Object and Pattern Detection in Video Sequences

by

Constantine Phaedon Papageorgiou

B.S., Mathematics/Computer Science (1992)

Carnegie Mellon University

Submitted to the

Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1997

Signature of author _____
Department of Electrical Engineering and Computer Science
May 14, 1997

Certified by _____
Tomaso Poggio
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Object and Pattern Detection in Video Sequences

by

Constantine Phaedon Papageorgiou

Submitted to the
Department of Electrical Engineering and Computer Science

May 14, 1997

in partial fulfillment of the requirements for the Degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

This thesis presents a general trainable framework for object detection in static images of cluttered scenes and a novel motion based extension that enhances performance over video sequences. The detection technique we develop is based on a wavelet representation of an object class derived from a statistical analysis of the class instances. By learning an object class in terms of a subset of an overcomplete dictionary of wavelet basis functions, we derive a compact representation of an object class which is used as input to a support vector machine classifier.

The paradigm we present successfully handles the major difficulties of object detection: overcoming the in-class variability of complex classes such as faces and pedestrians and providing a very low false detection rate, even in unconstrained environments.

We demonstrate the capabilities of the technique in two domains whose inherent information content differs significantly. The first system is face detection; we extend the methodology to the domain of people which, unlike faces, vary greatly in color, texture, and patterns. Unlike previous approaches, this system learns from examples and does not rely on any a priori (hand-crafted) models or motion-based segmentation.

The thesis also introduces a motion-based extension to enhance the performance of the detection algorithm over video sequences. This module is based on the realization that in regions of motion, the likely classes of objects are limited, so we can relax the strictness of the classifier. This does not compromise performance over non-moving objects. The results presented here suggest that this architecture may be extended to other domains.

Thesis Supervisor: Professor Tomaso Poggio

Department of Brain and Cognitive Sciences

Biography

Constantine Phaedon Papageorgiou was born on June 30, 1971 in Southfield, Michigan and grew up in Wellesley, Massachusetts. He attended Carnegie Mellon University from September 1989 to December 1992 where he earned a Bachelor's degree in Mathematics/Computer Science, graduating with University Honors and School of Computer Science Research Honors and was named an Andrew Carnegie Society Scholar. From January 1993 to August 1995, Constantine worked in the Speech and Language Processing Department at BBN Inc. in Cambridge, Massachusetts. He entered the graduate program in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology in September 1995 where he has been working in Professor Tomaso Poggio's group at the Center for Biological and Computational Learning and the Artificial Intelligence Laboratory. His research interests include pattern and object recognition and time series prediction, with application areas in image processing and finance.

M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian Detection Using Wavelet Templates. In Proceedings of *Computer Vision and Pattern Recognition 1997*, 1997.

C. Papageorgiou. High Frequency Time Series Analysis and Prediction Using Markov Models. In Proceedings of *Computational Intelligence in Financial Engineering 1997*, 1997.

C. Papageorgiou. Japanese Word Segmentation by Hidden Markov Model. In Proceedings of *ARPA Human Language Technology Conference*, 1994.

C. Papageorgiou and K. Carley. A Cognitive Model of Decision Making: Chunking and the Radar Detection Task. Carnegie Mellon University Computer Science Technical Report CMU-CS-93-228, 1993.

Acknowledgments

I would like to thank my thesis advisor, Tomaso Poggio, for his support at the Center for Biological and Computational Learning at MIT; Tommy originally suggested the idea of exploring object detection in video sequences. The core object detection system described in this thesis was developed with Michael Oren; without his work and patient explanations of wavelets, this thesis would not have come to fruition. I would also like to thank the rest of the group at CBCL — Nicholas Chan, Marcus Dill, Theodoros Evgeniou, Tony Ezzat, Federico Girosi, Vinay Kumar, Blake LeBaron, Joerg Lemm, Sayan Mukherjee, Jon Murnick, Edgar Osuna, Max Riesenhuber, Pawan Sinha — for making working in lab so stimulating, interesting, and entertaining.

To my parents

Thalia and John

and my sisters and brother

Elena, Antigone, and Demetrios

Contents

Chapter 1

Introduction

The amount of audio and visual information available has exploded in recent years, to the point where manually searching and cataloging the vast number of databases available has become impossible. One can expect this problem to only get worse, with the drop in prices in memory and the advent of new, inexpensive digital video recording techniques. This is a serious problem with databases of still images; for databases of video sequences, this problem is compounded several times over. Imagine trying to manually search a single video sequence for all images of Bill Clinton; at a rate of 30 frames per second, even one hour of video presents a formidable, time-consuming task. Our goal is to develop a robust, automatic method for searching video sequences for a specific class of objects, for example, faces or people. This technology represents the first step in a system that would find all the frames showing Bill Clinton, as hinted at above. This automatic detection module would first find the subset of frames containing images of people, and then would pass the subimages of people to a recognition module that could compare the person to an internal database of people that it "knows", to determine which ones are Bill Clinton.

1.1 The Problem

The work in this thesis addresses the problem of object and pattern detection in video sequences of cluttered scenes. This problem is a version of the detection problem for

single images; in this case, though, the dynamic motion information is available to any algorithm that tackles object detection in video sequences. In this sense, the detection in video task is more simple than detecting objects in static images. Regardless of the extra information available, the object detection task is still difficult; we now discuss the characteristics of the problem that make it so challenging.

Trying to detect real-world objects of interest, such as faces and people, poses especially challenging problems since these non-rigid objects are hard to model and there is significant variety in color and texture. In contrast to the case of pattern classification where we need to decide between a relatively small number of classes, the detection problem requires us to differentiate between the object class and the rest of the world. As a result, the class model must accommodate the intra-class variability without compromising the discriminative power in distinguishing the object within cluttered scenes. Furthermore, the need for a precise and tight class model is magnified due to the difficulty of the decision problem in the case of detection.

In Figure 1-1, we highlight the difficulty of the general detection problem by showing a subset of the training examples used as prototypes for a pedestrian object class. We can see that that the pedestrians vary greatly in color and texture and there is no consistency in the background. The model we develop should be able to model this variability.

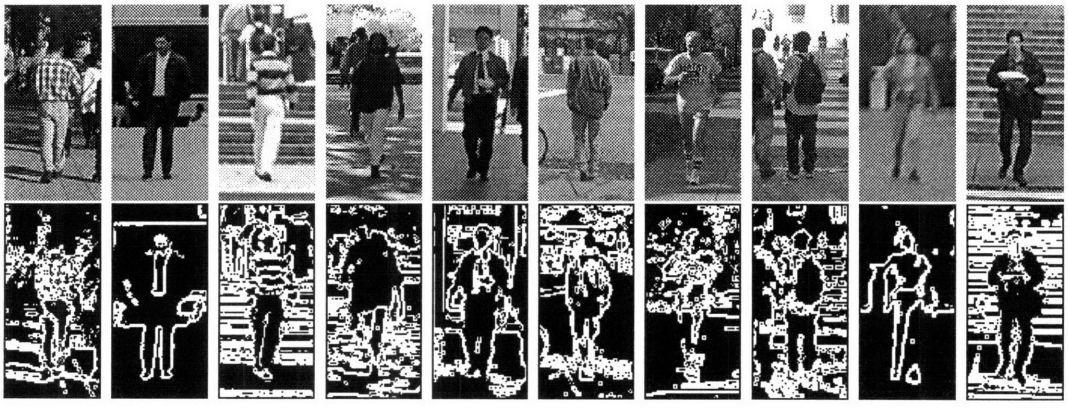


Figure 1-1: The top row shows examples of images of people in the training database. The examples vary in color, texture, view point (either frontal or rear) and background. The bottom row show edge detection of the pedestrians. Edge information does not characterize the pedestrian class well.

One appealing framework in which the problem can be cast is that of maximum likelihood estimation (MLE). This is a powerful technique, since we can obtain an estimate of the likelihood from our data set, but it suffers from the problem that it assumes there is one and only one class object in the scene. For the more general and much more difficult case, this set of approaches cannot be used since it is not known how many class objects are present in the scene, if any.

Consequently, the classification of each pattern in the image must be done independently; this makes the decision problem susceptible to missed instances of the class and false positives. As mentioned above, our dual goal is to develop a system that misses very few of the desired objects while generating very few false detections. How do we quantify "very few", though? A false detection rate of one for every 10,000 patterns looked at seems very low, but when we realize that a typical image we will be analyzing contains on the order of 500,000 subwindows, this means that could expect 50 false detections per image. When we look at video sequences, the cumulative number of false detections across a sequence of frames would seriously limit the practicality of using such a system. Hence, one of the goals underlying our work is to provide a framework whose accuracy is tunable to the current task.

1.2 Previous Work

In this section, we describe prior work in object detection in static images and in video sequences that is related to our technique.

1.2.1 Object Detection in Static Images

The basic problem of object detection in images is of central importance to any image understanding system. Typically, the systems that have been developed fall into one of two categories: template-based approaches that attempt to match or fit a prototype template to different parts of the image (Betke and Makris, 1995[2], Yuille *et al.*, 1992[31]) or image invariance methods that base a matching on a set of image pattern relationships (eg. brightness levels) that, ideally, uniquely determine the objects being searched for (Sinha, 1994[22][23]).

More recently, systems for detecting unoccluded vertical frontal views of human faces in images have been developed using example-based approaches by Sung and Poggio, 1994[26], Moghaddam and Pentland, 1995[15], Rowley *et al.*, 1995[20], Vaillant *et al.*, 1994[28], and Osuna *et al.*, 1997[17]. 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. The system of Sung and Poggio models a set of faces and non-faces as clusters in a high-dimensional space. For a given pattern, the system uses two distance measures for each cluster to differentiate face patterns from non-faces: a Mahalanobis-like distance in the space of the largest eigenvectors and the Euclidean distance in the space of the smallest eigenvectors. A neural network classifier is trained on these distance measurements. To find all the faces in an image, the algorithm iterates over each sub-image of the image, calculating the distance measures for input into the neural network. The other face detection systems cited above are similar: Moghaddam and Pentland use an eigenface representation derived from a principal components analysis of faces, Rowley and Vaillant use neural networks with different receptive fields, and Osuna uses a support vector machine to classify the patterns.

An approach that differs from the standard pixel-based representation was taken by Sinha, 1994[22][23] who introduced the idea of the "template ratio" — encoding a human face as a set of binary relationships between the average intensities of 11 regions. The assumption behind the template ratio was that these relationships will hold regardless of significant changes in illumination direction and magnitude. For example, the eye sockets are almost always darker than the forehead or the cheeks. The success and robustness of the template ratio approach for face detection indicates that a representation based on the encoding of differences in average intensities of different regions is a promising direction. However, no rigorous mathematical formulation and learning algorithm for the derivation of the template were presented; the face regions covered by the template and the relationships were hand-crafted. Also, the detection algorithm was based on simple template matching whose expressive power is restricted to relatively simple and fixed relationships.

