Finding People in Images and Videos

Navneet Dalal

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Spécialité : **Imagerie, Vision et Robotique**

dans le carde de l'École Doctorale
Mathématiques, Sciences et Technologie de l'Information

présentée et soutenue publiquement
par

Navneet DALAL

le 17 Juillet, 2006

Finding People in Images and Videos

JURY

M.	James L. CROWLEY	Président
M.	Martial HEBERT	Rapporteur
M.	Luc Van GOOL	Rapporteur
M.	Shai AVIDAN	Examinateur
Mme.	Cordelia SCHMID	Directeur de thèse
M.	William J. TRIGGS	Directeur de thèse

Thèse préparée dans le laboratoire GRAVIR – IMAG au sein du
Projet LEAR, INRIA Rhône-Alpes
655 avenue de l'Europe, 38334 Saint Ismier, France.

Résumé

Cette thèse propose une solution pour la détection de personnes et de classes d'objet dans des images et vidéos. Le but principal est de développer des représentations robustes et discriminantes de formes visuelles, qui permettent de décider si un objet de la classe apparaît dans une région de l'image. Les décisions sont basées sur des vecteurs de descripteurs visuels de dimension élevée extraits des régions. Afin d'avoir une comparaison objective des différents ensembles de descripteurs, nous apprenons une règle de décision pour chaque ensemble avec un algorithme de type machine à vecteur de support linéaire. Piloté entièrement par les données, notre approche se base sur des descripteurs d'apparence et de mouvement de bas niveau sans utiliser de modèle explicite pour l'objet a détecter. Dans la plupart des cas nous nous concentrons sur la détection de personnes – classe difficile, fréquente et particulièrement intéressante dans applications come l'analyse de film et de vidéo, la détection de piétons pour la conduite assistée ou la surveillance. Cependant, notre méthode ne fait pas d'hypothèse forte sur la classe à reconnaître et elle donne également des résultats satisfaisants pour d'autres classes comme les voitures, les motocyclettes, les vaches et les moutons.

Nous apportons quatre contributions principales au domaine de la reconnaissance visuelle. D'abord, nous présentons des descripteurs visuels pour la détection d'objets dans les images statiques : les grilles d'histogrammes d'orientations de gradients d'image (en anglais, HOG – Histogrammes of Oriented Gradients). Les histogrammes sont évalués sur une grille de blocs spatiaux, avec une forte normalisation locale. Cette structure assure à la fois une bonne caractérisation de la forme visuelle locale de l'objet et la robustesse aux petites variations de position, d'orientation spatiale, d'illumination locale et de couleur. Nous montrons que la combinaison de gradients peu lissés, une quantification fine de l'orientation et relativement grossière de l'espace, une normalisation forte de l'intensité, et une méthode évoluée de ré-apprentissage des cas difficiles permet de réduire le taux de faux positifs par un à deux ordres de grandeur par rapport aux méthodes précédentes.

Deuxièmement, afin de détecter des personnes dans les vidéos, nous proposons plusieurs descripteurs de mouvement basés sur le flot optique. Ces descripteurs sont incorporés dans l'approche précédente. Analogues aux HOG statiques, ils substituent aux gradients d'image statique les différences spatiales du flot optique dense. L'utilisation de différences minimise l'influence du mouvement de la caméra et du fond sur les détections. Nous évaluons plusieurs variations de cette approche, qui codent soit les frontières de mouvement (motion boundaries), soit les mouvements relatifs des paires de régions adjacentes. L'incorporation du mouvement réduit le taux de faux positifs d'un ordre de grandeur par rapport à l'approche précédente.

Troisièmement, nous proposons une méthode générale pour combiner les détections multiples basées sur l'algorithme "mean shift" pour estimer des maxima de densité à base de noyaux. L'approche tient compte du nombre, de la confiance et de l'échelle relative des détections.

Finalement, nous présentons un travail en cours sur la façon de créer de un détecteur de personnes à partir de plusieurs détecteurs de parties – en occurrence le visage, la tête, le torse, et les jambes.

Abstract

This thesis targets the detection of humans and other object classes in images and videos. Our focus is on developing robust feature extraction algorithms that encode image regions as high-dimensional feature vectors that support high accuracy object/non-object decisions. To test our feature sets we adopt a relatively simple learning framework that uses linear Support Vector Machines to classify each possible image region as an object or as a non-object. The approach is data-driven and purely bottom-up using low-level appearance and motion vectors to detect objects. As a test case we focus on person detection as people are one of the most challenging object classes with many applications, for example in film and video analysis, pedestrian detection for smart cars and video surveillance. Nevertheless we do not make any strong class specific assumptions and the resulting object detection framework also gives state-of-the-art performance for many other classes including cars, motorbikes, cows and sheep.

This thesis makes four main contributions. Firstly, we introduce grids of locally normalised Histograms of Oriented Gradients (HOG) as descriptors for object detection in static images. The HOG descriptors are computed over dense and overlapping grids of spatial blocks, with image gradient orientation features extracted at fixed resolution and gathered into a high-dimensional feature vector. They are designed to be robust to small changes in image contour locations and directions, and significant changes in image illumination and colour, while remaining highly discriminative for overall visual form. We show that unsmoothed gradients, fine orientation voting, moderately coarse spatial binning, strong normalisation and overlapping blocks are all needed for good performance. Secondly, to detect moving humans in videos, we propose descriptors based on oriented histograms of differential optical flow. These are similar to static HOG descriptors, but instead of image gradients, they are based on local differentials of dense optical flow. They encode the noisy optical flow estimates into robust feature vectors in a manner that is robust to the overall camera motion. Several variants are proposed, some capturing motion boundaries while others encode the relative motions of adjacent image regions. Thirdly, we propose a general method based on kernel density estimation for fusing multiple overlapping detections, that takes into account the number of detections, their confidence scores and the scales of the detections. Lastly, we present work in progress on a parts based approach to person detection that first detects local body parts like heads, torso, and legs and then fuses them to create a global overall person detector.

Acknowledgements

First and foremost I want to express my gratitude and thanks to my thesis directors, Bill Triggs and Cordelia Schmid. I feel Bill has been an ideal adviser. He has been a great mentor, a collaborator, a guide, and a friend. This thesis would not have been completed without his commitment and diligent efforts which not only influenced the content of the thesis but also the language in which it has been conveyed. He not only showed how to exhaustively explore different methodologies and analyse results, but also imbibed in me the need for perfection not only when performing research but also when communicating the results. I am also grateful to Cordelia who always encouraged my initiatives to make the PhD research more applicable. I thank her for offering me the opportunity to do a PhD on such an interesting and challenging topic and providing me the platform for this thesis. She has also been very supportive for letting me take time out of my thesis and venture into writing a book. I can never forget the support Bill and Cordelia showed towards the fag end of my thesis when they learned about my job offer and the constraint of finishing the remaining part of research and the dissertation in two months time. I would promptly receive detailed reviews, comments and corrections of the chapters despite the amount of work at their end, reflecting their commitment and devotion.

Thanks to my thesis rapporteurs, Martial Hebert and Luc Van Gool, for earnestly reading the thesis and their detailed comments on the manuscript in a relatively short time. I would like to thank Shai Avidan, my thesis examiner, for his interest in my work and James Crowly, the thesis committee president, for making his time available for me.

I am also grateful to Radu Horaud, my MSc. thesis adviser. Radu has always been very open and a great support. He encouraged me to go to Siemens Corporate Research for summer internship and collaborate with Dorin Comaniciu, from whose association I have tremendously benefited, mostly from Dorin's combination of mathematical rigour, his intuitive ideas and his original views.

I would like to thank all my friends and colleagues from LEAR and MOVI for making INRIA a fun and interesting place to work. Much respect to my office mates Ankur, Matthijs and Guillaume for putting up with me during these past three years. I shared many interesting discussions and work time fun with them. The help received from them both personally and professionally is also immense, and Matthijs aide has not stopped pouring even after my shift from Grenoble. I also want to thank Srikumar, Gyuri, Eric, Joost, Jakob, Vittorio, Marcin, Diane, Stan, Cristi, João, David, Magda, Marta, Edmond, Rémi, Fédéric, Dana, Jianguo, and Pau for the lively working atmosphere and interesting lunchtime discussions. I would also like to thank Anne for bearing with me for all the administrative goof ups.

I can never forget my good friends Peter, Irina, Trixi and Markus for making my stay at Grenoble so much more pleasant. They introduced me to the French culture and language

though none of them are French themselves! They were like a second family to me and were always ready to help, guide, and ”feed” me. I still can not help missing delicious Tiramisu that Irina used to bake.

I cannot end without thanking my parents and my brother, for their absolute confidence in me. My final words go to Priyanka, my wife. In my line of work spouses suffer the most. Thank you for putting up with my late hours, many spoiled weekends and vacations, and above all for staying by my side.

A conclusion is simply the place where someone got tired of thinking.

– Arthur Block

Contents

1

Introduction

Computers have become ubiquitous in our daily lives. They perform repetitive, data intensive and computational tasks, more efficiently and more accurately than humans. It is natural to try to extend their capabilities to perform more intelligent tasks such as analysis of visual scenes or speech, logical inference and reasoning – in brief the high-level tasks that we humans perform subconsciously hundreds of times every day with so much ease that we do not usually even realise that we are performing them. Take the example of the human visual system. Our daily lives are filled with thousands of objects ranging from man made classes like cars, bicycles, buildings, tables, chairs to natural ones like sheep, cows, trees, leaves, rocks, mountains and humans. Any given class has a huge intra-class variation. For example "car" can be used to denote many four wheeled vehicles, including various sub categories like a sedan, hunchback, station wagon or SUV. The exact type, colour and viewpoint of a car is irrelevant to the decision that an object is a car. Similarly, we are able to detect people under widely varied conditions – irrespective of the colour or kind of clothing, pose, appearance, partial occlusions, illumination or background clutter. Computers are currently far behind humans in performing such analysis and inference.

Thus one goal of researchers working in computer vision and machine intelligence has been to grant computers the ability to see – visual analysis and interpretation of images or videos. One of the primary tasks is the detection of different classes of objects in images and videos. Such a capability would have many applications, for example in human computer interaction, robotics, automatic analysis of personal or commercial digital media content, automated manufacturing processes, and smart autonomous vehicles.

This chapter introduces the problem of object detection – in particular detection of people –, discusses the challenges involved, and briefly presents our approach highlighting the contributions of the thesis. Throughout this dissertation, object detection refers to detecting object categories, not particular objects. We start in Sect. 1.1 with a general description of our goal in object detection, and then describe the key applications of person detection, which is the problem that we focus on in this thesis. Section 1.2 discusses the difficulties of visual object detection. Section 1.3 presents a brief background and gives some general perspectives on object detection and Sect. 1.4 summarises our own approach. We conclude with a discussion of the major contributions and an outline of the structure of the dissertation in Sect. 1.5 and Sect. 1.6.

1.1 The Goal

This thesis targets the problem of visual object detection in images and videos. In particular, it addresses the issue of building object detectors from a computer vision point of view, where the detectors search given images or videos for objects and localise them. For a more precise definition of our goal we can view an object detector as a combination of two key building blocks: a feature extraction algorithm that encodes image regions or parts of videos as feature vectors, and a detector that uses the computed features to provide object/non-object decisions. Our main contributions relate the first part – the encoding of image or video regions into feature vectors. This is fundamental to creating robust object detectors as unlike text documents where two words either match exactly or are different, matching or categorising objects in images is inherently ambiguous. Many factors contribute to this ambiguity, including the image formation processes, variations in illumination, partial occlusions, intra-class variation and context. Here context denotes the overall detection goal. For example if we are looking for our car in a parking lot, the goal becomes to locate a *particular car* of a specific make, model and colour. Compare this to a scenario where someone is looking at a painting and is asked to locate a car. The goal in this case is to look for *cars* in a broader perspective – something which resembles a car in form, possibly with a texture quite unlike real cars. Colour and model become irrelevant. Thus the matching criterion has changed owing to the context. This thesis focuses on general purpose object detectors that do not make strong contextual assumptions. More details of the challenges involved are given in Sect. 1.2. The introduction of more robust discriminant image descriptors simplifies the classification task allowing objects to be discriminated more easily with less training data and less sophisticated learning methods.

1.1.1 Person Detection and its Applications

Although our feature sets are use full in general, our experiments will focus on the problem of *finding people in images and videos*. Person detection is a challenging task, with many applications that has attracted lot of attention in recent years. Consider the case of personal digital content analysis, where typical content is images taken during a vacation, at a party or at some family occasion. Statistics show that even digital camera owners who use their cameras only occasionally can take as many as 10,000 photos in just 2-3 years, at which point it becomes tedious to manually search and locate these photos. Intelligent digital content management software that automatically adds tags to images to facilitate search is thus an important research goal. Most of the images taken are of people, so person detection will form an integral part of such tools. For commercial film and video contents, person detection will form an integral part of applications for video on demand and automatic content management. In conjunction with face and activity recognition, this may facilitate search for relevant contents or searches for few relevant sub-sequences. Figure 1.1 shows some images containing people from a collection of personal digital images. In this thesis we will mainly study the detection of fully visible people in more or less upright poses. The human body model and appearance are relatively constrained in such poses. One can thus learn relevant feature vectors or descriptors for either the whole body image or for various sub-parts (e.g. legs, arms) and build a detector based on these.

Person detectors are also being explored for the detection of pedestrians by smart cars. Typically, information from multiple sensors such as stereo and infra-red cameras is fused and domain specific knowledge such as the fact that pedestrians often traverse cross walks is exploited, but performance is still far below than needed for such systems to be used in the real world. More robust person detectors would certainly help to improve the overall system performance.

Fig. 1.1. *Some images from a collection of personal digital photos. This collection, the INRIA static person detection data set, is the benchmark data for most of the analysis in this thesis. The collection contains people with a wide range of variation in pose, appearance, clothing, illumination and background. The subjects are always upright but some images contain partial occlusions.*

Another application is in video surveillance and security where real-time systems are needed to analyse and process video sequences for intrusion detection.

1.2 Challenges

The foremost difficulty in building a robust object detector is the amount of variation in images and videos. Several factors contribute to this:

- Firstly, the image formation process suppresses 3-D depth information and creates dependencies on viewpoint such that even a small change in the object's position or orientation

w.r.t. the camera centre may change its appearance considerably. A related issue is the large variation in scales under which an object can be viewed. An object detector must handle the issues of viewpoint and scale changes and provide invariance to them.

- Secondly, most natural object classes have large within-class variations. For example, for humans both appearance and pose change considerably between images and differences in clothing create further changes. A robust detector must try to achieve independence of these variations.

- Thirdly, background clutter is common and varies from image to image. Examples are images taken in natural settings, outdoor scenes in cities and indoor environments. The detector must be capable of distinguishing object class from complex background regions.

  The previous two difficulties present conflicting challenges, that must be tackled simultaneously. A detector that is very specific to a particular object instance will give less false detections on background regions, but will also miss many other object instances while an overly general detector may handle large intra-class variations but will generate a lot of false detections on background regions.

- Fourthly, object colour and general illumination varies considerably, for example direct sunlight and shadows during the day to artificial or dim lighting at night. Although models of colour and illumination invariance have made significant advances, they still are far from being effective solutions when compared to human and mammalian visual systems which are extremely well adapted to such changes, *c.f*. Land [1959a,b], Daw [1984], Ingle [1985]. Thus a robust object detector must handle colour changes and provide invariance to a broad range of illumination and lighting changes.

- Finally, partial occlusions create further difficulties because only part of the object is visible for processing.

Figure 1.1 provides some examples illustrating these difficulties for person detection. Note the amount of variation in the images, in particular the range of human poses, the variations in appearance and illumination, the differences in clothing and background clutter and the range of different scenes in which the images are taken.

Another challenge is the amount of high-level context and background information that humans can deal with but that computers still lack. Figure 1.2 highlights some instances where humans use background information and reasoning to prune false detections and to choose correct ones. Figure 1.2(a,b) show cases (left bounding boxes in both images) where person-like image regions are detected, but where taking into account background information, perceived depth, and subtle details of the appearance we can conclude that these are in fact images of real sized people but not actual people. Both images also contain cases (right bounding boxes) where even though only a partly occluded silhouette is available, we are still able to conclude that these are actual people because given the overall surrounding and context no other phenomenon explains them.

Object detection in videos creates additional challenges, even though the motion fields in videos provide extra information that is not present in static images. This is especially true in our scenario where moving cameras and backgrounds are allowed. Figure 1.3 shows a few pairs of consecutive frames taken from commercial film and video sequences. Compared to Fig. 1.1 the data set has even more variations in human pose. The first challenge is how to effectively cancel out the camera motion, as this can vary considerably from one sequence to another. The second difficulty is that humans are capable of generating a very large range of motion fields,

Fig. 1.2. *Some examples of person detection in images where humans can use overall image context, high-level inference and logical reasoning to make accurate decisions about ambiguous object instances.*

Fig. 1.3. *Some pairs of consecutive images from the INRIA moving person database. The database contains shots with moving subjects, cameras and backgrounds. Approximately 70% of shots contain a moving camera or background or both.*

owing to their articulated structure. The range of viewpoints further compounds this problem. The third issue is that motion vectors such as computed optical flow or fitted parametric models are often noisy and error prone. For these reasons, the detector must use a robust features set of motion to achieve reliable detection.

1.3 Some Background on Object Detection

As mentioned earlier an object detector can be viewed as a combination of an image feature set and a detection algorithm. Feature extraction involves sparse or dense representation of image regions as feature vectors; and the detector architecture specifies exactly how the spatial occurrences of these feature vectors w.r.t. to each other are exploited to obtain detection decisions.

Feature extraction typically captures intensity patterns, texture details, and/or shape and contour information. There are two contrasting views in computer vision on how to compute feature vectors:

- One approach is based on sparse features extracted from a set of salient image regions. The motivation is that not all image regions contain useful information: many are uniform, textureless, or too cluttered to use. Support for this approach comes from physiological studies based on human eye-tracking, where it was inferred that gaze preferably fixates on image regions with corners and multiple superimposed orientations [Zetzsche et al. 1998, Barth et al. 1998], and that local spatial contrast is significantly higher at these points than at random locations, while image uniformity and pixel correlations are significantly lower [Reinagel and Zador 1999].

- The alternative approach is to densely compute feature vectors on image regions. The point of view here is that at the early stages of visual scene analysis, all image regions might be of equal importance and it is best not to lose small details; instead let the latter stages decide which regions are the most relevant. Support comes from studies of the mammalian visual system: the first level of visual coding in mammals involves the computation of dense and overlapping centre-surround receptive-fields of different scales [Fischer 1973, Hubel and Wiesel 1974, Hubel 1995, Chapter 3].

Note that the differences between the sparse and dense approaches are not as great as they may seem as the detection of salient regions requires a dense scan of the input image. The difference is that the criterion used to scan the images for salient regions can be (and usually is) different from the information encoded in the final feature vector. For example, interest point based approaches to salient region detection are usually tuned to fire on blob like structures (one such detector is the Laplacian of Gaussians) while their feature vector computations are tuned to encode gradient or contour information (*e.g.* using SIFT [Lowe 2004] or shape context [Belongie et al. 2001]). Clearly these two criteria are distinct. In any case, in the sparse approach the detector stage only receives the filtered responses of the salient region detector – it has no control over how to chose these salient regions in the first place.

Regarding the detector, several different models and techniques have been studied in the literature. We broadly divide them into two categories:

- One common approach is to learn to recognise classes of similar image regions that commonly occur in the given object class. Broadly this can be termed a *parts* based approach. It fits well with a structural notion of objects: "I can see two wheels, one handle bar, some rods connecting two wheels, a seat and no motor. So it is a bicycle". Details include exactly how the parts are detected and which model is used to learn these occurrences. For example, in a star network each part is defined and located with respect to a central hub, in a fully connected network spatial co-occurrences of each pair of parts are modelled, and in a tree structured network parts at leaves are connected to other part(s) which can be further chained to form a complete articulated structure. The notion of what defines a part is also somewhat blurred. Some approaches attempt to detect physical parts (*e.g.* if

goal is to find people then the subject's head, legs, arms and torso are parts), while others define small image regions or simply use the salient image regions that are detected in a sparse image representation as parts.

- Another perhaps simpler, approach is to implicitly encode spatial information in the form of rigid templates of feature vectors. This scheme is usually based on densely computed image representations, but sparse representations can also be used. The overall detection is then provided by an explicit template matching or by use of state-of-the-art machine learning algorithms such as kernel-based Support Vector Machines (SVM).

Both generative and discriminative approaches can be used in the detection stage. Typically generative approaches use Bayesian graphical models with Expectation-Maximisation (EM) to characterise these parts and to model their co-occurrences. Discriminative approaches use machine learning techniques to classify each feature vector as belonging to the object or not.

1.4 Overview of Our Approach

Our approach to object detection is based on scanning a detection window over the image at multiple positions and scales, in each position runs an object/non-object classifier. We thus adopt a *bottom-up* approach that uses low-level information such as appearance and motion vectors to detect objects in images and videos, and that does not deal with general context or prior information, sometimes termed top-down information in the vision community. A dense feature-based representation is computed in each image region and passed to a region classifier that represents spatial information implicitly by position in the computed feature vector. This is a purely data-driven approach where any invariance that we want the detector to learn needs to be present in the training data used to learn the classifier. Image regions are represented by a dense and overlapping grid of features extracted at fixed resolution and gathered into a high-dimensional feature vector, and classification is performed using a machine learning-based algorithm. The approach performs implicit feature selection and the expectation is that if the encoding of images into feature vectors is sufficiently discriminative then the machine learning algorithm will be able to learn to recognise the object class.

We adopt this relatively simple architecture in order to concentrate on one of the fundamental problems of object detection – what kind of feature representation or *descriptors* to use. Linear Support Vector Machines (SVMs) [Vapnik 1995, Cristianini and Shawe-Taylor 2000, Schölkopf and Smola 2002] are used as the classifiers as they offer the state-of-the-art performance and are fast to run. Although this is one of the simplest approaches to object detection, we will show that with appropriate features it can provide state-of-the-art results for object detection tasks.

For static images, we propose locally normalised Histograms of Oriented Gradients (HOG) as descriptors. The HOG descriptors are computed from image gradients and are designed to be robust to (a) small changes in image contour locations and directions, (b) significant change in image illumination and colour, while (c) remaining as discriminative and separable as possible. They compute weighted histograms of gradient orientations over small spatial neighbourhoods, gather these neighbouring histograms into local groups and contrast normalise them. They are reminiscent of edge orientation histograms [Freeman and Roth 1995, Freeman et al. 1996], SIFT descriptors [Lowe 2004] and shape contexts [Belongie et al. 2001], but they are computed on a dense grid of uniformly spaced cells and they use overlapping descriptors for improved performance.

Once we have the window level classifier, to obtain detections over a test image we scan the detection window across the images at all positions and scales, giving a detection score at each point. This typically results in multiple overlapping detections, around each true object instance in the image. These need to be fused for the final results. We offer a general solution based on kernel density estimation that also takes into account the number of detections, their confidence scores and the scales of the detections. Basically, negative scores from the linear SVM are zeroed and a 3D position-scale mean shift process [Comaniciu 2003b] is run to identify significant local peaks in the resulting score. If above threshold, these are declared as positive detections.

For the detection of moving objects in videos, we propose descriptors based on oriented histograms of differential optical flow. As for the static descriptors, these motion based vectors are weighted histograms computed over small spatial neighbourhoods and they use overlapping local normalisation for improved performance. However instead of image gradients, the votes in the histogram are computed from local differentials of dense optical flow. These are designed to provide as much invariance to camera motion as possible. Strong normalisation ensures that the features are relatively invariant to the magnitude of the flow vectors and coarse orientation binning provides robustness to errors in the estimation of optical flow. Currently the method uses individual pairs of consecutive images and we do make any attempt to enforce temporal continuity of detections.

1.5 Summary of Contributions

This thesis studies the question of feature sets for robust visual object recognition. It makes three major contributions to this goal:

- The most important contribution is the *dense* and *overlapping* encoding of image regions based on HOG descriptors. The proposed descriptors are computed using fine grained features and they form robust feature spaces for visual recognition, significantly improving the detector performance compared to previous state-of-the-art algorithms for many object classes. Several schemes and variants of the HOG architecture are proposed and compared.

- The second contribution are the new differential flow based motion descriptors. These descriptors encode the noisy optical flow estimates into robust feature vectors in a manner that is comparatively insensitive to the exact camera motion. Several variants are proposed, some capturing motion boundaries while others aim at encoding the relative motions of adjacent image regions.

- We also describe a general algorithm for fusing multiple overlapping detections and non-maximum suppression. Our experiments include some interesting conclusions such as the extent to which fine-grained scans and relatively conservative smoothing helps to improve the overall object detection performance for a given binary window classifier.

The thesis also provides guidelines on how to choose the various parameters of the feature sets and what effects each of these have on the overall detection performance. All of the conclusions are validated with extensive experimental tests. Using the same detection and classification architecture, the proposed feature sets improve the overall detection performance by more than an order of magnitude over previous feature sets for object detection and they are shown to work well for a number of object classes. We also propose a parts-based approach

where each part detector is itself a stand alone detector using the proposed HOG architecture. The combined detector encodes the spatial positions of part occurrences implicitly and thus further improves the detection performance.

