the positive set. As negative training set, a collection of images containing no people at all is used. In our case, randomly extracted rectangular regions are used coming from the same image sequence that provides the set of positives. The most important features are determined using the AdaBoost technique described in section 2.4. Based on the obtained feature set, the system is trained i.e. the values of these features are calculated for the positive and negative training sets and the SVM is trained with the resulting feature vectors.
- **Stage 2: Discriminator.** To build a discriminator, a collection of exemplars for a specific actor is acquired to serve as positive training set. A negative training set is set up containing also exemplars for other characters. Given these training sets feature selection is carried out (section 2.4). Then, the corresponding feature values are determined and the SVM is trained to get the discriminator.

The presented approach can be extended when there are more than two individuals. A possible approach is to build additional two-class discriminators and to combine them in a multi-class discriminator. Another possibility is to extend the hierarchy further. As we know that the object classes – individual people in our case – possess similarities, it is preferable to first focus on the features common to all of them and only then on the specific differences. Once we detect that some image window does not contain a person, we do not need to spend any more effort. On the other hand when we have the information that a person is present, we only need to investigate the additional, characteristic features that distinguish one from the others. With this approach we hope to avoid calculating redundant information.

Another aspect of hierarchical recognition is its connection to the scene description. Maybe only a quick overview is needed such as: “There are two people in this scene”. Or else more detailed information is requested like who are in the scene and what they are doing. With the proposed hierarchical approach the computational load can be better tuned towards such different queries.

4 Pose Estimation

The task of pose estimation in our approach is addressed in the same way as the problem of recognizing people described above. We use a combination of the wavelet transform and SVM classification like in the pedestrian detector. Based on that technique, several pose specific detectors are created. In fact, they are discriminators each of which is responsible for separating one pair of poses. The final pose can then be inferred by combining these two-class classifiers to form a multi-class classifier.

To build the pose discriminators, a separate training set is acquired for each of the poses that shall be estimated. One pair of these sets is taken at a time and the according discriminator is set up. In a first step the most important features are determined using the mentioned AdaBoost feature selection (section 2.4). Then the SVM is trained with the reduced feature set.

We have developed detectors for the heading orientation of a person (which we will refer to as ‘pose’ here). To this end separate training sets have been collected for frontal views of a person and for situations where the person is facing either to the left or to the right side. As will be seen in the results section, the responses of these detectors are graded rather than all-or-nothing. This opens further prospects for a fine-grained orientation reading based on the combined output. This possibility has not been explored yet.

5 Alternative Tree Approach

Our most complete system of detectors extracts both person identities and poses. As part of this work we have also elaborated an alternative to the described hierarchical approach we have investigated another form of a tree-structured classifier. In contrast to the former it employs a series of discriminators at the beginning in order to determine a suitable detector for the image window at hand (figure 2). The principle is to recursively split up categories of classes before tackling the actual detection problem. This approach relies on similarities among different classes which allow to group them together.

In our case we are interested in pose and identity of the person. Therefore we first use a pose discriminator. In the following step a discriminator for different people in that pose is applied. A leaf node is reached which contains a person- and pose-specific detector. This final classifier decides upon the actual occurrence of such an appearance.

6 Tracking

To speed up the detection, the SVM is integrated into a particle filter [5], also known as Condensation approach in the computer vision community. Furthermore, the tracking reduces false detections which creep into the detection more easily when no temporal knowledge is used. The Condensation algorithm can

