International Journal of Computer Vision 38(1), 15–33, 2000

A Trainable System for Object Detection

CONSTANTINE PAPAGEORGIOU AND TOMASO POGGIO
Center for Biological and Computational Learning, Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA
cpapa@ai.mit.edu
tp@ai.mit.edu

Abstract. This paper presents a general, trainable system for object detection in unconstrained, cluttered scenes. The system derives much of its power from a representation that describes an object class in terms of an overcomplete dictionary of local, oriented, multiscale intensity differences between adjacent regions, efficiently computable as a Haar wavelet transform. This example-based learning approach implicitly derives a model of an object class by training a support vector machine classifier using a large set of positive and negative examples. We present results on face, people, and car detection tasks using the same architecture. In addition, we quantify how the representation affects detection performance by considering several alternate representations including pixels and principal components. We also describe a real-time application of our person detection system as part of a driver assistance system.

Keywords: computer vision, machine learning, pattern recognition, people detection, face detection, ear detection

1. Introduction

As the amount of image and video information available increases, robust, configurable object detection systems for managing this data will become indispensable. There has been an explosion in the amount of information presented on the Internet as it is quickly transitions from a text-based medium to one of image and video content; object detection systems will be used to search through the growing number of image and video databases. This technology will also be used in surveillance applications, driver assistance systems, and as front ends to recognition systems.

This paper addresses the problem of object and pattern detection in static images of unconstrained, cluttered scenes. We contrast detection with the problem of recognition, where the goal is to identify specific instances of a class. A face *detection* system knows how to differentiate faces from "everything else", while a face *recognition* system knows the difference between my face and other faces. The detection of real-world objects of interest, such as faces, people, and cars, poses challenging problems: these objects are difficult to model with significant variety in color and texture, the backgrounds against which the objects lie are often complex and cluttered, and characteristics like lighting, size, and number of objects cannot be accounted for in any but the most contrived situations.

Our technique uses a descriptive model of an object class that is rich enough to effectively model any of the possible shapes, poses, colors, and textures of an object. At the same time, the technique is general enough that it can easily be transfered to a new class of objects.

The system derives much of its power from a new representation that describes an object class in terms of a large set of local oriented intensity differences between adjacent regions; this representation is efficiently computable as a Haar wavelet transform. Images are mapped from the space of pixels to that of an overcomplete dictionary of Haar wavelet features that provides a rich description of the pattern. This representation is able to capture the structure of the class of objects we would like to detect while ignoring the noise inherent in the images. The use of an overcomplete dictionary is inspired by image reconstruction techniques; our goal is to do classification and, to this end, the

overcomplete dictionary provides us with a richer expressive language in which we can compare complex patterns.

We will be using an example-based learning approach where a model of an object class is derived implicitly from a training set of examples. In this way, specializing this general system to a specific domain involves plugging in a new set of training data without modifying the core system or handcrafting a new model. The specific learning engine we use is a support vector machine (SVM) classifier. This classification technique has a number of properties that make it particularly attractive and has recently received much attention in the machine learning community.

There is a large body of previous work in object detection; of particular relevance to this paper is the work on face detection in static images. Recently, example-base approaches have achieved a high degree of success in this field (Sung and Poggio, 1994; Moghaddam and Pentland, 1995; Rowley et al., 1998; Vaillant et al., 1994; Osuna et al., 1997b). These view-based approaches can handle detecting faces in cluttered scenes, and have shown a reasonable degree of success when extended to handle non-frontal views. In contrast to face detection, detecting people in static images has, until now, not been successfully tackled. Current people detection systems (Wren et al., 1995; Haritaoglu et al., 1998; Heisele and Wohler, 1998; McKenna and Gong, 1997; Shio and Sklansky, 1991; Rohr, 1993; Hogg, 1983) typically assume any of several restrictive assumptions, that the people are moving, there is a static background with a fixed camera, implement tracking and not true detection, use hand-crafted models, or they make assumptions on the number of people in the scene. In Forsyth and Fleck (1997, 1998), they describe a system that uses color, texture, and geometry to localize horses and naked people in static images. The system is mainly targeted towards retrieving images with a single object of interest. Methods of learning these "body plans" of hand coded hierarchies of parts from examples are described in Forsyth and Fleck (1997). Our system makes none of these assumptions and results in a highly robust people detection technique for static images. Car detection is also a domain receiving increased attention; (Bregler and Malik, 1996) describe a system using mixtures of experts on second order Gaussian features to identify different classes of cars (detection has been subsumed) and (Lipson, 1996; Lipson et al., 1997) describes a system that uses a deformable template for side view car detection. In Beymer et al. (1997), they present a traffic monitoring system that has a car detection module that locates corner features in highway sequences and groups features for single cars together by integrating information over time. The system of Betke et al. (1997) and Betke and Nguyen (1998) uses corner features and edge maps combined with template matching to detect cars in highway video scenes.

This paper describes our general framework for object detection in the context of face, people, and car detection. We provide an in-depth description of our core system in Section 2, along with details on wavelets (Section 2.1), our particular dictionary of wavelet features (Section 2.2), and the support vector machine classification technique (Section 2.3). In Section 3, we compare and contrast wavelets with other possible representations, including pixels and principal components. A real-time implementation of our people detection system as part of a driver assistance system is described in Section 4. We conclude with related areas that we are currently pursuing and directions for future work.

2. Architecture and Representation

The architectural overview of our system is provided in Fig. 1 as applied to the task of people detection and shows the training and testing phases. In the training step, the system takes as input 1) a set of images of the object class that have been aligned and scaled so that they are all in approximately the same position and the same size and 2) a set of patterns that are not in our object class. An intermediate representation that encapsulates the important information of our object class is computed for each of these patterns, yielding a set of positive and negative feature vectors. These feature vectors are used to train a pattern classifier to differentiate between in-class and out-of-class patterns.

In the testing phase, we are interested in detecting objects in out-of-sample images. The system slides a fixed size window over an image and uses the trained classifier to decide which patterns show the objects of interest. At each window position, we extract the same set of features as in the training step and feed them into our classifier; the classifier output determines whether or not we highlight that pattern as an in-class object. To achieve multiscale detection, we iteratively resize the image and process each image size using the same fixed size window.

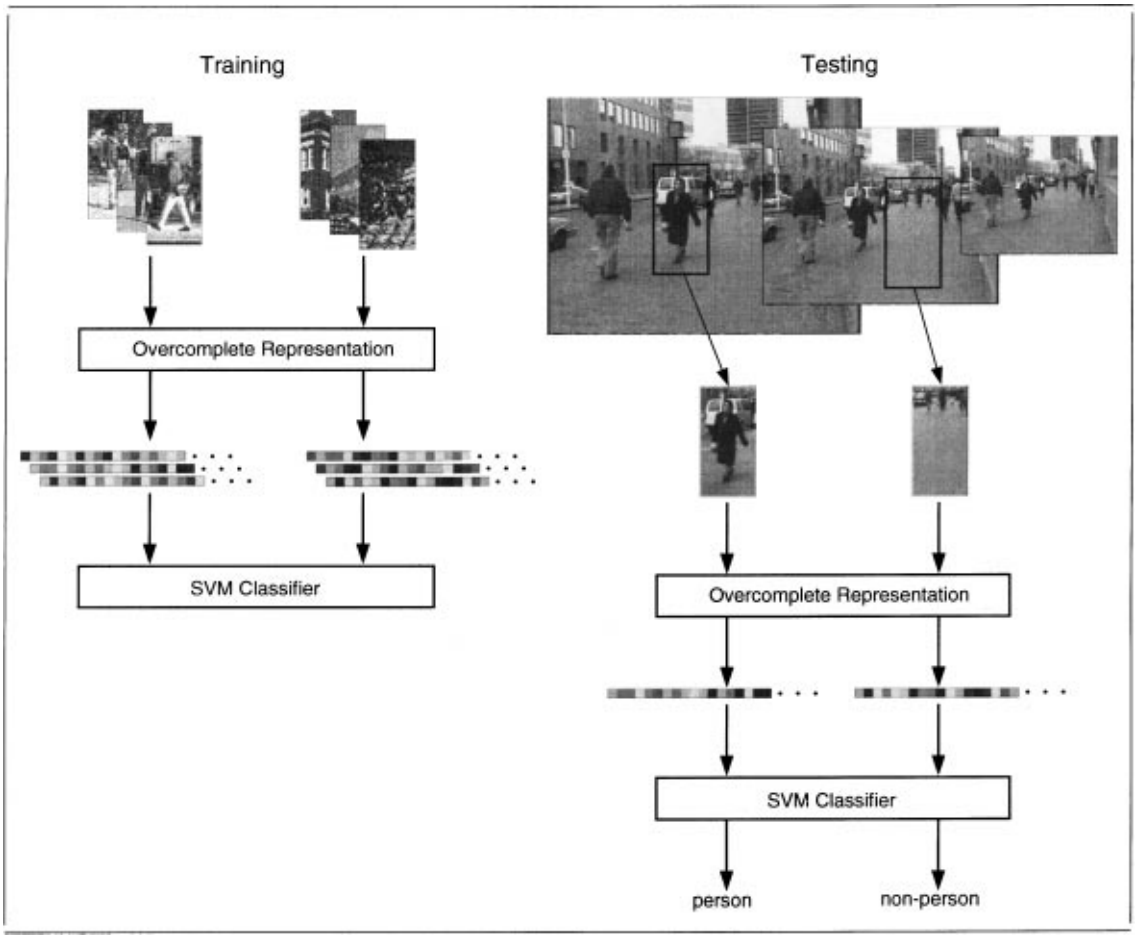

Figure 1. The training and testing phases of our system.