The thesis also provides two challenging annotated data sets for the training and testing of person detectors to the computer vision community: the INRIA static person data set for detectors working on individual images, and the INRIA moving person data set for detectors using video sequences.

1.6 Outline of the Dissertation

This chapter introduced the object detection problem, and described the key applications of person detection and presented the overall goals of the thesis. It then continued with a brief background of objection detection in computer vision, outlined our approach to the problem, and summarised our main contributions. The remaining chapters are organised as follows:

- **Chapter 2** describes the state of the art in object detection, focusing particularly on persons detection. It first describes previous work on image description, then summarises the key contributions on detection models. It also presents our motivation for using dense feature sets for object detection.
- **Chapter 3** presents a high-level overview of our approach to object detection. It does not give implementation level details but it describes the overall detection framework and gives an overview of the key experimental results. It thus provides an introduction to the detection framework and a summary of the key results of the thesis.
- **Chapter 4** describes the computation of HOG feature vectors in detail. It discusses several variants of HOG taking "person detection" (the detection of mostly visible people in more or less upright poses) as a test case, and studies the effects a number of implementation choices on detector performance. The results show that grids of HOG descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale image gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalisation in overlapping descriptor blocks are all important for good results. We also analyse what kinds of image features the detector keys on, and present brief guidelines on how to choose the descriptor parameters for a given object class. The chapter concludes with a summary of the HOG encoding algorithm and the overall classifier learning algorithm.
- **Chapter 5** presents the algorithm for fusing multiple overlapping detections. It first lays down some essential criteria that any fusion algorithm should fulfil and then proposes a kernel density based fusion in 3-D position and scale space as an appropriate solution. The chapter studies in detail how the parameters of the fusion algorithm affect the detector performance and provides some guidelines on their optimal values. It also presents the overall detection algorithm. Although we developed our feature selection approach for the problem of detecting people in images and videos, we also present results that show that the approach is general and gives state-of-the-art performance for several other man-made objects such as car, and motorbikes and for natural object classes such as sheep and cows.

- **Chapter 6** studies the use of motion features to detect people in videos and investigates feature sets for robust detection of standing or moving people in videos. We propose several variants of oriented histograms of differential optical flow and empirically study how the performance depends on the proposed variants. We show that strong normalisation, comparatively coarser orientation binning, and a basic unregularised optical flow estimation method based on brightness constancy works best. All of the variants are tested on several databases including a challenging test set taken from feature films and containing wide ranges of pose, motion and background variations, including moving cameras and backgrounds.
- **Chapter 7** describes an extension of our framework to parts-based detection of people in images. We investigate several approaches to parts-based detection and propose a sparse spatial histogram based integration scheme in which only the part with the highest confidence values vote. We show that by combining specialised head and shoulders, torso, and legs detectors, the parts based approach can outperform the monolithic person detector.
- **Chapter 8** summarises our approach and our key results, and provides a discussion of the advantages and limitations of the work. It also provides some suggested directions for future research in this area.

2

State of the Art

Object detection in images and videos has received a lot of attention in the computer vision and pattern recognition communities in recent years. As mentioned in Chapter 1 it is a challenging problem with many potential applications. This chapter reviews the state of the art in automatic object detection and localisation, with particular attention to human detection.

As discussed in Sect. 1.3, most of the cited works on object detection can be classified according to:

- The image descriptors or feature vectors that they use
- The detection framework that is built over these descriptors

We review the relevant work according to these categories. Section 2.1 covers the different image feature sets that have been used including ones specific to human detection. Section 2.2 provides an overview of the key approaches used to build detectors. Section 2.3 and Sect. 2.4 provide, respectively, states of the art on person detection in videos and on approaches to fuse multiple overlapping detections. Section 2.5 concludes by presenting motivations behind our approach and reviewing its relation to prior work.

2.1 Image Features

The image feature set needs to extract the most relevant features for object detection or classification while providing invariance to changes in illumination, differences in viewpoint and shifts in object contours. To achieve this, rather than directly using raw images intensities or gradients, one often uses some form of more advanced local image descriptors. Such features can be based on points [Harris and Stephens 1988, Mikolajczyk and Schmid 2002], blobs (Laplacian of Gaussian [Lindeberg 1998] or Difference of Gaussian [Lowe 2001]), intensities [Kadir and Brady 2001, Ullman et al. 2001, Vidal-Naquet and Ullman 2003], gradients [Ronfard et al. 2002, Mikolajczyk et al. 2004], colour, texture, or combinations of several or all of these [Martin et al. 2004]. The final descriptors need to characterise the image sufficiently well for the detection and classification task in hand.

We will divide the various approaches into two broad categories: sparse representations based on points, image fragments or part detectors; and dense representations using image intensities or gradients.

2.1.1 Sparse Local Representations

Sparse representations are based on local descriptors of relevant local image regions. The regions can be selected using either key point detectors or parts detectors.

Point Detectors

The use of salient local points or regions for object detection has a long history [Schiele and Crowley 1996a,b,c, Schmid and Mohr 1997, Weber et al. 2000, Lowe 2001, Agarwal and Roth 2002, Fergus et al. 2003, Dorkó and Schmid 2003, Lowe 2004, Opelt et al. 2004, Leibe et al. 2005, Mikolajczyk et al. 2005]. These approaches extract local image features at a sparse set of salient image points – usually called points of interest or *key points*. The final detectors are then based on feature vectors computed from these key point descriptors. The hypothesis is that key point detectors select stable and more reliable image regions, which are especially informative about local image content. The overall detector performance thus depends on the reliability, accuracy and repeatability with which these key points can be found for the given object class and the informativeness of the points chosen. Commonly used key point detectors include Förstner-Harris [Förstner and Pertl 1986, Förstner and Gülch 1987, Harris and Stephens 1988], Laplacian [Lindeberg 1998] or Difference of Gaussians (DoGs) [Lowe 2004], and scale invariant Harris-Laplace [Mikolajczyk and Schmid 2004]. Some point detectors such as DoGs or Harris-Laplace also provide additional local scale and/or dominant orientation information. One advantage of sparse key point based approaches is the compactness of the representation: there are many fewer key point descriptors than image pixels, so the latter stages of the classification process are speeded up. However note that most key point detectors are designed to fire repeatedly on particular objects and may have limitations when generalising to object classes or categories, *i.e.* they may not be repeatable for general object classes.

Depending on the object class, most approaches can use any one of these key point detectors or combinations of several of them. For example Fergus et al. [2003] used the entropy based region detector of Kadir and Brady [2001] for object representation, Weber et al. [2000] and Agarwal and Roth [2002] use the Förstner and Gülch [1987] interest point operator, while Dorkó and Schmid [2003] and Opelt et al. [2004] use Harris [Harris and Stephens 1988], Harris-Laplace [Mikolajczyk and Schmid 2002] and Laplacian of Gaussian (LoG) [Mikolajczyk and Schmid 2002] based interest point detectors.

Regarding the computation of feature vectors or descriptors over the local image regions surrounding the key points, many approaches have been tried. Currently the most popular approaches are image gradient based descriptors such as the *Scale Invariant Feature Transformation (SIFT)* [Lowe 1999, 2004] and shape contexts [Belongie et al. 2001, 2002]. Both compute local histograms of image gradients or edges. SIFT uses the local scale and dominant orientation given by the key point detector to vote into orientation histograms with weighting based on gradient magnitudes. It thus computes scale and rotation invariant feature vectors. The scale information is also used to define an appropriate smoothing scale when computing image gradients. SIFT computes histograms over rectangular grids, whereas shape contexts use log-polar grids. The initial shape context method [Belongie et al. 2002] used edges to vote into 2-D spatial histograms, but this was later extended to generalised shape contexts by Mori and Malik [2003] who use gradient orientations to vote into 3-D spatial and orientation histograms with gradient magnitude weighting similar to SIFT.

Part or Limb Detectors

Local "parts"-based detectors are also widely used in object and human recognition systems [Forsyth and Fleck 1997, Ioffe and Forsyth 1999, Schneiderman and Kanade 2000, Ronfard et al. 2002, Ramanan and Forsyth 2003, Sigal et al. 2003, Schneiderman and Kanade 2004]. For example, Forsyth and Fleck [1997], Ioffe and Forsyth [1999, 2001a,b] and Ramanan and Forsyth [2003] use explicit human body segments (forearm, upper arm, upper leg, lower leg, torso, etc.) which are assumed to be well represented by cylinders. Parallel edge detectors are then used to detect the corresponding image segments, and body-geometry based detectors are built by using articulation constraints or graphical models to constrain the relative geometry of the limbs. 3D limb detectors have also been used, *c.f*. Sigal et al. [2003]. One problem with these approaches is that the assumption that limbs can be represented by parallel lines is rather simplistic and its scalability to real world examples is questionable. This may explain the lack of extensive testing on real world images in these works.

2.1.2 Dense Representation of Image Regions

Another approach is to extract image features densely (often pixel-wise) over an entire image or detection window and to collect them into a high-dimensional descriptor vector that can be used for discriminative image classification or labelling window as object or non-object, *c.f*. Sect. 2.2.1. Typically the representation is based on image intensities, gradients or higher order differential operators.

Regions and Fragments Based on Image Intensity

One of the primary works using simple image intensities is the "eigenfaces" approach of Sirovitch and Kirby [1987] and Turk and Pentland [1991], where the pixels of fixed-resolution face images are rearranged to forms large feature vector and Principal Component Analysis (PCA) is used to characterise the main variations of the ensemble of face vectors. Another work using intensity images is the face detection system of Rowley et al. [1998] who locally correct the lighting of the images by performing histogram equalisation before passing them to a neural network classifier [Bishop 1995] for face/non-face detections. Ullman et al. [2001], Vidal-Naquet and Ullman [2003] use simple object fragments based on image intensity patterns. Fragments are generated randomly, with the most relevant ones being selected greedily by maximising the mutual information between the fragment and the class. Pairwise statistical dependencies between fragments is also used to create non-linear tree based models. The authors show improved detection performance relative to wavelet based descriptors (described below), but limit their evaluations to relatively rigid object classes such as faces and cars.

Edge and Gradient Based Detectors

Image edges and gradient filters have also been used for object detection. A popular approach is the pedestrian detection system of Gavrila and Philomin [1999], who propose to extract edge images and match them to a set of learned exemplars using chamfer distance. This has been recently used in a practical real-time pedestrian detection system [Gavrila et al. 2004]. Other approaches using image gradient descriptors are Ronfard et al. [2002] and Mikolajczyk et al. [2004]. Ronfard et al. [2002] build an articulated body detector by incorporating SVM based

limb classifiers built over 1st and 2nd order Gaussian filters in a dynamic programming framework similar to those of Felzenszwalb and Huttenlocher [2000] and Ioffe and Forsyth [2001b][1]. Mikolajczyk et al. [2004] design features specifically tuned for human detection. They propose 7 parts-based detectors: front and profile view of head, upper body and legs. Gradient images are used to compute multi-scale descriptors based on histogramming dominant orientations at each position, similar in spirit to SIFT [Lowe 1999].

Wavelet Based Detectors

Some well known approach to object detection are described in Papageorgiou and Poggio [2000], Mohan et al. [2001], Viola and Jones [2001]. These approaches use dense encoding of image regions based on operators similar to Haar wavelets.

Papageorgiou and Poggio [2000] use absolute values of Haar wavelet coefficients at different orientations and scales as their local descriptors. Images are mapped from pixel space to an over-complete dictionary of Haar wavelets that is rich enough to describe patterns. Horizontal, vertical, and diagonal wavelets are used. To obtain an over-complete basis, the wavelets are computed with overlapping supports. Haar wavelets can be calculated efficiently, while still remaining rich enough to encode visually significant patterns, and the use of over-completeness provides a reasonable degree of translation invariance. The descriptor vectors are used in a kernelised Support Vector Machine (SVM) framework (see Sect. 2.2.1), so the final decision criterion is a sum of weighted kernel distances from selected training examples. However we find that linear SVMs (weighted sums of rectified wavelet outputs) give similar results and are much faster to calculate. Papageorgiou and Poggio [2000] show results for pedestrian, face, and car detection. Mohan et al. [2001] add simple part based descriptors to this approach. Four part detectors are built for pedestrians: head, left arm, right arm and leg detectors. Each detector is a part classifier built on Papageorgiou and Poggio's Haar wavelets. The responses of the part detectors are checked for a proper geometric configuration, and the final classification is performed by using a SVM on their outputs. A highly optimised version of this method is presented in de Poortere et al. [2002].

Rather than using a single complex classifier, Viola and Jones [2001] and Viola et al. [2003] build a more efficient progressive rejection based classification chain that uses a generalisation of Haar wavelets — differences of rectangular regions arranged in several Haar and bar-like arrangements — as features. The classifiers become progressively more complex with depth in the chain. Each stage is designed to reject as many of the remaining negative cases as possible, while still retaining all but a negligible fraction of the positives. To train each stage, features with all possible rectangle dimensions are tested and the sample reweighting procedure AdaBoost [Schapire 2002] is used as a greedy feature selection method. This approach was used to build a real time face detector in Viola and Jones [2001], and extended to pedestrian detection from video in Viola et al. [2003]. For improved accuracy, the pedestrian method includes temporal information (motion descriptors based on difference of rectangular region sums between the current frame and the next) in the descriptor pool.

[1] This approach falls in both the sparse class (when seen from the parts based point of view) and the dense one (when considering how parts are selected in the first place).

2.2 Classification Methods

Classification methods based on local part descriptors can be divided into discriminative approaches such as Support Vector Machines, and generative ones such as graphical models.

2.2.1 Discriminative Approaches

Machine learning techniques such as SVMs [Vapnik 1995, Cristianini and Shawe-Taylor 2000, Schölkopf and Smola 2002, Tipping 2001] and Boosting [Schapire 2002] have become popular as classifiers for object recognition owing to their ability to automatically select relevant descriptors or features from large feature sets, their superior performance, and their relative ease of use.

Support Vector Machine (SVM) Classifiers

SVM have been widely used for object recognition for the past decade. They find a separating hyperplane that maximises the margin (gap) between the object class and non-object class in either the input feature space or a kernelised version of this. In Papageorgiou and Poggio [2000], Papageorgiou et al. [1998], Mohan et al. [2001], SVM are used as classifiers over parts-based descriptors. Mohan et al. [2001] created a two stage cascade of SVM classifiers. The first stage creates part (head, left arm, etc) detectors from Haar wavelets. The second combines the part detections to obtain the final object detector. SVMs are also used as an intermediate stage in Ronfard et al. [2002], Dorkó and Schmid [2003]. In Ronfard et al. [2002], 15 SVM and Relevance Vector Machine (RVM) based limb detectors are created based on first and second order image gradients, but the final classifier is based on dynamic programming over assemblies of limb detections as in Ioffe and Forsyth [2001b,a]. Dorkó and Schmid [2003] use SVM based classifiers over interest points as intermediate part detectors for general object recognition, and test two types of final classifiers: (a) likelihood ratios for detecting parts $\frac{P(part=1|object=1)}{P(part=1|object=0)}$, and (b) mutual information between detected parts and object classes.

Cascaded AdaBoost

AdaBoost combines a collection of weak classifiers to form a stronger one. In vision, it is used particularly to build cascades of pattern rejecters, with at each level of the cascade choosing the features most relevant for its rejection task. Although AdaBoost cascades are slow to train, owing to their selective feature encoding they offer significant improvement (compared to SVMs) in the run-time of the final detectors.

Viola and Jones [2001], Viola et al. [2003] use AdaBoost to train cascades of weak classifiers for face and pedestrian detection, using spatial and temporal difference-of-rectangle based descriptors. Opelt et al. [2004] use a similar AdaBoost framework for their interest point based weak classifiers. Schneiderman and Kanade [2000, 2004] propose a more elaborate classification model. They define parts as functions of specific groups of wavelet coefficients, represented with respect to a common coordinate frame. This implicitly captures the geometric relationships between parts. Independent ("naive Bayes") combinations of likelihood ratios $\frac{P(part|object=1)}{P(part|object=0)}$ are combined to form the final classifier. The original detector did not use AdaBoost, but in Schneiderman and Kanade [2004], conditional probability scores are estimated using a modification of AdaBoost. Mikolajczyk et al. [2004] also use likelihood ratios $\frac{P(descriptor|object=1)}{P(descriptor|object=0)}$ as weak classifiers, but relax the independence assumption by using likelihoods over pairs of descriptors as

weak classifiers. They again use AdaBoost to combine weak classifiers linearly to create strong ones. Finally, a coarse-to-fine strategy is used for fast detection.

Recently Zhu et al. [2006] used histograms of oriented gradient features proposed in this thesis with a cascade of rejecters based approach. They use an integral array representation [Viola and Jones 2001] and AdaBoost to achieve a significant improvement in run time compared to our approach while maintaining similar performance levels.

Other Methods

Agarwal and Roth [2002] use the perceptron-like method Winnow as the underlying learning algorithm for car recognition. The images are represented as binary feature vectors and classification is done by using a learned linear function over the feature space.

2.2.2 Bayesian and Graphical Models

Ullman et al. [2001] and Vidal-Naquet and Ullman [2003] show that image fragments selected by maximising the mutual information between the fragment and the class label provide an informative and independent representation. They use Naïve Bayes classification and show that using tree based Bayesian networks over the same fragment set does not give a noticeable improvement in classification results.

Weber et al. [2000] use Bayesian generative models learned with EM to characterise classes, and use likelihood ratios $\frac{P(part|object=1)}{P(part|object=0)}$ for classification. Fergus et al. [2003] also use likelihood ratios, but with a more elaborate model of conditional probabilities that includes the position and scale of the features as well as their appearance.

2.3 Person Detectors in Videos

A person detector that incorporates motion descriptors has been proposed by Viola et al. [2003]. They build a human detector for static-camera surveillance applications, using generalised Haar wavelets and block averages of spatiotemporal differences as image and motion features and a computationally efficient rejection chain classifier [Baker and Nayar 1996, Viola and Jones 2001, Sun et al. 2004] trained with AdaBoost [Schapire 2002] feature selection. The inclusion of motion features increases the performance by an order of magnitude relative to a similar static detector, but the adoption of a static camera greatly simplifies the problem because the mere presence of motion already provides a strong cue for human presence. Other approaches to motion descriptors are the phase based features of Fleet and Jepson [1993]. Other surveillance based detectors include the flow-based activity recognition system of Haritaoglu et al. [2000]. Efros et al. [2003] used appearance and flow features in an exemplar based detector for long shots of sports players, but quantitative performance results were not given.

For pedestrian detection in video sequences, Gavrila et al. [2004] use a static appearance based Chamfer matching system [Gavrila 1999] in conjunction with texture classification, stereo verification and tracking of detections over time to consolidate the results. Shashua et al. [2004] follow a similar approach. Images are scanned for candidate regions followed by single frame classification of the candidate regions. The final results are obtained by consolidating the single frame detections using a multi-frame approval stage consisting of dynamic gait pattern analysis, tracking of detections and motion analysis coupled with camera ego-motion estimation.

2.4 Fusion of Multiple Detections

Binary classifier based object detectors sweep a detection window across the image typically at multiple scales. This usually produces multiple overlapping detections for each object in the image and these must be merged. Rowley et al. [1998] proposed a heuristic method for fusing overlapping detections. The number of detections within a specified neighbourhood is computed and if it is greater than a threshold, the centroid of these detections is taken as the location of the detection result. Centroids are computed in 3-D position and scale space. The number of detections gives the detection score. The scale value of the centroid defines the bounding region for the detected object. Each centroid is examined to check whether its bounding region overlaps with other centroids. Overlapping centroids with lower score are removed and the remaining centroids constitute the final result. Viola and Jones [2004] proposed a simpler method. The set of detections is partitioned into disjoint subsets, and each partition corresponds to a single final detection. Two detections are taken to be in the same subset if their bounding regions overlap. The final detection and location region is the average of all of the detections in the set.

2.5 Motivations of Our Approaches

Our approach is based on a *dense* and *overlapping* encoding of image regions into histogram of oriented gradient descriptors. Although Haar wavelet like features in Papageorgiou and Poggio [2000], Mohan et al. [2001], Viola and Jones [2001] form dense and overcomplete representations, they do not exploit the lessons learnt from SIFT [Lowe 2004], shape context [Belongie et al. 2001] and similar features. These descriptors perform remarkably well in pattern recognition [Schmid et al. 2005], object detection [Opelt et al. 2004, Lowe 2004, Mikolajczyk et al. 2005], shape matching [Mori and Malik 2001, Belongie et al. 2002], image classification and matching [Mikolajczyk and Schmid 2005], and texture representation and recognition [Lazebnik et al. 2003, 2005]. This thesis builds on these recent advances and proposes a monolithic encoding of image regions based on SIFT like features for object detection within a framework similar to that of Papageorgiou and Poggio [2000], Mohan et al. [2001]. The approach is simple and compliments bag-of-keypoints based approaches [Agarwal and Roth 2002, Fergus et al. 2003, Opelt et al. 2004, Leibe et al. 2005, Mikolajczyk et al. 2005] by explicitly encoding spatial information. Although it is not robust to occlusions and does not support parts based detectors as such, recent results of Everingham et al. [2006a,b] show that the new approach outperforms the feature point based methods for scenarios involving fully visible and unoccluded object classes.

Also some object classes have lot of within-class variation and key point based descriptors have limited performance in this case. In particular, for humans it is unclear which key point detector to use. All the standard ones return either points in textured regions or blob like structures. On humans, texture is seen mainly on clothing so detections are not repeatable across changes of clothing. Variations in pose tend to confuse the blob based detectors. In a recent work, Leibe et al. [2005] build a profile-view pedestrian detection system using Difference of Gaussian as the key point detector and SIFT [Lowe 2004] as the image descriptor, but their test set contains only a limited range of poses and some repeatability is expected. Not surprisingly, they find that the most informative key points correspond to the crossing of legs in their profile views and that it is useful to combine different key point detectors to achieve better performance [Leibe 2006].

2.5.1 Relationship to Prior Work

Features related to our proposed orientation histograms have been used extensively in computer vision [McConnell 1986, Bichsel 1991, Freeman and Roth 1995, Freeman et al. 1996, Lowe 1999, 2004, Belongie et al. 2002]. McConnell [1986] proposed edge-slope histograms over spatial neighbourhoods for pattern recognition tasks, which was later extended to orientation histograms of image gradients in Bichsel [1991], Freeman and Roth [1995]. Freeman and Roth [1995] used it for hand recognition system. They computed very fine (36 bins) orientation histograms of thresholded image gradients over the whole image (images rescaled to 106×80 pixels), and then blurred them in orientation domain to provide invariance to small changes in orientations. The final recognition task compared the computed feature vector on the test image to the feature vectors computed during the training stage by nearest neighbourhood search. However these approaches only reached maturity recently when combined with local spatial histogramming and normalisation in Lowe's SIFT approach to wide baseline image matching [Lowe 1999, 2004]. In this approach SIFT provides the image patch representation for matching scale-invariant key points and have performed very well as descriptors for key points. Here we build on this powerful SIFT description, but use it in a dense pixel-wise manner to describe an object window. This way we avoid the well-know drawbacks of interest points for object class, for which they are less well suited. This thesis and other recent comparison results, *c.f*. Everingham et al. [2006a,b], show that dense representations based on HOG like features significantly outperform most object detection approaches for many object classes.

3

Overview of Detection Methodology and Results

Whatever the [visual] *cortex is doing, the analysis must be local.*
– David Hubel *in* Eye, Brain, and Vision

Our contribution to visual object detection centres around the robust encoding of images and videos. The proposed encoding schemes are based on dense representations that map local image regions to high-dimension feature spaces. To encode static images, we propose oriented histograms of image gradients – the *appearance* channel – and for videos we augment appearance with a *motion* channel – oriented histograms of differential optical flow. This thesis focuses on various aspects of the image encoding. It proposes and provides an in depth study of different image encoding schemes, but on the other hand deliberately adopts a standard and a relatively simple learning and testing framework for object detection. This chapter introduces our detection framework, provides an overview of our encoding scheme, describes our linear SVM based learning process, and summarises the key findings. This sets the stage for the following three chapters, which provide detailed technical presentation of the different stages of the detection process, the implications of various choices made during image encoding, and reemphasise the lessons learned.

We separate the presentation into two stages, learning and detection. Section 3.1 gives a first overview of the complete architecture. Section 3.2 gives the schema for the proposed feature extraction and region classification algorithms. During the detection phase detections at multiple positions and scales need to be fused; this is summarised in Sect. 3.3. A sketch of the learning process used is given in Sect. 3.4. An executive-level comparison with other existing descriptor types is given in Sect. 3.5, which also gives an overview of the key factors contributing to the performance of the proposed oriented histogram descriptors.