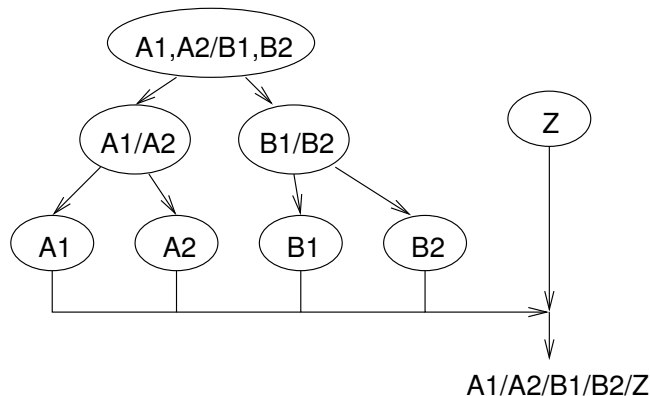


Fig. 2. Alternative tree approach. The structure represents a chain of discriminators that split up categories of classes and lead into several class specific detectors at the tree leaves.

represent non-linear problems and non-Gaussian densities by propagating multiple alternate hypotheses (particles) simultaneously. Basically, Condensation is used as the general tracking framework, while the role of the detector is to provide a measure for each of the particles of the Condensation tracker. The physical properties of our objects are represented in a first order motion model for position and a zero order model for the size. Therefore, the state vector used by the Condensation tracker consists of position, velocity and scale of the object. To model the stochastic diffusion, a Gaussian probability distribution is used.

To determine the weights of each sample, the SVM evaluation function is calculated for the rectangular window specified by the corresponding state vector. Subsequently, the weights are normalized and transformed by an exponential function. This is necessary because the SVM outputs do not represent a probability distribution. The exponential function allows to shape the distribution in a flexible way so as to either focus on the few best scoring particles only or to make a more uniform selection among the particles.

The initial particles of the Condensation tracker are spread at positions where the target is expected to appear, while a target is recognized on the basis of the SVM output.

7 Experimental Results

The experiments are demonstrated on sitcoms as they offer a constrained world. They feature only a small number of characters and a small number of sets in comparison to films or real-life scenes. However, they are challenging as camera movements, different non-static backgrounds and appearance changes have to be managed.

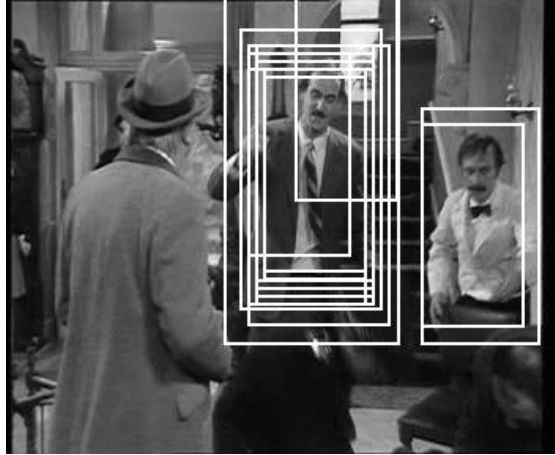


Fig. 3. Basil detector using the direct approach.

The following results are presented on an episode of *Fawlty Towers* from which training and test sets have been extracted for each of the main actors. The extraction has been done manually by specifying rectangles containing the upper part of the body of a specific actor. The cropping rectangle was chosen to have a fixed ratio of height/width of 2:1. It has been scaled for each exemplar so as to contain the body from the hip up to the head in the vertical direction and the shoulders in the horizontal direction. A small gap of about 10% of the person’s size has been left out between the rectangle border and the person in order not to destroy potentially valuable silhouette properties.

That way a total of roughly 1000 exemplars per character have been collected from one half of an episode to serve as training set. The other half of the same episode has been used to create the test set. The individual sets have also been split up into pose specific sets containing *left*, *right* and *frontal* views. Additionally, for all sets a corresponding subset has been created consisting only of images that differ substantially in their background. These have been used for feature selection because otherwise a lot of non-person areas would have been considered important. All the following results were produced by processing grey scale images, as the use of color information brings only minor improvements [2].