This section addresses the key issues in the development of our trained pattern classifier: the representation and the learning engine.

2.1. Wavelets

The ultimate goal in choosing a representation for an object detection system is finding one that yields high inter-class variability, while at the same time achieving low intra-class variability. Since object classes like people can be quite complex, this is a nontrivial task. To encode the visual structure of an object class, our representation must identify features at a resolution where there will be some consistency throughout the object class, while at the same time ignoring noise. The representation we use, Haar wavelets, identifies local, oriented intensity difference features at different scales and is efficiently computable. The Haar wavelet is perhaps the simplest such feature with finite support. We transform our images from pixel space to the space of wavelet coefficients, resulting in an overcomplete dictionary of features that are then used as training for a classifier.

This section describes the underlying representation that we use for extracting object features, the Haar wavelet. We also describe a denser (redundant) transform that we use to provide a richer feature set and to achieve the spatial resolution we need to accomplish detection.

2.1.1. The Haar Wavelet. Wavelets provide a natural mathematical structure for describing our patterns; a more detailed treatment can be found in Mallat (1989). These vector spaces form the foundations of

the concept of a multiresolution analysis. We formalize the notion of a multiresolution analysis as the sequence of approximating subspaces $V^0 \subset V^1 \subset V^2 \subset \ldots V^j \subset V^{j+1} \ldots$; the vector space V^{j+1} can describe finer details than the space V^j, but every element of V^j is also an element of V^{j+1}. A multiresolution analysis also postulates that a function approximated in V^j is characterized as its orthogonal projection on the vector space V^j.

As a basis for the vector space V^j, we use the *scaling functions*,

$$\phi_i^j = \sqrt{2^j}\phi(2^j x - i), \quad i = 0, \ldots, 2^j - 1, \qquad (1)$$

where, for our case of the Haar wavelet,

$$\phi(x) = \begin{cases} 1 & \text{for } 0 \le x < 1 \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

Next we define the vector space W^j that is the orthogonal complement of two consecutive approximating subspaces, $V^{j+1} = V^j \oplus W^j$. The W^j are known as *wavelet subspaces* and can be interpreted as the subspace of "details" in increasing refinements. The wavelet space W^j is spanned by a basis of functions,

$$\psi_i^j = \sqrt{2^j}\psi(2^j x - i), \quad i = 0, \ldots, 2^j, \qquad (3)$$

where for Haar wavelets,

$$\psi(x) = \begin{cases} 1 & \text{for } 0 \le x < \frac{1}{2} \\ -1 & \text{for } \frac{1}{2} \le x < 1 \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

The sum of the wavelet functions form an orthonormal basis for $L_2(R)$. It can be shown (under the standard conditions of multiresolution analysis) that all the scaling functions can be generated from dilations and translations of one scaling function. Similarly, all the wavelet functions are dilations and translations of the mother wavelet function. Figure 2(a) shows the scaling

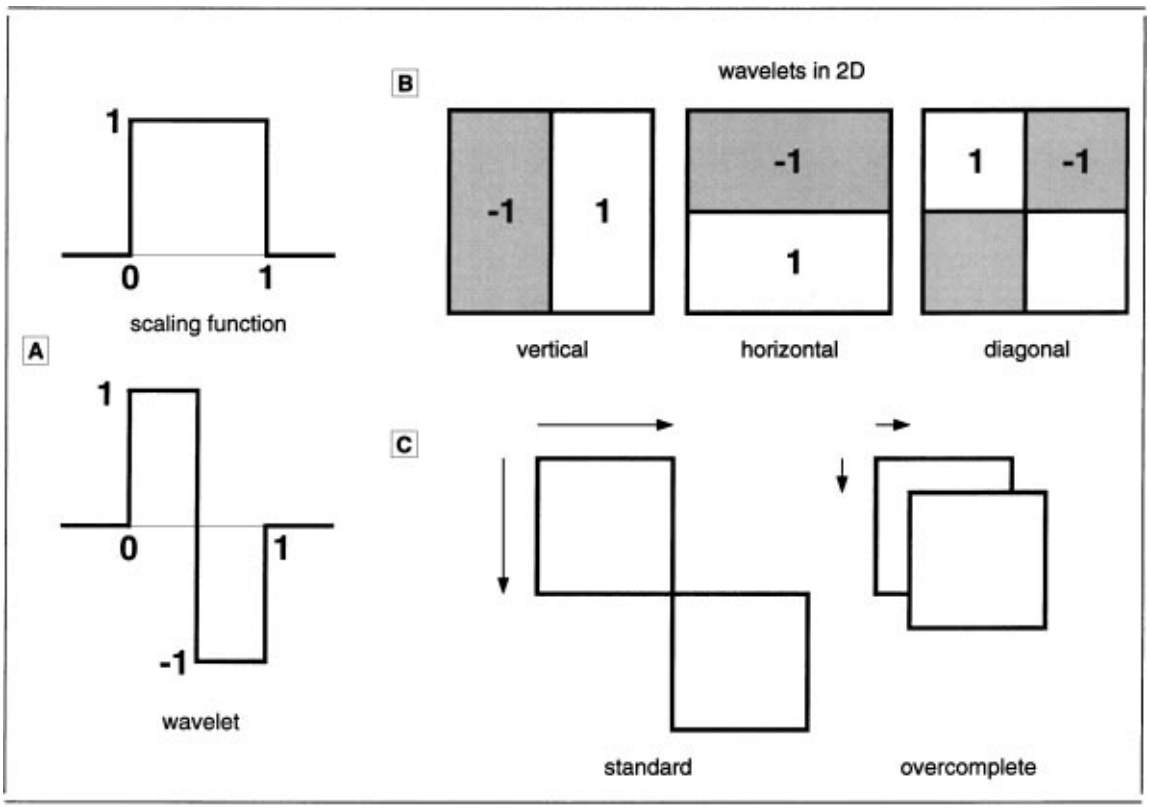

Figure 2. The Haar wavelet framework; (a) the Haar scaling function and wavelet, (b) the three types of 2-dimensional non-standard Haar wavelets: vertical, horizontal, and diagonal, and (c) the shift in the standard transform as compared to our quadruply dense shift resulting in an overcomplete dictionary of wavelets.

and wavelet functions. The approximation of some function $f(x)$ in the space V^j is found to be:

$$A_j(f) = \sum_{k \in Z} \overbrace{\langle f(u), \phi_k^j(u) \rangle}^{\lambda_{j,k}} \phi_k^j(x) \tag{5}$$

where we let the inner product be denoted by $\lambda_{j,k}$ for future use. Similarly, the projection of $f(x)$ on W^j is:

$$D_j(f) = \sum_{k \in Z} \overbrace{\langle f(u), \psi_k^j(u) \rangle}^{\gamma_{j,k}} \psi_k^j(x) \tag{6}$$

where, in this case, the inner product is denoted by $\gamma_{j,k}$.

The structure of the approximating and wavelet subspaces leads to an efficient cascade algorithm for the computation of the scaling coefficients, $\lambda_{j,k}$, and the wavelet coefficients, $\gamma_{j,k}$:

$$\lambda_{j,k} = \sum_{n \in Z} h_{n-2k} \lambda_{j+1,n} \tag{7}$$

$$\gamma_{j,k} = \sum_{n \in Z} g_{n-2k} \lambda_{j+1,n} \tag{8}$$

where $\{h_i\}$ and $\{g_i\}$ are the filter coefficients corresponding to the scaling and wavelet functions. Using this construction, the approximation of a function $f(x)$ in the space V^j is:

$$A_j(f) = \sum_{n \in Z} \lambda_{j,k} \sqrt{2^j} \phi(2^j x - k) \tag{9}$$

Similarly, the approximation of $f(x)$ in the space W^j is:

$$D_j(f) = \sum_{n \in Z} \gamma_{j,k} \sqrt{2^j} \psi(2^j x - k) \tag{10}$$

Since we use the Haar wavelet, the corresponding filters are: $h = \{\ldots, 0, \frac{1}{2}, \frac{1}{2}, 0, 0, \ldots\}$ and $g = \{\ldots, 0, -\frac{1}{2}, \frac{1}{2}, 0, 0, \ldots\}$. The scaling coefficients are simply the averages of pairs of adjacent coefficients in the coarser level while the wavelet coefficients are the differences.

It is important to observe that the discrete wavelet transform (DWT) performs *downsampling* or *decimation* of the coefficients at the finer scales since the filters h and g are moved in a step size of 2 for each increment of k.