3.1 Overall Architecture

The overall object detection architecture is built around a method for classifying individual image regions. This is divided into two phases. The *learning phase* creates a binary classifier that provides object/non-object decisions for fixed sized image regions ("*windows*"); while the *detection phase* uses the classifier to perform a dense multi-scale scan reporting preliminary object

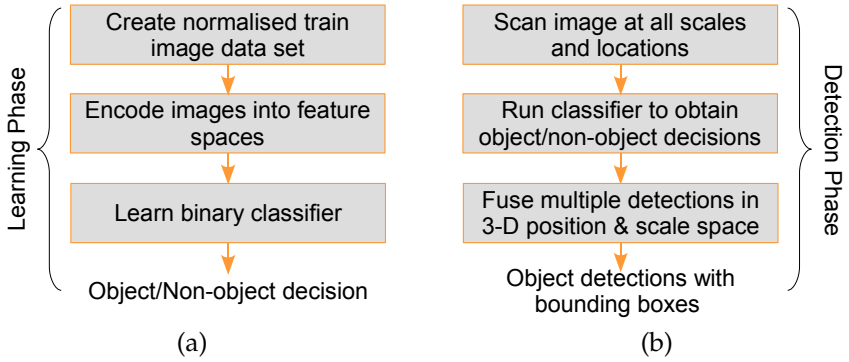


Fig. 3.1. *Overall object detection architecture. (a) The learning phase extracts robust visual features from fixed size training windows, and trains a binary object/non-object classifier over them. (b) The detection phase uses the learned binary classifier to scan the test image at all locations and scales for object/non-object decisions. These preliminary decisions are later fused to produce the final object detections.*

decisions at each location of the test image. These preliminary decisions are then fused to obtain the final object detections. Both the *learning phase* and the *detection phase* contain three stages. Figure 3.1 depicts these. Overall this defines a fixed and relatively simple architecture for object detection. The final detector performance depends on the accuracy and reliability of the binary classifier and on how multiple detections are fused during the detection phase.

The first stage of learning is the creation of the training data. The positive training examples are fixed resolution *image windows* containing the centred object, and the negative examples are similar windows that are usually randomly subsampled and cropped from set of images not containing any instances of the object. The binary classifier is learned using these examples. Ideally, each positive window contains only one instance of the object, at a size that is approximately fixed w.r.t. the window size. In some cases, the windows contain only a limited number of point of views of the object, *e.g.* Rowley et al. [1998], Papageorgiou and Poggio [2000], Mohan et al. [2001], Viola and Jones [2001]. Details of the data sets used and how we annotate the images are given in Appendix A.

This simple window architecture has various advantages. It allows a conventional classifier to be used for detection and relieves the classifier of the responsibility to be invariant to changes in position and scale (although invariance to other types of transformations, changes in pose and viewpoint, and illumination still has to be assured). It also means that the classifier works in relative coordinates (feature position relative to the centre of the current window) which allows relatively rigid template-like feature sets to be used. On the other hand it means that the classifier is run on a large number of windows, which can be computationally expensive and which makes the overall results very sensitive to the false positive rate of the classifier (*c.f.* the arguments for filtering-based approaches such as Viola and Jones [2001]). In fact for a 640×480 pixels image, there can be 10,000 windows per image, so useful range of false positive rates is below 10^{-4} or 10^{-5} false positives per window – well below the rates usually tested in ROC plots.

The image feature extraction process maps image windows to a fixed size feature space that robustly encodes visual form. This process is the core topic of the thesis and is discussed in more detail below. These feature vectors are fed into a pattern recognition style classifier. Any classifier can be used for the purpose, but SVM or AdaBoost are common. In this thesis we have chosen to focus mainly on the issue of robust feature sets for object recognition, so we have

selected a simple, reliable classification framework as a baseline classifier for most of the experiments. We use *linear* SVM as our baseline binary classifier as it proved to be the most accurate, reliable and scalable of the classifiers tested in our initial experiments. Three properties of linear SVM make it valuable for comparative testing work: it converges reliably and repeatably during training; it handles large data sets gracefully; and it has good robustness towards different choices of feature sets and parameters. As the linear SVM works directly in the input feature space, it ensures that the feature set is as linearly separable as possible, so improvements in performance imply an improved encoding. The resulting system gives "*monolithic*" object/non-object decisions over the detection window – it does not provide parts based detection and typically not very robust to visibility or partial occlusions. Nevertheless, the monolithic approach gives a high quality detector that is difficult to beat for fully visible people. Chapter 7 details our work in progress on a more flexible, parts-based approach. Section 3.4 provides details on exact learning process used throughout the thesis.

During detection, the input test image is scanned at all scales and locations. For each scale and location, the feature vector is computed over the detection window, just as in the learning phase, and the binary classifier is run to produce object/non-object decision for the window. Image regions that contain objects typically produce multiple firings and it is necessary to fuse these overlapping detections into a single coherent one. The overall detection score depends both on how finely the test image is scanned and how the detections are fused. Section 3.3 provides an overview of the multi-scale fusion and detection process whose detailed analysis is postponed until Chapter 5. As the detector scans thousands of windows for each image, it must thus have a very low false positive rate per window to achieve good image-level performance. The below classifiers achieve 80–90% true positive recall rates even at 10^{-4} or 10^{-5} false positives per window.

For object detection in videos, the overall architecture remains the same except that the detector input is a set of consecutive image windows (usually just two below). The feature set includes both appearance descriptors (over the current image) and motion ones (over consecutive images).

3.2 Overview of Feature Sets

Our proposed image feature sets are based on *dense* and *overlapping* encoding of image regions using "*Histogram of Oriented Gradient (HOG)*" descriptors. The descriptors can be broadly divided in two categories: static and motion HOGs. Static HOGs are computed over individual images and form the *appearance channel*. Motion HOGs are computed over a set of consecutive images of a video sequence and form the *motion channel*, which is used only for detections on videos. Both static and motion HOGs are based on oriented histograms, the key difference being that static ones are computed over image gradients whereas motion ones over differential optical flow. Figure 3.2 gives an overview of overall encoding process. The general schema for static and motion HOGs are described in the next two sections while detailed discussions of the implementation are postponed until Chapters 4 and 6 respectively.

3.2.1 Static HOG Descriptors

This section gives an overview of the static HOG feature extraction chain. The method is based on evaluating a dense grid of well-normalised local histograms of image gradient orientations

over the image windows. The hypothesis is that local object appearance and shape can often be characterised rather well by the distribution of local intensity gradient or edge directions, even without precise knowledge of the corresponding gradient or edge positions. Figure 3.3 presents the complete processing chain of the feature extraction algorithm.

We now sketch and motivate each step of this process.

- The first stage applies an optional global image normalisation equalisation that is designed to reduce the influence of illumination effects. In practice we use gamma (power law) compression, either computing the square root or the log of each colour channel. Image texture strength is typically proportional to the local surface illumination so this compression helps to reduce the effects of local shadowing and illumination variations.

- The second stage computes first order image gradients. These capture contour, silhouette and some texture information, while providing further resistance to illumination variations. The locally dominant colour channel is used, which provides colour invariance to a large extent. Variant methods may also include second order image derivatives, which act as primitive bar detectors – a useful feature for capturing, *e.g.* bar like structures in bicycles and limbs in humans.

- The third stage aims to produce an encoding that is sensitive to local image content while remaining resistant to small changes in pose or appearance. The adopted method pools gradient orientation information locally in the same way as the SIFT [Lowe 2004] feature. The image window is divided into small spatial regions, called "*cells*". For each cell we accumulate a local 1-D histogram of gradient or edge orientations over all the pixels in the cell. This combined cell-level 1-D histogram forms the basic "*orientation histogram*" representation. Each orientation histogram divides the gradient angle range into a fixed number of predetermined bins. The gradient magnitudes of the pixels in the cell are used to vote into the orientation histogram. Figure 3.3 illustrates the notion of a cell and orientation histogram within it.

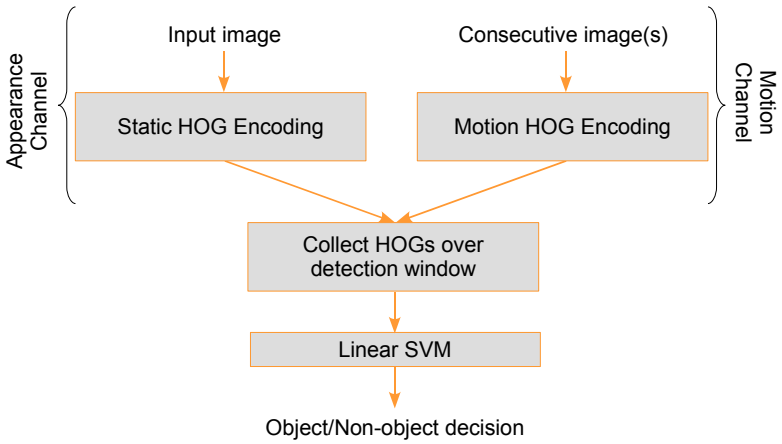


Fig. 3.2. *An overview of HOG feature extraction and object detection chain over an image window. The appearance channel uses the current image and computes static HOGs over it. The motion channel computes motion HOGs over set of consecutive images in the video sequence. Both HOGs are collected over the detection window and the combined vectors are fed to a classifier for object/non-object classification.*

- The fourth stage computes normalisation, which takes local groups of cells and contrast normalises their overall responses before passing to next stage. Normalisation introduces better invariance to illumination, shadowing, and edge contrast. It is performed by accumulating a measure of local histogram "energy" over local groups of cells that we call "*blocks*". The result is used to normalise each cell in the block. Typically each individual cell is shared between several blocks (as shown in Fig. 3.3), but its normalisations are block dependent and thus different. The cell thus appears several times in the final output vector with different normalisations. This may seem redundant but Sect. 4.3.4 shows that this improves the performance. We refer to the normalised block descriptors as *Histogram of Oriented Gradient (HOG)* descriptors.
- The final step collects the HOG descriptors from all blocks of a dense overlapping grid of blocks covering the detection window into a combined feature vector for use in the window classifier.

In practice, the implementation differs slightly from that presented in Fig. 3.3. Certain stages are optimised for efficiency and we have tested several variants of HOG descriptor, *e.g.* with different spatial organisations of the cells into blocks. Details of these are given in Sect. 4.1.

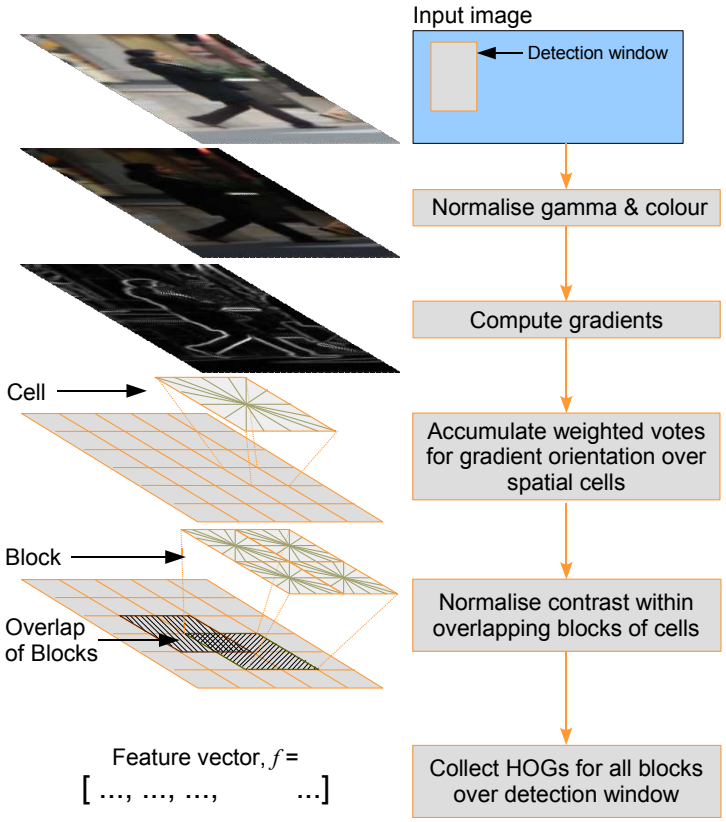


Fig. 3.3. *An overview of static HOG feature extraction. The detector window is tiled with a grid of overlapping* ***blocks****. Each block contains a grid of spatial* ***cells****. For each cell, the weighted vote of image gradients in orientation histograms is performed. These are locally normalised and collected in one big feature vector.*

The HOG representation has several advantages. The use of orientation histograms over image gradients allows HOGs to capture local contour information, *i.e.* the edge or gradient structure, that is very characteristic of local shape. In conjunction with the spatial quantisation into cells, it allows them to capture the most relevant information with controllable precision and invariance (*e.g.* by changing the number of bins in orientation histograms and the cell size). Translations and rotations make little difference so long as they are much smaller than the local spatial or orientation bin size. For example in the human detector we find that rather coarse spatial sampling, fine orientation sampling and strong local photometric normalisation turns out to be the best strategy, presumably because this permits limbs and body segments to change appearance and move from side to side provided that they maintain a roughly upright orientation. Gamma normalisation and local contrast normalisation contribute another key component: illumination invariance. The use of overlapping of blocks provides alternative normalisations so that the classifier can choose the most relevant one. These steps ensure that as little information as possible is lost during the encoding process. Overall the encoding focuses on capturing relevant fine grained features and adding the required degree of invariance at each step.

3.2.2 Motion HOG Descriptors

Our proposed motion features use oriented histograms of differential optical flow. The overall schema is similar to that presented in Fig. 3.3, except it replaces the gradient computation (step two in Fig. 3.3) with a two step process: involving the computation of flow and the estimation of some kind of flow differential. We use differential flow because our goal is object detection in video streams involving possibly moving cameras and/or backgrounds, and these motions need to be cancelled out for good detector performance. The overall chain is shown in Fig. 3.4.

The first change is computation of dense subpixel-accurate optical flow. Variants include "cell" level motion estimation, providing one motion vector estimate for each cell, as in MPEG-4 block coding[1]. Section 6.3 presents details of the flow methods. Flow differentials are then computed. Details are discussed in Chapter 6, which compares several schemes for cancelling camera motion and estimating local flow differences. For now it is only important to know that signal is based on the relative motion of different pixels or regions in the detector block. The remaining steps are the same as in the static HOG encoding.

As in static case, the motion descriptor needs to be invariant to small changes in the motion vectors. This is achieved by estimating "*motion cells*" – voting for oriented motion information over cell level histograms. The magnitude and orientation of the motion vectors are estimated from differential flows, just as we estimate the orientation and magnitude for image gradients. To ensure that the results are not sensitive to the magnitude of the motion, the cells are grouped into blocks and each block is normalised locally. The final step collects the blocks of motion HOGs into a single large vector. The fact that the overall schemata are similar for the static and motion HOGs highlights the power of orientation histograms as a versatile tool for the robust encoding of appearance and motion information.

3.3 Fusion of Multiple Detections

During the detection phase, the binary window classifier is scanned across the image at multiple scales. This typically produces multiple overlapping detections for each object instance. These

[1] In MPEG-4 encoding terminology, what we term as cells is known as blocks.

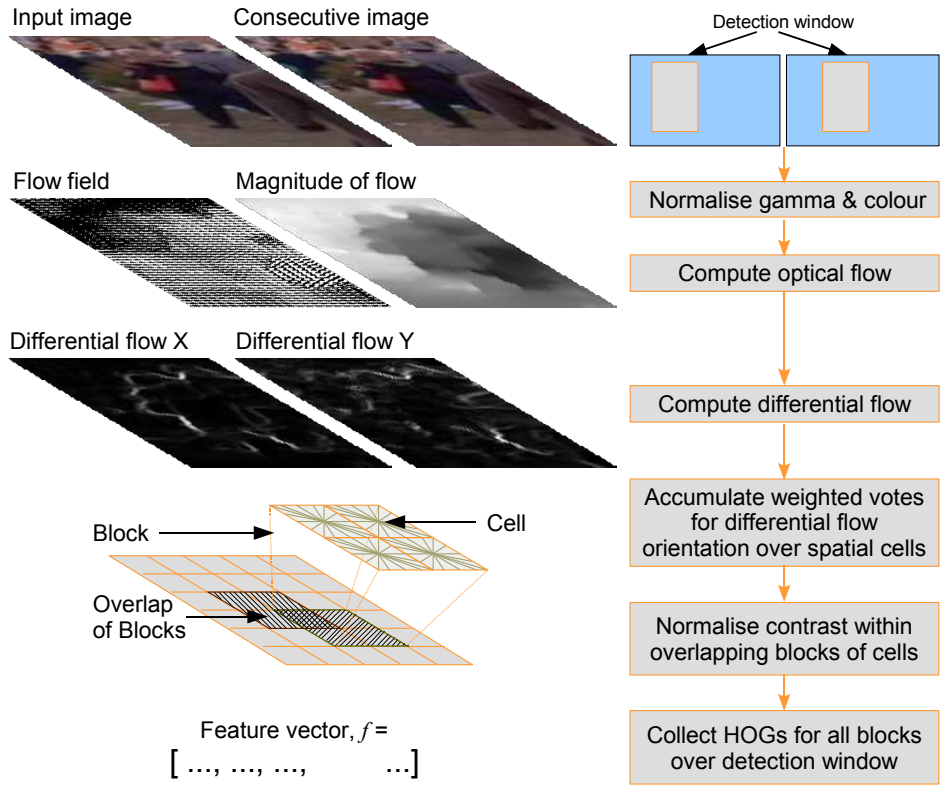


Fig. 3.4. *The feature extraction process for the motion channel. Rather than the image gradients used by the HOG, the motion HOG estimates optical flow and computes differentials or local differences of it. The remaining steps are the same as in the static HOG chain in Fig. 3.3.*

detections need to be fused together. Heuristic methods are often used for this, *c.f*. Sect. 2.4. We propose somewhat more principled solution based on representing detections in a position scale pyramid. Each detection provides a weighted point in this 3-D space and the weights were the detection's confidence score. A non parametric density estimator is run to estimate the corresponding density function and the resulting modes (peaks) of the density function constitute the final detections, with positions, scales and detection scores given by value of the peaks. We will call this process as *non-maximum suppression*. Figure 3.5 illustrates the steps. In practice we use Gaussian kernel mean shift for the mode estimation. Details include how fine the scale and position steps need to be, how to map linear classifier outputs to confidence scores, and the amount of smoothing required when evaluating density estimates. The mathematical and algorithmic details are presented in Chapter 5. For the time being, assume that after non-maximum suppression the detector provides results as detection scores and bounding boxes representing the objects.

3.4 The Learning Process

We use linear Support Vector Machines as our benchmark classifiers. Kernel SVMs can also be used but their run time is significantly higher. Each binary classifier is trained in two stages. First, we train a preliminary detector on the positive training windows and an initial set of neg-

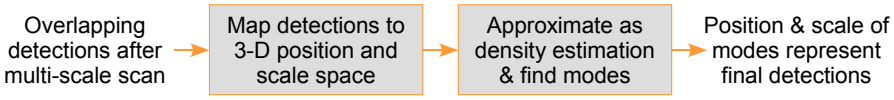


Fig. 3.5. *An overview of the steps of the non-maximum suppression algorithm. The detection window is scanned across the image at all positions and scales, and non-maximum suppression is run on the output pyramid to detect object instances.*

ative windows. The (centred and normalised) positive windows are supplied by the user, and the initial set of negatives is created once and for all by randomly sampling negative images. A preliminary classifier is thus trained using these. Second, the preliminary detector is used to exhaustively scan the negative training images for *hard examples* (false positives). The classifier is then re-trained using this augmented training set (user supplied positives, initial negatives and hard examples) to produce the final detector. The number of hard examples varies from detector to detector, and in particular depends on the initial detector's performance. If too many large number of hard examples are generated, the set of hard examples is subsampled so that the descriptors of the final training set fit into 1.7 GB of RAM for SVM retraining. This retraining process significantly and consistently improves the performance of all the tested detectors (approximately reducing false positive rates by an order of magnitude in most cases). Additional rounds of retraining make little or no difference to the overall results and so we do not use them. The restriction to a limited amount of RAM implies that the larger the descriptor vector, the smaller the number of hard examples that can be included. Thus large descriptor vectors may be at a disadvantage. However, we think that this is fair as memory is typically the main resource limitation during training.

3.5 Overview of Results

Before presenting the detailed implementation and performance analysis in the following chapters, we motivate the work by summarising the overall performance of the HOG based detectors relative to that of some other existing methods. We use Recall-Precision (RP) curves and average precision (AP) for the evaluations. The complete approach is evaluated, not just the window classifier, *i.e.* the detector scans the test image at multiple-scales and performs non-maximum suppression to obtain the final detections and their associated bounding boxes. The precise definition of these criteria is given in Appendix B.

For static images we implemented Haar wavelet, PCA-SIFT and shape context based descriptors, and compared them with our two best static HOG variants. Figure 3.6(a) presents the RP for various detectors on the INRIA static person data set (see Appendix A). The HOG-based descriptors significantly outperform the wavelet, shape context and PCA-SIFT based ones. The two variants of HOG descriptors used here are standard rectangular HOG (R-HOG) and bar (second derivative) HOG (R2-HOG). R-HOG is based on rectangular blocks of descriptors as above. R2-HOG also uses rectangular blocks but it augments R-HOG (which uses first order image gradients) with second order derivatives. R-HOG gives average precision of 0.755 (*i.e.* 24.5% of detected rectangles do not contain people) at a recall rate of 0.7 (*i.e.* 70% of people present are found). This is the default static image detector throughout this thesis. Compared to R-HOG, R2-HOG improves the AP by 0.01. Replacing linear SVM in the default R-HOG with Gaussian kernel SVM improves the AP to 0.762 from 0.74, but at the cost of a very significant increase in run-time. Our extended Haar wavelet scheme is similar to that used in Mohan et al.

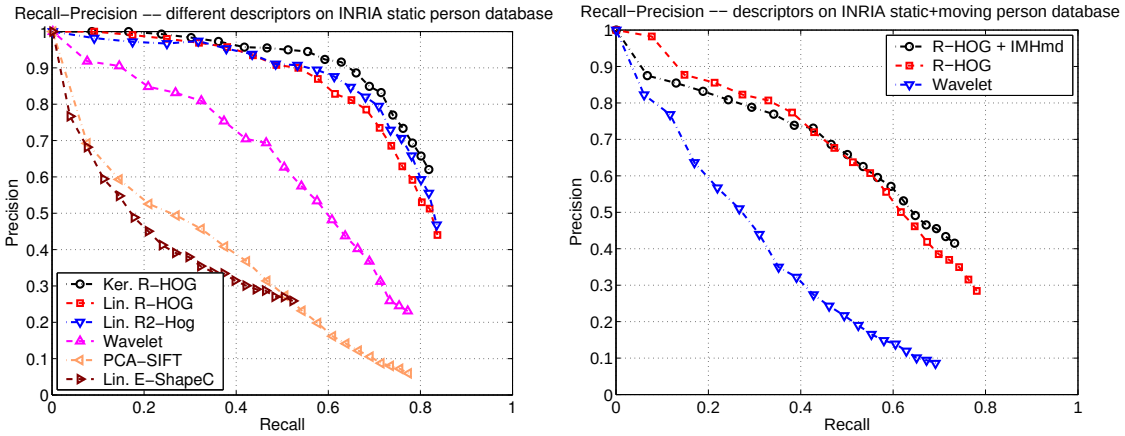


Fig. 3.6. *The performance of selected detectors on the INRIA static (left) and static+moving (right) person data sets. For both of the data sets, the plots show the substantial overall gains obtained by using HOG features rather than other state-of-the-art descriptors. (a) Compares static HOG descriptors with other state of the art descriptors on INRIA static person data set. (b) Compares combined the static and motion HOG, the static HOG and the wavelet detectors on the combined INRIA static and moving person data set.*

[2001] but also includes both 1st and 2nd-order derivative filters at 45° interval and the corresponding 2nd derivative xy filter. It yields AP of 0.53. Shape contexts based on edges (E-ShapeC) perform considerably worse with an AP of 0.25. However, Chapter 4 will show that generalised shape contexts [Mori and Malik 2003], which like standard shape contexts compute circular blocks with cells shaped over a log-polar grid, but which use both image gradients and orientation histograms as in R-HOG, give similar performance. This highlights the fact that orientation histograms are very effective at capturing the information needed for object recognition.

For the video sequences we compare our combined static and motion HOG, static HOG, and Haar wavelet detectors. The detectors were trained and tested on training and test portions of the combined INRIA static and moving person data set. Details on how the descriptors and the data sets were combined are presented in Chapter 6. Figure 3.6(b) summarises the results. The HOG-based detectors again significantly outperform the wavelet based one, but surprisingly the combined static and motion HOG detector does not seem to offer a significant advantage over the static HOG one: The static detector gives an AP of 0.553 compared to 0.527 for the motion detector. These results are surprising and disappointing because Sect. 6.5.2, where we used DET curves (*c.f.* Sect. B.1) for evaluations, shows that for exactly the same data set, the individual window classifier for the motion detector gives significantly better performance than the static HOG window classifier with false positive rates about one order of magnitude lower than those for the static HOG classifier. We are not sure what is causing this anomaly and are currently investigating it. It seems to be linked to the threshold used for truncating the scores in the mean shift fusion stage (during non-maximum suppression) of the combined detector.

Pablo Picasso, *Portrait of Wilhelm Uhde*, Oil on canvas, 1910. Private Collection.

4

Histogram of Oriented Gradients Based Encoding of Images

This chapter describes the proposed Histogram of Oriented Gradient (HOG) feature set. It details several variants of HOG descriptors, with differing spatial organisation, gradient computation and normalisation methods and providing in depth study of the effects of various parameters on detection performance. The main conclusion is that HOG encoding provides excellent detection performance relative to other existing feature sets including Haar wavelets [Papageorgiou et al. 1998, Papageorgiou and Poggio 2000, Mohan et al. 2001, Viola and Jones 2001].