7.1 People Recognition

Figure 3 shows a result for the direct approach to people detection and recognition. The rectangles indicate the regions where the algorithm has detected the character ‘Basil’. As can be seen there are several detections that are correct, varying slightly in scale and positions. By some simple post-processing it is possible to combine those overlapping regions into one single result. There are some false detections however, of which some point to another character of the series named ‘Manuel’. This indicates that a recognizer for a particular person also

responds to other people. That becomes even more clear when looking at the histogram of output values when applying the Basil detector on test sets containing random patterns or exemplars of Manuel or Basil (figure 4). It can be seen that the Basil recognizer responds to a substantial amount of the Manuel test set depending on the chosen threshold. For example when choosing a threshold value of -0.7 , we get 98% detections of Basil with 4.1% false detections on the random set and 75% response on the Manuel test set.

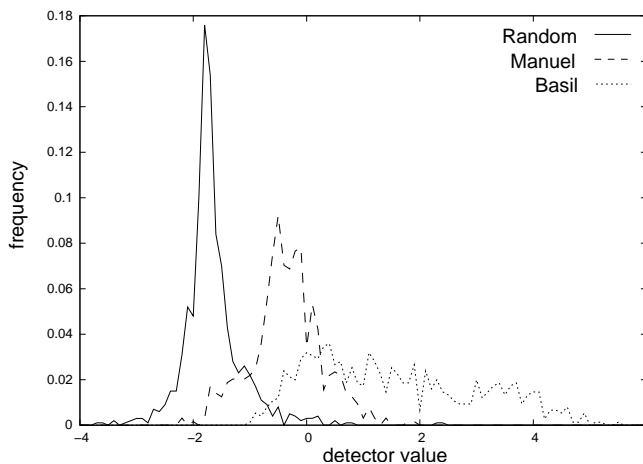


Fig. 4. Histogram of the detector values of the Basil detector (direct approach) on various test sets. A positive value indicates “Basil” while a negative value means “not Basil”.

A comparison of the direct approach for people recognition with the hierarchical approach is presented in figure 5 in terms of ROC curves. They show the recognition performance for the two characters Basil and Manuel. The two direct detectors are trained using the 80 most important features. The hierarchical approach also uses 80 features in both stages. This has been chosen to facilitate the determination of the computational complexity because all kernel evaluations in equation 1 are equally expensive and the complexity depends only on the number of support vectors. Table 1 shows the exact number of support vectors computed for the chosen setup. The total effort needed is about the same for both approaches. In detail it depends on the probability of a frame to contain Basil, Manuel or neither of them. For example the hierarchical scheme is faster for detecting that no exemplar is present as it only needs to evaluate one SVM in this case. That is primarily advantageous when scanning then whole image as there are mostly non-person frames to evaluate.

Comparing the two approaches, the ROC curves show that the hierarchical approach is superior in the range of 70-95% recognition rate. An important property of both methods is that the percentage of false positives increases quickly

when going towards very high detection rates. To avoid the undesirable effect of too many false positives one can reduce the detection rate slightly below the achievable maximum. Interestingly, at this point the hierarchical approach turns out to be much better in the sense that it shows a very low ratio of false positives. Looking at the horizontal distance of the two curves and considering the logarithmic scale of the abscissa, the amount of false positives is several times higher for the direct approach. In the context of video analysis the loss of detections can be compensated by the fact that a single shot tends to be represented by several keyframes, thereby giving the detectors repeated chances to detect the actor. On the other hand, an avalanche of false positives renders automated retrieval tools like these close to useless.

Direct:	Basil Manuel	110 70	} 180
Hierarchical:	Basil/Manuel combined Basil vs Manuel	157 30	} 187

Table 1. Number of support vectors for the direct recognition and the hierarchical approach.