2.1.2. 2-Dimensional Wavelet Transform. The natural extension of wavelets to 2D signals is obtained by taking the tensor product of two 1D wavelet transforms. The result is the three types of wavelet basis functions shown in Fig. 2. The first type of wavelet is the tensor product of a wavelet by a scaling function, $\psi(x, y) = \psi(x) \otimes \phi(y)$; this wavelet encodes a difference in the average intensity along a vertical border and we will refer to its value as a *vertical* coefficient. Similarly, a tensor product of a scaling function by a wavelet, $\psi(x, y) = \phi(x) \otimes \psi(y)$, is a *horizontal* coefficient, and a wavelet by a wavelet, $\psi(x, y) = \psi(x) \otimes \psi(y)$, is a *diagonal* coefficient since this wavelet responds strongly to diagonal boundaries.

Since the wavelets that the standard transform generates have irregular support, we use the non-standard 2D DWT where, at a given scale, the transform is applied to each dimension sequentially before proceeding to the next scale (Stollnitz et al., 1994). The results are Haar wavelets with square support at all levels, shown in Fig. 2(b).

2.1.3. Quadruple Density Transform. For the 1D Haar transform, the distance between two neighboring wavelets at level n (with support of size 2^n) is 2^n. To obtain a denser set of basis functions that provide a richer model and finer spatial resolution, we need a set of redundant basis functions, or an overcomplete *dictionary*, where the distance between the wavelets at level n is $\frac{1}{4}2^n$ (Fig. 2c). The straightforward approach of shifting the signal and recomputing the DWT will not generate the desired dense sampling. Instead, this can be achieved by modifying the DWT. To generate wavelets with *double density*, where wavelets of level n are located every $\frac{1}{2}2^n$ pixels, we simply do not downsample in Eq. (8). To generate the *quadruple density* dictionary, first, we do not downsample in Eq. (7), giving us double density scaling coefficients. Next, we calculate double density wavelet coefficients on the two sets of scaling coefficients—even and odd—separately. By interleaving the results of the two transforms we get quadruple density wavelet coefficients. For the next level $(n+1)$, we keep only the even scaling coefficients of the previous level and repeat the quadruple transform on this set only; the odd scaling coefficients are dropped off. Since only the even coefficients are carried along at all the levels, we avoid an "explosion" in the number of coefficients, yet obtain a dense and uniform sampling of the wavelet coefficients at all the levels. As with the regular DWT, the time complexity is $O(n)$ in

the number of pixels n. The extension of the quadruple density transform to 2D is straightforward.

2.2. The Wavelet Representation

The Haar transform provides a multiresolution representation of an image with wavelet features at different scales capturing different levels of detail; the coarse scale wavelets encode large regions while the fine scale wavelets describe smaller, local regions. The wavelet coefficients preserve all the information in the original image, but the coding of the visual information differs from the pixel-based representation in two significant ways.

First, the wavelets encode the difference in average intensity between local regions along different orientations, in a multiscale framework. Constraints on the values of the wavelets can express visual features of the object class; strong response from a particular wavelet indicates the presence of an intensity difference, or boundary, at that location in the image while weak response from a wavelet indicates a uniform area.

Second, the use of an overcomplete Haar basis allows us to propagate constraints between neighboring regions and describe complex patterns. The quadruple density wavelet transform provides high spatial resolution and results in a rich, overcomplete dictionary of features. Instead of quadruple density wavelets, it is possible to use just the double density wavelets that overlap by 50%; we expect that the quadruple density version should give us better performance, though this is an untested assumption.

Our main motivation for using wavelets is that they capture visually plausible features of the shape and interior structure of objects that are invariant to certain transformations. The result is a compact representation where dissimilar example images from the same object class map to similar feature vectors.

With a pixel representation, what we would be encoding are the actual intensities of different parts of the patterns—a simple example makes it clear that this encoding does not capture the important features for detection. Take, for instance, our example of two data points of the same class where one is a dark body on a white background and the other is a white body on a dark background. With an intensity based representation (like pixels), each of these examples maps to completely different feature vectors. A representation that encodes local, oriented, intensity differences (like Haar wavelets) would yield similar feature vectors where the features corresponding to uniform regions are zero and those corresponding to boundaries are non-zero. In fact, since in our representation we encode only the magnitude of the intensity difference, the feature vectors for this simple two example case would be identical.

We do not use all the very fine scales of wavelets as features for learning since these scales capture high frequency details that do not characterize the class well; for instance, in the case of people, the finest scale wavelets may respond to checks, stripes, and other detail patterns, all of which are not features that are characteristic to the entire class. Similarly, the very coarse scale wavelets are not used as features for learning since their support will be as large as the object and will therefore not encode useful information. So, for the object detection system we have developed, we throw out the very fine and very coarse wavelets and only use 2 medium scales of wavelets as features for learning. These scales depend on the object class and the size of the training images and are chosen a priori.

In the following sections, we show how our wavelet representation applies to faces, people, and cars; this coding of local intensity differences at several scales provides a flexible and expressive representation that can characterize each of these complex object classes. Furthermore, the wavelet representation is computationally efficient for the task of object detection since we do not need to compute the transform for each image region that is examined but only once for the whole image and then process the image in the space of wavelets.

2.2.1. Analyzing the Face Class. For the face class, we have a training set of 2,429 gray-scale images of faces—this set consists of a core set of faces, with some small angular rotations to improve generalization—and 24,730 non-face patterns. These images are all scaled to the dimensions 19×19 and show the face from above the eyebrows to below the lips; typical images from the database are shown in Fig. 3. Databases of this size and composition have been used extensively in face detection (Sung, 1995; Rowley et al., 1998; Osuna et al., 1997a). For the size of patterns our face system uses, we have at our disposal wavelets of the size 2×2, 4×4, 8×8, and 16×16. Instead of using the entire set of wavelets, we a priori limit the dictionary to contain the wavelets of scales 2×2 and 4×4, since coarser features do not contain significant information for detection purposes. At the scale 4×4 pixels, there are 17×17 features in quadruple density for each wavelet class and at 2×2 pixels there are 17×17

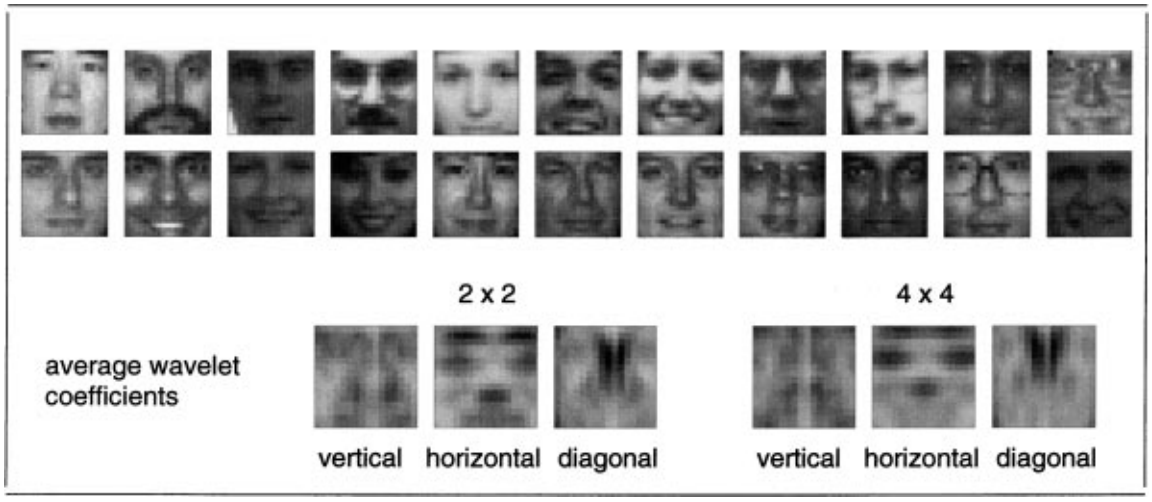

Figure 3. Example images from the database of faces used for training and the corresponding ensemble average features. The training images are gray level of size 19 × 19 pixels. The average feature values are coded in gray level and are displayed in their proper spatial configuration. Features whose values are close to the average value of one are coded as gray, coefficients that are above the average are darker and those below the average are lighter. We can observe strong features in the eye areas and the nose. The cheek area is an area of almost uniform intensity, that is, the coefficients in the cheek regions have below average values.

features in double density for each class, for a total of 1,734 coefficients.

The raw value of a coefficient may not necessarily be indicative of a boundary—a weak coefficient in a relatively dark image may still indicate the presence of an intensity difference that is significant for the purposes of classification. To reduce these effects on the features used for classification, we normalize a coefficient's value against the other coefficients in the same area. For the normalization step, we compute the average of each wavelet's class ($\{vertical, horizontal, diagonal\} \times \{2, 4\}$) over the current pattern and divide the wavelet response at a certain spatial location by its corresponding class average. We calculate the averages separately for each class since the power distribution between the different classes may vary.

After the normalization, the average value of a coefficient for random patterns should be 1. Three classes of feature magnitudes will emerge: ensemble average values much larger than 1 indicate strong intensity difference features that are consistent along all the examples, values that are much less than 1 indicate consistent uniform regions, and values that are close to 1 are associated with inconsistent features, or random patterns.