We start by describing the tested variants of HOG descriptors in Sect. 4.1. For comparison we also implemented and tested several other state-of-the-art descriptors. Section 4.2 provides details of these. Implementation details, including optimal values for the various parameters of the HOG feature set, are described in Sect. 4.3 which also details the influence of each stage of the computation on detection performance. We conclude that fine grained features – essentially fine-scale gradients, and fine orientation binning – relatively coarse spatial binning and high-quality local contrast normalisation in overlapping descriptor blocks are all important for good results. The overall performance results, initially mentioned in Sect. 3.5, are quantified in Sect. 4.4. Most of the sections of this chapter use human detection as a benchmark case and Sect. 4.5 gives insight into the kinds of features the classifier cues on for humans. The HOG feature set also gives state-of-the-art performance for other object classes, and Sect. 4.6 summarises these results. The chapter concludes with a summary of the final learning algorithm in Sect. 4.7 and a discussion of it in Sect. 4.8.

4.1 Static HOG Descriptors

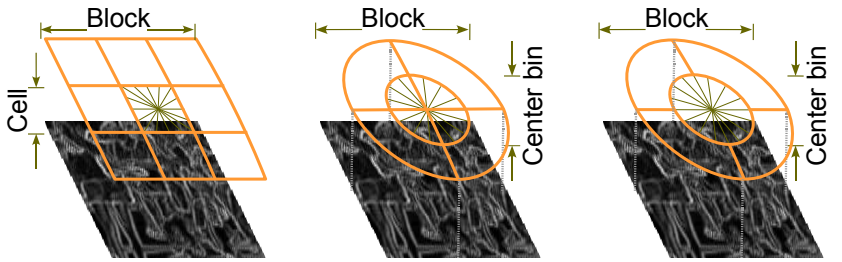
Histogram of Oriented Gradient descriptors provide a dense indeed overlapping description of image regions. This section describes four variants of the HOG encoding and presents the key parameters involved in each variant. Comparisons of results are postponed until Sect. 4.3.5. All of the variants share the same basic processing chain described in Sect. 3.2.1, *i.e.* they all are computed on a dense grid of uniformly spaced cells, they capture local shape information by encoding image gradients *orientations* in histograms, they achieve a small amount of spatial invariance by locally pooling these histograms over spatial image regions, and they employ overlapping local contrast normalisation for improved illumination invariance.

4.1.1 Rectangular HOG (R-HOG)

R-HOG descriptor blocks use overlapping square or rectangular grids of cells. They are the default descriptor for all the experiments in this thesis. The descriptor blocks are computed over dense uniformly sampled grids and are usually overlapped. Each block is normalised independently. We use square R-HOGs and compute $\varsigma \times \varsigma$ grids (which define the number of cells in each block) of $\eta \times \eta$ pixel cells each containing β orientation bins, where ς, η, β are parameters. First order image gradients are used to compute oriented histogram voting. Figure 4.1(a) shows a 3×3 cell ($\varsigma = 3$) R-HOG descriptor.

R-HOGs are similar to SIFT descriptors [Lowe 2004] but are used quite differently. SIFTs are computed at a sparse set of scale-invariant key points, rotated to align their dominant orientations and used individually, whereas R-HOGs are computed in dense grids at a single scale without dominant orientation alignment. The grid position of the block *implicitly* encodes spatial position relative to the detection window in the final code vector. SIFTs are optimised for sparse wide baseline matching, R-HOGs for dense robust coding of spatial form. Other precursors to R-HOG include the edge orientation histograms of Freeman and Roth [1995], *c.f.* Sect. 2.5.1.

As in Lowe [2004], we find that it is useful to down-weight pixels near the edges of the R-HOG block by applying a Gaussian spatial window to each pixel before accumulating orientation votes into cells.



(a) R-HOG/SIFT (b) C-HOG (c) Single centre C-HOG

Fig. 4.1. *Variants of proposed HOG descriptors. (a) A rectangular HOG (R-HOG) descriptor with 3×3 blocks of cells. (b) Circular HOG (C-HOG) descriptor with the central cell divided into angular sectors as in shape contexts. (c) A C-HOG descriptor with a single central cell.*

4.1.2 Circular HOG (C-HOG)

In circular HOG (C-HOG) block descriptors, the cells are defined into grids of log-polar shape. The input image is covered by a dense rectangular grid of centres. At each centre, we divide the local image patch into a number of angular and radial bins. The angular bins are uniformly distributed over the circle. The radial bins are computed over log scales, resulting in increasing bin size with increasing distance from the centre. This implies that more pixels are averaged in the outer cells than in the inner cells, so that the descriptor resolution decreases when moving away from the centre. This provides a natural down-weighting of the pixels near the borders and it turns out that C-HOGs give similar performance, with or without Gaussian spatial down-weight of image gradients.

C-HOGs are reminiscent of shape contexts [Belongie et al. 2001] except that each spatial cell contains a stack of gradient-weighted orientation cells instead of a single orientation-independent edge-presence count. Thus they resemble generalised shape contexts [Mori and Malik 2003]. The motivation for the log-polar grid is that it allows fine coding of nearby structure to be combined with coarser coding of wider context. Further support comes from the fact that the transformation from the visual field to the V1 cortex in primates is logarithmic [Schwartz 1977]. However Sect. 4.3.5 shows that small descriptors[1] with very few radial bins give the best performance, so in practice there is little inhomogeneity or context. This suggests that it is probably better to consider C-HOG's simply as an advanced form of centre-surround coding.

The C-HOG layout has four spatial parameters: the numbers of angular and radial bins; the radius of the central bin in pixels; and the expansion factor for subsequent radii. Two variants of the C-HOG geometry were tested: the first divides central cell into angular sectors as in generalised shape contexts; the second has a single circular central cell, similar to the GLOH feature of Mikolajczyk and Schmid [2005]. Figure 4.1(b,c) illustrates the two variants. To anticipate the results in Sect. 4.3.5, single circular-centre C-HOGs have fewer spatial cells and thus lower feature overall dimension, which can be advantageous. Also, for small central cell sizes, the divided central cell variant may over partition the image, which may be counter productive.

4.1.3 Bar HOG

Bar HOG descriptors are computed similarly to the gradient HOG ones, but use oriented second derivative (bar) filters instead of first derivatives. As with image gradients, we compute the dominant second derivative orientation. For this we estimate the maximum or minimum of the second order derivative response w.r.t. angle at a given smoothing scale. The corresponding angle indicates the dominant direction. Similar to the gradient HOG descriptor, we can now vote into an orientation histogram at the dominant angle using the absolute value of the corresponding maximum or minimum.

The maximum response orientation and magnitude can be computed analytically and efficiently using the steerable property of Gaussian derivative filters. The oriented second order image derivative at an angle θ can be computed from the three basis filter responses [Freeman and Adelson 1991]: I_{xx}, I_{xy} and I_{yy}, where I_{xx}, I_{xy}, I_{yy} are the second order derivatives of the image I along corresponding orientations. The complete interpolation formula giving the estimate of second order derivative at an angle θ is given as

$$I''_\theta = \cos^2\theta \;\; I_{xx} - 2\cos\theta\sin\theta \;\; I_{xy} + \sin^2\theta \;\; I_{yy} \tag{4.1}$$

The dominant orientation is given by zeroing the derivative of (4.1) w.r.t. θ:

$$\theta = \frac{1}{2}\arctan\left(\frac{2I_{xy}}{I_{yy} - I_{xx}}\right) \tag{4.2}$$

The main motivation for using bar HOG is that gradients and contours are only a part of the image information. Many objects, especially articulated animals and human beings though being articulated objects can be modelled as connected bar and blob like structures (*c.f.* Sminchisescu and Triggs [2001], Ioffe and Forsyth [2001a]). The success of these approaches shows the validity of the hypothesis. The second order gradients act as primitive detectors for bars encoding their dominant orientations into HOG descriptors. Additional support comes from the

[1] 16×16 diameter C-HOGs for a 64×128 pixel window containing the person.

fact that the mammalian visual system also contains cells with bar like receptive-fields [Hubel 1995, Chapter 4]. Thus, for certain classes like people or bicycles, we find that it helps to include primitive bar and blob detectors in the feature set.

The overall bar HOG descriptor uses both first and second order image gradients – it includes a HOG feature set for votes over oriented second derivative (bar) filters as well as one for standard image gradients. The resulting generalised orientation histogram descriptor is denoted R2-HOG. The smoothing scale used to compute the second derivatives I_{xx}, I_{xy} and I_{yy} controls the bar detector scale. For Gaussian smoothing of scale ρ, the maximum response occurs for strips of width 2ρ. The optimal value of ρ thus depends on the object and the normalised scale. For the normalised person data set with 64×128 windows, *c.f.* Appendix A, human limbs are typically 6–8 pixels wide and $\rho = 3$ or $\rho = 4$ gives the best results.

4.1.4 Centre-Surround HOG

The block based architecture of R-HOG and C-HOG involves nearly redundant computation. R-HOGs apply a Gaussian down-weighting centred on the current block at each point and then make weighted votes into cell orientation histograms. If there were no Gaussian down-weighting, the R-HOG computation could be optimised by computing orientation histograms once for each cell. But we might be interested in using cells of different widths, *c.f.* Zhu et al. [2006], and for this the R-HOG architecture is not optimal. Similarly C-HOG requires the computation of weights over a non-uniform block (log-polar) at each point. The centre-surround HOG architecture avoids these block specific computations, which allows to be optimised for very rapid computation via integral histograms [Porikli 2005].

Centre-Surround HOGs implement an alternative centre-surround style cell normalisation scheme. They create β "*orientation images*", where β is the number of orientation bins. The image gradient orientation at each pixel is soft quantised into β bins using linear interpolation and weighted using gradient magnitude and stored in the corresponding orientation image. The set of β orientation images is analogous to the oriented histogram in R-HOG or C-HOG. A similar architecture has been proposed by Zhu et al. [2006], who use integral histograms [Porikli 2005] to rapidly tile the orientation images without linear interpolation voting with a grid of cells of any desired dimension. At each pixel in the dense grid where we want to evaluate a HOG descriptor, local normalisation is computed by dividing by the total "*surround*" energy (summed over all orientation images and pooled spatially using Gaussian weighting – equivalent to convolution of image gradient with Gaussian) in the neighbouring region. A cell-like notion, which forms the "*centre*", is introduced by convolving each orientation image with a Gaussian at a scale smaller than surround convolution. Thus like R-HOG and C-HOG there is overlapping of blocks, but each cell is coded only once in the final descriptor. Variants include using several normalisations for each cell employing different centre and surround centres or pooling scales.

4.2 Other Descriptors

In the evaluations below we compare HOG descriptors to several other descriptors from the literature. This section provides details of these.

4.2.1 Generalised Haar Wavelets.

Papageorgiou et al. [1998], Papageorgiou and Poggio [2000], Mohan et al. [2001] proposed a Haar-wavelet [Mallat 1989] based pedestrian detection system. They used rectified responses from three first-order over-complete or overlapping Haar wavelets $\left[\begin{smallmatrix}1 & 1\\ -1 & -1\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}-1 & 1\\ -1 & 1\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix}-1 & 1\\ 1 & -1\end{smallmatrix}\right]$ as a feature set. Our experiments (*c.f.* Fig. 4.9(a)) show that performance can be improved significantly by using an augmented set of 1st- and 2nd-order wavelets. We use an extended set of oriented 1st and 2nd derivative box filters at 45° intervals and the corresponding 2nd derivative xy filter. The output vector is contrast normalised for illumination invariance. The complete features are rectified responses over two scales, 9×9 and 12×12. Figure 4.2 shows these filters.

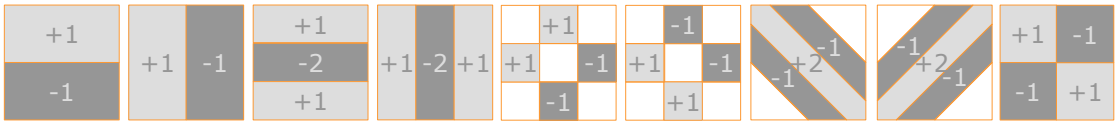
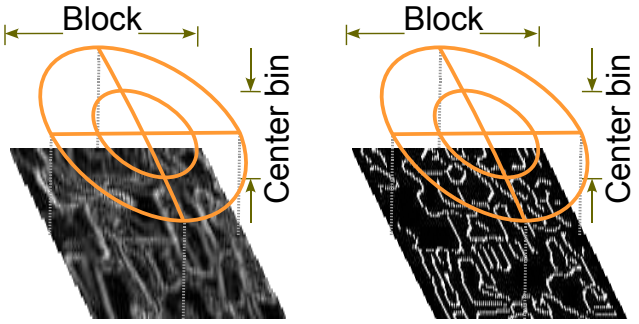


Fig. 4.2. *Generalised 1st- and 2nd-order Haar wavelet operators.*

4.2.2 Shape Contexts

The original shape contexts [Belongie et al. 2001, 2002, Mori and Malik 2001] use binary edge-presence voting into log-polar spaced histogram bins, irrespective of edge orientation. We simulate this using our C-HOG descriptor with just 1 orientation bin. Both gradient-strength (G-ShapeC) and edge-presence (E-ShapeC) based votings are tested. Fig. 4.3 shows the two variants. In the case of E-ShapeC, the edge thresholds were chosen automatically to maximise detection performance. The values selected were somewhat variable, in the region of 20–50 graylevels. Both variants were normalised locally to achieve illumination invariance as in C-HOG.



(a) Gradient shape context (b) Edge shape context

Fig. 4.3. *Gradient-strength shape context (G-ShapeC) and edge-presence (E-ShapeC) shape context variants. Neither uses orientation voting (they have only one orientation bin).*

4.2.3 PCA-SIFT

PCA-SIFT descriptors are based on projecting gradient images onto a basis learned from training images using PCA [Ke and Sukthankar 2004]. Ke and Sukthankar computed PCAs on x- and y- image gradients over an $N \times N$ image window and found that they outperformed SIFT for key point based matching, but this has been disputed [Mikolajczyk and Schmid 2005]. Our motivation for using PCA-SIFT is its similarity to reduce dimension appearance models to the kind popularised by Sirovitch and Kirby [1987], Turk and Pentland [1991], Belhumeur et al. [1997]. However PCA-SIFT computes bases over image gradients instead image intensity. Our implementation uses 16×16 blocks with the same derivative scale, overlap, *etc.*, as our HOG descriptors. Each descriptor block is normalised independently. The PCA basis is calculated using only the positive training images.

4.3 Implementation and Performance Study

We now give details of our different HOG implementations and systematically study the effects of various choices on detector performance. We discuss each step of the descriptor processing chain presented in Fig. 3.3 in turn, and provide detailed experimental results to support our conclusions. We also compare the performance of different HOG variants. Related factors such as size and the amount of background context in each test window, and the classifier used for detection are also studied.

For all of the experiments in this and the following section, we restrict our self to studying the performance of the window level classifier, using Detection Error Tradeoff (DET) to characterise performance. The use of the window-level classifier in the overall detector is studied in Chapter 5. DET curves plot miss-rate against false positives per window. We opted for DET curves because to plot Recall-Precision for the overall classifier the non-maximum suppression stage must be included, which would add another layer of processing with a new set parameters to the tests. Appendix B provides more details on DET curves.

Default Detector.

As a yardstick for the purpose of comparison, throughout this section we compare results to our default detector which has the following properties: input image in RGB colour space (without any gamma correction); image gradient computed by applying $[-1, 0, 1]$ filter along x- and y-axis with no smoothing; linear gradient voting into 9 orientation bins in 0°–180°; 16×16 pixel blocks containing 2×2 cells of 8×8 pixel; Gaussian block windowing with $\sigma = 8$ pixel; *L2-Hys* (Lowe-style clipped L2 norm) block normalisation; blocks spaced with a stride of 8 pixels (hence 4-fold coverage of each cell); 64×128 detection window; and linear SVM classifier. We often quote the performance at 10^{-4} false positives per window (FPPW) – the maximum false positive rate that we consider to be useful for a real detector given that 10^3–10^4 windows are tested for each image.

4.3.1 Gamma/Colour Normalisation

We evaluated several input pixel representations including grayscale, RGB and LAB colour spaces optionally with power law (gamma) equalisation. When available, colour information

always helps, *e.g.* for the person detector RGB and LAB colour spaces give comparable results, while restricting to grayscale reduces performance by 1.5% at 10^{-4} FPPW.

We tried two variants of gamma normalisation: square root and log compression of each colour channel. One can justify log compression on the grounds that image formation is a multiplicative process, so assuming that illumination is slowly varying taking the log factors albedo from illumination. Similarly photon noise in the CCD detectors is proportional to the square root of intensity so taking the square root makes the effective noise approximately uniform[2]. For most object classes, square root compression improved the detection performance by a slight margin compared to unnormalised colour channels. The margin is perhaps only small because the subsequent descriptor normalisation achieves similar results. For the person detector, square root compression continues to improve performance at low FPPW (by 1% at 10^{-4} FPPW). However log compression is too strong and performance drops by 2% at 10^{-4} FPPW. Our experience is that square root gamma compression gives better performance for man made object classes such as bicycles, motorbikes, cars, buses, and also people (whose patterned clothing results in sudden contrast changes). For animals involving lot of within-class colour variation such as cats, dogs, and horses, unnormalised RGB turns out to be better, while for cows and sheep square root compression continues to yield better performance. This is perhaps as much a question of the typical image background as of the object class itself.

4.3.2 Gradient Computation

We computed image gradients using optional Gaussian smoothing followed by one of several discrete derivative masks and tested how performance varies. For colour images (RGB or LAB space), we computed separate gradients for each colour channel and took the one with the largest norm as the pixel's gradient vector. Several smoothing scales were tested including σ=0 (none). Smoothing while computing gradients significantly damages the performance. For Gaussian derivatives, moving from σ=0 to σ=2 nearly doubles the miss rate from 11% to 20% at 10^{-4} FPPW. Figure 4.4(a) summarises the effect of derivative scales on the performance. Derivative masks tested included various 1-D point derivatives, 3×3 Sobel masks and 2×2 diagonal ones $\left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ (these are the most compact centred 2-D derivative masks). The simple $[-1, 0, 1]$ masks give the best performance. Table 4.1 shows how performance varies as we change derivative masks. Using smoothed versions such as cubic-corrected or 3×3 Sobel masks systematically reduces performance, as do the 2×2 diagonal masks. Uncentred $[-1, 1]$ derivative masks also decrease performance, presumably because orientation estimation suffers as a result of the x and y filters being based at different centres.

Overall detector performance is sensitive to the way in which gradients are computed and the simplest scheme of centred 1-D $[-1, 0, 1]$ masks at σ=0 works best. The use of any form of smoothing or of larger masks of any type seems to decrease the performance. The most likely reason for this is that fine details are important: images are essentially edge based and smoothing decreases the edge contrast and hence the signal. A useful corollary is that the optimal image gradients can be computed very quickly and simply.

4.3.3 Spatial / Orientation Binning

The next step is the fundamental nonlinearity of the descriptor. Each pixel contributes a weighted vote for orientation based on the orientation of the gradient element centred on it.

[2] If the pixel intensity is $I \pm \sigma\sqrt{I}$, then $\sqrt{I \pm \sqrt{I}} \approx \sqrt{I}\sqrt{1 \pm \sigma/\sqrt{I}} \approx \sqrt{I}\,(1 \pm \sigma/(2\sqrt{I})) = \sqrt{I} \pm \sigma/2$.

Mask Type	1-D centred	1-D uncentred	1-D cubic-corrected	2×2 diagonal	3×3 Sobel
Operator	$[-1, 0, 1]$	$[-1, 1]$	$[1, -8, 0, 8, -1]$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$, $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Miss rate at 10^{-4} FPPW	11%	12.5%	12%	12.5%	14%

Table 4.1. *Different gradient masks and their effect on detection performance. All results are without Gaussian smoothing ($\sigma = 0$).*

The votes are accumulated into orientation bins over local spatial regions that we call *cells*, see Sect. 3.2.1 and Fig. 3.3. The cells can be either rectangular or radial (log-polar sectors). The orientation bins are evenly spaced over 0°–180° ("unsigned" gradient – a gradient vector and its negative vote into the same bin) or 0°–360° ("signed" gradient). To reduce aliasing, votes are interpolated trilinearly between the neighbouring bin centres in both orientation and position. Details of the trilinear interpolation voting procedure are presented in Appendix D. The vote is a function of the gradient magnitude at the pixel, either the magnitude itself, its square, its square root, or a clipped form of the magnitude representing soft presence/absence of an edge at the pixel. In practice, using the magnitude itself gives the best results. Taking the square root of magnitude reduces performance slightly, while using binary edge presence voting decreases it significantly (by 5% at 10^{-4} FPPW).

Fine orientation coding turns out to be essential for good performance for all object classes, whereas (see below) spatial binning can be rather coarse. As Fig. 4.4(b) shows for the person data set, increasing the number of orientation bins improves performance significantly up to about 9 bins, but makes little difference beyond this. This is for bins spaced over 0°–180°, *i.e.* the 'sign' of the gradient is ignored. Including signed gradients (orientation range 0°–360°, as in the original SIFT descriptor) decreases the performance, even when the number of bins is also doubled to preserve the original orientation resolution. For humans, the wide range of clothing and background colours presumably makes the signs of contrasts uninformative. However note that including sign information does help significantly in some other object recognition tasks, particularly for man made objects such as cars, bicycles, and motorbikes.

4.3.4 Block Normalisation Schemes and Descriptor Overlap

Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast. Hence effective local contrast normalisation turns out to be essential for good performance. A number of different normalisation schemes were evaluated. Most of them are based on grouping cells into larger spatial blocks and contrast normalising each block separately. In fact, the blocks are typically overlapped so that each scalar cell response contributes several components to the final descriptor vector, each normalised with respect to a different block. This may seem redundant but good normalisation is critical and including overlap significantly improves the performance. Fig. 4.4(c) shows that performance increases by 4% at 10^{-4} FPPW as we increase the overlap from none (stride 16) to 16-fold area / 4-fold linear coverage (stride 4).

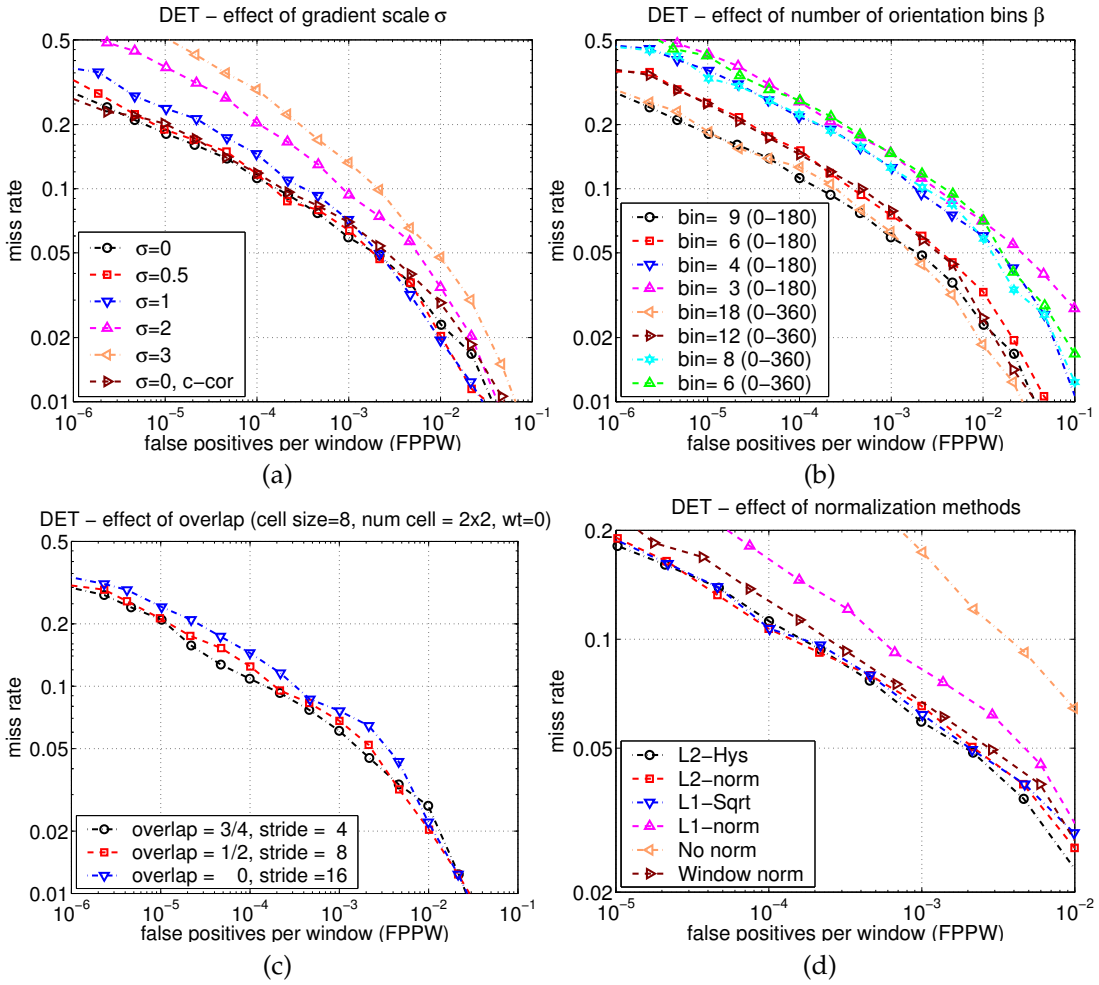


Fig. 4.4. *An evaluation of the effects of the major parameters of R-HOG descriptor on the person detection data set. For details see the text. (a) Using fine derivative scale significantly increases the performance. ('c-cor' is the 1D cubic-corrected point derivative,* c.f. *Sect. 4.3.2). (b) Increasing the number of orientation bins increases performance significantly up to about 9 bins (resolution of* 20° *per bin) spaced over* 0°–180°*. Including signed gradients (orientation range* 0°–360°*) with 18 bins (resolution fixed to* 20°*) decreases performance. (c) Using overlapping descriptor blocks reduces the miss rate by around 5%. (d) The effect of different block normalisation schemes (see Sect. 4.3.5). L2-norm with hysteresis thresholding (L2 Hys), L2-norm and L1-norm followed by square root of descriptor vector (L1-sqrt) perform equally well for person object class.*

We evaluated four different block normalisation schemes for each of the above HOG geometries. Let $\mathbf{v}$ be the unnormalised descriptor vector, $\|\mathbf{v}\|_k$ be its k-norm for k=1, 2, and ϵ be a small normalisation constant to avoid division by zero. The four schemes are:

- *L2-norm*, $\mathbf{v} \leftarrow \mathbf{v}/\sqrt{\|\mathbf{v}\|_2^2 + \epsilon^2}$;
- *L2-Hys* , L2-norm followed by clipping (limiting the maximum values of $\mathbf{v}$ to 0.2) and renormalising, as in Lowe [2004];
- *L1-norm*, $\mathbf{v} \leftarrow \mathbf{v}/(\|\mathbf{v}\|_1 + \epsilon)$;

- *L1-sqrt*, $\mathbf{v} \leftarrow \sqrt{\mathbf{v}/(\|\mathbf{v}\|_1 + \epsilon)}$, *i.e.* L1-norm followed by square root, amounts to treating the descriptor vectors as probability distributions and using the Bhattacharya distance between them.