1.2.2 Object Detection in Video Sequences

The detection of objects in video has seen a high degree of interest in recent years. We describe several relevant systems here.

Most early systems that detect objects in video sequences have focused on using motion and simple shapes or constraints to find people. Tsukiyama and Shirai, 1985[27] use simple shape descriptions to determine the location of leg motion against a white background and a distance measure is utilized to determine the correspondences between moving regions in consecutive images. This system can handle multiple people in an image, but requires a stationary camera and only uses leg movement to track people. Leung and Yang, 1987[12][11] use a voting process to determine candidate edges for moving body parts and a set of geometric constraints to determine actual body part locations. This architecture also assumes a fixed camera; another important restriction is that it is only able to deal with a single moving person.

The use of 3D models has been prominent in finding people in video sequences. This type of system, while adequate for specific, well-defined domains, involves using a lot of domain specific information in the development of the model and is not

easily portable to new domains. Hogg, 1983[7] describes a system that is based on modeling a human figure as a hierarchical decomposition of 3D cylinders, using dynamic constraints on the movement of the limbs as well. Edge detection is used to determine the possible locations of body parts and a search tree is used to determine the location that maximizes a "plausibility" measure, indicating the likelihood that there is a person at this location. Rohr, 1993[19] develops a system using similar 3D cylindrical models of the human body and kinematic motion data. Model contours are matched with edges that are found in an image using a grid search method. A Kalman filter is used to determine the exact position and pose of the walking person across multiple frames. Both these architectures assume a fixed camera and a single moving person in the image.

Wren *et al.*, 1995[30] describe a system for the real-time tracking of the human body. The model of the person they develop uses a maximum a posterior (MAP) approach to segment a person into blobs corresponding to different regions of the body. This system relies on two key assumptions: that the camera and background are fixed and that there is a single person in the image. These are clearly restrictive assumptions for a general purpose person tracker.

McKenna and Gong, 1997[14] describe a system that tracks people in video and automatically detects a face for each person found in the images. The algorithm assumes a fixed camera and, after detecting motion, clusters the motion information to separate different bodies of motion. They use a Kalman filter to track the different people and implement a radial basis function network to detect the faces. As noted, the system assumes a fixed camera and may have problems detecting people that are not moving.

To track moving objects, Heisele *et al.*, 1997[6] use the clusters of consistent color to track moving objects. Initially, the system computes the color clusters for the first image in a sequence. The system recomputes the cluster centroids for subsequent images, assuming a fixed number of clusters. To track an object, the clusters corresponding to that object are manually labeled in an initial image and are tracked in subsequent frames – the user is in effect performing the first detection manually. The

authors highlight, as future work, investigating object detection with this algorithm. An important aspect of this system is that, unlike other systems described in this section, this technique does not assume a stationary camera.

Campbell and Bobick, 1995[4] take a different approach to analyzing human body motion. They present a system for recognizing different body motions using constraints on the movements of different body parts. Motion data is gathered using ballet dancers with different body parts marked with sensors. The system uses correlations between different part motions to determine the "best" recognizer of the high-level motion. They use this system to classify different ballet motions. Lakany and Hayes, 1997[10] also use moving light displays (MLDs) combined with a 2D FFT for feature extraction to train a neural network to recognize a walker from his/her gait.

All these systems have succeeded to varying degrees but have relied on the following restrictive features:

- explicit modeling of the domain;
- stationary camera and a fixed background;
- marking of key moving features with sensors/lights;
- implement tracking of objects, not detection of specific classes.

Model-based approaches need a large amount of domain specific knowledge while marking features is impractical for real world use. The tracking systems have problems handling the entrance of new people into the scene; to overcome this problem, a tracking system would need to emulate a detection system. This work will overcome these problems by introducing an example-based approach that learns to recognize patterns and avoids the use of motion and explicit segmentation. The motion-based extension presented in this thesis is used solely to enhanced detection accuracy, without compromising accuracy over non-moving objects.

1.3 Our Approach

The approach taken in this thesis is that the system we develop will learn to perform the detection task from examples for static images and then will be enhanced with a motion module. Learning based techniques are used in many pattern classification problems in a wide range of areas, from image processing to time series prediction. The problem of learning from examples is formulated as one where the system attempts to derive an input/output mapping, or equivalently, a model of the domain, from a set of training examples. This type of approach is particularly attractive for several reasons. First and foremost, by learning the characteristics of a problem from examples, we avoid the need for explicitly handcrafting a solution to the problem. A handcrafted solution may suffer from the users imposition of what he thinks the important features or characteristics of a decision problem are. With a learning-based approach, exactly the important features and relationships of a decision problem are automatically abstracted away as a trained model. On the other hand, learning based approaches suffer from the problem of overtraining or overfitting, where the model has learned the decision problem "too well" and is not able to generalize to new data.

For a learning based system to be successful, we must choose an appropriate set of features or representation that will allow the system to learn the decision problem. As we mentioned before, many of the static detection systems use pixel-based image representation. The pixel values are used directly to represent the objects of interest, typically faces. While this approach is satisfactory in the case of faces which are relatively rigid objects, the systems fail to generalize to other domains. The fundamental problem is that the raw pixel intensities fail to capture the common structure of complex class instances.

One might consider using an edge-based representation; Figure 1-1 shows images of people that are used to train our system with their corresponding edge maps, derived after processing them with a Sobel edge detector. This is not feasible because these algorithms only examine a small local neighborhood to determine if a pixel is part of an edge; it should be clear from the variability in clothing patterns that, at a fine scale,

edge detection will result in many spurious patterns. It is clear from these images that an approach based on analyzing and matching these fine scale edges is unlikely to succeed. These images show that, not only are the edges unreliable information, but also that different instances of the people class can appear very differently and it will be hard, if not impossible, to derive a class model from edges. There is no consistency across this sample of the class, so it is not evident how we should specify our model. The above examples of pixel-based and edge-based representations are given to illustrate that the appropriate representation of the class objects is the the crux of developing a trainable detection system. To make the model learning and classifier training feasible, a new representation must be invoked.

In Section 1.2, we describe the work of Sinha, 1994[22][23]; this idea of using relationships that express differences between intensities of neighboring regions suggests use of basis functions that encode these differences. The Haar wavelet is a particular simple family of such basis functions that we choose for our system. In our work, we use the wavelet representation to capture the structural similarities between various instances of an object class. Another important feature of our work is the use of an overcomplete, or redundant, set of basis functions; this is important in capturing global constraints on the object shape and for providing adequate spatial resolution. We introduced this idea in [16], where we applied the idea of using wavelets for detection for the first time and showed how the wavelet based representation is both efficiently learnable and provides a model of an object class that has significant discriminative power; the application domain was pedestrian detection in static images. The Haar wavelet representation has also been used for image database retrieval, Jacobs *et al.*, 1995[8], where the largest wavelet coefficients are used as a measure of similarity between two images. Our results on object detection using the wavelet representation demonstrate that it may be a promising framework for computer vision applications.

In this thesis, the capabilities of our technique are highlighted by applying it to two classes, faces and pedestrians. These two domains are characterized by different types of information. Human faces share common pattern structure, as noted earlier,

with eye regions being darker than the forehead, and so on. As such, the face detection system learns commonalities within the boundaries of the face. On the other hand, the pedestrian class can only be defined by the general boundary of the body – clearly, there is no way to characterize the many colors, textures, and patterns of clothing that exist. We will show that the wavelet framework and learning algorithm that our approach is based on is able to handle these domains whose inherent information content differs widely.

This thesis presents a novel extension that uses motion cues to improve detection accuracy. The motion module is a general one that can be used with many detection algorithms – we apply it in our wavelet framework for pedestrian detection as a testbed. In the case of video sequences, we can utilize motion information to enhance the robustness of the detection module. We generate the flow field between two consecutive frames, defined as the pixelwise correspondences between the two frames. From this low-level map we can extract higher order information on regions of the sequence that are moving. We compute the flow between consecutive images and detect discontinuities in this flow field that indicate probable motion of objects relative to the background. In these regions of motion, the likely class of objects is limited, so we can relax the strictness of the classifier. It is important to observe that we do not assume a static camera nor do we need to recover camera ego-motion, rather, we use the dynamic motion information to assist the classifier. Additionally, the use of motion information does not compromise the ability of the system to detect non-moving people.

The organization of the paper is as follows: in Chapter 2 we introduce the wavelet based representation that is the core of the pattern recognition algorithm. A short review of the discrete wavelet transform is given and we also present its extension to the dense wavelet transform. Chapter 3 presents the learning algorithm that automatically learns the important characteristics of an object class. Chapter 4 describes the support vector machine learning algorithm we use. The system architecture is described in Chapter 5. The experimental results are detailed in Chapter 6. In Chapter 7, we describe the novel motion-based module that allows us to apply the detection

technique to video sequences, with higher accuracy than a system that analyzes each frame statically. Chapter 8 summarizes the results of the thesis, presents several potential applications of the technique, and describes directions for future work.

Chapter 2

Wavelets

This section describes the underlying representation that we use for extracting object features, the Haar wavelet; a more detailed treatment can be found in Mallat, 1989[13]. We also describe a denser (redundant) transform that we use to achieve the spatial resolution we need to accomplish detection and define the wavelet basis.

2.1 The Haar Wavelet

Wavelets provide a natural mathematical structure for describing our patterns. 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 $V^0 \subset V^1 \subset V^2 \subset \ldots V^j \subset \ldots V^{j+1} \ldots$; the vector space V^{j+1} can describe finer details that 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*,

$$\phi_i^j = \sqrt{2^j}\phi(2^j x - i), i = 0, \ldots, 2^j - 1, \tag{2.1}$$

where, for our case of the Haar wavelet,

$$\phi(x) = \begin{cases} 1 & for\ 0 \leq x < 1 \\ 0 & otherwise \end{cases} \tag{2.2}$$

Next we define the vector space W^j that is the orthogonal complement of two consecutive approximating subspaces, $V^{j+1} = V^j \bigoplus W^j$. 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,

$$\psi_i^j = \sqrt{2^j}\psi(2^j x - i), i = 0, \ldots, 2^j, \tag{2.3}$$

where for Haar wavelets,

$$\psi(x) = \begin{cases} 1 & for\ 0 \leq x < \frac{1}{2} \\ -1 & for\ \frac{1}{2} \leq x < 1 \\ 0 & otherwise \end{cases} \tag{2.4}$$

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

$$A_j f = \sum_{k \in Z} \overbrace{< f(u), \phi_k^j(u) >}^{\lambda_{j,k}} \phi_k^j(x) \tag{2.5}$$

and similarly, the projection of $f(x)$ on W^j is:

$$D_j f = \sum_{k \in Z} \overbrace{< f(u), \psi_k^j(u) >}^{\gamma_{j,k}} \psi_k^j(x) \tag{2.6}$$

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, $\gamma_{j,k}$:

$$\lambda_{j,k} = \sum_{n \in Z} h_{n-2k} \lambda_{j+1,n} \tag{2.7}$$

$$\gamma_{j,k} = \sum_{n \in z} g_{n-2k} \lambda_{j+1,n} \tag{2.8}$$

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:

$$A_j f = \sum_{n \in Z} \lambda_{j,k} \sqrt{2^j} \phi(2^j x - k) \tag{2.9}$$

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

$$D_j f = \sum_{n \in Z} \gamma_{j,k} \sqrt{2^j} \psi(2^j x - k) \tag{2.10}$$

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.