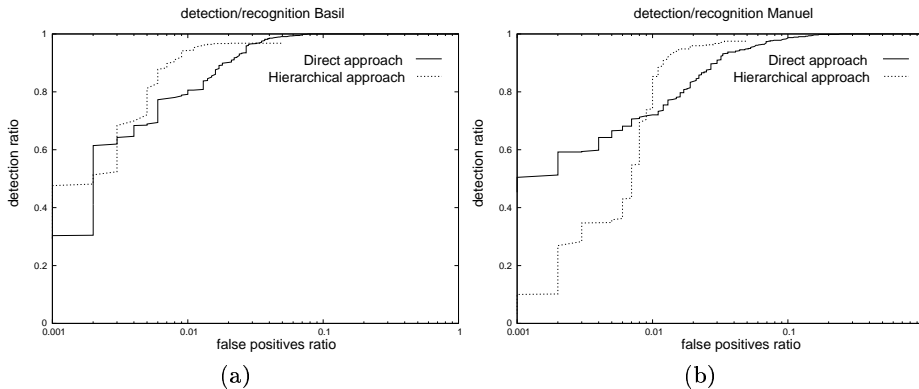


Fig. 5. Comparison of the direct and the hierarchical approach for recognition, ROC curves are for the detection and recognition of Basil (a) and Manuel (b).

7.2 Pose Estimation

For pose estimation three discriminators have been trained. Each one is responsible for discriminating one particular pair of poses, *left* \Leftrightarrow *front*, *left* \Leftrightarrow *right* and *front* \Leftrightarrow *right*. At present we use a single character for pose estimation

assuming that the recognition has already been done. In the feature selection step the 40 most important features have been selected. This is less compared to the detection and recognition task where 80 features are used. Our experiments have shown that it is much easier to determine poses than to detect people.

An example for pose estimation is shown in figure 6, on a sequence of images where Manuel is performing a turn, from facing left over a frontal pose to facing right. The curve shows the corresponding output of the three 2-class pose detectors over time. To get a multi-class estimate of the pose one is required to evaluate two of them at each frame. Doing this results in the interpretation *left* \rightarrow *front* \rightarrow *right* with transition indices 10 (left/front passing the threshold level) and 21 (front/right passing the threshold level). This quite accurately corresponds to the true evolution of the pose.

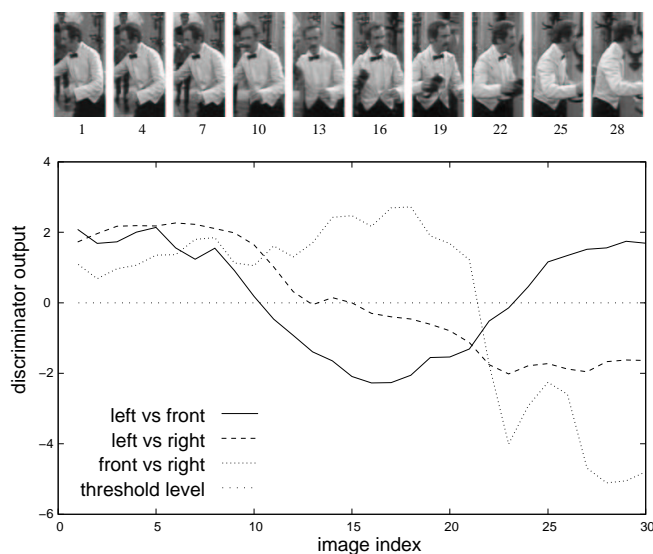


Fig. 6. Response of the 2 class pose estimators on a characteristic sequence.

7.3 Alternative Tree Approach

To investigate the second hierarchical approach an experimental setup has been used that is able to discriminate two types of poses and two characters of Fawly Towers. The pose types are frontal and profile views of a person. The characters to be discriminated are Basil and Manuel. So the resulting tree classifier looks as in Figure 7.

This setup is compared to a straightforward detector in terms of ROC curves. For the comparison both approaches were trained such that the overall computational complexity was roughly the same. The analysis takes into account the

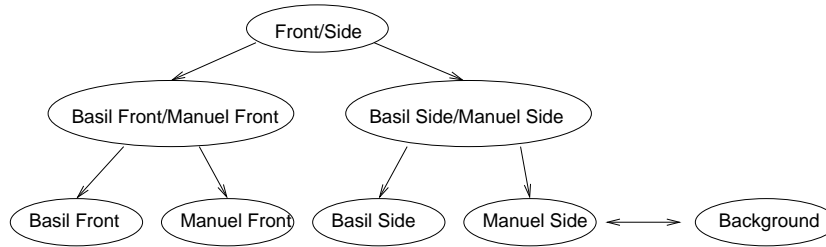


Fig. 7. Concrete setup of the second variant of the tree classifier used in the experiments.

detection aspect but does not incorporate the person/pose recognition as that information is unavailable for a single class detector. Results are shown in figure 8.