To visualize the detected face features we code the ensemble average of the wavelet coefficients using gray level and draw them in their proper spatial layout in Fig. 3. Coefficients with values close to 1 are plotted in gray, those with values larger than 1 are darker, and those with values less than 1 are lighter. It is interesting to observe the emerging patterns in the facial features. The vertical wavelets capture the sides of the nose, while the horizontal wavelets capture the eye sockets, eyebrows, and tip of the nose. Interestingly, the mouth is a relatively weak feature compared to the others. The diagonal wavelets respond strongly to the endpoint of facial features.

2.2.2. Analyzing the People Class. For learning the people class, we have collected a set of 1,800 color images of people in different poses (Fig. 4) and use the 1,800 mirror images as well and 16,726 non-people patterns. All of the images are normalized to the dimensions 128 × 64 and the people images are aligned such that the bodies are centered and approximately the same size (the distance from the shoulders to feet is about 80 pixels).

As in the case of faces, to code features at appropriate scales for people detection—scales at which we expect relevant features of people to emerge—we restrict the system to the wavelets at scales of 32 × 32 pixels (15 × 5 features for each orientation) and 16 × 16 pixels (29 × 13 for each orientation).

In our people detection system, our training database is of color images. For a given pattern, we compute the quadruple density Haar transform in each color channel

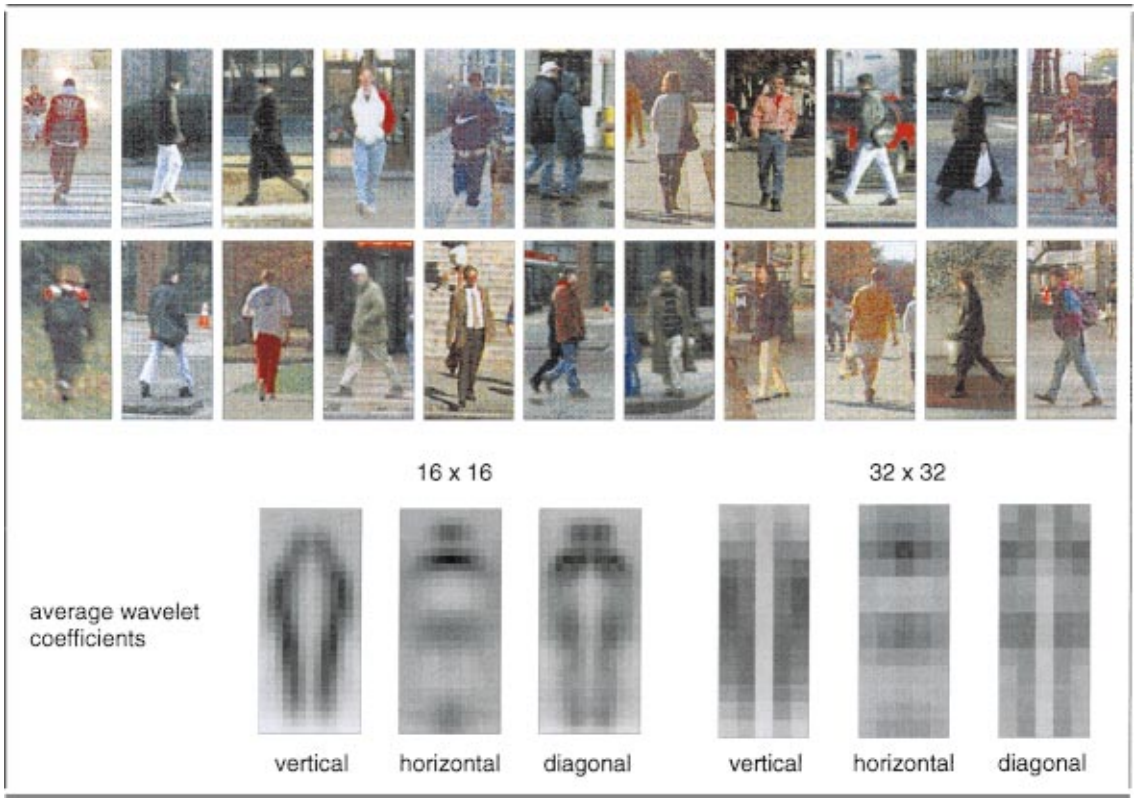

Figure 4. Example images from the database of people used for training and the corresponding ensemble average features. The training images are 128 × 64 color images. As in the case of faces, the average feature values are coded in gray level and are displayed in their proper spatial configuration. The wavelets identify the significant visual boundary information present in the people images: the vertical wavelets respond to the sides of the body, the horizontal wavelets respond to the top of the head and the shoulders, and the diagonal wavelets respond to the head, shoulders, hands, and feet.

(RGB) separately and take as the coefficient value at a specific location and orientation the one largest in absolute value among the three channels, providing the system with the most visually significant information. This technique maps the original color image to a pseudo-color channel that gives us 1,326 wavelet coefficients, the same number as if we had been using gray level images.

To visualize the patterns that emerge using this wavelet representation for people, we can code the average values of the coefficients in gray level and display them in the proper spatial layout as we did for the faces. Figure 4 shows each average wavelet displayed as a small square where features close to 1 are gray, stronger features are darker, and weaker features are lighter. As with faces, we observe that each class of wavelet coefficients is tuned to a different type of structural information. The vertical wavelets capture the sides of the people. The horizontal wavelets respond to the shoulders and to a weaker belt line. The diagonal wavelets are tuned to "corner features", i.e. the shoulders, hands, and feet. The 16×16 scale wavelets provide fine spatial resolution of the body's overall shape and smaller scale details, such as the head and extremities, are clearly evident.

2.2.3. Analyzing the Car Class. The car detection system uses a database of 516 frontal and rear color images of cars, normalized to 128×128 and aligned such that the front or rear bumper is 64 pixels across. For training, we use the mirror images as well for a total of 1,032 positive patterns and 5,166 negative patterns. The two scales of wavelets we use for detection are 16×16 and 32×32. Like the processing for people, we collapse the three color channel features into a single channel by using the maximum wavelet response

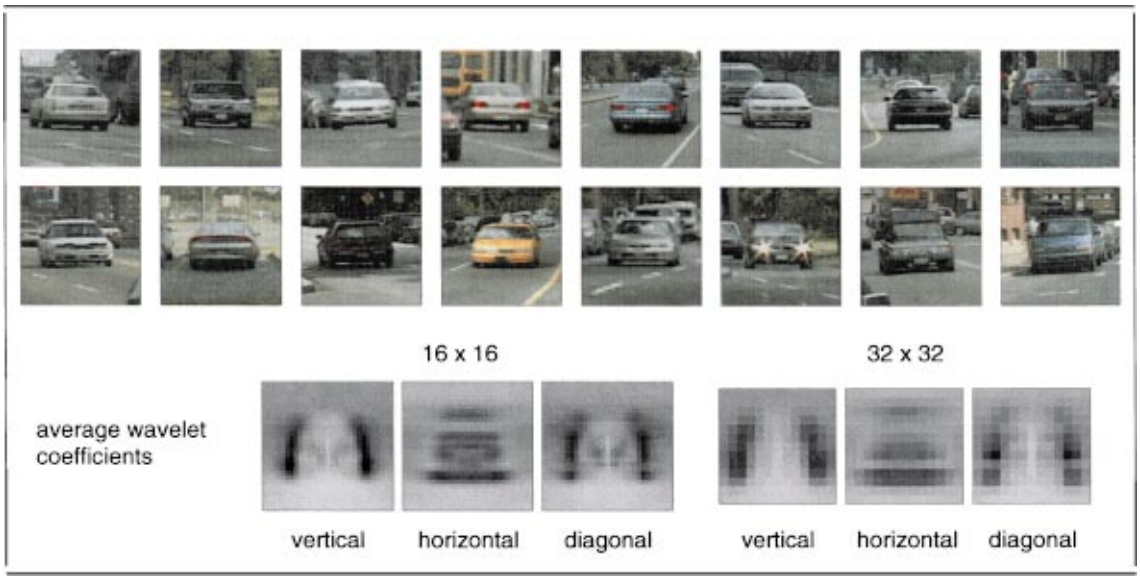

Figure 5. Example images from the database of cars used for training and the corresponding ensemble average features. The training images are 128×128 color images. The gray level coding of the average feature values show that the wavelets respond to the significant visual characteristics of cars: the vertical wavelets respond to the sides of the car, the horizontal wavelets respond to the roof, underside, top of the grille and bumper area, and the diagonal wavelets respond to the corners of the car's body. At the scale 16×16, we can even see evidence of what seems to be license plate and headlight structures in the average responses.

of each channel at a specific location, orientation, and scale. This gives us a total of 3,030 wavelet features that are used to train the SVM.

The average wavelet feature values are coded in gray level in Fig. 5. As with both faces and cars, much of the characteristic structure of cars is evident in these averages.

2.2.4. Discussion. Comparing the database of people (Fig. 4) to the database of faces (Fig. 3) illustrates an important fundamental difference in the two classes. In the case of faces, there are clear patterns within the face, consisting of the eyes, nose and mouth; these patterns are common to all the examples. This is not the case with full body images of people. The people do *not* share any common color or texture. Furthermore, the people images have a lot of spurious details such as jackets, ties, and bags. On the other hand, we would expect that people can be characterized quite well by their fairly similar overall body shape, or "silhouette". Our approach treats these two cases where there is different underlying information content in the object classes in a uniform manner. Frontal and rear views of cars have both a certain amount of common interior structure (top of grille, license plates, headlights) as well as fairly uniform outer boundaries; we will also see that cars are handled equally well in this framework.