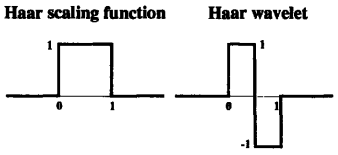


Figure 2-1: Haar scaling function and Haar wavelet.

2.2 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 Figure 2-2. 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 *corner* coefficient since this wavelet responds strongly to corners.

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[24]). The results are Haar wavelets with square support at all scales. In doing the non-standard transform, we apply quadruple density transforms in each of the dimensions.

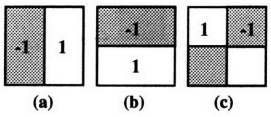


Figure 2-2: The 3 types of 2-dimensional non-standard Haar wavelets; (a) "vertical", (b) "horizontal", (c) "corner".

2.3 Dense Wavelet Transform

The standard Haar basis is not dense enough for our application; for the 1D transform, the distance between two neighboring wavelets at level n (with support of size 2^n) is 2^n. For better spatial resolution, we need a set of redundant basis functions, or an overcomplete *dictionary*, where the distance between the wavelets at scale n is $\frac{1}{4}2^n$. We call this a *quadruple density* dictionary (see Figure 2-3). As one can easily observe, the straightforward approach of shifting the signal and recomputing the

DWT will *not* generate the desired dense sampling. Instead, this can be obtained by modifying the DWT. To generate wavelets with *double density*, where wavelets of level n are centered every $\frac{1}{2}2^n$, we simply do not downsample in Equation 2.8. To generate the quadruple density dictionary, we do not downsample in Equation 2.7 and get double density scaling coefficients. The next step is to 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 scale, 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 scales, we avoid an "explosion" in the number of coefficients, yet provide a dense and uniform sampling of the wavelet coefficients at all the scales. As with the regular DWT, the time complexity is $O(n)$ in the number of pixels n. The extension of the quadruple transform to 2D is straightforward.

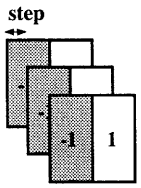


Figure 2-3: Quadruple density 2D Haar basis. The dense sampling provide adequate spatial resolution and the overlapping supports facilitates the definition of complex constaints on the object patterns.

Chapter 3

Learning

3.1 Learning the Object Class Representations

In this section we describe the wavelet representation and how it can be learned from examples. We illustrate the techniques on two different classes of objects: faces and full-body pedestrians.

3.1.1 The Wavelet Representation

The Haar 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: the coefficients encode the difference in average intensity between different regions along different orientations. Furthermore, this coding is done in different scales. The constraints on the values of the coefficients can express well-defined constraints on the object. For example, a low value of a wavelet coefficient indicates a uniform area and a strong value of a "corner" coefficient indicates an actual corner in the image. Since the precise value of a coefficient may be irrelevant, we analyze its relative value compared to the other coefficients after we perform a normalization described in the next section. It is also straightforward to encode the relationships between the ratio of intensities, instead of the differences; this is accomplished by simply computing the wavelet transform on the log of the image values. The wavelet

transform also provides a multiresolution representation with coefficients in different scales capturing different levels of detail; the coarse scale coefficients capture large regions while the fine scale coefficients represent smaller, local regions.

Another important aspect of this representation is the use of an overcomplete (redundant) Haar basis which allows us to propagate constraints between neighboring regions and to describe complex patterns. We choose the quadruple density wavelet transform since it is found to provide adequate spatial resolution. As is demonstrated in the following section, the use of difference or ratio coding of intensities in different scales provides a very flexible and expressive representation that can characterize 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 look at different sets of coefficients for different spatial locations.

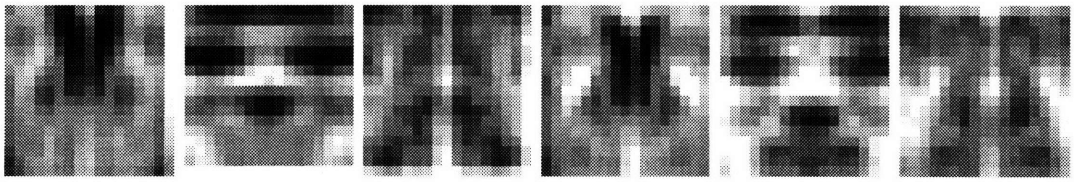
Given an object class, the central problem is how to learn which are the relevant coefficients that express structure common to the entire object class and which are the relationships that define the class. Currently, we divided the learning into a two-stage process: identifying the significant wavelet coefficients and learning the relationships between the coefficients. This section describes the first stage of learning; the second stage is described in the next section.

Figure 3-1: The databases of faces used for training. The images are gray level of size 19×19 pixels.

3.1.2 Learning the Face Class

1.05	1.42	1.73	1.97	2.07	1.97	1.70	1.48	1.46	1.61	1.89	2.08	2.08	1.93	1.65	1.27	1.12
1.04	1.30	1.45	1.55	1.62	1.56	1.37	1.26	1.26	1.30	1.44	1.56	1.56	1.50	1.36	1.15	1.09
0.86	0.98	1.03	1.00	0.95	0.82	0.69	0.69	0.72	0.71	0.79	0.92	1.01	1.04	1.03	0.94	0.94
1.10	1.27	1.31	1.27	1.08	0.83	0.70	0.67	0.70	0.77	0.85	1.05	1.26	1.38	1.42	1.33	1.28
1.54	1.81	1.91	1.83	1.61	1.29	0.99	0.84	0.83	0.96	1.20	1.53	1.81	1.99	2.01	1.82	1.72
1.50	1.78	1.90	1.78	1.54	1.24	0.90	0.73	0.72	0.82	1.12	1.45	1.72	1.91	1.89	1.67	1.56
0.99	1.19	1.30	1.20	1.00	0.80	0.59	0.54	0.55	0.54	0.71	0.94	1.14	1.28	1.24	1.06	0.97
0.57	0.68	0.75	0.70	0.60	0.51	0.48	0.56	0.60	0.56	0.58	0.62	0.70	0.77	0.74	0.64	0.62
0.60	0.73	0.81	0.81	0.79	0.83	0.96	1.15	1.21	1.08	0.92	0.82	0.82	0.85	0.81	0.73	0.70
0.86	1.01	1.04	0.99	0.98	1.11	1.39	1.69	1.73	1.48	1.16	0.96	0.90	0.99	1.06	1.01	0.95
0.93	1.01	0.97	0.86	0.84	1.02	1.35	1.64	1.68	1.45	1.11	0.84	0.79	0.92	1.04	1.03	0.99
0.80	0.83	0.85	0.79	0.71	0.75	0.93	1.12	1.15	0.99	0.81	0.75	0.80	0.87	0.87	0.84	0.81
0.62	0.66	0.76	0.85	0.85	0.82	0.85	0.96	0.98	0.90	0.87	0.90	0.91	0.82	0.70	0.63	0.61
0.56	0.56	0.68	0.82	0.89	0.87	0.84	0.87	0.89	0.86	0.90	0.95	0.89	0.73	0.59	0.54	0.56
0.61	0.54	0.62	0.77	0.85	0.85	0.83	0.86	0.88	0.85	0.87	0.91	0.85	0.71	0.59	0.57	0.64
0.72	0.58	0.58	0.74	0.90	0.92	0.87	0.87	0.88	0.87	0.91	0.93	0.83	0.68	0.61	0.67	0.79
0.44	0.35	0.32	0.36	0.43	0.47	0.47	0.50	0.51	0.48	0.46	0.44	0.38	0.35	0.35	0.41	0.47

Table 3.1: Ensemble average of normalized horizontal coefficients of scale 4×4 of images of faces. Meaningful coefficients are the ones with values much larger or smaller than 1. Average values close to 1 indicates no meaningful feature.



(a) (b) (c) (d) (e) (f)

Figure 3-2: Ensemble average values of the wavelet coefficients for faces coded using gray level. Coefficients whose values are close to the average value of 1 are coded gray, the ones which are above the average are darker and below the average are lighter. We can observed strong features in the eye areas and the nose. Also, the cheek area is an area of almost uniform intensity, ie. below average coefficients. (a)-(c) vertical, horizontal and corner coefficients of scale 4×4 of images of faces. (d)-(f) vertical, horizontal and corner coefficients of scale 2×2 of images of faces.

For the face class, we have a set of 2429 gray-scale images of faces; this set consists of a core set of faces, with some small angular rotations to improve generalization. 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 Figure 3-1. Databases of this size and composition have been used extensively in face detection [25] [20] [17] and we keep this data format for comparison purposes. For the coefficient analysis, we use the wavelets at scales of 4×4 pixels (17×17 coefficients of quadruple density for each wavelet class) and 2×2 pixels (17×17 in double density for each class) since their dimensions correspond to typical facial features for 19×19 face images. We have a total of 1734 coefficients.

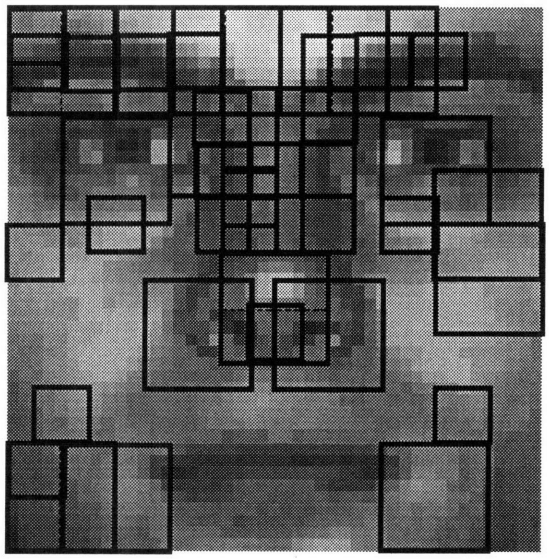


Figure 3-3: The significant wavelet bases for face detection that are uncovered through our learning strategy, overlayed on an example image of a face.