Figure 4.4(d) shows that L2-Hys, L2-norm and L1-sqrt all perform equally well for the person detector, while simple L1-norm reduces performance by 5% at 10^{-4} FPPW and omitting normalization entirely reduces it by 27%. For other object classes there is no consensus as to which normalisation scheme to use. For some classes L2-Hys seems to have an edge, while for others, such as cars and motorbikes, *L1-sqrt* gives the best results. More on this in Sect. 4.6. Some regularization ϵ is needed because we evaluate descriptors densely, including on empty patches, but the results are insensitive to ϵ's value over a large range. If pixel values are in the range $[0, 1]$, ϵ in range $1e^{-3}$–$5e^{-2}$ works best for all classes.

4.3.5 Descriptor Blocks

Section 4.1 presented two classes of block geometries – square or rectangular R-HOGs partitioned into grids of square or rectangular spatial cells, and circular C-HOG blocks partitioned into cells in log-polar fashion – and two kinds of normalisation: the block normalisation of R-HOG or C-HOG and the centre-surround normalisation of centre-surround HOG. This section studies the performance of R-HOG, C-HOG and centre-surround HOG as the descriptor parameters varies.

R-HOG.

Figure 4.5(a) plots the miss rate at 10^{-4} FPPW w.r.t. the cell size in pixels and the block size in cells. For human detection, 3×3 cell blocks of 6×6 pixel cells perform best with 10.4% miss-rate at 10^{-4} FPPW. Our standard 2×2 cell blocks of 8×8 cells are a close second. In fact, 6–8 pixel wide cells do best irrespective of the block size – an interesting coincidence as human limbs are about 6–8 pixels across in our images (see Fig. 4.5(b)). We find 2×2 and 3×3 cell blocks work best. Adaptivity to local imaging conditions is weakened when the block becomes too big, and when it is too small (1×1 cell block, *i.e.* normalisation over cell orientation histogram alone) valuable spatial information is suppressed.

As in Lowe [2004], down-weighting pixels near the edges of the block by applying a Gaussian spatial window to each pixel before accumulating orientation votes into cells improves performance – here by 1% at 10^{-4} FPPW for a Gaussian with $\sigma = 0.5 * \text{block\_width}$.

Multiple block types with different cell and block sizes can be included to provide encoding at multiple scales. Augmenting the feature space by including 3×3 cell blocks in addition to 2×2 cells one (cell size $\eta = 8 \times 8$) improves performance only marginally. Including cells and blocks at different scales ($\eta = 8 \times 8$, $\varsigma = 2 \times 2$ and $\eta = 4 \times 4$, $\varsigma = 3 \times 3$) improves performance by around 3% at 10^{-4} FPPW, *c.f.* Fig. 4.6(a). However the combination of $\eta = 8 \times 8$, $\varsigma = 2 \times 2$ and $\eta = 4 \times 4$, $\varsigma = 4 \times 4$ – with block size of 16×16 pixels in both cases – brings only slight improvement. This suggests that multiple block encoding should target both different cell (spatial pooling and block normalisation) sizes. However such multiple encodings greatly increase the descriptor size so it might be preferable to perform multilevel encoding using a feature selection mechanism such as AdaBoost in order to avoid explicit encoding of excessively large feature vectors.

Besides the square R-HOG blocks, we also tested vertical (2×1 cell) and horizontal (1×2 cell) blocks and a combined descriptor including both vertical and horizontal pairs. Figure 4.6(b)

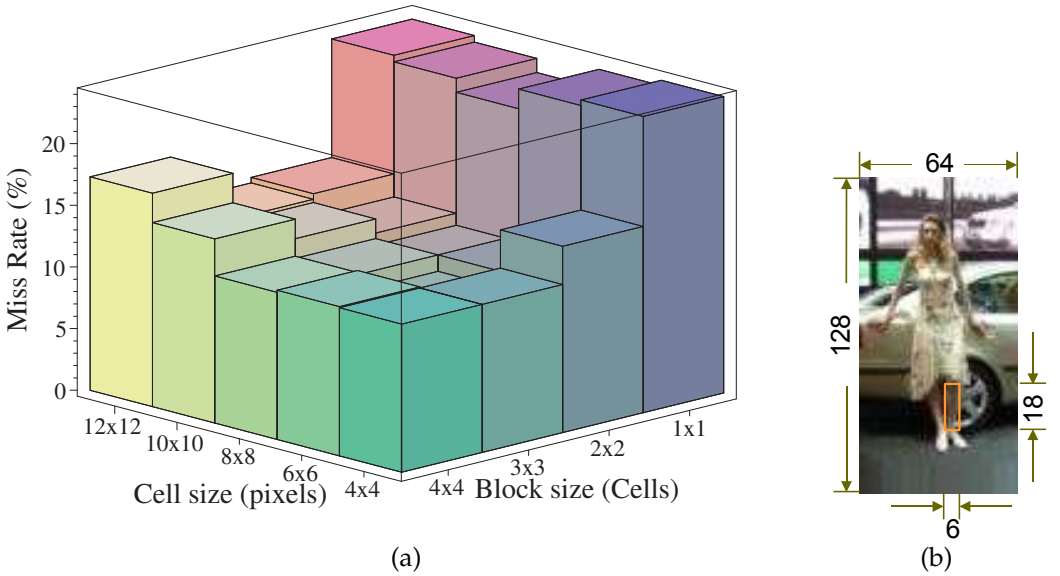


Fig. 4.5. *Effect of cell size and number of cells in the block on detection performance. (a) The miss rate at* 10^{-4} *FPPW as the cell and block sizes change. The stride (block overlap) is fixed at half of the block size.* 3×3 *blocks of* 6×6 *pixel cells perform best, with 10.4% miss rate. (b) Interestingly human limbs in* 64×128 *pixel normalized images are also around 6 pixels wide and 18 pixels long.*

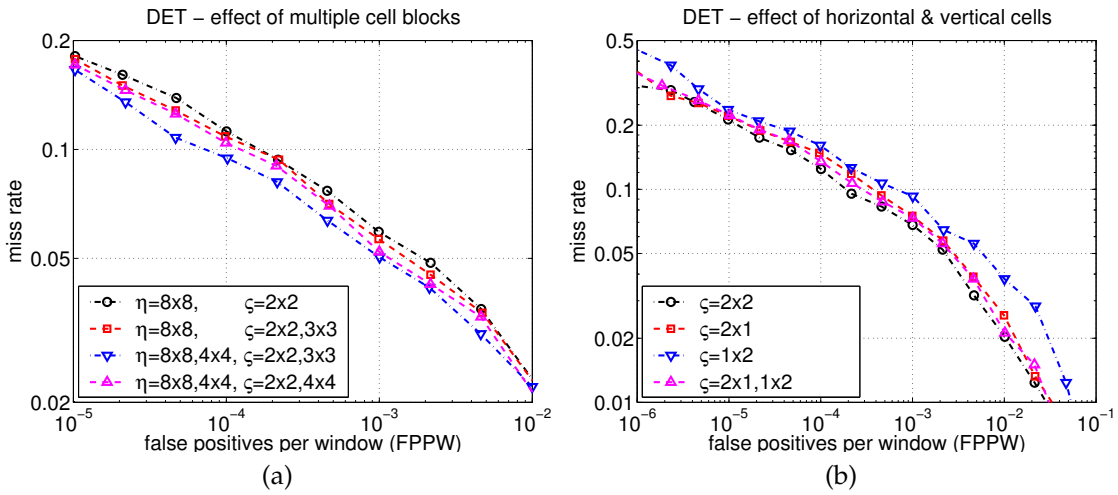
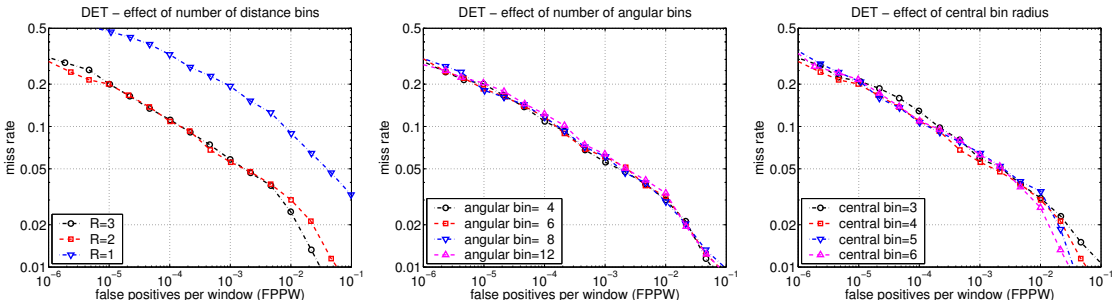


Fig. 4.6. *The effect of multiple-scale and rectangular blocks on detection performance. (a) Multiple block types with different cell and block sizes improve performance. (b) Vertical* (2×1) *blocks are better than horizontal* (1×2) *blocks.*

presents the results. Vertical and vertical+horizontal pairs are significantly better than horizontal pairs alone, but not as good as 2×2 or 3×3 cell blocks. Performance drops by 1% at 10^{-4} FPPW.

C-HOG.

The two variants of the C-HOG geometry, ones whose central cell is divided into angular sectors (Fig. 4.1(b)) and ones with a single circular central cell (Fig. 4.1(c)), perform equally well. We use single circular-centre variants as our default C-HOG variant, as these have fewer spatial cells than the divided centre ones. At least two radial bins (a centre and a surround) and four angular bins (quartering) are needed for good performance. Including additional radial bins does not change the performance much, *c.f*. Fig. 4.7(a), while increasing the number of angular bins decreases performance (by 1.3% at 10^{-4} FPPW when going from 4 to 12 angular bins), *c.f*. Fig. 4.7(b). Figure 4.7(c) shows that a central bin of 4 pixels radius gives the best results, but 3 and 5 pixel radii gives similar results. Increasing the expansion factor (log space radial increment) from 2 to 3 leaves the performance essentially unchanged. With these parameters, neither Gaussian spatial weighting (as in R-HOG) nor inverse weighting of cell votes by cell area changes the performance, but combining these two reduces it slightly. These values assume fine orientation sampling (we used 9 bins as default). Shape contexts (1 orientation bin) require much finer spatial subdivision to work well, usually 3 radial bins and as many as 12 angular bins, but their performance is still much lower than that of C-HOG. Figure 4.9 shows these results. Another variant of C-HOG, EC-HOG, uses binarised edges (*i.e*. thresholded gradients) to vote into the orientation histogram. Section 4.4 shows that such thresholding decreases performance.



(a) (b) (c)

Fig. 4.7. *Performance variations as a function of the C-HOG descriptor parameters. All results are with 9 orientation bins. (a) The effect of the number of radial bins (R) on performance. (b) Increasing the number of angular bins decreases performance marginally. It is best to use 4 bins (quartering). (c) The optimal central bin radius is 4 pixels.*

Centre-surround HOG.

Figure 4.4(c) ("*window norm*") shows that using centre-surround HOG decreases performance relative to the corresponding block based scheme (by 2% at 10^{-4} FPPW, for Gaussian pooling with $\sigma = 1$ cell widths). The reason is that there are no longer any overlapping blocks, so each cell is coded only once in the final descriptor. Including several normalisations for each cell

based on different pooling scales σ, but each centred on the cell location, provides no perceptible change in performance. So it seems that it is the existence of several pooling regions with *different* spatial offsets relative to the cell that is important in R-HOG, not the pooling scale. This point is further clarified in Sect. 4.5.

The centre-surround scheme has other advantages, notably that it can be optimised for much faster run-time. Recently, Zhu et al. [2006] showed that centre-surround HOG, in conjunction with integral histograms [Porikli 2005] and AdaBoost [Freund and Schapire 1996a,b, Schapire 2002], can be used to build a near real-time filter cascade style detector.

4.3.6 Detector Window and Context

Our 64×128 detection window includes about 16 pixels of margin around the person on all four sides. Figure 4.8(a) shows that this border provides a significant amount of context that helps detection. Decreasing it from 16 to 8 pixels (48×112 detection window) decreases performance by 4% at 10^{-4} FPPW. Keeping a 64×128 window but increasing the person size within it (again decreasing the border) causes a similar loss of performance, even though the resolution of the person has actually increased. So it is best to devote some of the window resolution feature vector components to coding context not subject.

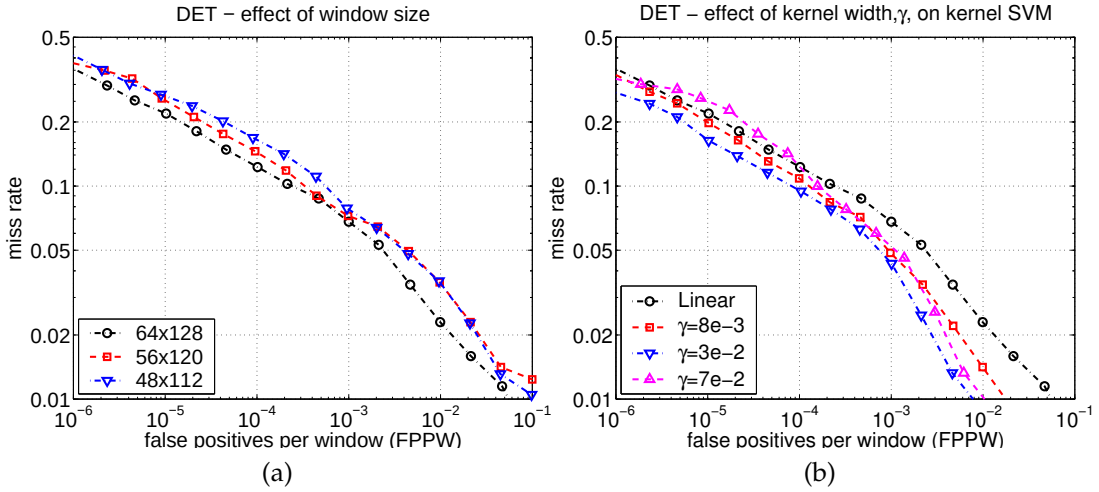


Fig. 4.8. *Effect of detection window size and kernel of SVM on the performance. (a) Reducing the 16 pixel margin around the* 64×128 *detection window decreases the performance by about 4%. (b) Using a Gaussian kernel SVM,* $\exp(-\gamma\|\mathbf{x_1} - \mathbf{x_2}\|^2)$*, improves the performance by about 3%.*

4.3.7 Classifier

By default we use a soft (C=0.01) linear SVM trained with SVMLight [Joachims 1999]. We modified SVMLight to reduce memory usage for problems with large dense descriptor vectors. Using a Gaussian kernel SVM increases performance by about 3% at 10^{-4} FPPW (see Figure 4.8(b)), but at the cost of a much higher run time.

Another possibility is to use cascaded AdaBoost as the classifier. This uses multi-level cascade of weak rejecters, each trained by combining elementary features using AdaBoost to form

a strong non-linear classifier. It avoids computation of unnecessary HOG features and offers significant run time improvements compared to SVMs. A comparative study of multiple scale HOG blocks in conjunction with AdaBoost is presented in Zhu et al. [2006].

4.4 Overall Results

Figure 4.9 shows the performance of the various window classifiers on the MIT and INRIA static person data sets. For details on these data sets see Appendix A. The main conclusions are similar to those presented in Sect. 3.5 for the complete final person detector. Overall the HOG-based classifiers greatly outperform the wavelet, PCA-SIFT and shape context ones, giving near-perfect separation on the MIT test set and at least an order of magnitude reduction in False Positive Per Window (FPPW) on the significantly harder INRIA static one.

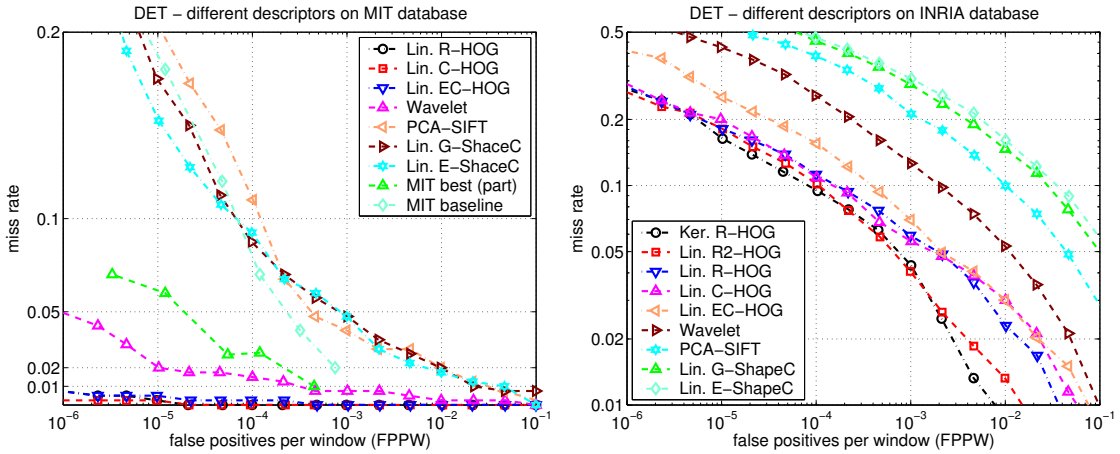


Fig. 4.9. *The performance of selected detectors on the (a) MIT and (b) INRIA static data sets. The HOG encoding outperforms all of the others. See the text for details.*

Figure 4.9(a) shows that the inclusion use of second order derivative like filters and oblique orientations significantly improve the performance of Haar wavelet descriptors (by 6.5% at 10^{-4} FPPW) compared to original MIT ones. However HOGs still outperform wavelets. The plot also shows MIT's best parts based and monolithic detectors. The points were interpolated from Mohan et al. [2001]. However note that an exact comparison is not possible as we do not know training-test partition of the Mohan et al. [2001] database and the negative images used are not available). The performances of the final R-HOG and C-HOG detectors are very similar, with C-HOG having a slight edge but being slower to compute owing to its log-polar grid. Figure 4.9(b) shows that the R2-HOG based descriptor including primitive bar detectors further improves the performance (by 2% at 10^{-4} FPPW). However it also doubles the feature dimension. Replacing the linear SVM with a Gaussian kernel one improves performance by about 3% at 10^{-4} FPPW, at the cost of much higher run times[3]. Keeping all other parameters the same, but using binary

[3] Note that we used hard examples generated by the *linear* R-HOG classifier to train the kernel R-HOG one, as kernel R-HOG generates so few false positives that its hard example set is too sparse to improve

edge voting (EC-HOG) instead of gradient magnitude weighted voting (C-HOG) decreases the performance by 5% at 10^{-4} FPPW. Omitting the orientation information decreases performance very substantially for both edges (E-ShapeC) and gradients (G-ShapeC) by 33% at 10^{-4} FPPW even when additional spatial or radial bins are added in an attempt to compensate [4]. PCA-SIFT also performs poorly. One reason is that, in comparison to Ke and Sukthankar [2004], many more (80 of 512) eigenvectors have to be retained to capture the same proportion of the variance. This may be because in the absence of a key point detector, the spatial registration is weaker so that patches vary much more. Increasing the number of eigenvalues further does not change detection performance.

4.5 Visual Cues for Person Detection

Figure 4.10 tries to give some intuition on the features that HOG based person detectors cue their decisions on. Figure 4.10(a) shows the average gradient image over all of the training examples. This suggests that the contours of the subject's head, shoulders, and legs and the point where the feet touch the ground are most likely to be relevant cues for classification. This is indeed the case. We can use the coefficients of the trained linear SVM classifier as a measure of how much weight each cell of each block has in the final discrimination decision and which orientations are the most important for each cell. Consider the R-HOG detector with overlapping blocks. Examination of Fig. 4.10(b,g) shows that the most important cells are the ones that frequently contain major human contours (especially the head and shoulders and the feet), normalised w.r.t. blocks lying *outside* the contour. In other words — despite the complex, cluttered backgrounds that are common in our INRIA static person training set — the detector cues mainly on the contrast of silhouette contours against the background, not on internal edges within the foreground (subject) or on silhouette contours normalised against the foreground. This point is pictorially highlighted in Fig. 4.10(d). It may be that patterned clothing and pose variations make internal regions unreliable as cues, or that foreground-to-contour transitions are confused by smooth shading and shadowing effects. Moreover, Fig. 4.10(c,h) illustrate that gradients inside the person (especially vertical ones) typically count as negative cues, presumably in order to suppress false positives in which long vertical lines trigger vertical head and leg cells.

Figure 4.11 shows some additional examples of descriptor encoding. A close look suggests that the linear SVM is able to map the positive example images to a set of canonical contours corresponding to humans. The last image in each triplet shows that the SVM maps different poses to similar set of canonical contours emphasising the subjects head, shoulders and legs.

4.6 Experiments on Other Classes

HOG descriptors are equally well suited to the detection of other object classes, and give state-of-the-art performance for many of the object classes that we tested. This section summarises

the generalisation significantly. This seems to be a general rule: using a classifier that is too strong in the first round training provides too few new hard negatives and gives less good results than retraining based on the negatives from a weaker classifier

[4] For both E-ShapeC and G-ShapeC, 12 angular and 3 radial intervals with inner centre circle of radius 4 pixels and outer circle of radius 16 pixels give the best results.

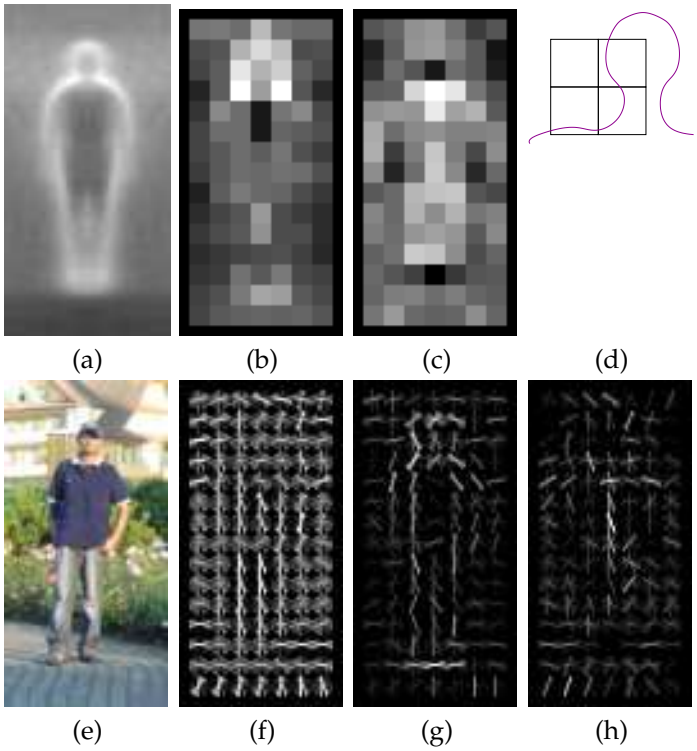


Fig. 4.10. *For the person class, the HOG classifiers cue mainly on silhouette contours, especially the head, shoulders and feet. More precisely the chosen cells are ones on the contour, normalised using blocks centred on the image background just* outside *the contour. (a) The average gradient image over the training examples. (b) Each "pixel" shows the maximum positive SVM weight in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) A sketch portraying the most relevant blocks – those lying just outside the contour. (e) A test image. (f) Its computed R-HOG descriptor. (g,h) The R-HOG descriptor weighted respectively by the positive and negative SVM weights. Only the dominant orientation is shown for each cell.*

the key HOG parameters for several other object classes. We optimised all of the key parameters for each object class in the Pascal[5] Visual Object Challenge (VOC) 2006[6]. This challenge targets image classification and localisation for 10 different classes: bicycle, bus, car, cat, cow, dog, horse, motorbike, person and sheep. Table 4.2 summarises the main changes that occurred. The overall conclusion is that most of the parameters are very similar to those for person class, and those that do vary can be easily grouped and structured. This can help us by providing quick first guess of the HOG parameters for any given new object class. The VOC object classes can be broadly divided into two groups: natural objects such as horses, cows and sheep, and man made objects such as cars, motorbikes and buses. We treat the person class as an exception and place it in a separate category: even though people are natural objects whose articulations in result in characteristics similar to the natural object category and their clothing results in appearance features similar to the man made object category. We now comment on how performance varies

[5] PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning) is a European Commission funded Network of Excellence programme.