There is certain a priori knowledge embedded in our choice of the wavelets. The use of the absolute value of the coefficient may be essential in the case of people since the direction of the intensity difference of a certain feature's orientation is not important; a dark body against a light background and a light body against a dark background should be represented as having the same information content. Furthermore, we compute the wavelet transform for a given pattern in each of the three color channels and then, for a wavelet at a specific location and orientation, we use the one that is largest in magnitude amongst the three channels. This is based on the observation that there is little consistency in color between different people and allows the system to key off of the most visually significant features. This same prior assumption is used for our car detection system as well.

Once we have generated the feature vectors for an object class and have done the same for a set of images not in our object class, we use a learning algorithm that learns to differentiate between the two classes. The particular learning engine we use is a support vector machine, described below.

2.3. Support Vector Machine Classification

The second key component of our system is the use of a trainable pattern classifier that learns to differentiate between patterns in our object class and all other patterns. In general terms, these supervised learning techniques rely on having a set of labeled example patterns from which they derive an implicit model of the domain of interest. The particular learning engine we use is a support vector machine (SVM) classifier.

Support vector machines (SVM) is a technique to train classifiers that is well-founded in statistical learning theory; for details, see Vapnik (1995), Burges (1998) and Vapnik (1998). One of the main attractions of using SVMs is that they are capable of learning in *high-dimensional spaces* with very few training examples. They accomplish this by minimizing a bound on the empirical error and the complexity of the classifier, at the same time.

This concept is formalized in the theory of uniform convergence in probability:

$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \Phi\left(\frac{h}{\ell}, \frac{-\log(\eta)}{\ell}\right) \qquad (11)$$

with probability $1-\eta$. Here, $R(\alpha)$ is the expected risk, $R_{\text{emp}}(\alpha)$ is the empirical risk, ℓ is the number of training examples, h is the VC dimension of the classifier that is being used, and $\Phi(\cdot)$ is the VC confidence of the classifier. Intuitively, what this means is that the uniform deviation between the expected risk and empirical risk decreases with larger amounts of training data ℓ and increases with the VC dimension h. This leads us directly to the principle of structural risk minimization, whereby we can attempt to minimize at the same time both the actual error over the training set and the complexity of the classifier; this will bound the generalization error as in Eq. (11). It is exactly this technique that support vector machines approximate.

This controling of both the training set error *and* the classifier's complexity has allowed support vector machines to be successfully applied to very high dimensional learning tasks; (Joachims, 1997) presents results on SVMs applied to a 10,000 dimensional text categorization problem and (Osuna et al., 1997b) show a 283 dimensional face detection system.

The support vector machine algorithm formulates the training problem as one that finds, among all possible separating surfaces, the one that maximizes the distance between the closest elements of the two classes. In practice, this is determined through solving a quadratic programming problem.

Using the SVM formulation, the general form of the decision function for a point $\mathbf{x}$ is:

$$f(\mathbf{x}) = \theta\left(\sum_{i=1}^{\ell} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right) \qquad (12)$$

where ℓ is the number of training data points, α_i are Lagrange parameters obtained in the optimization step, and $\theta(\cdot)$ is a threshold function. The kernel $K(\cdot,\cdot)$ defines a dot product between projections of the arguments in some feature space; it is in this (typically high dimensional) feature space that a separating hyperplane is found. Different kernels induce different types of classifiers. For example, with $K(\mathbf{x},\mathbf{y}) = \mathbf{x}\cdot\mathbf{y}$ the separating surface is a hyperplane in the space of $\mathbf{x}$, $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}+1)^n$ leads to an nth degree polynomial classifier, and $K(\mathbf{x},\mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2)$ gives a Gaussian radial basis function.

Once the optimization problem has been solved, it is usually the case that most of the parameters α_i are zero. The decision surface therefore only depends on a smaller number of data points with non-zero α_i; these data points are called *support vectors*.

For our detection problem, where we use a quadratic classifier, the decision surface is:

$$f(\mathbf{x}) = \theta\left(\sum_{i=1}^{N_s} \alpha_i y_i (\mathbf{x}\cdot\mathbf{x}_i + 1)^2 + b\right) \qquad (13)$$

where i is now an index into just the N_s support vectors.

3. Experiments

In Figs. 6–8 we present examples of our trainable object detection system as applied to the domains of face, people, and car detection, respectively. We reiterate that the system makes no a priori assumption on the scene structure or the number of objects present and does not use any motion or other dynamical information. The performance of each of these particular instantiations of detection systems could easily be improved by using more training data. We have not sought to push the limits of performance in particular domains; rather, our goal has been to show that this uniform architecture for object detection leads to high performance in several domains.

The dense Haar transform captures a rich set of features that allows the SVM classifier to obtain a powerful class model; the wavelets respond to significant visual

Figure 6. Results of our face detection system on a set of out-of-sample images. A, C, E, F, G, H, I, J, K, L, M, N are from the test database of Sung & Poggio; B, D are from www.starwars.com; O is from www.corbis.com. Missed faces (B, F, I, J, K, M) are due to significant head rotations that were not present in the training data. False positives (D, E, F, N) are due to insufficient training data and can be eliminated by using more negative training data.

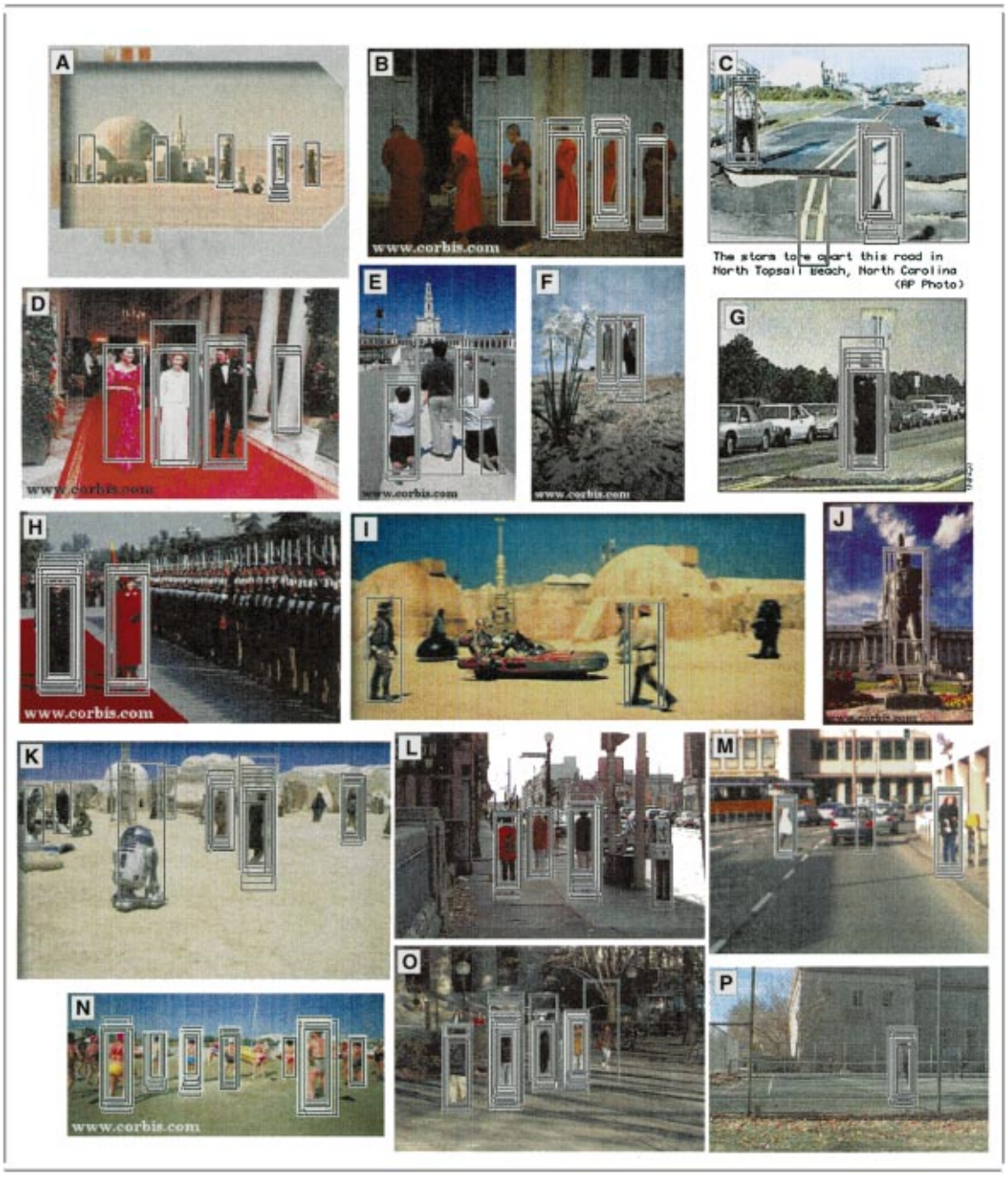

Figure 7. Results of people detection on out-of-sample images. A, I, K are from www.starwars.com; B, D, E, F, H, J, N are from www.corbis.com; C, G are from www.cnn.com; L, O, P were taken in Boston and Cambridge; M was provided by DaimlerChrysler. Missed detections are due to the person being too close to the edge of the image (B) or when the person has a body shape not represented in the training data (I). False positives often look very similar to people (A) or are due to the presence of strong intensity differences (D, E, K, L, M, O).