Our goal is to learn the significant subset of coefficients which convey the most important information on the structure of the face class. One could attempt to determine this set from a visual analysis of the database and by using knowledge of the structure of the face. However, it is clear that this ad hoc approach cannot replace a more rigorous analysis; this will also become essential when we try to analyze more complex objects such as people. Our approach is to identify which coefficients are consistent along all the examples of the class. These coefficients can either be very small along the ensemble of objects, indicating areas with almost uniform intensity, or can have significant non-zero values, indicating a feature such as corner or vertical boundary. Coefficients whose values change randomly between different instances of the class do not indicate a class feature and can be discarded from consideration. For the learning and detection steps we use the absolute value of the coefficients; this simplification will also be important for the case of people detection.

The basic analysis in identifying the important coefficients consists of two steps: first, we normalize the wavelet coefficients relative to the rest of the coefficients in the patterns; second, we analyze the averages of the normalized coefficients along the ensemble.

The normalization step involves computing the average of each coefficient's class ($\{vertical, horizontal, corner\} \times \{2, 4\}$) over all the object patterns and dividing every coefficient by its corresponding class average. We calculate the averages separately for each class since the power distribution between the different classes may vary. To begin specifying the wavelet representation, we calculate the average of each normalized coefficient over the set of objects. As an example of the effect of this processing, Table 3.1 shows the average coefficient values for the set of horizontal Haar coefficients of scale 4×4 for the face class. After the normalization, the average value of a coefficient for random patterns should be 1. We can observe three types of coefficients: coefficients whose ensemble average values are much larger than 1, indicating strong coefficients that are consistent along all the examples, coefficients whose values are much less than 1, indicating uniform regions, and coefficients whose values are close to 1. The last group contains coefficients whose values are not consistent along the ensemble and therefore can be considered as irrelevant coefficients. From the former two groups we choose the strongest and weakest coefficients as the consistent prominent features. The above analysis is done to the different coefficient types at different scales. To illustrate the detected features we code the ensemble average of the coefficients using gray level and draw them in their proper spatial layout, shown in Figure 3-2. Coefficients with values close to 1 are plotted in gray, those with values larger than 1 are plotted darker, and those with values less than 1 are lighter. It is interesting to observe the emerging patterns in the facial features. The vertical coefficients, Figure 3-2(a),(d), capture the sides of the nose, while the horizontal coefficients, Figure 3-2(b),(e), capture the eye sockets, eyebrows, and tip of the nose. The mouth is found to be a relatively weak feature compared to the others. The corner coefficients, Figure 3-2(c),(f) respond strongly to the endpoint of facial features. We also conduct a similar analysis with the wavelets of the log of the intensities (these are related to the ratio of intensities). Results of this statistical analysis are similar to the intensity differencing wavelets, indicating that, for pedestrians and faces, the difference and ratio versions capture essentially identical information. An analysis using the sigmoid function as a "soft threshold" on the normalized coefficients yields

equivalent results. In general, the learning of the coefficients can be based on different statistical analyses of the ensemble coefficients.

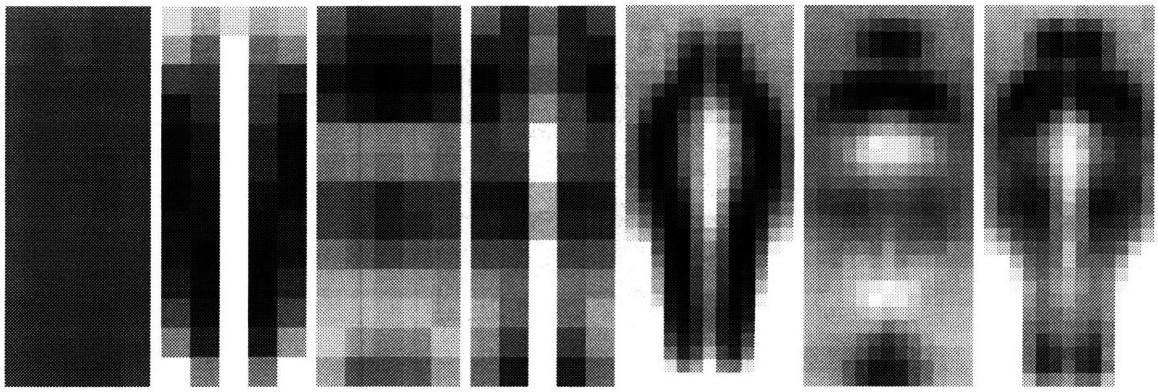
From this statistical analysis, we derive a set of 41 coefficients, from both the coarse and finer scales, that consistently recover the significant features of the face. These significant bases consist of 6 vertical, 7 horizontal, and 6 corner coefficients at the scale of 4×4 and 10 vertical, 11 horizontal, and 1 corner coefficients at the scale of 2×2. Figure 3-3 shows a typical human face from our training database with the significant 41 coefficients drawn in the proper configuration.

3.1.3 Learning the Pedestrian Class

For learning the pedestrian class, we have collected a set of 924 color images of people (Figure 1-1). All the images are scaled and clipped to the dimensions 128×64 such that the people are centered and approximately the same size (the distance from the shoulders to feet is about 80 pixels). Since there can be variations in scale and size between peoples during the detection process we cannot use very small scale coefficients and need to allow a tolerance of few pixels. Therefore, in our analysis, we restrict ourselves to the wavelets at scales of 32×32 pixels (one array of 15×5 coefficients for each wavelet class) and 16×16 pixels (29×13 for each class). For each color channel (RGB) of every image, we compute the quadruple dense Haar transform and take the coefficient value to be the largest absolute value among the three channels, yielding 1326 wavelet coefficients. The use of the largest coefficient in absolute value among the RGB channels is based on the observation that there is no consistency in color between the different people and the most robust visual information is the differentiation between a person's overall shape and the background. The use of the absolute value of the coefficient is essential in the case of pedestrians since the signs of the coefficients are meaningless; a dark body against a light background should be interpreted the same way as a light body against a dark background.

To visualize the emerging patterns for the different classes of coefficients we can color code the values of the coefficients and display them in the proper spatial layout, as we did for the faces. Each coefficient is displayed as a small square where coefficients

close to 1 are gray, stronger coefficients are darker, and weaker coefficients are lighter.



(a) (b) (c) (d) (e) (f) (g)

Figure 3-4: Ensemble average values of the wavelet coefficients coded using gray level. Coefficients whose values are above the template average are darker, those below the average are lighter. (a) vertical coefficients of random scenes. (b)-(d) vertical, horizontal and corner coefficients of scale 32×32 of images of people. (e)-(g) vertical, horizontal and corner coefficients of scale 16×16 of images of people.

Figures 3-4(a)-(d) show the color coding for the arrays of coarse scale coefficients (32×32) and Figures 3-4(e)-(g) show the arrays of coefficients of the finer scale, (16×16). Figure 3-4(a) shows the vertical coefficients of random images; as expected this figure is uniformly gray. The corresponding images for the horizontal and corner coefficients, not shown here, are similar. In contrast to the random images, the coarse scale coefficients of the people, Figures 3-4(b)-(d), show clear patterns. It is interesting to observe that each class of wavelet coefficients is tuned to a different type of structural information. The vertical wavelets, Figure 3-4(b), capture the sides of the pedestrians. The horizontal wavelets, Figure 3-4(c), respond to the line from shoulder to shoulder and to a weaker belt line. The corner wavelets, Figure 3-4(d), are better tuned to corners, for example, the shoulders, hands, and feet. The wavelets of finer scale in Figures 3-4(e)-(g) provide better spatial resolution of the body's overall shape and smaller scale details such as the head and extremities appear clearer.

The result of the analysis described above is a set of 29 coefficients that are consistent along the ensemble either as indicators of "change" or "no-change". 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 3-5 shows the coefficients in their proper

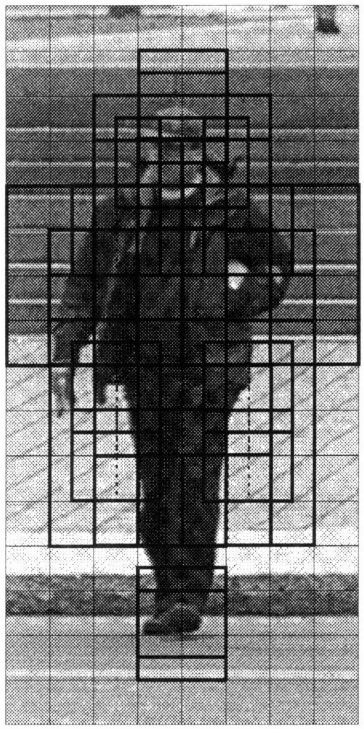


Figure 3-5: The significant wavelet bases for pedestrian detection that are uncovered through our learning strategy, overlayed on an example image of a pedestrian.

spatial locations, overlayed on an image from the training database. The identified set of coefficients is used as a feature vector for a classification algorithm that is trained to differentiate pedestrians from non-pedestrians.

3.1.4 Discussion

We have decomposed the learning of an object class into a two-stage learning process. In the first stage, described in this section, we perform a dimensionality reduction where we identify the most important coefficients from the original full set of wavelet coefficients consisting of three types in two scales. The relationships between the coefficients which define the class model are learned in the second stage using a support vector machine (SVM). Based on our initial experiments, it is doubtful that successful learning of the relationships between coefficients' values could be achieved on the original full set of coefficients – for our two domains, these have dimensions of 1326 and 1734 – without introducing several orders of magnitude of additional training

data. Most of these coefficients do not necessarily convey relevant information about the object class we are learning but, by starting with a large overcomplete dictionary, we would not sacrifice details or spatial accuracy. The above learning step extracts the most prominent features and results in a significant dimensionality reduction.

Comparing the database of people, Figure 1-1, to the database of faces, Figure 3-1, illustrates an important fundamental difference in the two classes. In the case of faces, we can find 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, there a lot of spurious details such as jackets, ties, bags and more. On the other hand, the overall body has a typical shape (or "silhouette") that characterizes people well. It should be observed that in the case of faces, the inner facial features are picked up by our learning algorithm while in the case of people it is the body shape that is identified. Our approach treats these two different cases in almost uniform manner.

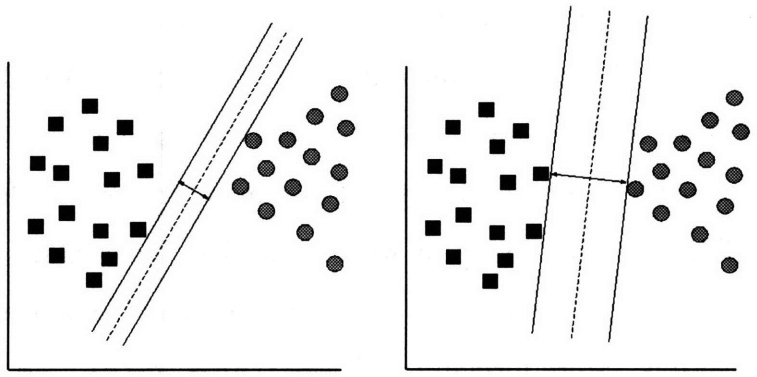
Chapter 4

Support Vector Machine Classifier

As described in the previous section, the decision task, whether a given window contain a member of the target class or not, is the most difficult task and crux of the detection system. In Chapter 3 we describe the identification of the significant coefficients that characterize the object classes. These coefficients can be used as feature vector for various classification methods.