[6] Details and results of the Pascal VOC 2006 challenge are available from `http://www.pascal-network.org/challenges/VOC/voc2006/index.html`

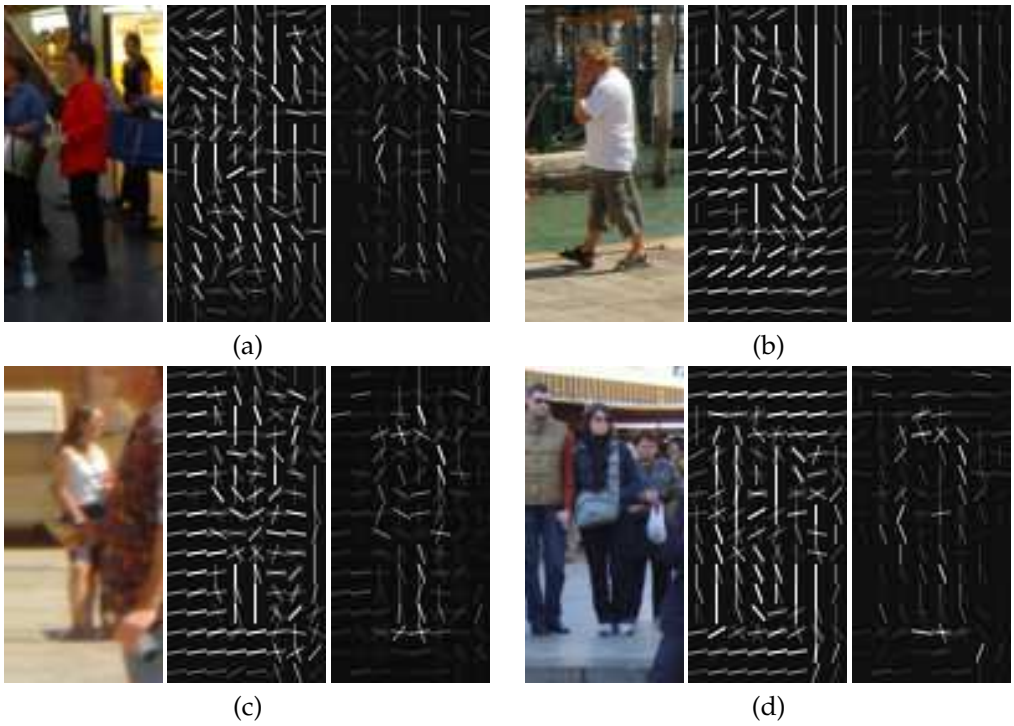


Fig. 4.11. *Visual cues encoded in the R-HOG descriptor. In each triplet, we display from left to right: (1) the input image, (2) the corresponding R-HOG feature vector (only the dominant orientation of each cell is shown), (3) the dominant orientations selected by the SVM (obtained by multiplying the feature vector by the corresponding weights from the linear SVM).*

as different parameters are changed. The person class has already been discussed in detail, here we just recall that all four of the combinations RGB or $\sqrt{\text{RGB}}$ with L1-Sqrt or L2-Hys provide

Table 4.2. *Key HOG parameters used for several different object classes. Experimental results are given in Sect. 4.6.*

Class	Window Size	Avg. Size	Orientation Bins	Orientation Range	Gamma Compression	Normalisation Method
Person	64×128	Height 96	9	(0−180°)	$\sqrt{\text{RGB}}$	L2-Hys
Car	104×56	Height 48	18	(0−360°)	$\sqrt{\text{RGB}}$	L1-Sqrt
Motorbike	120×80	Width 112	18	(0−360°)	$\sqrt{\text{RGB}}$	L1-Sqrt
Bus	120×80	Height 64	18	(0−360°)	$\sqrt{\text{RGB}}$	L1-Sqrt
Bicycle	104×64	Width 96	18	(0−360°)	$\sqrt{\text{RGB}}$	L2-Hys
Cow	128×80	Width 96	18	(0−360°)	$\sqrt{\text{RGB}}$	L2-Hys
Sheep	104×60	Height 56	18	(0−360°)	$\sqrt{\text{RGB}}$	L2-Hys
Horse	128×80	Width 96	9	(0−180°)	RGB	L1-Sqrt
Cat	96×56	Height 56	9	(0−180°)	RGB	L1-Sqrt
Dog	96×56	Height 56	9	(0−180°)	RGB	L1-Sqrt

similar results. Also for the image window sizes shown in the table, for all classes, cells of 8×8 or 6×6 pixels and blocks of 2×2 or 3×3 cells give good performance.

In the man made object category, for buses and motorbikes L1-Sqrt normalisation gives significantly better results than L2-Hys (miss rate increases by 5% and 3%, respectively, at 10^{-4} FPPW) and $\sqrt{\text{RGB}}$ gamma compression results in slight edge over the RGB image. For cars all four combinations (RGB/$\sqrt{\text{RGB}}$ + L1-Sqrt/L2-Hys) give similar results, with $\sqrt{\text{RGB}}$ and L1-Sqrt having a slight edge. Bicycles are an exception. The combination $\sqrt{\text{RGB}}$ and L1-Sqrt decreases the performance by more than 6% at 10^{-4} FPPW. The RGB and L2-Hys combination performs best, with the miss rate increasing by 1–2% for the remaining two. Signed gradients (over 0°–360° range) and 20° wide orientation bins give good performance for all objects in the man made category. Unsigned gradients decrease performance by 3% at 10^{-4} FPPW for bicycles and cars, 17% at 10^{-4} FPPW for buses, and give essentially same results for motorbikes. For motorbikes and bicycles, the R2-HOG (primitive bar detector) feature set further improves the results, respectively, by 3% and 1% at 10^{-4} FPPW.

In the natural object category, most of the images in cow and sheep classes contained side views of cream and light-brown coloured animals. Thus the gradient sign becomes relevant and significantly outperforms unsigned gradients, with performance improving by as much as 7% and 14% at 10^{-4} FPPW for cows and sheep, respectively. For both classes L2-Hys is best, improving results by more than 3% at 10^{-4} FPPW. However the horse, cat and dog classes are much harder to learn as the images contained large variations in pose, viewpoint and colour of the animals. The overall conclusion is that unsigned gradients over RGB channels and L1-Sqrt normalisation give the best results. Surprisingly, compared to L1-Sqrt, L2-Hys normalisation significantly decreased the performance for these classes. It is best to use the RGB image, probably because this allows fine texture to be captured during the gradient computation for these classes.

In summary, for man made object classes first try $\sqrt{\text{RGB}}$ with L1-Sqrt normalisations and 20° wide orientation bins over 0°–360° range, while for natural objects it is worth tyring both RGB with L1-Sqrt or $\sqrt{\text{RGB}}$ with L2-Hys normalisations. If there is lot of intra-class variations in colour, use unsigned gradients (orientations in range 0°–180°) with 20° wide bins in the histogram, else use signed gradients (range 0°–360°) with the same bin width. Section 5.5 presents the overall recall-precision curves for these object classes.

4.7 Encoding and Learning Algorithms

Figure 4.12 provides a step by step guide to the HOG descriptor calculation for R-HOG, R2-HOG or C-HOG descriptors. For all descriptors, the whole image is first preprocessed ("Common initial steps" in the algorithms below), and for any user supplied window location the descriptor computation is then performed on demand. The complete learning algorithm is provided in Fig. 4.13.

4.8 Discussion

This chapter has presented several key results. The fact that HOG descriptors greatly outperform wavelets and that any significant degree of smoothing before calculating gradients damages the HOG results emphasises that much of the available image information is from

Input: The scaled input image at the current scale and image window resolution
Output: The encoded feature vector for any user supplied location

Common initial steps:
(a) Optional: Gamma normalise each colour channel of the input image
(b) For each colour channel, convolve with $[-1, 0, 1]$ mask along x and y axis, and compute gradients. The channel with the largest magnitude gives the pixel's dominant orientation and magnitude.
(c) If using R2-HOG, smooth the image using Gaussian kernel of width ρ, compute the derivatives I_{ij}, $i, j \in \{x, y\}$, apply (4.1)–(4.2) and compute dominant orientation θ and magnitude as above

Descriptor computation: For each user supplied window locations
(a) Divide the image window into a dense uniformly sampled grid of points, and for each point
(b) If using R-HOG
  (1) Divide the $\varsigma\eta \times \varsigma\eta$ square pixel image region centred on the point into cells
  (2) Apply a Gaussian window with sigma $= 0.5 \times \varsigma\eta$ to image gradients in the block
  (3) Create a $\varsigma \times \varsigma \times \beta$ spatial and orientation histogram
  (4) For each pixel in the block, use trilinear interpolation to vote into the histogram using gradient magnitude
(c) If using R2-HOG, follow the same step as in R-HOG independently for both first and second order image gradients, *i.e.* create two 3-D histograms
(d) If using C-HOG,
  (1) Divide the image region centred on the point into a log-polar circular block; create angular and radial bins dividing the block into cells
  (2) Create a β bin orientation histogram for each cell
  (3) For each pixel in the block vote into the cell histograms using trilinear interpolation in log-polar-orientation space

Common final steps:
(a) Apply *L2-Hys* or *L1-Sqrt* normalisation independently to each block; if using R2-HOG, apply normalisation to each 3-D histogram independently
(b) Collect the HOGs of all blocks lying in the image window into one big descriptor vector

Fig. 4.12. *A summary of the R-HOG, R2-HOG and C-HOG encoding algorithm. For details on trilinear interpolation, see Appendix D.*

abrupt edges at fine scales. Blurring this in the hope of reducing the sensitivity to noise or spatial position is a mistake. In fact the amount of blur that the Haar wavelet like descriptors introduce effectively destroys much of the available object information. Rather then blurring before taking gradients and rectifying, the gradients should be calculated at the finest available scale (in the current pyramid layer), rectified or used for orientation voting, and only then blurred spatially. Given this, quite coarse spatial pooling can be used to obtain a degree of translation invariance without losing the ability to cue on sharp intensity transitions. In practice 6–8 pixel wide cells (one limb width) suffice.

On the other hand it pays to sample orientation rather finely: both wavelets and shape contexts lose out significantly here. For all classes, 20° wide orientation bins give good performance. The range of orientation bins (0–180° or 0–360°) is dictated by the object class – signed gradient information is usually preferable if relative differences in colour are important, *e.g.* lighter rims surrounded by dark wheels. Preserving gradient sign information does not seem to help for human detection, perhaps because humans wear clothes of all colours, while it does help for the

Input: Normalised and fixed resolution (width W_n and height H_n) positive windows; negative training images
Output: Trained binary classifier for object/non-object decisions on $W_n \times H_n$ image windows

Create initial negative examples once and for all by randomly selecting window locations on each negative image *First phase learning*:
(a) Calculate the supplied descriptor for all positive images
(b) Learn a linear SVM classifier on the supplied descriptor vectors

Generate hard negative examples: Perform a multi-scale scan for false positives on all the negative images
(a) Let the start scale be $S_s = 1$ and compute the end scale $S_e = \min\left(W_i/W_n, H_i/H_n\right)$, where W_i, H_i are the image width and height, respectively
(b) Compute the number of scale steps S_n to process

$$S_n = \text{floor}\left(\frac{\log(S_e/S_s)}{\log(S_r)} + 1\right)$$

where S_r is the multi-scale step constant
(c) For each scale $S_i = [S_s, S_s S_r, \ldots, S_n]$
(1) Rescale the input image using bilinear interpolation
(2) Apply the encoding algorithm and densely scan the scaled image with stride N_s for object/non-object detections
(3) Push all detections with $t(w_i) > 0$ (*i.e. hard examples*) to a list

Second phase learning:
(a) Estimate total number of hard examples that can be stored in RAM

$$\#\,\text{HardExamples} = \frac{\text{Total RAM}}{\text{Feature vector size}} - \#\,\text{PositiveExamples} - \#\,\text{NegativeExamples}$$

(b) If there are more hard examples than this, uniformly sample this number of hard examples and include them in the negative training set
(c) Learn the final SVM classifier on the positive windows, the initial negative examples and the generated hard examples

Fig. 4.13. *Complete window classifier learning algorithm.*

man made object classes and for consistently coloured natural ones. For certain classes, such as bicycles, including second order derivatives (bar detector) further improves the performance.

At the prepossessing stage, the uncompressed RGB gradient image can be used as the default for all object classes. However square root compression of image intensity often helps and is worth testing for any new object class.

Strong *local* contrast normalisation is essential for good results. One might have thought that a single large many celled HOG descriptor covering the whole detection window would give the best performance, but the results show that a more local normalisation policy improves the performance significantly. However it is still best to normalise over a finite spatial patch: normalising over orientations alone (a HOG block with a single spatial cell) worsens performance. The way that normalisation is done is also important. Lowe's L2-norm followed by hysteresis thresholding and renormalisation and L1-norm followed by square root of descriptor vector usually perform better than other normalisations we have tried. However it is seldom clear in

advance which one to use. Section 4.6 provides some guidelines, but it is best to test the two variants for each new object class.

Better results can be achieved by normalising each element (edge, cell) *several times* with respect to different local supports, and treating the results as independent signals. In our standard detector, each HOG cell appears four times with different normalisations. This may seem redundant as the only difference in their votes is the different normalisation in the four parent HOG blocks, but including this 'redundant' information improves performance of human detection from 84% to 89% at 10^{-4} FPPW. Physiological studies also highlight the fact that mammalian visual system has overlapping cells in its primary cortex [Fischer 1973, Hubel and Wiesel 1974]. Traditional centre-surround style schemes are probably not the best choice, but they offer significant speed advantages. If used in conjunction with AdaBoost, they offer comparable performance with significant speed up.

In summary, we have shown that using locally normalised histogram of gradient orientation features similar to SIFT descriptors [Lowe 2004] in dense overlapping grids gives very good results for person detection, reducing false positive rates by more than an order of magnitude relative to the best Haar wavelet based detector from Mohan et al. [2001]. The HOG encoding is generally useful and gives equally good performance for many other object classes. We systematically studied the influence of the various HOG encoding parameters for several object classes, ranging from man made objects like cars, bikes to natural object classes like cows, horses. Most of the HOG parameters were found to be remarkably stable and object type and expected image size gives useful hints about appropriate initial values for the remaining parameters.

Overall we studied the influence of various descriptor parameters and concluded that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalisation in overlapping descriptor blocks are all important for good performance.

René Magritte, *Carte Blanche*, Oil on canvas, 1965. ©National Gallery of Art, Washington.

5

Multi-Scale Object Localisation

Chapter 4 detailed the HOG feature set and showed evaluation results on window level object/non-object classifiers. This is only one component of the overall object detector. The final goal is to detect and precisely localise any objects that appear in the image. For detectors based on a binary object/non-object classifier, the detector scans the image with a detection window at all positions and scales, running the classifier in each window and fusing multiple overlapping detections to yield the final object detections. This fusion is an important stage of the process and how it should be performed is unclear. Many previous works either omit the issue entirely [Papageorgiou and Poggio 2000, Mohan et al. 2001, Fleuret and Geman 2001] or use somewhat heuristic methods [Rowley et al. 1998, Schneiderman and Kanade 2004, Viola and Jones 2004, Intel OpenCV 2006].

This chapter proposes a generalised solution to the fusion of multiple overlapping detections. Section 3.3 has already provided an overview of the approach; this chapter details it and studies its performance. In Sect. 5.1, we start with a discussion of the general characteristics that an ideal solution should have and propose a robust fusion of overlapping detections in 3-D position and scale space as the solution. The problem is posed as one of kernel-density estimation. We treat it as a suppression of non-maximum responses. So the solution is given by locating modes of the density estimate. For this we use a mean shift based mode detection procedure [Comaniciu 2003b,a]. Section 5.2 provides the mathematical details. The multi-scale scan and the non-maximum suppression procedure introduce some additional parameters, such as the scale resolution and the spatial and scale kernel widths for mean shift. Using the person detection as an example, Sect. 5.3 describes how best to chose these parameters. The main conclusion is that a fine grained scan of position and scale is necessary for good performance. The complete detection algorithm is presented in Sect. 5.4. Section 5.5 presents the overall results for other object (non-human) classes.

5.1 Binary Classifier for Object Localisation

Scanning detection window based object detection and localisation requires multiple overlapping detections to be merged. Our solution is based on the following two hypotheses:

- If the detector is robust, it should give a strong positive (though not maximum) response even if the detection window is slightly off-centre or off-scale on the object.
- A reliable detector will not fire with same frequency and confidence for non-object image windows.

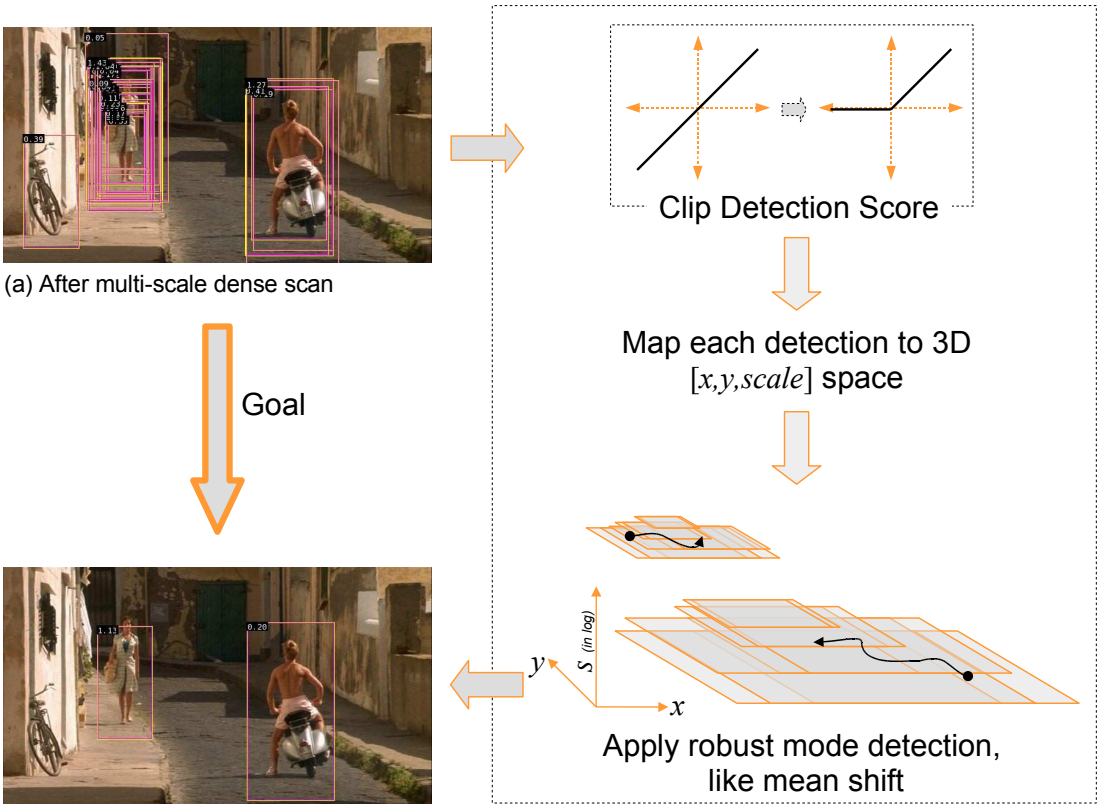


Fig. 5.1. *Non-maximum suppression for fusion of overlapping detections. (a) A typical result after scanning the binary classifier across the test image at all positions and scales. (b) The final fusion of the overlapping detections. (c) The steps of the non-maximum suppression algorithm. The first step maps the linear SVM scores to positive values. Negative values are zeroed via clipping function* $\max(score, 0)$*. The second step maps each detection to 3-D position and scale space. The third and final step applies the mean shift mode detection algorithm to each detection. The mean shift procedure provides local smoothing of the detections. The result is that overlapping detections (nearby ones in 3-D position-scale space) cluster together. The mode of each cluster provides the final detection result.*

The first hypothesis assumes that the detector response degrades gradually under small changes in object position or scale, but that the maximum response occurs only at the right position and scale. The second hypothesis implies that false positives are mainly due to accidental alignments, so that their probability of occurring consistently at several adjacent scale levels and positions is low. Figure 5.1(a) shows such an example. The detector fires multiple times in the neighbourhood of the true people, but it also miss-fires once for the bicycle at the bottom-left. The linear SVM output is high for the true positives and relatively low for the false positive. The goal is to fuse detections and to achieve results similar to Fig. 5.1(b).

An ideal fusion method would incorporate the following characteristics:

1. The higher the peak detection score, the higher the probability for the image region to be a true positive.

2. The more overlapping detections there are in the neighbourhood of an image region, the higher the probability for the image region to be a true positive.

3. Nearby overlapping detections should be fused together, but overlaps occurring at very different scales or positive positions should not be fused.

The third characteristic is based on the observation that the windows used to learn binary classifiers can be larger than the object to allow some context[1]. Thus there may be scenarios where detection windows overlap for nearby objects. Heuristic fusion approaches [Rowley et al. 1998, Schneiderman and Kanade 2004, Viola and Jones 2004, Intel OpenCV 2006] will not work for these cases. For example, Viola and Jones [2004] partitions the set of detections are into disjoint subsets, each subset providing a single final detection. As the training windows are larger than the actual person in our person data set, some detections may be overlapping and classified as single detection by Viola and Jones [2004] method even though the images contain two people occurring at very different scales (one detection occurring in another detection).

We represent detections using kernel density estimation (KDE) in 3-D position and scale space. KDE is a data-driven process where continuous densities are evaluated by applying a smoothing kernel to observed data points. The bandwidth of the smoothing kernel defines the local neighbourhood. The detection scores are incorporated by weighting the observed detection points by their score values while computing the density estimate. Thus KDE naturally incorporates the first two criteria. The overlap criterion follows from the fact that detections at very different scales or positions are far off in 3-D position and scale space, and are thus not smoothed together. The modes (maxima) of the density estimate correspond to the positions and scales of final detections.

The kernel width should be chosen to meet several criteria. It should not be less than the spatial and scale stride at which window classifiers are run, nor less than the natural spatial and scale width of the intrinsic classifier response (the former should obviously be chosen to be less than the latter). Also it should not be wider than the object itself so that nearby objects are not confused.

Notation.

Let $\mathbf{x}_i = [x_i, y_i]$ and s'_i denote the detection position and scale, respectively, for the i-th detection. The detection confidence score is denoted by w_i. For KDE, the weights w_i should be greater than zero and the feature space (3-D position and scale space) should be homogeneous. Thus if the raw input scores have negative values we need to map them through a positive transformation function $t(w_i)$ to get the weights for the KDE, and the detections are represented in 3-D space as $\mathbf{y} = [x, y, s]$, where $s = \log(s')$. This ensures detections homogeneity in the 3-D space, because when scanning the classifier across multiple scales, the scale steps usually follow a geometric sequence.