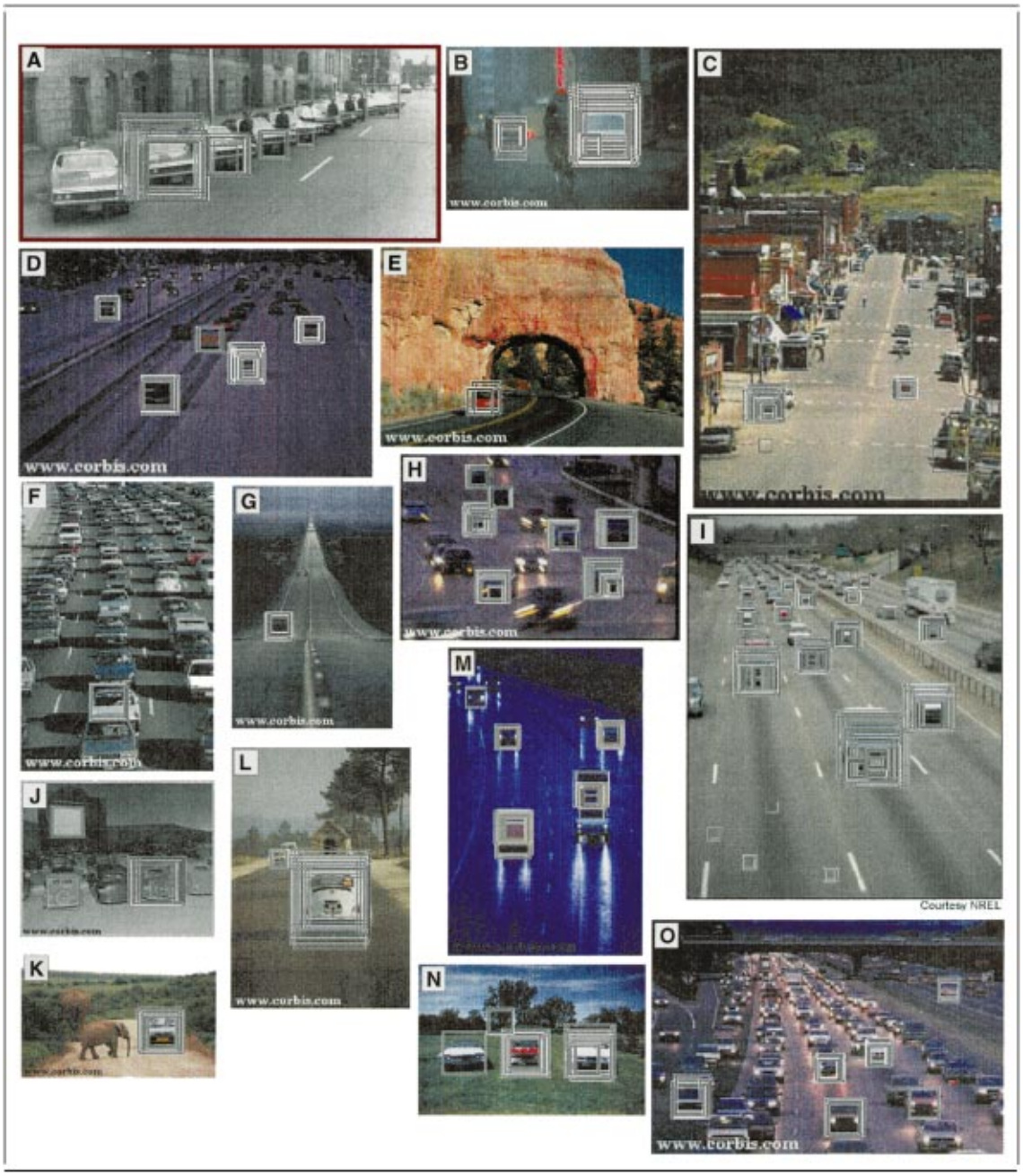

Figure 8. Results of car detection on out-of-sample images. A is from www.lewistonpd.com; B, C, D, E, F, G, H, J, K, L, M, O are from www.corbis.com; I is from www.enn.com; N is from www.foxglove.com. Missed positive examples are due to occlusions (A, F, O) or where a car is too close to the edge of the image (A). False positives (C, J, I, N) are due to insufficient training and can be eliminated with more negative training patterns.

features while smoothing away noise. This choice of features is a priori, however; this section presents the results of many tests comparing different features for object detection. There are many possible alternate representations that have been used in the literature, including pixels and PCA, and these different representations are compared in our detection framework. Another decision we made was to ignore the sign of the wavelets and use their absolute value; this is tested against the signed values. In addition, for people detection, our training set is in color; we empirically quantify the improvement in performance using color data as opposed to gray level data.

In the results presented in this section, our people detection system is trained on 1,848 positive patterns (924 frontal and rear people images and their mirror images) and 11,361 non-people patterns and tested on 123 images containing people and 794,906 non-people patterns. The face detection system is trained on 2,429 face images and 13,229 non-face patterns and tested on 105 images containing faces and 3,909,200 non-face patterns. The car detection system is trained on 1,032 frontal and rear color images of cars (516 examples and their mirrors) and 5,166 non-car patterns and tested on 90 images containing cars and 600,272 non-car patterns.

3.1. Pixels, Wavelets, PCA

Our main premise for choosing a wavelet based representation is that intensity differences between local adjacent regions contain higher quality information for the purpose of object detection than other traditional representations. Pixel representations capture the "most local" features. These have been used extensively for face detection but due to the variability in the people patterns, we would expect pixel representations to fail for people detection. At the other end of the locality spectrum are global representations like PCA which encodes a class in terms of basis functions that account for the variance in the data set. We can change the class of features to see which yields the best performance. For the people and car detection systems, we use the 1,769 overlapping 8×8 averages instead of pixels for a more fair comparison that uses similar numbers of features; furthermore, these averages are histogram equalized in the same manner as the pixel representation. For faces, we use pixels.

3.2. Signed vs. Unsigned Wavelets

The features our system uses do not contain information on the sign of the intensity gradient, but are the absolute values of the wavelet responses. With these features, we are solely describing the strength of the intensity differences. For an object class like people, where a dark body on a light background has the same information as a light body on a dark background and there is little consistency in the intensities, the sign of the gradient should not matter. On the other hand, if we consider face patterns, there is consistent information in the sign of the gradient of the intensity differences. For instance, the eyes are darker than the cheeks and the forehead and the mouth is darker than the cheeks and the chin; these types of relationships have been explored in Sinha (1994). We might expect that using the sign information ($+$ or $-$) would enhance results in this case.

3.3. Complete vs. Overcomplete

The motivation for using the overcomplete Haar wavelet representation is to provide a richer set of features over which the system will learn and, ultimately, a more accurate description of a person. We test this against the standard complete Haar representation.

3.4. Color vs. Gray Level

For color images in the case of people detection, we collapse information from the three color channels into a single pseudo-channel that maintains the strongest local intensity differences. It is intuitively obvious that color images contain much richer information than the corresponding gray-scale versions. We present experiments that quantify the inherent information content in using color images as opposed to gray level for object detection.

3.5. Faces, People, and Cars

Our ROC curves highlight the performance of the detection system as accuracy over out-of-sample data against the rate of false positives, measured as the number of false positives per pattern examined. The ROC curves that compare different representations for the face detection system are shown in Fig. 9. The representations used for face detection are raw pixels (361

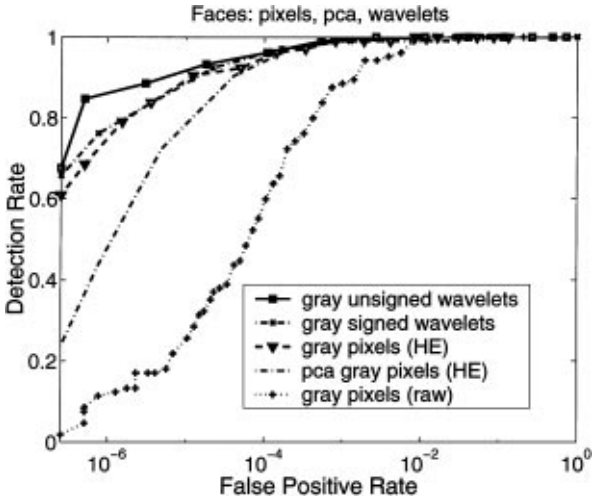

Figure 9. ROC curves for face detection comparing different features using pixel features as a benchmark.