The classification technique we use is the support vector machine (SVM) developed by Vapnik *et al.*, 1992[3][29]. This recently developed technique has several features that make it particularly attractive. Traditional training techniques for classifiers, such as multilayer perceptrons (MLP), use empirical risk minimization and only guarantee minimum error over the training set. In contrast, the SVM machinery uses structural risk minimization which minimizes a bound on the generalization error and therefore should perform better on novel data. Another aspect of the SVM technique that makes it appealing is that there is only one tunable parameter, C, a penalty term for misclassifications. This is contrasted with other conventional classifiers like neural networks, where there are a large number of tunable parameters that can greatly affect the performance of the classifier, such as network topology and the learning rate. We first illustrate the SVM algorithm for the simple case of two linearly separable classes and then describe the full algorithm for nonlinear decision surfaces.

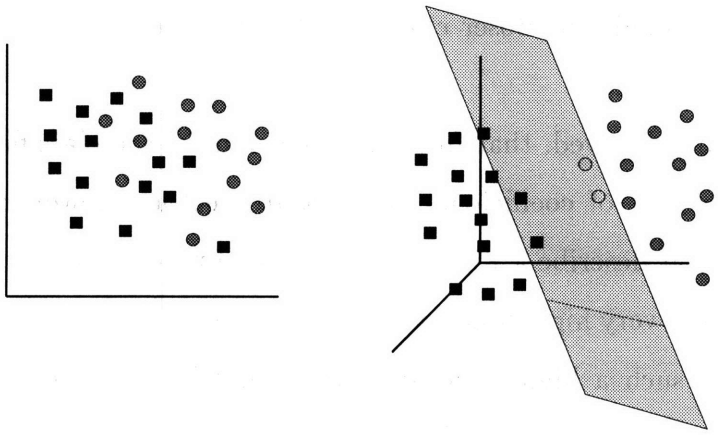
As we already hinted at, the goal of the SVM algorithm is to find a classifier that minimizes the generalization error over a set of labeled examples, $\{(x_i, y_i)\}_{i=1}^{l}$. For



(a) small margin (b) large margin

Figure 4-1: The separating hyperplane in (a) has small margin; the hyperplane in (b) has larger margin and should generalize better on out-of-sample data.

the linearly separable case, it is clear that the hyperplane that accomplishes this is the one that maximizes the margin between the two classes, where margin is the sum of the distances from the hyperplane to the closest point in each of the two classes. This concept is illustrated in Figure 4-1.



(a) original data set (b) mapped feature space

Figure 4-2: The original data set may not be linearly separable; the support vector machine uses a nonlinear kernel to map the data points into a very high dimensional feature space in which the classes have a much greater chance of being linearly separable.

Finding the best linear separating hyperplane will obviously not work for a problem whose solution has a nonlinear decision surface. Herein lies the key in the formu-

lation of the SVM algorithm. An interesting aspect of the SVM is that its decision surface depends only on the inner product of the feature vectors. This leads to an important extension since we can replace the Euclidean inner product by any symmetric positive-definite kernel $K(x, y)$ [18]. Instead of working in the original feature space of variables, x, the SVM uses this nonlinear kernel to project the original set of variables into a high dimensional feature space in which the problem has a greater chance of being linearly separable. More formally, $x \epsilon \Re^d \Rightarrow z(x) \equiv (\phi_1(x), ..., \phi_n(x)) \epsilon \Re^n$. This mapping of the feature vectors into a higher dimensional space significantly increases the discriminative power of the classifier. Changing the kernel function leads to different well known classifiers such as Gaussian RBFs, polynomial classifiers of various degrees, and MLPs. For our classification problem, we find that using a polynomial of degree two as the kernel provides good results.

The detection architecture is not hardcoded to use the support vector machinery; indeed, we could use any classification algorithm but choose the SVM because of its solid theoretical foundations and its initial success when applied to tasks such as handwritten digit recognition (Boser *et al.*, 1992[3]) and face detection (Osuna *et al.*, 1997[17]).

It should also be observed, that from the viewpoint of the classification task, we could use the whole set of coefficients as a feature vector. However, using all the wavelet functions that describe a window of 128×64 pixels in the case of pedestrians would yield vectors of very high dimensionality, as we mentioned earlier. The training of a classifier with such a high dimensionality would in turn require too large an example set. The dimensionality reduction stage of Chapter 3 serves to select the basis functions relevant for this task and to reduce their number considerably.

Chapter 5

System Architecture

5.1 The Detection System

Once we have identified the important basis functions we can use classification technique – for this thesis, we use a support vector machine – to learn the relationships between the wavelet coefficients that define the object class. In this section, we present the overall architecture of the detection system and the training process.

5.1.1 System Architecture

The system detects objects in arbitrary positions in the image and in different scales. To accomplish this task, the system is trained to detect a member of an object class centered in a detection window of a certain size — 19×19 for faces and 128×64 for pedestrians. This training stage is the most difficult part of the system training and once it is accomplished the system can detect objects at arbitrary positions by scanning all possible locations in the image by shifting the detection window. This is combined with iteratively resizing the image to achieve multi-scale detection. For our experiments with faces, we detected faces from the minimal size of 19×19 to 5 times this size by scaling the novel image from 0.2 to 1.0 times its original size, at increments of 0.1. For pedestrians, the image is scaled from 0.2 to 2.0 times its original size, again in increments of 0.1. At any given scale, instead of recomputing

the wavelet coefficients for every window in the image, we compute the transform for the whole image and do the shifting in the coefficient space.

For the face detection system, since we are using the coefficients of scale 2×2 that are computed at double density, a shift of one coefficient in the finer scale corresponds to a shift of one pixel in the image; a shift in the coarser scale coefficients of 4×4 (in quadruple density) also corresponds to a shift of one pixel in the image space. Thus, the spatial resolution for detecting faces is one pixel. For the pedestrian detection system, a shift of one coefficient in the finer scale corresponds to a shift of 4 pixels in the image and a shift in the coarse scale corresponds to a shift of 8 pixels. Since most of the coefficients in the wavelet bases are at the finer scale (the coarse scale coefficients hardly change with a shift of 4 pixels), we achieve an effective spatial resolution of 4 pixels by working in the wavelet coefficient space.

5.1.2 System Training

We train our systems using databases of positive examples gathered from outdoor and indoor scenes. The initial negative in the training database are patterns from natural scenes not containing people or faces. A combined set of positive and negative examples for a single class form the initial training database for the classifier. A key issue with the training of detection systems is that, while the examples of the target class are well defined, there are no typical examples of the negative example class – this class is immense. The main idea in overcoming this problem of defining this extremely large negative class is the use of "bootstrapping" training (Sung and Poggio, 1994[26]). In the context of the pedestrian detection system, after the initial training, we run the system over arbitrary images that do not contain any people. Any detections are clearly identified as false positives and are added to the database of negative examples and the classifier is then retrained with this larger set of data. These iterations of the bootstrapping procedure allows the classifier to construct an incremental refinement of the non-pedestrian class until satisfactory performance is achieved. This bootstrapping technique is illustrated in Figure 5-1 for pedestrians; the implementation for the face detection version is equivalent.

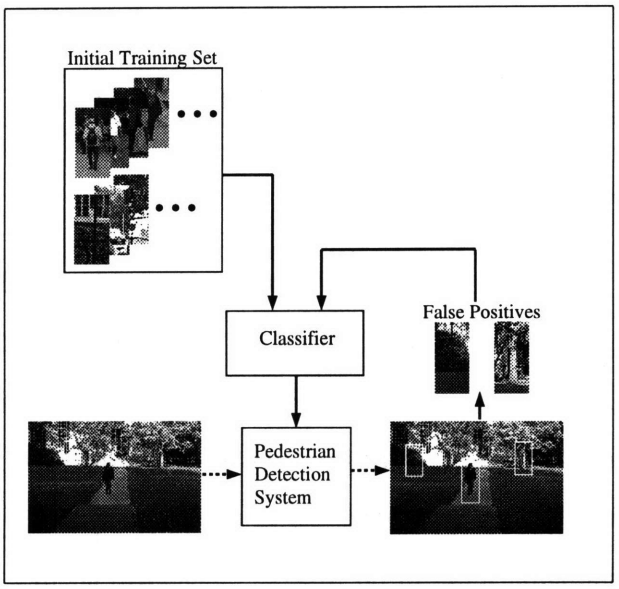


Figure 5-1: Incremental bootstrapping to improve the system performance.

Chapter 6

Experimental Results

6.1 The Experimental Results

6.1.1 Face Detection

To evaluate the face detection system performance, we start with a database of 924 positive examples and 1000 negative examples. The system then undergoes the bootstrapping cycle detailed in Section 5.1.2. For this system, the support vector classifier undergoes 5 bootstrapping steps, ending up with a total of 6000 negative examples. Out-of-sample performance is evaluated using a set of 131 faces. We evaluate the rate of false detections using a set of 50 images of natural scenes that do not contain either faces or people; a total of $9,427,479$ patterns are classified by the face detection system. To give a complete characterization of performance of the detection system, we run a large set of tests using different classification thresholds. This provides the only true way of measuring the performance of such a system; rather than give a single performance result, we compute the full spectrum of the accuracy/false detection rate tradeoff inherent in such a system. The ROC curve measuring this performance is shown in Figure 6-1. We can see that, if we allow one false detection per 5,500 windows examined, the rate of correctly detected faces is 75%. On the hand, a much more stringent system that only allows one false positive for every 50,000 windows examined has an accuracy of 40%.

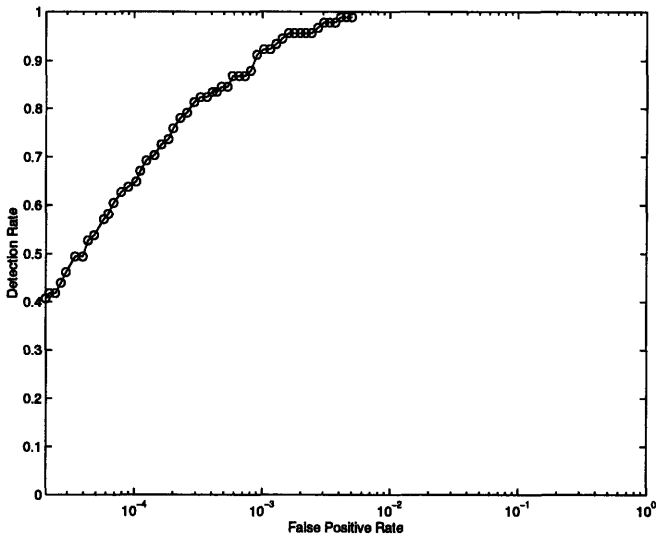


Figure 6-1: ROC curve for the frontal face detection system. The detection rate is plotted against the false detection rate, measured on a logarithmic scale. The false detection rate is defined as the number of false detections per inspected window.