5.2 Non-maximum Suppression

Let $\mathbf{y}_i$, $i = 1 \ldots n$ be the set of detections (in 3-D position and scale space) generated by the detector. Assume that each point also has an associated symmetric positive definite 3×3 bandwidth or covariance matrix $\mathbf{H}_i$, defining the smoothing width for the detected position and scale

[1] *E.g.* we have 16 pixel margin on each side for the person class in the MIT pedestrian and the INRIA static person data set, *c.f.* Appendix A.

estimate. Overlapping detections are fused by representing the n points as a kernel density estimate and searching for local modes. If the smoothing kernel is a Gaussian, the weighted kernel density estimate at a point $\mathbf{y}$ is given by (*c.f.* Comaniciu [2003b,a])

$$\hat{f}(\mathbf{y}) = \frac{1}{n(2\pi)^{3/2}} \sum_{i=1}^{n} |\mathbf{H}_i|^{-1/2} t(w_i) \exp\left(-\frac{\mathrm{D}^2\left[\mathbf{y}, \mathbf{y}_\mathrm{i}, \mathbf{H}_\mathrm{i}\right]}{2}\right) \tag{5.1}$$

where

$$\mathrm{D}^2\left[\mathbf{y}, \mathbf{y}_\mathrm{i}, \mathbf{H}_\mathrm{i}\right] \equiv (\mathbf{y} - \mathbf{y}_\mathrm{i})^\top \mathbf{H}_\mathrm{i}^{-1} (\mathbf{y} - \mathbf{y}_\mathrm{i}) \tag{5.2}$$

is the Mahalanabois distance between $\mathbf{y}$ and $\mathbf{y}_i$. The term $t(w_i)$ provides the weight for each detection. The gradient of (5.1) is given by

$$\begin{aligned}
\nabla\hat{f}(\mathbf{y}) &= \tfrac{1}{n(2\pi)^{3/2}} \sum_{i=1}^{n} |\mathbf{H}_i|^{-1/2} \mathbf{H}_i^{-1} (\mathbf{y}_i - \mathbf{y})\, t(w_i) \exp\left(-\frac{\mathrm{D}^2\left[\mathbf{y}, \mathbf{y}_\mathrm{i}, \mathbf{H}_\mathrm{i}\right]}{2}\right) \\
&= \tfrac{1}{n(2\pi)^{3/2}} \left[\sum_{i=1}^{n} |\mathbf{H}_i|^{-1/2} \mathbf{H}_i^{-1} \mathbf{y}_i\, t(w_i) \exp\left(-\frac{\mathrm{D}^2\left[\mathbf{y}, \mathbf{y}_\mathrm{i}, \mathbf{H}_\mathrm{i}\right]}{2}\right)\right. \\
&\qquad \left. - \left\{\sum_{i=1}^{n} |\mathbf{H}_i|^{-1/2} \mathbf{H}_i^{-1}\, t(w_i) \exp\left(-\frac{\mathrm{D}^2\left[\mathbf{y}, \mathbf{y}_\mathrm{i}, \mathbf{H}_\mathrm{i}\right]}{2}\right)\right\} \mathbf{y}\right]
\end{aligned} \tag{5.3}$$

Let ϖ_i be the weights defined as

$$\varpi_i(\mathbf{y}) = \frac{|\mathbf{H}_i|^{-1/2} t(w_i) \exp\left(-\mathrm{D}^2\left[\mathbf{y}, \mathbf{y}_\mathrm{i}, \mathbf{H}_\mathrm{i}\right]/2\right)}{\sum_{i=1}^{n} |\mathbf{H}_i|^{-1/2} t(w_i) \exp\left(-\mathrm{D}^2\left[\mathbf{y}, \mathbf{y}_\mathrm{i}, \mathbf{H}_\mathrm{i}\right]/2\right)} \tag{5.4}$$

and satisfy $\sum_{i=1}^{n} \varpi_i = 1$. Dividing (5.3) by (5.1) and using (5.4) we have

$$\frac{\nabla\hat{f}(\mathbf{y})}{\hat{f}(\mathbf{y})} = \sum_{i=1}^{n} \varpi_i(\mathbf{y})\, \mathbf{H}_i^{-1} \mathbf{y}_i - \left(\sum_{i=1}^{n} \varpi_i(\mathbf{y})\, \mathbf{H}_i^{-1}\right) \mathbf{y} \tag{5.5}$$

Let

$$\mathbf{H}_h^{-1}(\mathbf{y}) = \sum_{i=1}^{n} \varpi_i(\mathbf{y})\, \mathbf{H}_i^{-1} \tag{5.6}$$

be the data weighted harmonic mean of the covariance matrices $\mathbf{H}_i$ computed at $\mathbf{y}$. From (5.5) and (5.6), the variable bandwidth mean shift vector is defined as

$$\mathbf{m}(\mathbf{y}) = \mathbf{H}_h \frac{\nabla\hat{f}(\mathbf{y})}{\hat{f}(\mathbf{y})} \equiv \mathbf{H}_h(\mathbf{y}) \left[\sum_{i=1}^{n} \varpi_i(\mathbf{y})\, \mathbf{H}_i^{-1} \mathbf{y}_i\right] - \mathbf{y} \tag{5.7}$$

At the mode location, the gradient $\nabla\hat{f}(\mathbf{y}) = 0$, implying $\mathbf{m}(\mathbf{y}) = 0$. Equation (5.7) suggests that the mode can be iteratively estimated by computing

$$\mathbf{y}_m = \mathbf{H}_h(\mathbf{y}_m) \left[\sum_{i=1}^{n} \varpi_i(\mathbf{y}_m)\, \mathbf{H}_i^{-1} \mathbf{y}_i\right] \tag{5.8}$$

starting from some $\mathbf{y}_i$ until $\mathbf{y}_m$ does not change anymore.

It can be proved, *c.f.* Comaniciu [2003a], that equations (5.8) and (5.6) define a method for fusioning of multiple overlapping detections $\mathbf{y}_i$, $i = 1, \ldots, n$ where $\mathbf{y}_m$ is the fusion estimate and

$\mathbf{H}_h$ is the convex combination of the detection covariances. $\mathbf{H}_h$ is a consistent and conservative estimate of the true uncertainty of the $\mathbf{y}_m$. The derivation is the same as in Comaniciu [2003b,a], except that we have an additional $t(w_i)$ term throughout.

A mode seeking algorithm can be derived by iteratively computing the variable bandwidth mean shift vector (5.7) for each data point until it converges, *i.e.* the new mode location (5.8) does not change anymore. For each of the n point the mean shift based iterative procedure is guaranteed to converge to a mode[2]. For more details on the mean shift mode seeking algorithm and its properties see Comaniciu et al. [2001], Comaniciu [2003b]. Typically several points would converge to the same mode and the set of all modes represents the final detection results. The location of each mode in 3-D gives the detection location and scale, and the value of the peak represents the detection score.

Detection Uncertainty Matrix $\mathbf{H}_i$.

One key input to the above mode detection algorithm is the amount of uncertainty $\mathbf{H}_i$ to be associated with each point. We assume isosymmetric covariances, *i.e.* the $\mathbf{H}_i$'s are diagonal matrices. Let $\texttt{diag}[\mathbf{H}]$ represent the 3 diagonal elements of $\mathbf{H}$. We use scale dependent covariance matrices such that

$$\texttt{diag}[\mathbf{H}_i] = [(\exp(s_i)\sigma_x)^2, (\exp(s_i)\sigma_y)^2, (\sigma_s)^2] \tag{5.9}$$

where σ_x, σ_y and σ_s are user supplied smoothing values. Scaling the smoothing values σ_x and σ_y by the term $\exp(s_i)$ increases the spatial uncertainty of the detected point. This should be the natural behaviour – the larger the detection scale the higher the uncertainty in the location estimate. The exponential is used as the scale values s_i are on a log scale.

Transformation function $t(w_i)$.

Another input is the transformation function $t(w_i)$. Mean shift requires positive weights $t(w_i) > 0, \forall w_i$. We studied three functions:

- Hard clipping of negative scores to zero. The function $t(w)$ becomes

$$t_{hc}(w) = \begin{cases} 0 & \text{if } w < c \\ w - c & \text{if } w \geq c \end{cases} \tag{5.10}$$

  where c is a threshold. For linear SVMs we usually use $c = 0$. Function (5.10) introduces rejection mechanism where most of the (negative) classified windows are discarded at the first step, and the mean shift is performed only on the relatively few detections greater than the threshold c.

- Soft clipping of negative scores. The function $t(w)$ becomes

$$t_{sc}(w) = a^{-1} \log(1 + \exp(a(w + c))) \tag{5.11}$$

  where a defines how aggressively the function $t_{sc}(w)$ approaches the hard clipping function and parameter c defines the shift.

- Fit a parametric form of sigmoid function to the linear SVM scores w_i during the training, and use the output of this. The functional form of $t(w)$ becomes

[2] In theory it can converge to a saddle point also, however we find that for the fusion of overlapping detections this occurs very rarely.

$$t_s(w) = \frac{1}{1 + \exp(a(w + c))} \tag{5.12}$$

where parameters a and c are evaluated to give the best probability estimates for the training outputs. We use the algorithm from Platt [2000] to compute a and c from the training data. The function (5.12) maps all of the detections (even negative ones) to probability scores and thus preserves more information. However this does not necessarily imply that it achieves better recall and/or precision rates, see Sect. 5.3 for details.

Figure 5.2 compares the three transformation functions.

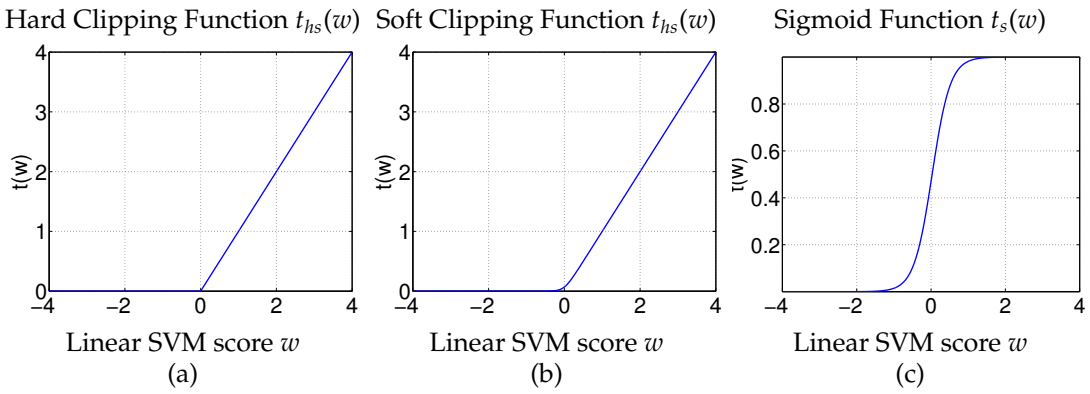


Fig. 5.2. *Three classes of transformation functions $t(w)$. (a) The hard clipping function as defined in (5.10). $c = 0$ gives good performance for most test cases and is used as the default. (b) The soft clipping function as defined in (5.11). We use $a = 10$ and $c = 0$ as default values. (c) The sigmoid function (5.12) with $a = -4.1$ and $c = -0.029$. The parameters a and c are estimated using Platt's method. The values here correspond to the R-HOG descriptor on the person class.*

5.3 Evaluation of Non-maximum Suppression

The scanning and non-maximum suppression process introduces a few additional parameters into the overall detection system: (a) the *scale ratio* S_r used to compute the scale-steps during multi-scale scan; (b) the *window stride* N_s used to scan the positions (*i.e.* the distance between two consecutive classifier windows) at each scale level; and (c) the *smoothing parameters* σ_x, σ_y, and σ_s used in the non-maximum suppression stage. These parameters can have a significant impact on performance so proper evaluation is necessary. We used the default person detector from Sect. 4.3 for the evaluations in this section. For all of the results here, unless otherwise noted, a scale ratio of 1.05, a stride of 8 pixels, and $\sigma_x = 8$, $\sigma_y = 16$, $\sigma_s = \log(1.3)$ are used as default values.

5.3.1 Transformation Function $t(w)$

Figure 5.3(a) compares the effect of the transformation function $t(w)$. The simple hard clipping (5.10) and soft clipping (5.11) functions outperform the sigmoid function. The integrated Average Precision (AP) (*c.f.* Appendix B) for the hard clipping and soft clipping is 0.739 and 0.746,

respectively, whereas sigmoid AP stands at 0.667. The values AP for the soft and hard clipping are quite close but it should be noted that even small changes in AP values are significant[3]. The value of c is usually classifier dependent, but $c = 0$ gives good results for most linear SVM based classifiers. For the default person detector, our experiments show that $c = 0$ is the optimal value. Figure 5.3(a) shows that soft clipping gives better recall than hard clipping as it uses more information, however a large scale value $a \geq 10$ (and hence a t function with a sharp "knee") is must. At $a = 5$ drop in the precision is significant, with the AP reducing to 0.661. The reason for this drop is, that for small a, the softmax function maps many more negative scores to small but nonnegligible positive weights and the large number of regions creates spurious detections around negative score local maxima. These detections are ignored only at low recall values (recall < 0.5). Similar reasoning applies to the sigmoid function (5.11). The advantage of hard clipping is that its rejection mechanism drastically reduces the number of detections to be fused, thus significantly speeding up the mean shift based mode detection process. For the remaining evaluations in this section we use the hard clipping function because it is fast and gives good results without any fine tuning.

5.3.2 Scale Ratio and Window Stride

Fine scale sampling turns out be crucial for good performance in non-maximum suppression. A scale ratio of 1.01 gives the best performance, but significantly slows the overall process. Changing the scale ratio from 1.01 to 1.05 reduces the AP by 0.015 to 0.739. For scale ratios larger than 1.05, the AP drops more quickly reducing to 0.708 at 1.10 and eventually to 0.618 at 1.30. Figure 5.3(b) summarises the results. It is equally important to use a fine spatial stride. Fine steps always yield better recall rates and usually also improve the precision marginally. Decreasing the stride from 8 to 2 improves the recall from 0.837 to 0.856, while increasing it to 16 (twice the descriptor cell width and one-fourth of the object width) drastically reduces it. Table 5.1 summarises the effect of stride on AP and maximum recall.

Table 5.1. *The effect of window stride on average precision and maximum recall. Fine spatial sampling always helps. Increasing stride to values larger than the cell size reduce the performance significantly.*

Stride	2	4	8	12	16
Average Precision	0.747	0.735	0.739	0.702	0.655
Maximum Recall	0.856	0.852	0.837	0.796	0.742

Note that the false detection rate in classifier window DET curves is not indirectly related to recall precision and overall detector performance. For example, with a scale ratio of 1.05 and a window stride of 8 pixels the detector tests approximately 26.9 million windows for all test images in INRIA static data set. Reducing the scale ratio to 1.01 and the window stride to 2, the number of tested windows increases to 1976 million, all the while improving the AP to 0.768, without the false positive rates per classifier window being changed at all.

[3] For example if the precision is exactly the same for two recall-precision curves, but the recall is larger for one of them, an improvement in AP of 0.01 at a precision at highest recall value (at recall $= 0.83$ for hard clipping case in Fig. 5.3(a)) of approximately 0.5, implies a 2% improvement in recall.

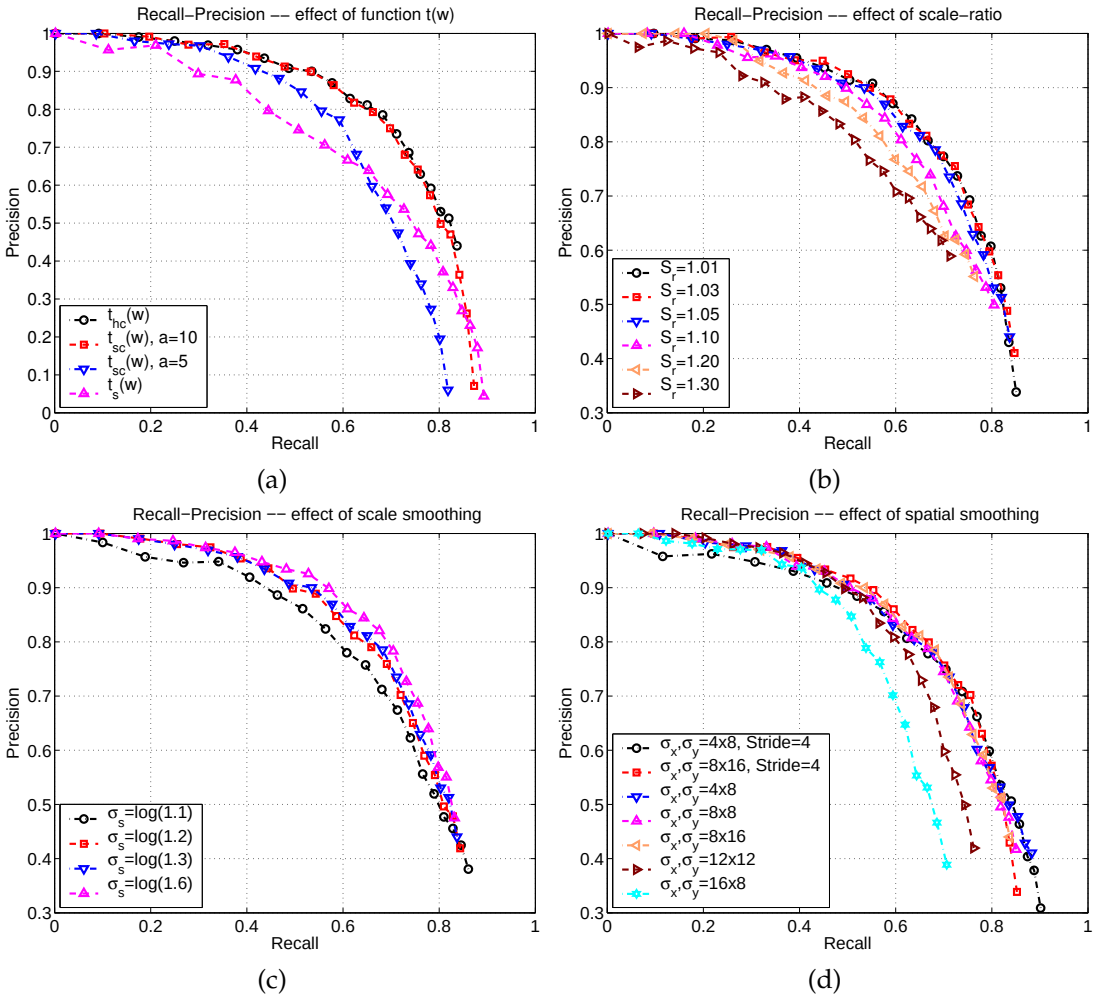


Fig. 5.3. *Recall-precision curves showing the effect of different non-maximum suppression parameters on overall performance. (a) Comparison of different $t(w)$ functions. The hard clipping and soft clipping ($a = 10$) functions perform well. (b) Fine scale sampling is important for good performance. (c) Scale smoothing in the range of* log(1.3)–log(1.6) *gives best performance and is independent of the scale ratio used. (d) There is a trade off in spatial smoothness. For values that are too low or too high, the performance decreases. Spatial smoothness equal to the window stride gives good performance for all cases.*

5.3.3 Smoothing Parameters

Figure 5.3(c) shows the influence of scale smoothing σ_s on the performance. Insufficient scale smoothing $\sigma_s = \log(1.1)$ reduces the performance: Although it results in a marginally higher overall recall (probably because fewer of the detections are fused together), the precision drops because multiple detections for the same ground truth object are not fused. $\sigma_s = \log(1.1)$ results in AP of 0.715, rising to AP of 0.750 for $\sigma_s = \log(1.6)$. Scale smoothing of log(1.3)–log(1.6) gives good performance for most object classes.

Table 5.2. *Spatial smoothing proportional to the window stride gives the best results. Smoothing should be adapted to the window shape. For example, for upright humans, (64 × 128) smoothing $\sigma_x \times \sigma_y = 2A \times A$, $A \in \{4, 8\}$, decreases performance much more than smoothing of $A \times A$ or $A \times 2A$.*

Spatial Smoothing $[\sigma_x \times \sigma_y]$	4×4	4×8	8×8	8×12	8×16	12×12	12×16	12×24	8×4	16×8
Stride 4×4	0.738	**0.755**	0.748	0.746	0.751	0.688	0.687	0.686	0.726	0.614
Stride 8×8	0.737	**0.759**	0.743	0.740	0.739	0.679	0.677	0.677	0.728	0.617

(a) (b)

Fig. 5.4. *Classifier response for a dense scan (at every pixel) of the person detector at a scale level. (a) The input image. The highlighted image region corresponds to the detector's normalised image window size. (b) The detector response after a dense scan of the input image at the given scale level. The maximum of the response is indeed centred on the highlighted image region. Note the distribution of the response at centre of the image region. It is vertically stretched and proportional to the aspect ratio of the detection window.*

The optimal amount of spatial smoothing depends on the window stride and the descriptor cell size[4]. We evaluated spatial smoothing over a wide range for window stride of 4 and 8 pixels. Figure 5.3(d) presents the RP curves and Table 5.2 summarises the APs. An interesting observation is that smoothing values similar to the window stride give good performance, *e.g.* for a stride of both 4 and 8 pixels, $[\sigma_x, \sigma_y] = [4, 8]$ give the best performance. This agrees well with the actual distribution of the confidence scores at any scale level. Figure 5.4 gives an example. For all of the object classes tested, we find that the distribution of the confidence score near a peak is roughly proportional to the aspect ratio of the detection window. Smoothing in the opposite way reduces performance. For example, for our person class with normalised window size of 64 × 128, smoothing of 8 × 4 and 16 × 8 decrease performance substantially to AP of 0.728 and 0.617 respectively. Similarly, smoothing equal to the cell size gives good performance, while large amounts of smoothing decrease the performance.

[4] The spatial smoothing could also be dependent on the block stride used to compute descriptors. However the stride is related to both the cell size and the block size. The cell size controls the amount of spatial pooling and the block size controls the local normalisation. For a dense redundant representation, the block stride should not be more than half the block size, but taking it equal to or less than the cell size is usually preferable. For almost all of our experiments the block size was in fact the same as the cell size, so here we compare the window stride to the descriptor cell size only.

We also tested another variant of mode finding that applies explicit Gaussian smoothing to the detections by recursive filtering in 3-D scale space. Maxima are located by searching for local maxima over the 27 nearest neighbours for each tested location in 3-D. This method can accept signed linear SVM outputs without any transformation function $t(w)$, but its performance is significantly worse than mean shift based method with hard clipping. If the sigmoid function $t_s(w)$ is used, the maximas should be ideally close to those located by the mean shift procedure. However the disadvantage is that its implementation has numerical issues as both the computation of Gaussian smoothing across scales and detection of local maximas in 3-D space requires interpolation, and despite careful implementation these numerical issues remain. Hence we do not use it.

5.4 The Complete Detection Algorithm

The complete detection algorithm is presented in Fig. 5.4. The stride is usually taken to be equal to the cell size in R-HOG and to the diameter of the centre-cell in C-HOG. A scale stride S_r of 1.05 works well in practice. We use $\sigma_s = \log(1.3)$ and $[\sigma_x, \sigma_y]$ proportional to the aspect ratio of the detection window times the stride N_s, *e.g.* for the default person detector $[\sigma_x, \sigma_y] = [8, 16]$.

We stop mean shift iterations when the mean shift vector (5.7) is below some user supplied threshold. Due to numerical inaccuracies and irregular structure, after convergence, usually

Input:
(a) Test image
(b) Trained window classifier with normalised window of width W_n and height H_n
(c) Scale step size S_r, stride for spatial scan N_s, and sigma values σ_x, σ_y, and σ_s
Output: Bounding boxes of object detections

Initialise
(a) Let start scale $S_s = 1$; compute end scale $S_e = \min\left(W_i/W_n, H_i/H_n\right)$, where W_i and H_i are image width and height, respectively
(b) Compute the number of scale steps S_n to process

$$S_n = \text{floor}\left(\frac{\log(S_e/S_s)}{\log(S_r)} + 1\right)$$

For each scale $S_i = [S_s, S_s S_r, \ldots, S_n]$
(a) Rescale the input image using bilinear interpolation
(b) Extract features (Fig. 4.12) and densely scan the scaled image with stride N_s for object/non-object detections
(c) Push all detections with $t(w_i) > c$ to a list

Non-maximum suppression
(a) Represent each detection in 3-D position and scale space $\mathbf{y}_i$
(b) Using (5.9), compute the uncertainty matrices $\mathbf{H}_i$ for each point
(c) Compute the mean shift vector (5.7) iteratively for each point in the list until it converges to a mode
(d) The list of all of the modes gives the final fused detections
(e) For each mode compute the bounding box from the final centre point and scale

Fig. 5.5. *The complete object detection algorithm.*

several points in the basin of the attraction of a mode cluster around the true location of the mode. We group these mode candidates using a proximity measure. The final location is the mode corresponding to the highest density. Theoretically, (5.7) may converge to a saddle point, but for this application we do not observe this case in practice. In our experiments we find that the points take 4–5 iterations to converge to modes. Thus a naive implementation of mean shift is sufficient. However if the detector is a weak classifier, it may result in lots of detections and can considerably slow down the non-maximum suppression. Several tricks exist to speed up the mean shift iterations, for details see Comaniciu and Meer [2002].

Using the above algorithm and the default detector from Sect. 4.3, Fig. 5.6 shows some examples detections on the INRIA static person data set. The method detects people reliably in any upright pose, including people standing in crowds, and works for varying background clutter and in very dark and low contrast images. Typically the false positives result from head-shaped contours supported by vertical gradients below it while misses occur either if the persons' poses are different from those present in training examples or if people are too close to the image boundaries[5].

5.5 Overall Results for Other Object Classes

Section 4.6 presented the key parameter choices for other object (non-human) classes. This section provides the overall recall-precision curves for all of the object classes in which we participated in the PASCAL Visual Object Classes (VOC) challenge [Everingham et al. 2006a]. For each object class, we used the best parameter combination as described in Sect. 4.6 to train the classifier. Each classifier was learned on the supplied *Training Set* and parameter optimisation was performed using the *Validation Set*. The best detector was used to obtain results on the *Test Set*. The results presented here are on the *Validation Set* because at the time of writing this dissertation we did not have access to the ground-truth on the *Test Set*.