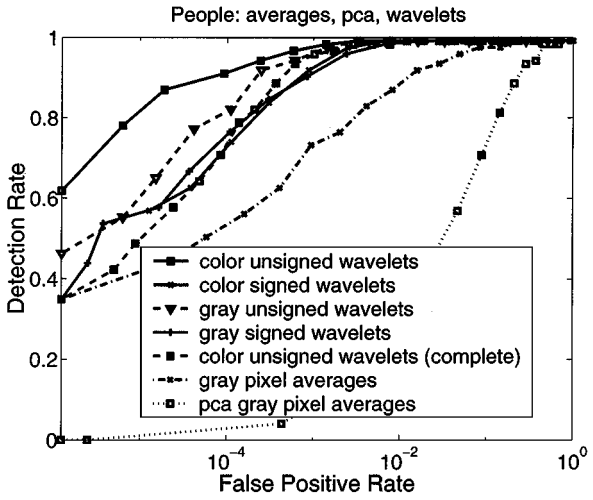

Figure 10. ROC curves for people detection comparing different features using pixel type features as a benchmark.

features), histogram equalized pixels (361 features), principal components of histogram equalized pixels (361 features), gray signed wavelets (1,740 features), and gray unsigned wavelets (1,740 features). Gray unsigned wavelets yield the best performance, while gray signed wavelets and histogram equalized gray pixels lead to the same level of performance, slightly worse than the gray unsigned wavelets; the version using principal components is less accurate than the histogram equalized pixels. That the unsigned wavelets perform better than the signed wavelets is somewhat counterintuitive; we had postulated that the sign of the wavelets contain important information for face detection since human faces have consistent patterns. Using the absolute magnitude of the wavelets may result in a representation with less variability than the signed version, while still encoding the important information for detection, allowing the classifier to find a better decision surface. To gauge the performance of the system, we can take a point on the ROC curve and translate the performance into real image terms. For instance, for a 90% detection rate, we must tolerate 1 false positive for every 100,000 patterns processed, or approximately 1 false positive per image.

The ROC curves for the people detection system are shown in Fig. 10. Here, using all the color features performs the best, where for instance a 90% detection rate leads to 1 false positive for every 10,000 patterns that are processed (about 3 false positives per image). Gray level wavelets perform significantly better than the corresponding gray level averages; here, unlike the case of face detection, the raw pixel values do not characterize the object class well. When we use the 1,769 PCAs of the 8×8 averages the performance is significantly worse. Figure 10 also supports our hypothesis on the necessity of an overcomplete versus a complete representation; the system starting from a complete representation (120 color wavelets) underperforms all of the systems based on the overcomplete representation. The signed versions of both the color and gray level wavelets perform worse than their unsigned versions. We hypothesize that the reason is the same as the case for faces, that the unsigned versions result in more compact representations over which it is easier to learn (see the intuition given in Section 2.2).

The preliminary ROC curve for our car detection system using unsigned wavelet features on color images is shown in Fig. 11.

4. A Real-Time Application

There are many possible applications of this technology, ranging from automotive assistance systems to surveillance. The only factor that is inhibiting our system from being used right now in such systems is the relatively slow processing speed. It is important to note that our full system is, for the most part, an unoptimized research tool; we have not invested significant amounts of effort in improving the core speed.

We have developed a modified version of our static people detection system that achieves real-time

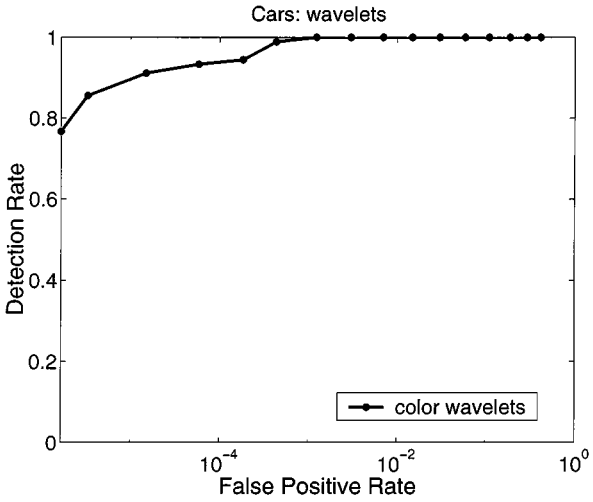

Figure 11. Preliminary ROC curve for car detection using wavelet features over color images.

performance. This section describes a real-time application of our technology as part of a larger system for driver assistance; the combined system, including our people detection module, is currently deployed "live" in a DaimlerChrysler S Class demonstration vehicle. The remainder of this section describes the integrated system.

4.1. Speed Optimizations

Our original unoptimized static people detection system using color images processes sequences at a rate of 1 frame per 20 minutes; this is clearly inadequate for any real-time automotive application. We have implemented optimizations that have yielded several orders of magnitude worth of speedups.

4.1.1. Gray Level Images. Our use of color images for people detection is predicated on the fact that, for people, the three different color channels (RGB) contain a significant amount of information that gets washed out in gray level images of the same scene. This use of color information results in significant computational cost; the resizing and Haar transform operations are performed on each color channel separately. In order to improve system speed, we modify the system to process intensity images.

4.1.2. Using a Subset of the Features. Instead of using the entire set of 1,326 wavelet features, the system undergoes a feature selection step where we pick just 29 of the more important features across our training set that encode the structure of the body. This changes the 1,326 dimensional inner product in Eq. (13) into a 29 dimensional inner product. These wavelets are currently manually chosen as the strongest and weakest wavelets that are consistent across the ensemble either as indicators of an intensity boundary or a uniform region. There are 6 vertical and 1 horizontal coefficients at the scale of 32×32 and 14 vertical and 8 horizontal at the scale of 16×16. Figure 12 shows the coefficients

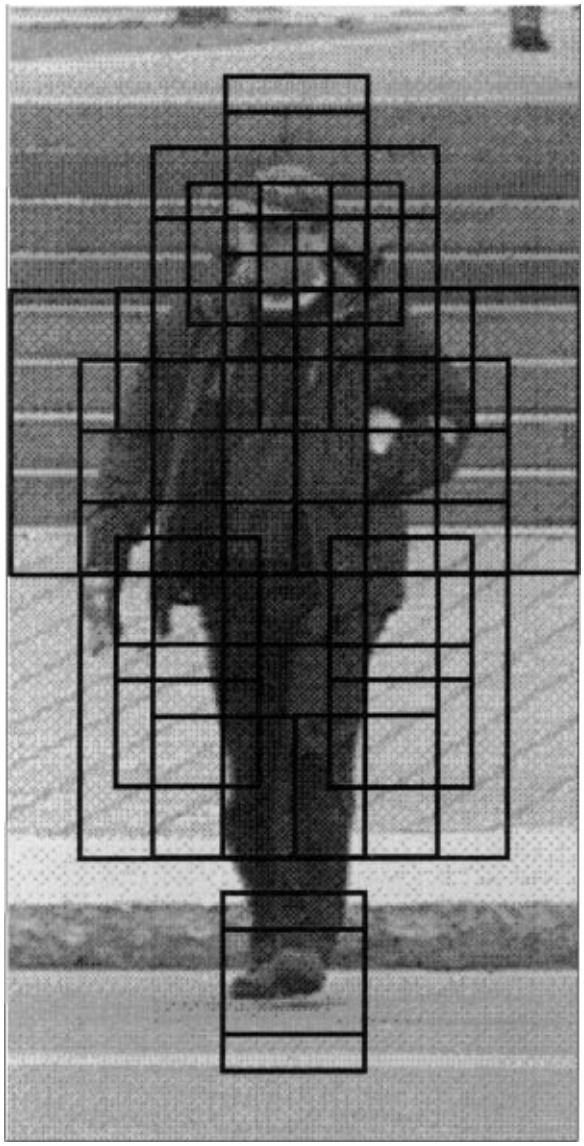

Figure 12. The reduced set of 29 wavelet features for fast people detection overlayed on an example image of a person.

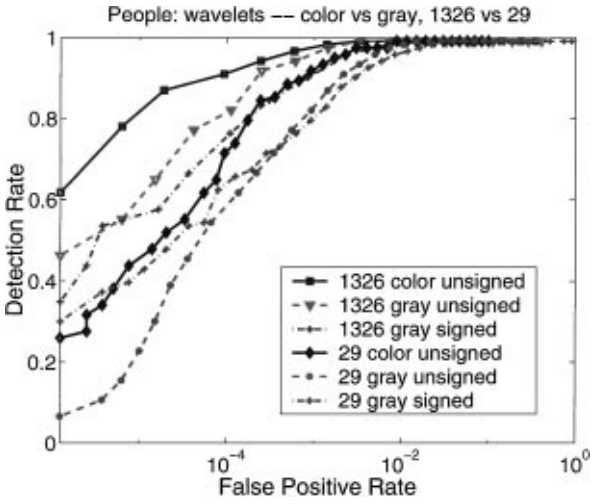

Figure 13. ROC curves for people detection comparing different wavelet features and different feature set sizes; in the version running in the experimental DaimlerChrysler car, we use the 29 gray unsigned version.

in their proper spatial locations, overlayed on an image from the training database. This sparser representation does not, of course, yield the same performance; Fig. 13 shows how our 29 gray unsigned wavelet version compares with other wavelet features and feature set sizes.