In Figure 6-2 we show the results of running the face detection system over example images. The woman in the left image is not detected due to the rotation of her head; currently, the system is not able to handle large rotations such as this, but with further training on an appropriate set of rotated examples, this type of rotation could be detected. In the image on the right, all the faces are detected correctly, but there is a single incorrect detection. Again, we expect that with further training, this will be eliminated.

6.1.2 People Detection

The frontal and rear pedestrian detection system starts with 924 positive examples and 789 negative examples and goes through 9 bootstrapping steps ending up with a set of 9726 patterns that define the non-pedestrian class. We measure performance on novel data using a set of 105 pedestrian images that are close to frontal or rear views; it should be emphasized that we do not choose test images of pedestrians in perfect frontal or rear poses, rather, many of these test images represent slightly rotated or walking views of pedestrians. As with the faces, we use a set of 50 natural scenes to measure the false detection rate; the pedestrian detection system looks at 2,789,081

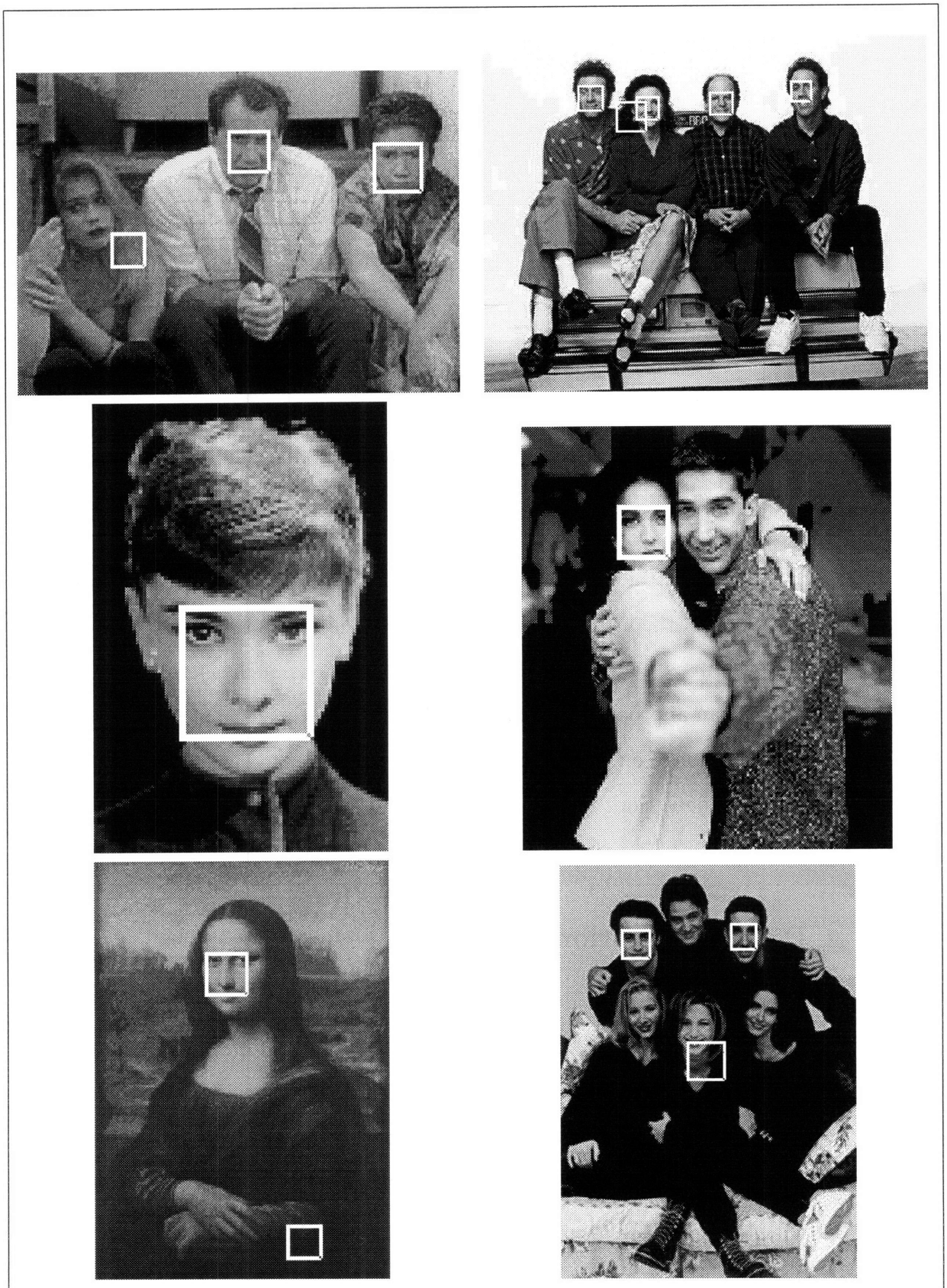


Figure 6-2: Results from the face detection system; the missed faces are due to either high degrees of rotation (top left, middle right, lower right) or occlusion (lower right).

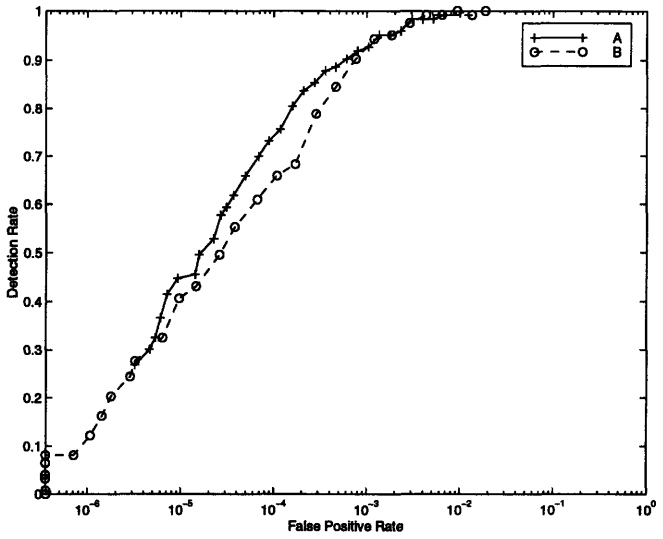


Figure 6-3: ROC curves for the frontal and rear view people detection system. The detection rate is plotted against the false detection rate, measured on a logarithmic scale. The false detection rate is defined as the number of false detections per inspected window. System B penalizes incorrect classifications of both positive and negative examples equally, while system A penalizes incorrectly classified positive examples five times more than negative examples.

patterns.

In general, the performance of any detection system exhibits a tradeoff between the rate of detection and the rate of false positives. Performance drops as we impose more stringent restrictions on the rate of false positives. To capture this tradeoff, we vary the sensitivity of the system by thresholding the output and evaluate the ROC curve, given in Figure 6-3 for two versions of the system. System B penalizes incorrect classifications of both positive and negative examples equally while system A penalizes incorrectly classified positive examples five times more than negative examples. The curve indicates that even larger penalty terms for the positive examples may improve accuracy significantly. From the curve, we can see, for example, that if we have a tolerance of one false positive for every 15,000 windows examined, we can achieve a detection rate of 70%. Figure 6-4 exhibits some typical images that are processed by the pedestrian detection system; the images are very cluttered scenes crowded with complex patterns. These images show that the architecture is able to effectively handle detection of people with different clothing under varying illumination

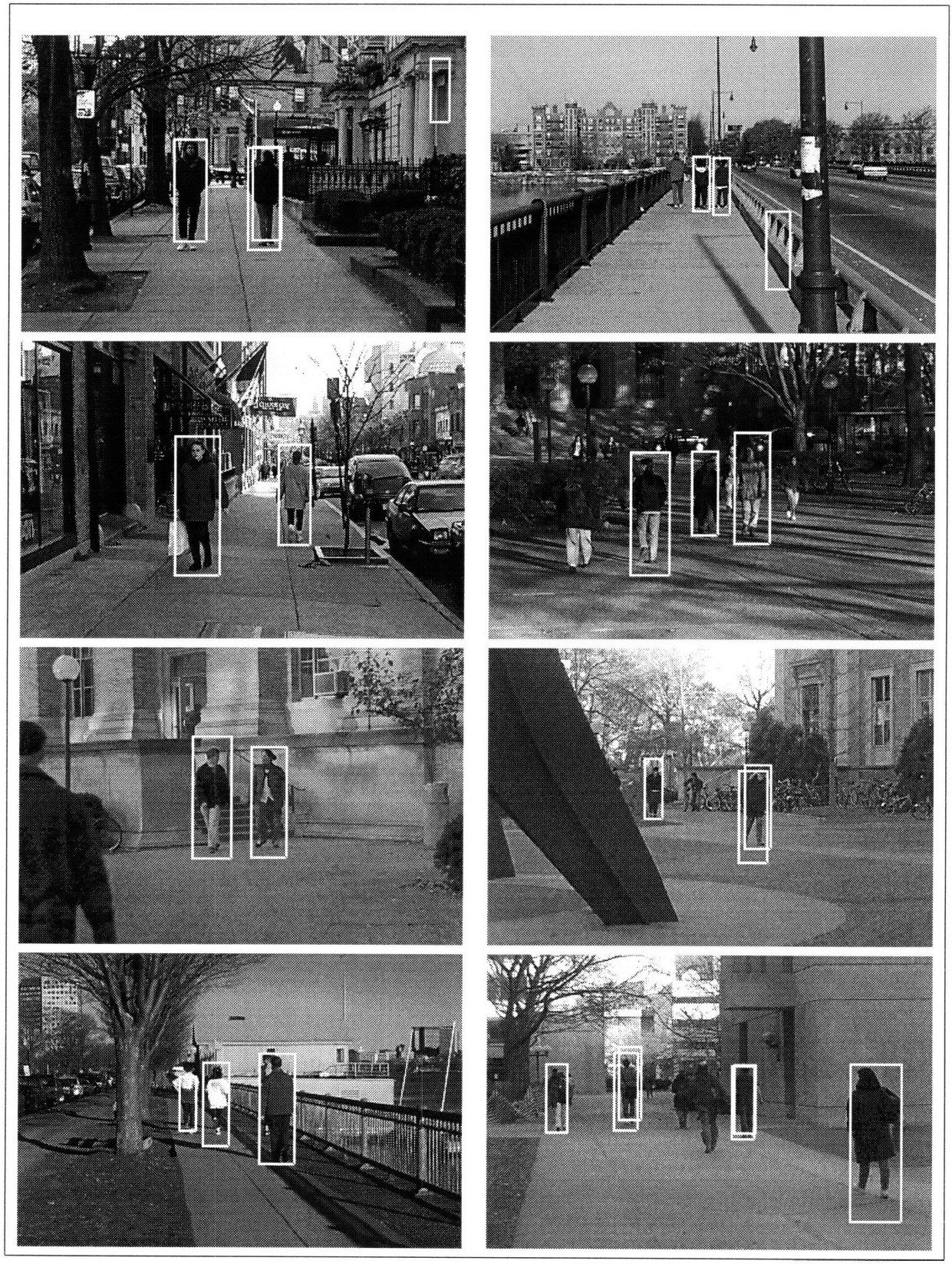


Figure 6-4: Results from the pedestrian detection system. These are typical images of relatively complex scenes that are used to test the system.

conditions.