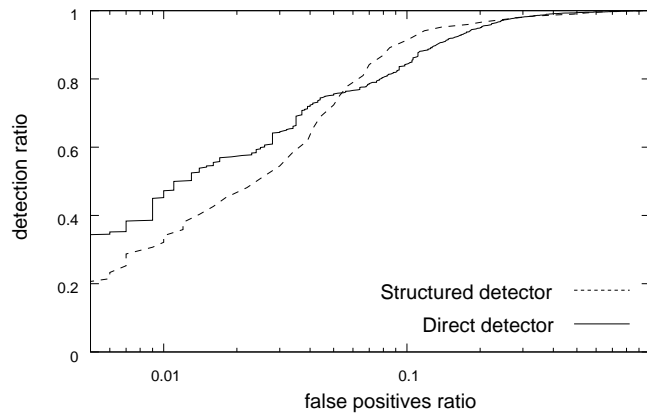


Fig. 8. Comparison of the tree-structured detector of Fig. 7 and a direct detector.

The graph reveals that there is no overall superiority of one of the approaches. Rather it depends on the desired detection rates. At rates above 80% the tree structure shows improved performance over the direct detector whereas at lower rates it is rather at a disadvantage.

7.4 Tracking

Figure 9 shows an example of our proposed tracking algorithm. The individual particles are shown as rectangles to bring out the characteristics of the processing. The first thing to note is that the actual tracking succeeds although the sequence offers some challenges. In the middle of the sequence the person to be tracked is partially occluded by some other persons in the foreground. That

problem is typically not handled very well in most approaches. In our example it is obvious that for a short interval the tracking becomes inaccurate, but after becoming completely visible again the correct focus is regained. Clearly, this shows the robustness of our method against partial occlusion.

For comparison, we also tested a gradient method combined with a Kalman filter. The prediction of the Kalman filter provides a rough state estimation while this initial result is then optimized along the gradient of the SVM function. However, we noticed that this tracking approach can easily get stuck in local minima while particle filters recover due to the multiple hypotheses.

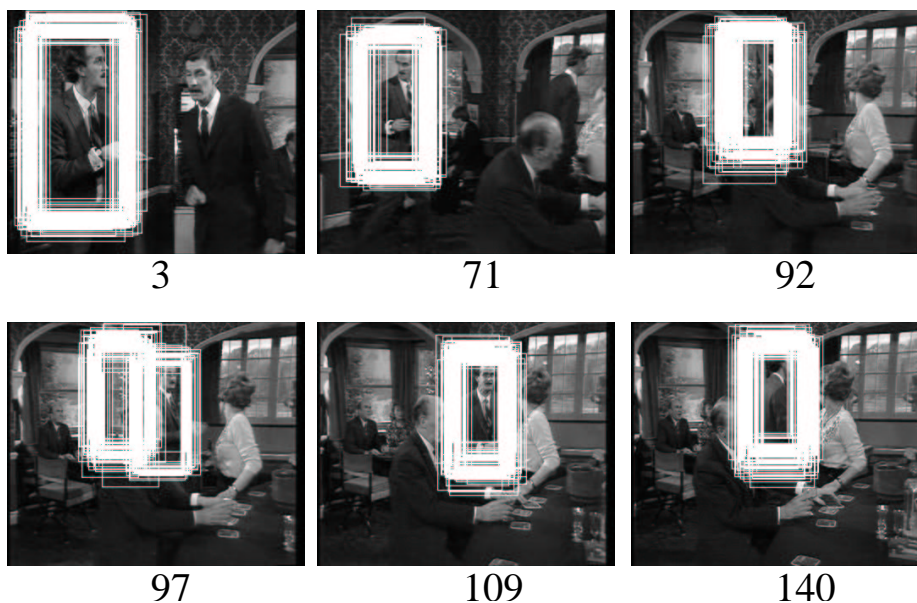


Fig. 9. Tracking Basil.