4.1.3. Reduced Set Vectors. From Eq. (13), we can see that the computation time is also dependent on the number of support vectors, N_s; in our system, this is typically on the order of 1,000. We use results from (Burges, 1996) to obtain an equivalent decision surface in terms of a small number of synthetic vectors. This method yields a new decision surface that is equivalent to the original one but uses just 29 vectors.

4.1.4. Focus of Attention. To further enhance the processing speed of the system, we can use a focus of attention module that concentrates processing only on areas of an image that are likely to contain people. This focus of attention can key off of different characteristics, including motion, distance, local image complexity, shape, and color (Itti et al., 1998; Itti and Koch, 1999).

4.2. Integration with the DaimlerChrysler Urban Traffic Assistant

To this end, we have integrated our people detection system with a stereo-based obstacle detection system in collaboration with DaimlerChrysler AG. DaimlerChrysler has obviously motivated interests in obstacle detection algorithms for automotive applications as a means to aid driving and, ultimately, to allow for autonomous driving. One of the important requirements of the system is that it is able to deal with both highway and urban scenes, the latter being much more complex than the former.

The DaimlerChrysler Urban Traffic Assistant (UTA) is a real-time vision system for obstacle detection, recognition, and tracking (Franke et al., 1998). UTA relies on 3D position and depth information, obtained using a binocular stereo vision system. To overcome the expensive correspondence problem, they have developed a feature based approach to stereo analysis that runs at 25 Hz on a 200 MHz PowerPC 604. The system clusters feature points that correspond to the same object, providing a rectangular bounding box around each obstacle in the scene.

Using this bounding box which closely outlines the shape of the obstacle, we expand this area to provide a larger region of interest in which we will run our people detection system; this is done to alleviate possible misalignments in the bounding box provided by the stereo system. Furthermore, the stereo system provides an accurate estimate of the distance to each object; using this information we can constrain the number of sizes at which we look for people to a small number, typically under three scales.

Within these regions of interest, we use our 29 gray level feature system with the reduced set method that lowers the number of support vectors to 29. In real-world test sequences processed while driving through Esslingen/Stuttgart, Germany, we are able to achieve rates of more than 10 Hz with under 15 ms per obstacle being spent in the people detection module.

5. Conclusion

We have described a general, trainable object detection system for static images; in this paper, results are shown for face, people, and car detection with excellent results. The system uses a representation based on an overcomplete dictionary of Haar wavelets that captures the significant information about elements of the object class. When combined with a powerful classification engine, the support vector machine, we obtain a detection system that achieves our goals of high accuracy with low rates of false positives. For face detection, typical out-of-sample performance is a detection

rate of 90% with 1 false positive for every 100,000 patterns that are processed and for people detection we can achieve 90% accuracy with 1 false positive for every 10,000 patterns processed. To our knowledge, this is the first people detection system described in the literature that is purely a pattern classification system and that does not rely on motion, tracking, background subtraction, or any assumptions on the scene structure.

Our results in car detection in static images using this trainable architecture are also novel. Due to the significant change in the 2D image information of cars under varying viewpoint, developing a pose invariant car detection system is likely to be significantly more difficult than a pose invariant (upright) people detection system, since the characteristic pattern of a person does not change significantly from different viewpoints. Instead of a full pattern approach, a component based approach to car detection that identifies different parts of a car—headlights, wheels, windshield, etc.—in the appropriate configuration may be more successful. Preliminary work on such a component based system for people detection is described in Mohan (1999).

While the core system we describe implements a brute force search in the entire image, the detector would be more appropriate as part of a larger system. For instance, if we incorporate a *focus of attention* module as in the case of the DaimlerChrysler integration, the system will be able to target specific areas in the scene. This results in both faster processing time and more robust performance.

The performance that this system achieves can be enhanced by incorporating dynamical information when we are processing video sequences. Several techniques that we are working on have already improved the performance to the point where our false positive rate is near zero.

Acknowledgments

This paper describes research done within the Center for Biological and Computational Learning in the Department of Brain and Cognitive Sciences and at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. This research is sponsored by grants from the National Science Foundation, ONR, and Darpa. Additional support is provided by Eastman Kodak Company, Daimler Chrysler, Siemens, ATR, AT&T, Compaq, Honda R&D Co., Ltd., Merrill Lynch, NTT and Central Research Institute of Electric Power Industry.

References

Betke, M., Haritaoglu, E., and Davis, L. 1997. Highway scene analysis in hard real-time. In *Proceedings of Intelligent Transportation Systems*.

Betke, M. and Nguyen, H. 1998. Highway scene analysis form a moving vehicle under reduced visibility conditions. In *Proceedings of Intelligent Vehicles*, pp. 131–136.

Beymer, D., McLauchlan, P., Coifman, B., and Malik, J. 1997. A real-time computer vision system for measuring traffic parameters. In *Proceedings of Computer Vision and Pattern Recognition*, pp. 495–501.

Bregler, C. and Malik, J. 1996. Learning appearance based models: Mixtures of second moment experts. In *Advances in Neural Information Processing Systems*.

Burges, C. 1996. Simplified support vector decision rules. In *Proceedings of 13th International Conference on Machine Learning*.

Burges, C. 1998. A tutorial on support vector machines for pattern recognition. In *Proceedings of Data Mining and Knowledge Discovery*, U. Fayyad (Ed.), pp. 1–43.

Forsyth, D. and Fleck, M. 1997. Body plans. In *Proceedings of Computer Vision and Pattern Recognition*, pp. 678–683.

Forsyth, D. and Fleck, M. 1999. Automatic detection of human nudes, *International Journal of Computer Vision*, 32(1):63–77.

Franke, U., Gavrila, D., Goerzig, S., Lindner, F., Paetzold, F., and Woehler, C. 1998. Autonomous driving goes downtown. *IEEE Intelligent Systems*, pp. 32–40.

Haritaoglu, I., Harwood, D., and Davis, L. 1998. W4: Who? When? Where? What? A real time system for detecting and tracking people. In *Face and Gesture Recognition*, pp. 222–227.

Heisele, B. and Wohler, C. 1998. Motion-based recognition of pedestrians. In *Proceedings of International Conference on Pattern Recognition*, pp. 1325–1330.

Hogg, D. 1983. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20.

Itti, L. and Koch, C. 1999. A comparison of feature combination strategies for saliency-based visual attention systems. In *Human Vision and Electronic Imaging*, vol. 3644, pp. 473–482.

Itti, L., Koch, C., and Niebur, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

Joachims, T. 1997. Text categorization with support vector machines. Technical Report LS-8 Report 23, University of Dortmund.

Lipson, P. 1996. Context and configuration based scene classification. Ph.D. thesis, Massachusetts Institute of Technology.

Lipson, P., Grimson, W., and Sinha, P. 1997. Configuration based scene classification and image indexing. In *Proceedings of Computer Vision and Pattern Recognition*, pp. 1007–1013.

Mallat, S. 1989. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693.

McKenna, S. and Gong, S. 1997. Non-intrusive person authentication for access control by visual tracking and face recognition. In *Audio- and Video-based Biometric Person Authentication*, J. Bigun, G. Chollet, and G. Borgefors (Eds.), pp. 177–183.

Moghaddam, B. and Pentland, A. 1995. Probabilistic visual learning for object detection. In *Proceedings of 6th International Conference on Computer Vision*.

Mohan, A. 1999. Robust object detection in images by components. Master's Thesis, Massachusetts Institute of Technology.

Osuna, E., Freund, R., and Girosi, F. 1997a. Support vector machines: Training and applications. A.I. Memo 1602, MIT Artificial Intelligence Laboratory.

Osuna, E., Freund, R., and Girosi, F. 1997b. Training support vector machines: An application to face detection. In *Proceedings of Computer Vision and Pattern Recognition*, pp. 130–136.

Rohr, K. 1993. Incremental recognition of pedestrians from image sequences. In *Proceedings of Computer Vision and Pattern Recognition*, pp. 8–13.

Rowley, H., Baluja, S., and Kanade, T. 1998. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38.

Shio, A. and Sklansky, J. 1991. Segmentation of people in motion. In *IEEE Workshop on Visual Motion*, pp. 325–332.

Sinha, P. 1994. Qualitative image-based representations for object recognition. A.I. Memo 1505, MIT Artificial Intelligence Laboratory.

Stollnitz, E., DeRose, T., and Salesin, D. 1994. Wavelets for computer graphics: A primer. Technical Report 94-09-11, Department of Computer Science and Engineering, University of Washington.

Sung, K.-K. 1995. Learning and example selection for object and pattern detection. Ph.D. Thesis, MIT Artificial Intelligence Laboratory.

Sung, K.-K. and Poggio, T. 1994. Example-based learning for view-based human face detection. A.I. Memo 1521, MIT Artificial Intelligence Laboratory.

Vaillant, R., Monrocq, C., and Cun, Y.L. 1994. Original approach for the localisation of objects in images. *IEE Proceedings Vision Image Signal Processing*, 141(4):245–250.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer Verlag.

Vapnik, V. 1998. *Statistical Learning Theory*. John Wiley and Sons: New York.

Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. 1995. Pfinder: Real-time tracking of the human body. Technical Report 353, MIT Media Laboratory.