Considering the complexity of these scenes and the difficulties of object detection in cluttered scenes, we consider the above detection rates to be high. We believe that additional training and refinement of the current systems will reduce the false detection rates further; initial work in using the same framework to detect side views of pedestrians is promising.

Chapter 7

Motion

This thesis presents a technique for boosting detection accuracy when we are processing video sequences; the system is applied to detecting pedestrians in cluttered scenes with promising results. In Section 1.2 we describe several systems that have been developed to detect walking people. All of the systems assume a static camera, clearly a severe restriction for a practical system that might be integrated in a driver assistance system, for instance; the one system that does not assume a static camera, that of Heisele *et al.*, 1997[6], is actually only doing tracking of manually labeled color blobs, so is not yet a viable pedestrian detection system. Since these systems heavily rely on motion to determine the location of pedestrians in the images, they would also have problems finding pedestrians that are not moving. This section describes a new module that can be integrated into a static detection system to improve results over video sequences, while maintaining the same degree of accuracy for objects that are not moving.

For this approach, we do not require that detailed motion information be recovered from a video sequence – our goal is to determine the general areas of motion in an image. The class of possible moving objects is limited in the video sequences we look at – typically, only cars and people will be moving – so we can use this a priori information to relax the strictness of the classifier in the regions of motion. It is important to reiterate that, in contrast to most other people detection systems that rely on motion to accomplish detection, our motion extension does not compromise

the ability of the system to detect non-moving objects.

We start our discussion with a description of the algorithms used to recover the motion in consecutive frames for the two cases of static and moving cameras and then describe the relaxation of the classifier.

7.1 Motion Estimation

Given that our camera is fixed, we are guaranteed that the background is static, so to find motion in this case, we do a simple differencing of consecutive frames to obtain the areas where there are moving objects. Due to camera distortion, the background pixel values recovered by the camera may not be exactly the same from frame to frame, so rather than a strict differencing, we threshold the results of the differencing to allow for small changes in illumination. This method has been used in several systems (Hogg, 1983[7], Wren *et al.*, 1995[30]) that also assume that the only moving objects will be people. It is easy to see that this is a trivial case; we can exactly recover where motion has occurred very easily.

In the case of moving camera or moving background, a simple change detection strategy will not work. This case of a moving camera or a moving background is handled by a class of algorithms called optical flow algorithms. Here, we are not guaranteed that the same pixel location in consecutive frames will correspond to the same object. On the other hand, we can be confident that in subsequent frames, the same location will correspond to a pixel in a small neighborhood of the original pixel. It is exactly this constraint that optical flow algorithms use to determine motion information.

More formally, the goal in attempting to recover motion is to determine the vectors Δx and Δy associated with a gray level image, B, that describe the position of each pixel relative to a reference image, A. For our case, A is frame t of the video sequence, and B is the subsequent frame t+1; for the static camera case, the vectors Δx and Δy will be vectors of zeros. This definition is summarized in the optical flow equation:

$$F(x, y, t+1) = F(x + \Delta x(x, y), y + \Delta y(x, y), t) \tag{7.1}$$

where $(\Delta x(x, y), \Delta y(x, y))$ is the flow field describing the pixelwise correspondences. The algorithm introduces two assumptions that make the solution tractable: the change in adjacent flow vectors varies slowly – a smoothness assumption – and the flow vectors themselves are small. The optical flow constraint equation,

$$\frac{\partial F}{\partial t}(x, y) = \Delta x(x, y)\frac{\partial F}{\partial x}(x, y) + \Delta y(x, y)\frac{\partial F}{\partial y}(x, y) \tag{7.2}$$

is a first-order approximation to the Taylor series expansion of equation 7.1 (which is an underdetermined equation in two unknowns) and is solved over small neighborhoods of points.

This approximation is accurate for displacements of less than one pixel, so the optical flow algorithm we use for our system (Bergen and Hingorani, 1990[1]) uses a coarse-to-fine strategy where flow fields are iteratively estimated for higher resolutions of a Gaussian pyramid representation of the image. This type of strategy enables facilitates the efficient computation of flow fields where the image-to-image displacements are quite large. Then, to obtain the fine pixelwise correspondences, the algorithm computes the level-by-level correspondences and adds the refinements in the flow fields together.

Since our detection system is based on static analysis of a single image, we do not need to recover full motion information; rather, we assist the detection module by indicating probable location of a moving object. This can be done without recovering camera ego-motion, which is a difficult task; our algorithm identifies the relative motion of moving objects relative to the background. This is accomplished by identifying discontinuities in the optical flow field, corresponding to boundaries of the objects, and then using morphological operators to define the full region of interest. To locate discontinuities in the flow field, we compute the L_2 norm of adjacent flow vectors,

$$L_2(I_{i,j}, I_{i,j+1}) = \sqrt{(\Delta x(i, j))^2 + (\Delta y(i, j))^2} \tag{7.3}$$

and threshold this distance measure such that if $L_2(I_{i,j}, I_{i,j+1}) > M$, we have found a discontinuity in the flow field.

The collection of the discontinuity points will be the boundary between the background and a moving object in the foreground. The output of this processing is a binary image where pixel values of one indicate the boundary of a moving and non-moving region and pixel values of zero indicates a non-boundary.

It is important to note that we do not need to recover detailed motion information; rather, we are simply looking for general regions in which we are confident that there are moving objects.

7.2 Morphological Processing

The output of the optical flow processing is a binary image where pixel values of one indicate the boundary of a moving and non-moving region and pixel values of zero indicates a non-boundary. Once we identify the motion boundaries, we must process them to identify the full region of interest. Since we cannot assume that a single outlined moving region corresponds to a single moving object, we must fully process the interior of the region to find any people and label the pixels within the moving boundaries with values of one as well. To accomplish this we make use of some simple concepts from mathematical morphology, under the assumption that there will be some moving areas (arms, etc.) within the boundaries of a moving region.

Mathematical morphology (Serra, 1982[21], Dougherty, 1992[5], Korn *et al.*, 1996[9]) is a quantitative theory that formalizes notions of shape and structure of objects. We use some simple transforms from mathematical morphology to process the image to yield a representation where pixel values of one indicate motion, not just the boundary of a motion/non-motion region. These operations are defined on $\Re^2$; the extensions to the discrete case are straightforward.

Let X denote a binary shape (ie. pixel values of one) in $\Re^2$ space, let $B^s = \{-b | b \epsilon B\}$ denote the symmetrical set of B with respect to the origin, and let X_b denote the translate of X by the vector b. We define the dilation of X by B by:

$$X \oplus B^s = \bigcup_{b \epsilon B} X_{-b} = \{(x,y) \epsilon \Re^2 | B_{(x,y)} \cap X \neq \emptyset\} \tag{7.4}$$

Dilation corresponds to growing a region by tracing the center of the structuring element, B, along the boundary and adding the points covered by B to the set X. The complementary operation to dilation is erosion,

$$X \ominus B^s = \bigcap_{b \epsilon B} X_{-b} = \{(x,y) \epsilon \Re^2 | B_{(x,y)} \subseteq X\} \tag{7.5}$$

which corresponds to shrinking a region by tracing the center of the structuring element, B, along the boundary and removing the points covered by B from the set X.

The structural element we use is a square with a side length of 10 pixels. Initially, the image is smoothed to eliminate noise in the flow field. Then the dilation transform is applied to the image to enlarge the area in which we will search for pedestrians. Figure 7-1 shows a sample image with the sequence of transforms that is applied.

7.3 Relaxing the Classifier

The result of the morphological processing is a binary image with the areas of motion clearly identified. Since our test images are of typical cluttered street scenes, the types of moving objects we encounter is very limited; usually, they will be either people or cars and we can use this hint to improve detection. By lowering the strictness of the classifier, we can detect pedestrians that would not normally be detected by the base system; typically they are pedestrians that are in an off-frontal pose and so are classified as non-pedestrians by the base system. The new threshold of classification for the moving regions is determined empirically, over a sequence of images.

We test the system over a sequence of 208 frames; the detection results are shown in Table 7.1. Out of a possible 827 pedestrians in the video sequence – including side views for which the system is not trained – the base system correctly detects 360 (43.5%) of them with a false detection rate of 1 per 236,500 windows. The system