8 Summary and Conclusions

We have presented an approach for the detection, recognition, tracking and pose estimation of people in video sequences with several new contributions which are

- improved feature selection and dimensionality reductions for the design of detectors,
- using a tree based classifier scheme to detect people and discriminate between different individuals and poses, whereby all the parameters are learned automatically,
- combining a wavelet and SVM based detector with particle filtering,

Results have been shown for analyzing sitcoms, which have the advantages that only a small number of locations and persons are used. Based on this convincing results further applications in surveillance, human-computer interfaces etc. can be envisioned.

Indeed, the training data have to be sufficiently discriminative in order to avoid that the detectors respond to the background. When one would want to discriminate between a larger quantity of poses it is advisable to investigate into a hierarchical feature selection as the detection rate is low otherwise. However, it is not obvious how to combine the poses so that a tree structure can be formed. The hierarchical pose estimation is part of our current studies.

Our future work will focus on the integration of the tracking with pose estimation, for example by including the pose in the state vector or by using model switching [6]. Additionally, we plan to recognize gestures from the poses. For example by clustering poses or by using the tracking framework once again to detect a gesture on the learned sequence of different poses.

Acknowledgment

The investigations reported in this contribution have been partially supported by the European Union (FP5-project ‘CogViSys’, IST-2000-29404).

References

1. M. C. Burl, T. K. Leung, and P. Perona. Face localization via shape statistic. In *International Workshop on Automatic Face and Gesture Recognition*, pages 154–159, 1995.
2. V. Depoortere, J. Cant, B. Van den Bosch, J. De Prins, R. Fransens, and L. Van Gool. Efficient pedestrian detection: A test case for svm based categorization. In *Cognitive Vision Workshop 02*, Zurich, Switzerland, September 2002.
3. T. Evgeniou, M. Pontil, C. Papageorgiou, and T. Poggio. Image representations for object detection using kernel classifiers. In *Asian Conference on Computer Vision*, pages 687–692, 2000.
4. J. Giebel and D. M. Gavrilu. Multimodal shape tracking with point distribution models’. In *The Annual Symposium for Pattern Recognition of the DAGM*, pages 1–8, 2002.
5. M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
6. M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *International Conference on Computer Vision*, pages 107–112, 1998.
7. M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *International Conference on Computer Vision*, pages 34–41, 2001.
8. C. Nakajima, M. Pontil, and T. Poggio. People recognition and pose estimation in image sequences. In *International Joint Conference on Neural Networks*, 2000.

9. E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *International Conference on Computer Vision*, pages 130–136, 1997.
10. E. Osuna and F. Girosi. Reducing the run-time complexity of support vector machines. In *International Conference on Pattern Recognition*, 1998.
11. C. Papageorgiou, T. Evgeniou, and T. Poggio. A trainable pedestrian detection system. In *IEEE Intelligent Vehicle Symposium*, pages 241–246, 1998.
12. H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, Jan 1998.
13. H. Schneiderman and T. Kanade. A statistical approach to 3D object detection applied to faces and cars. In *Proc. Computer Vision and Pattern Recognition*, pages 746–751, 2000.
14. R. Verma, C. Schmid, and K. Mikolajczyk. Face detection and tracking in a video by propagating detection probabilities. *Pattern Analysis and Machine Intelligence*, pages 1215–1227, 2003.
15. P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, pages 137–154, 2004.
16. G. Yang and T. Huang. Human face detection in a complex background. *Pattern Recognition*, 1994.