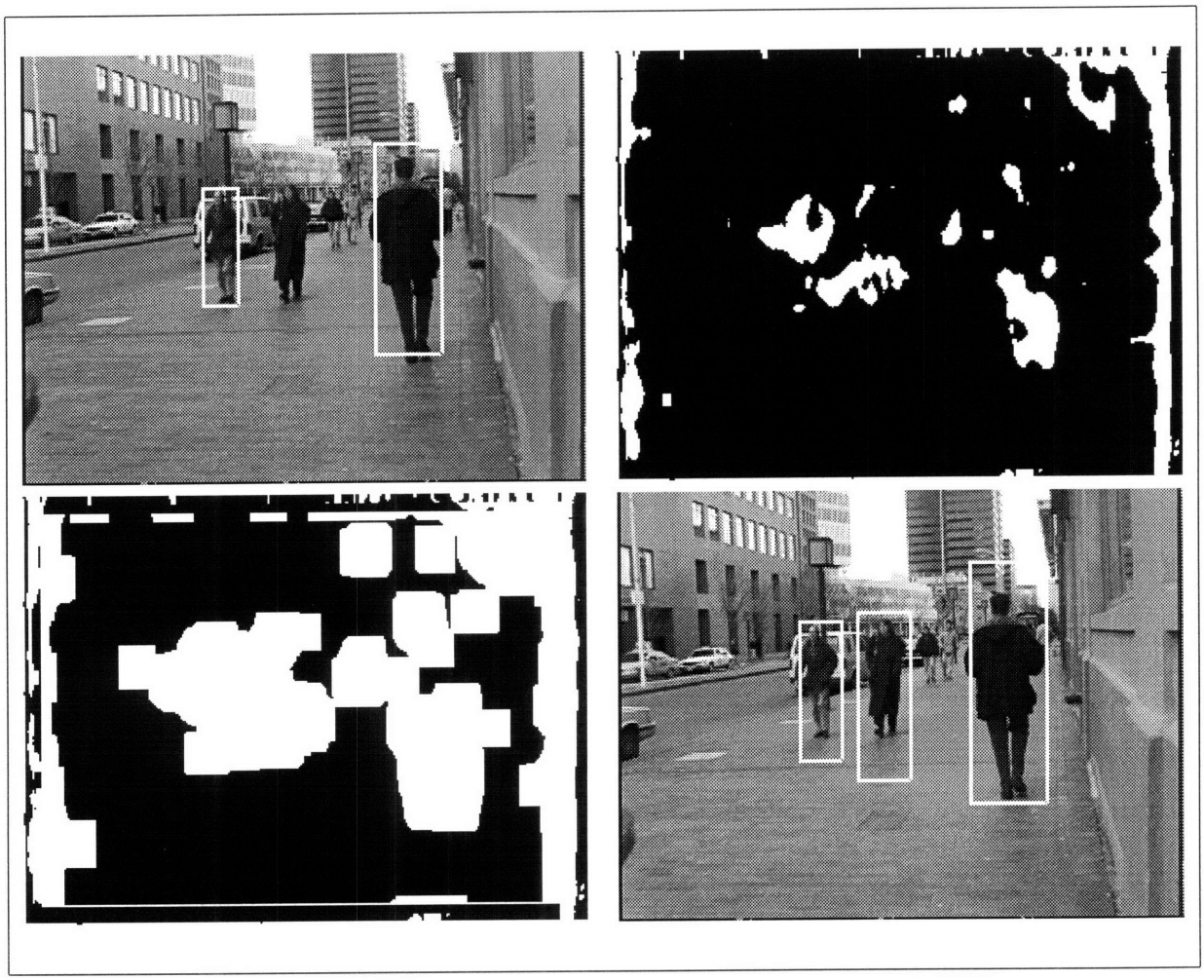


Figure 7-1: The sequence of steps in the motion-based module. The upper left image shows the static detection results, the upper right image shows the thresholded motion discontinuities, the lower left image shows the result of using the morphological operator, and the lower right image shows the improved detection results.

enhanced with the motion module detects 445 (53.8%) of the pedestrians, a 23.7 % increase in detection accuracy, while maintaining a false detection rate of 1 per 90,000 windows. It is important to iterate that the detection accuracy for non-moving objects is not compromised – in the areas of the image where the optical flow algorithm and subsequent morphological processing does not find motion, the classifier simply runs as before. Furthermore, the majority of the false positives in the system enhanced with the detection module were partial body detections, ie. a detection with the head cut off, which were still counted as false detections. Taking this factor into account, the false detection rate is even lower.

	Detection Rate	*False Positive Rate (per window)*
Base system	43.5%	1:236,500
Motion extension	53.8%	1:90,000

Table 7.1: Performance of the pedestrian detection system with the motion-based extensions, compared to the base system.

This relaxation paradigm has difficulties when there are a large number of moving bodies in the frame; the region of motion that is determined will be large and is susceptible to more false positives. Based on our results, though, we feel that this integration of a trained classifier with the module that provides motion cues could be extended to other systems as well.

Chapter 8

Conclusion

In this thesis, we describe a general framework for object detection in cluttered scenes based on the idea of an overcomplete wavelet representation and a novel motion based module that improves detection accuracy over video sequences when applied to detecting pedestrians. The motion-based extension introduced in this thesis allows us to use a moving camera and makes no assumptions on the motion of pedestrians in the scene, that is, the system is still able to detect non-moving pedestrians. Previous work in people detection has assumed a fixed camera and moving people; these limitations severely restrict the range of possible practical applications.

The wavelet representation we use yields not only a computationally efficient algorithm but an effective learning scheme as well. The core detection engine automatically learns the characteristics of an object class and is applied to both face and people detection with promising results. The success of the wavelet representation for face and people detection comes from its ability to capture high-level knowledge about the object class (structural information expressed as a set of constraints on the wavelet coefficients) and incorporate it into the low-level process of interpreting image intensities. Attempts to directly apply low-level techniques such as edge detection and region segmentation are likely to fail in the type of images we analyze since these methods are not robust, are sensitive to spurious details, and give ambiguous results. Using the wavelet representation, only significant information that characterizes the object class — as obtained in the learning phase — is evaluated and used.

We also present an extension that uses motion cues to improve pedestrian detection accuracy over video sequences. This module is appealing in that, unlike most systems, it does not totally rely on motion to accomplish detection; rather, it takes advantage of the a priori knowledge that the class of moving objects is limited while still being able to detect non-moving pedestrians.

We discuss several directions for future work and several possible applications of the techniques described in this thesis.

8.1 Future Work

To demonstrate the generality of the detection system we have developed, we should apply it to new object classes. Interesting classes of objects this could be applied to are cars, airplanes, and more views of pedestrians – side, sitting, walking, and so on.

The current system has difficulty detecting occluded objects. One way to deal with occluded objects could be to train several different lower level object detectors for different body parts and to combine the results using a voting process to create a pedestrian detector. For instance, we could develop arm, head, and leg detectors; if a certain number of detections in a proper configuration occur in a small area of an image, we would be confident that there was a person at that location. This type of system would effectively handle some occlusions.

The determination of the regions of motion for the motion-based module currently results in too much noise; improving this part of the system would result in improved performance in the form of higher detection accuracy and a lower false positive rate.

To be able and apply the techniques to practical areas, the detections need to be performed at near real-time frame rates of at least one or two frames per second. Currently, the system takes several minutes to process a single image; it should be possible to reduce this time to close to the desired rate with additional work.

It would also be interesting to try and learn to detect moving people by using the same architecture trained over the optical flow map of moving people. The wavelet representation would be finding differences in motion, rather than differences in in-

tensity.

8.2 Applications

The ability of our system to deal with both stationary and moving cameras, and stationary and moving objects, facilitates a wide range of applications. Our testbed system of pedestrian detection could be integrated into a driver assistance system that alerts the driver to pedestrians in the path of the automobile and also has surveillance applications.

This work has clear applications for the search and indexing of video databases; by using this system, we could quickly identify objects of interest and then pass the results to a dedicated recognition module to determine if the object is a specific instance, for instance, a specific person.

The strength of our system comes from the expressive power of the redundant wavelet representation – this representation effectively encodes the intensity relationships of certain pattern regions that define a complex object class. The encouraging results of our system in two different domains, faces and people, suggest that the detection approach described in this paper may well generalize to several other object detection tasks. When coupled with the novel motion-based module presented here, the system's performance is very encouraging over a pedestrian detection task.

Bibliography

[1] J.R. Bergen and R. Hingorani. Hierarchical motion-based frame rate conversion, April 1990.

[2] M. Betke and N. Makris. Fast object recognition in noisy images using simulated annealing. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 523–20, 1995.

[3] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optim margin classifier. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–52. ACM, 1992.

[4] L. Campbell and A. Bobick. Recognition of human body motion using phase space constraints. Technical Report 309, Media Laboratory, Massachusetts Institute of Technology, 1995.

[5] E.R. Dougherty. *An Introduction to Morphological Image Processing*, volume TT9. SPIE Press, 1992.

[6] B. Heisele, U. Kressel, and W. Ritter. Tracking non-rigid, moving objects based on color cluster flow. In *CVPR '97*, 1997. to appear.

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

[8] C.E. Jacobs, A. Finkelstein, and D.H. Salesin. Fast multiresolution image querying. *SIGGRAPH95*, August 1995. University of Washington, TR-95-01-06.

[9] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas. Fast nearest neighbor search in medical image databases. Computer Science Technical Report CS-TR-3613, University of Maryland, March 1996.

[10] H. Lakany and G. Hayes. An algorithm for recognising walkers. In J. Bigun, G. Chollet, and G. Borgefors, editors, *Audio- and Video-based Biometric Person Authentication*, pages 112–118. IAPR, Springer, 1997.

[11] M.K. Leung and Y-H. Yang. Human body motion segmentation in a complex scene. *Pattern Recognition*, 20(1):55–64, 1987.

[12] M.K. Leung and Y-H. Yang. A region based approach for human body analysis. *Pattern Recognition*, 20(3):321–39, 1987.

[13] S.G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–93, July 1989.

[14] S. McKenna and S. Gong. Non-intrusive person authentication for access control by visual tracking and face recognition. In J. Bigun, G. Chollet, and G Borgefors, editors, *Audio- and Video-based Biometric Person Authentication*, pages 177–183. IAPR, Springer, 1997.

[15] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. Technical Report 326, Media Laboratory, Massachusetts Institute of Technology, 1995. also in 5th ICCV June 1995.

[16] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *CVPR '97*, 1997.

[17] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. A.I. Memo 1602, MIT A. I. Lab., 1997.

[18] F. Riesz and B. Sz.-Nagy. *Functional Analysis.* Ungar, New York, 1955.

[19] K. Rohr. Incremental recognition of pedestrians from image sequences. *Computer Vision and Pattern Recognition*, pages 8–13, 1993.

[20] H.A. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. Technical Report CMU-CS-95-158, School of Computer Science, Carnegie Mellon University, July/November 1995.

[21] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1982.

[22] P. Sinha. Object Recognition via Image Invariants: A Case Study. In *Investigative Ophthalmology and Visual Science*, volume 35, pages 1735–1740. Sarasota, Florida, May 1994.

[23] Pawan Sinha. Qualitative image-based representations for object recognition. *MIT AI Lab-Memo*, No. 1505, 1994.

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

[25] K-K. Sung. *Learning and Example Selection for Object and Pattern Detection.* PhD thesis, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, December 1995.

[26] K-K. Sung and T. Poggio. Example-based learning for view-based human face detection. A.I. Memo 1521, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, December 1994.

[27] T. Tsukiyama and Y. Shirai. Detection of the movements of persons from a sparse sequence of tv images. *Pattern Recognition*, 18(3/4):207–13, 1985.

[28] R. Vaillant, C. Monrocq, and Y. Le Cun. Original approach for the localisation of objects in images. *IEE Proc.-Vis. Image Signal Processing*, 141(4), August 1994.

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

[30] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. Technical Report 353, Media Laboratory, Massachusetts Institute of Technology, 20 Ames Street, Cambridge, MA 02139, 1995.

[31] A. Yuille, P. Hallinan, and D. Cohen. Feature Extraction from Faces using Deformable Templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.