Data Preparation.

For training and validation we used the size-normalised object boxes from the positive *Training* and *Validation Set*. As described in Sect. 3.4, the corresponding negatives were sampled randomly from negative images. As in the person detector, we included an 8 or 16 pixel margin around the image windows for the cow, sheep, motorbike and bicycle classes to allow for context. For the remaining classes, most of the ground-truth object instances were too close to the image boundary and we found that including margins around the normalised windows resulted in added bias during the training, so no margins were used.

Summary of Results.

The PASCAL 2006 VOC challenge had 10 object classes. We participated in the following 7: person, car, motorbike, bicycle, bus, cow and sheep. Figure 5.7 shows the overall recall-precision for these object classes, separated into man made and natural object categories. The approach was unable to learn a detector with performance significantly above random for cats, dogs and horses. The probably reason for this is the large amount of shape variation in the examples from

[5] Windows close to the image boundaries are not evaluated in our person detector due to the required extra 8 pixel margin in the normalised image windows of our INRIA static person data set.

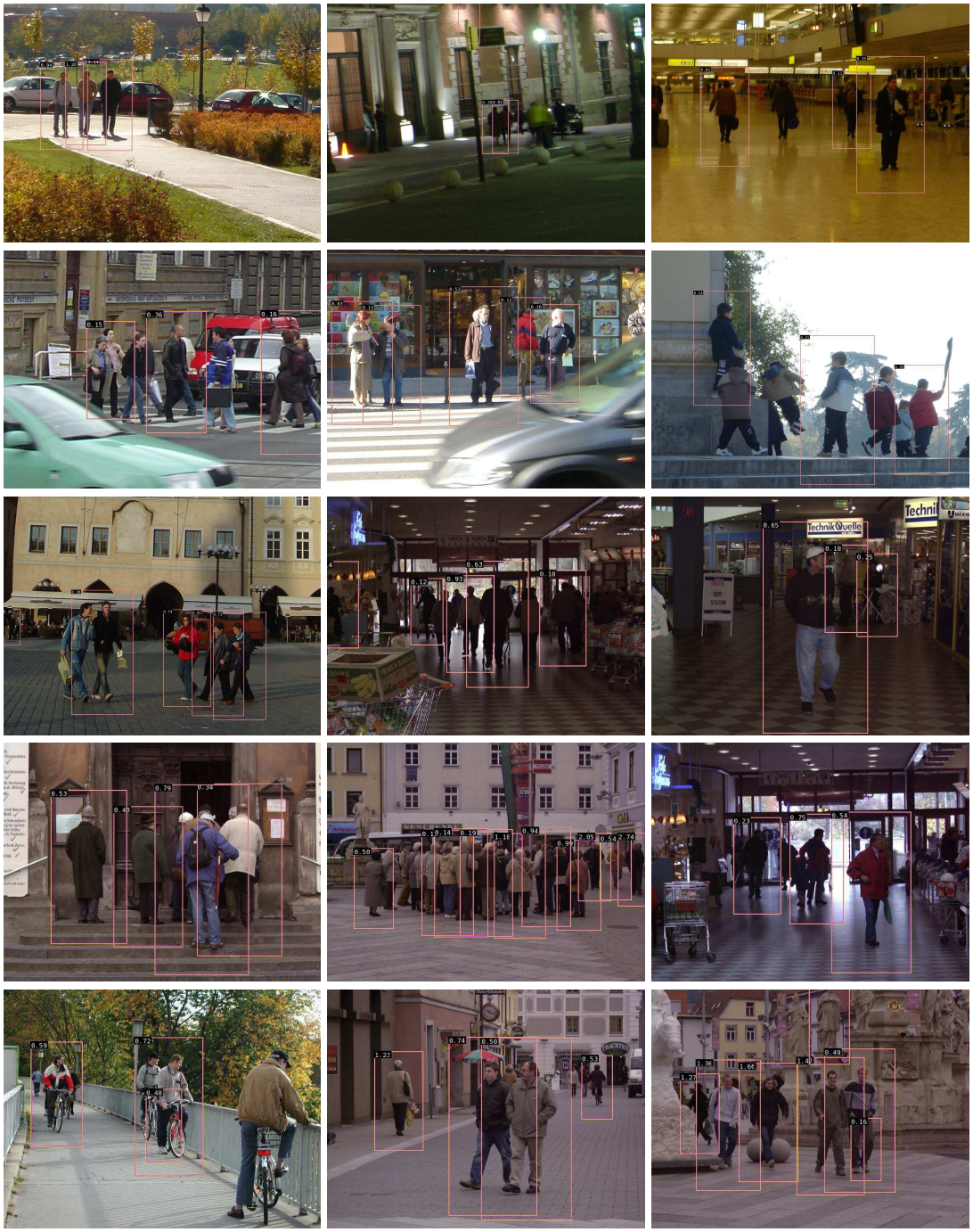


Fig. 5.6. *Some examples of detections on test images for the final person detector. The detector is able to detect people in crowds in varied articulated poses with different clothing, background clutter and illumination. Some images also show missed detections and/or false positives. The threshold used to display the detections in this figure corresponded to a recall rate of* 0.7 *at a precision of* 0.75*.*

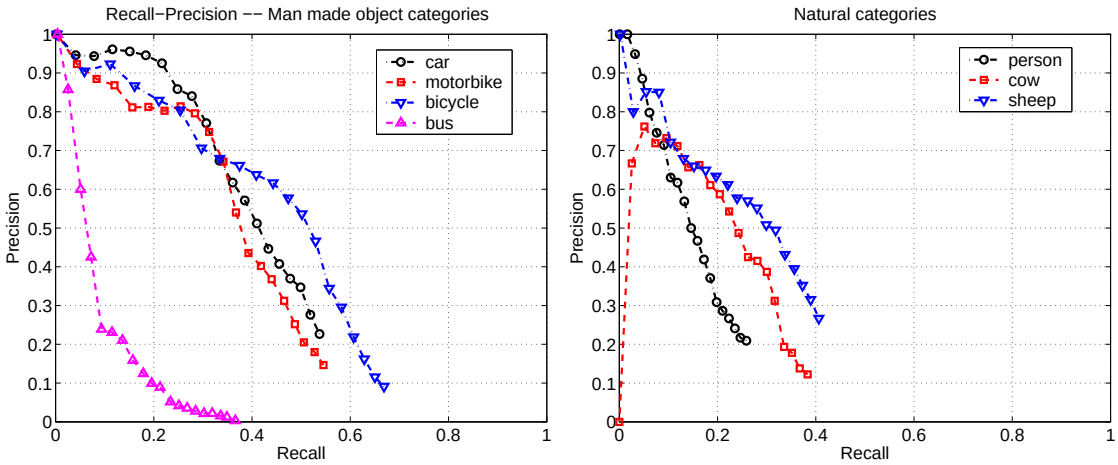


Fig. 5.7. *Overall recall-precision curves for the different object classes in the PASCAL 2006 VOC challenge. (a) Results on man made objects. (b) Results on natural objects.*

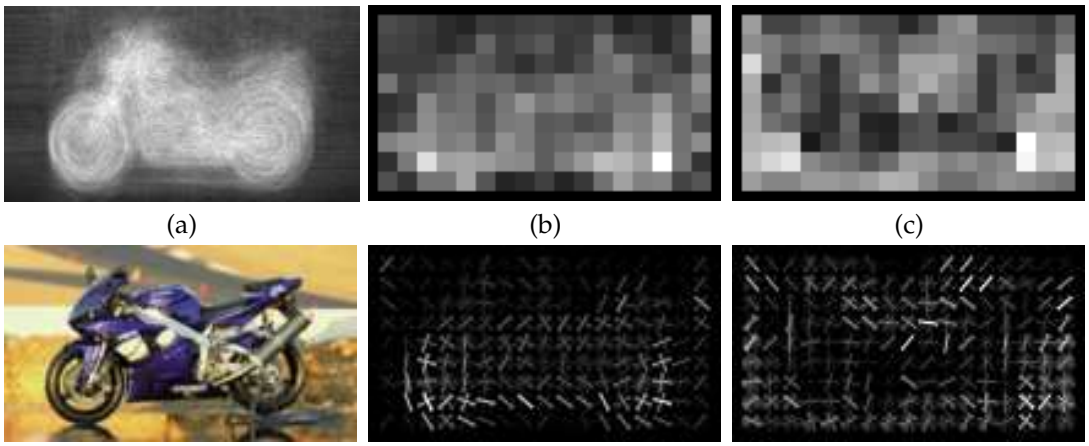
these classes. For example cat appear in a wide variety of body poses and orientations. On the other hand, our detector gave the best results in the object detection competition for the person, car, motorbike and sheep classes. For all of these classes the detector outperformed the other competing approaches by a significant margin. It was also ranked second for the bicycle class, being outperformed only by an AdaBoost based classifier that also used HOG descriptors. The performance remained comparable to the best detectors for buses and cows. A detailed comparison of our detectors with other approaches on the PASCAL test set is given in Everingham et al. [2006a]. Unlike other participants who learned different classifiers for different views such as left, right, front and back views, we learned only one linear classifier for each object class irrespective of the variation in views. This clearly highlights the discriminative power of the HOG descriptors.

Figure 5.8 and Fig. 5.9 show, respectively, the features that R-HOG descriptors cue their decisions on for motorbikes and cars. For motorbikes the blocks corresponding to the motor bikes wheel and rim are the most important, while for the cars roof, wheels and front/rear bonnets are the most important.

5.6 Conclusions

This chapter detailed our object detection and localisation algorithm for classification window scanning detectors. Although such object detectors abound in the literature [Rowley et al. 1998, Papageorgiou and Poggio 2000, Mohan et al. 2001, Fleuret and Geman 2001, Schneiderman and Kanade 2004, Viola and Jones 2004], we present a generalised method for the fusion of overlapping detections. The algorithm not only fuses multiple detections at nearby locations and scales and incorporates detection confidences, but also allows the detection of overlapping object instances occurring at very different scales.

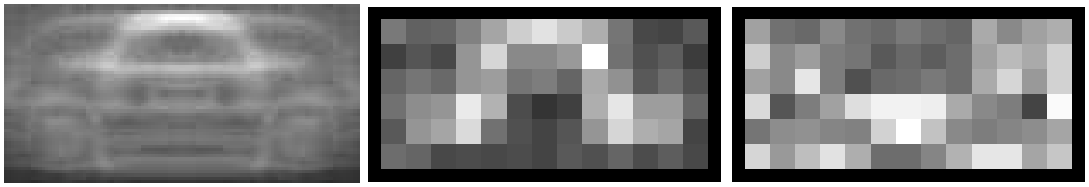
We studied the influence of all of the parameters that affect the performance of the fusion algorithm. The most important conclusion is that as the parameters change, the performance varies substantially even when exactly the same binary classifier is used for all evaluations. In



(a) (b) (c)

(d) (e) (f)

Fig. 5.8. *The feature information encoded by R-HOG for motorbikes. (a) The average gradient image over left-sided views in normalised motorbike images. (b) Each "pixel" shows the sum of all of the positive SVM weights in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) A test image. (e,f) The R-HOG descriptor weighted respectively by the positive and the negative SVM weights. Only the dominant orientation is shown for each cell.*



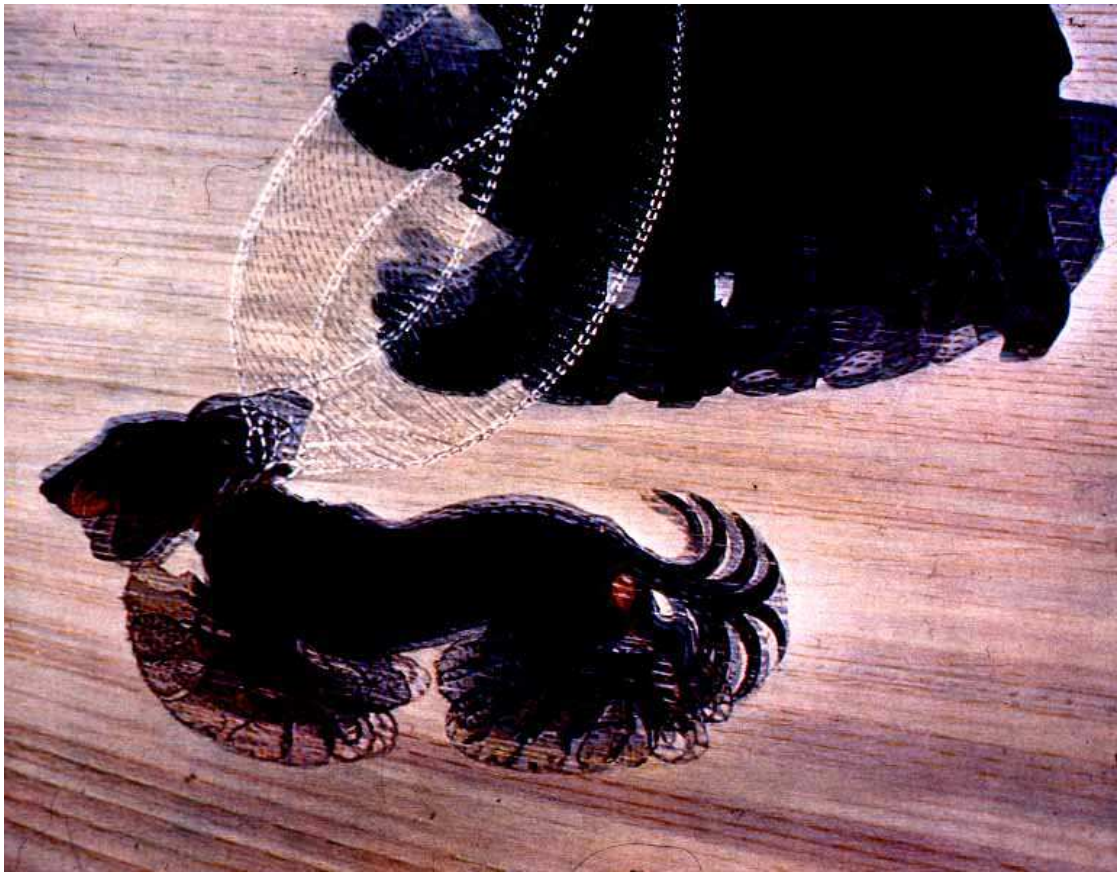
(a) (b) (c)

Fig. 5.9. *The feature information that the R-HOG detector cues on for cars. (a) The average gradient image over* 104×56 *normalised images from the Pascal 2006 VOC training set. The data set contains cars with front/rear, side and in-between views. (b) Each "pixel" shows the sum of all of the positive SVM weights in the block centred on the pixel. (c) Likewise for the negative SVM weights. For positive blocks, the car roof, wheels and front & rear bonnets weigh the most, whereas for negative ones the remaining regions weigh more.*

Chapter 4 we concluded that fine grained features help to improve performance, here we find that a multi-scale approach based on fine position and scale sampling gives best results. Ideally the test image should be scanned at all possible scale levels and all positions for each scale level, but this is impractical. A scale ratio of 1.05 and a window stride equal to the cell size or block stride seems to provide a good compromise. Values more than these significantly decrease the performance.

Another conclusion is that the amount of spatial smoothing when fusing detections in 3-D position and scale space should be conservative. Large amount of smoothing always decrease the performance. A value equal to the window stride provides good results. Interestingly, the spatial smoothing can be anisotropic; the best results corresponded to smoothing values proportional to the detection window aspect ratio. Experiments show that the distribution of detection scores at any scale level is elongated, with stretch of the distribution being proportional to the aspect ratio of detection window. We find that the same conclusion holds for other object classes

such as cars, motorbikes, horses. It is unclear why this is so. For cars we suspect that the reason is that only one detector was learned for all 360° views. The training images were normalised with respect to their height. Thus the variation along the width was much greater than the variation in height. The classifier adapted to these variations and thus is more robust to lateral shifts then to vertical shifts. Similar reasoning holds for motorbikes and bicycles, where the probability of having in-between views is high compared to exactly side or front/rear views. However it is unclear why the distribution for the person class is elongated. In Section 4.3.5 we found that vertical (2×1 cell) blocks outperform horizontal (1×2 cell) blocks. This implies the vertical encoding is more relevant and it may be that the classifier learned more variation for vertical scenarios. The fact that long vertical edges dominate for the human class may also be relevant.



Giacomo Balla, *Dynamism of a Dog on a Leash*, Oil on canvas, 1912. ©Albright-Knox Art Gallery, Buffalo, New York.

6

Oriented Histograms of Flow and Appearance for Detecting People in Videos

Motion information can be very characteristic, particularly for animal and human motions. Psychological experiments show that humans can easily recognise human actions from the motions of just a few dots placed at the joints of the actors[1]. Such experiments were popularised by Johansson [1973], who placed light emitters on their joints and filmed actors performing various activities like walking, jogging, dancing in the dark. Thus it is natural to ask: Can the detector performance be improved by including motion cues[2].

The goal of this chapter is to exploit motion cues to improve our person detector's performance for films and videos. Detecting people in video sequences introduces new problems. Besides the challenges already mentioned for static images such as variations in pose, appearance, clothing, illumination and background clutter, the detector has to handle the motion of the subject, the camera and the independent objects in the background. The main challenge is thus to find a set of features that characterise human motion well, while remaining resistant to camera and background motion.

This chapter introduces such a motion-based feature encoding scheme for the detection of standing and moving people in videos with possibly moving cameras and backgrounds. It presents features based on oriented histograms of various kinds of local differences or differentials of optical flow. We evaluate these features both independently and in combination with the static HOG appearance descriptors of Chapter 4. The new descriptors are designed to capture either motion boundaries or the relative motion of different limbs. The experiments show that they characterise human motion well while resisting background motions.

We start with a discussion of motion compensation in Sect. 6.1 and propose differentials of dense optical flow for this. Section 6.2 presents details of the proposed motion codings, which can broadly be divided in two categories: coding of motion boundaries and internal/relative dynamics. The section provides several variants differing mainly in how the spatial differentials are computed. A description of the different optical flow methods that we have evaluated is presented in Sect. 6.3. Our experiments show that using dense subpixel flow is essential for good performance, however noise is better than bias so heavily regularised flow is not required. Simple pixel-by-pixel multi-scale flow estimation work well. Section 6.4 discusses optimal parameter values for each scheme. Section 6.5 compares the proposed motion descriptors and studies the effect of representation choices on the performance. It shows that orientated his-

[1] There are other interesting observations as well. Human beings need only a fifth of a second to recognise the action [Johansson 1973]. If the actor is known to the viewer, the viewer is often able to recognise her or him [Cutting and Kozlowski 1977] or in the worst case determine his or her gender [Kozlowski and Cutting 1977, 1978, Cutting and Kozlowski 1978].

[2] Even in computer vision, Johansson's movies of dots in biological motion have been an inspiration, *e.g.* Song et al. [1999] proposed a probability based method for detecting and labelling human motions.

tograms of differential optical flow give the best overall detection performance and that the proposed motion descriptors reduces the false alarm rate by an order of magnitude in images with movement while maintaining the performance of the original static HOG method in stationary images. The complete motion HOG encoding algorithm is presented in Section 6.6. We conclude with a discussion in Sect. 6.7.

6.1 Formation of Motion Compensation

One way to approach motion encoding is to stabilise the backgrounds of the video sequences followed by a conventional motion-based detector such as Viola et al. [2003]. However, it is common to find sequences with non-rigid backgrounds or substantial camera translation and thus parallax in deep 3-D scenes. Another approach is to independently fit different parametric motion models to each new video sequence, *e.g*. homographies for purely rotating cameras or flat scenes, or fundamental matrices for translating and rotating cameras. However estimating such parametric models is sometimes challenging and it creates many different cases, each of which must be treated independently. There are also other issues such as singular cases while estimating parametric models and/or independently moving objects in the background. Given these challenges, the best compromise is to robustly estimate dense optical flow as it provides local motion estimates without (in principle) enforcing any prior motion model(s).

To motivate our approach to motion compensation, first note that the image flow induced by camera rotation (pan, tilt, roll) varies smoothly across the image irrespective of 3-D depth boundaries, and in most applications it is locally essentially translational because significant camera roll is rare. Thus, any kind of local differential or difference of flow cancels out most of the effects of camera rotation. The remaining signal is due to either depth-induced motion parallax between the camera, subject and background, or to independent motion in the scene. Differentials of parallax flows are concentrated essentially at 3-D depth boundaries, while those of independent motions are largest at motion boundaries. The 3-D depth and motion boundaries often coincide, *e.g*. for human subjects, both types of boundaries coincide with limb and body edges and flow differentials seem good cues for the outline of a person. However humans are also articulated objects with limbs that can move relative to one another. We expect internal dynamics such as relative limb motions to be quite discriminant for human movement. Differentials of flow taken within the subject's silhouette can capture these relative movements. Thus, flow-based features can focus either on coding motion (and hence depth) boundaries, or on coding internal dynamics and relative displacements of the limbs.

6.2 Motion HOG Descriptors

Motion HOG uses histogram of orientation based voting similar to that used in static HOG to provide a dense and overlapping feature set for video sequences. As in static HOG, the descriptors are normalised over local, overlapping blocks of spatial cells, and the resulting normalised histograms are concatenated to make the detection window descriptor vector used in the detector. However, instead of using image gradients, motion HOG uses differentials of optical flow – either flow orientation or oriented spatial gradients of flow components. The motion HOG architecture has already been presented in Sect. 3.2.2, here we describe the proposed feature extraction schemes.

Motion features take consecutive images of the video sequence as input. Although using more than two images should provide more information, it would imply a larger feature vector[3], which is disadvantageous in our SVM based limited-RAM learning methodology, *c.f.* Sect. 3.4. In the evaluation here, we limit ourselves to motion descriptors based on pairs of images.

Notation.

I^x, I^y denote images containing the x (horizontal) and y (vertical) components of optical flow, $I^{\mathbf{w}} = (I^x, I^y)$ denotes the 2D flow image ($\mathbf{w} = [x, y]^\top$), and $I^x_x, I^x_y, I^y_x, I^y_y$ denote the corresponding x- and y-derivative differential flow images. *E.g.*, $I^x_y = \frac{d}{dy} I^x$ is the y-derivative of the x component of optical flow.

6.2.1 Motion Boundary Based Coding

If the goal is to capture motion boundaries, it is natural to try to code the local orientations of motion edges by emulating the static-image HOG descriptors. The simplest approach is to treat the two flow components I^x, I^y as independent "images", take their local gradients separately, find the corresponding gradient magnitudes and orientations, and use these as weighted votes into local orientation histograms in the same way as for the standard gray scale HOG. We call this family of schemes *Motion Boundary Histograms (MBH)*. Figure 6.1 illustrates the kind of cues that MBH captures. Clearly the average gradients of both the x and the y flow fields, *c.f.* Figure 6.1(g,h), capture the subjects silhouette. Compared to the average gradient of the appearance channel, *c.f.* Fig. 4.10(a), the motion based gradients are more blurred. There may be several reasons for this. Firstly, errors in the flow estimates may cause imprecision in the motion boundary locations. Secondly, the motion boundaries highlight relative motions between different limbs and this may result in blurred boundaries. Evidence for this comes from the fact that the average gradient image for I^x is more blurred that that for I^y – presumably because human motion tends to be horizontal not vertical.

Variants of MBH differ in how orientation histograms are built. A separate histogram can be built for each flow component, or the two channels can be combined, *e.g.* by the winner-takes-all voting method that is used to handle colour channels in Sect. 4.3.2 or simply by summing the two histograms. We find that separate histograms are more discriminant. As in the static HOG, and irrespective of the optical flow method used, it is best to take spatial derivatives at the smallest possible scale ($[-1, 0, 1]$ mask) without any form of smoothing. Most of the parameter values turn out to be the same as for the static case. However, for descriptors computed using unregularised flow estimates the results are relatively sensitive to the epsilon parameter used during block normalisation. Relatively large values are usually preferable. For more details see Sect. 6.4.

6.2.2 Internal / Relative Dynamics Based Coding

Motion boundaries typically correspond with occlusion ones and as the static appearance descriptor already captures much of the available boundary information, the flow based descrip-

[3] We will see that the feature vector dimension for our motion descriptor computed over a pair of images is similar to that of the appearance descriptor. Thus the use of appearance and motion from two consecutive images approximately doubles the feature vector dimension, while the use of more than two consecutive images would increase it proportionally.

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

Fig. 6.1. *An illustration of the MBH descriptor on a pair of consecutive images. (a,b) The reference images at time t and $t+1$. (c,d) The computed optical flow, and flow magnitude showing motion boundaries. (e,f) The gradient magnitude (computed by applying centred $[-1,0,1]$ masks) of the flow field I^x, I^y for image pair (a,b). (g,h) The average gradient magnitudes of the flow fields I^x and I^y, respectively, over all training images*

tor should arguably focus more on capturing complementary information about internal or relative motions. This suggests that flow differences should be computed between pairs of nearby, but not necessarily neighbouring, points, and that angular voting should be based on the direction of the flow difference vector, not the direction of the spatial derivative displacement. So in opposition to MBH, we use (I^x_x, I^y_x) and (I^x_y, I^y_y) as the vectors for angular voting, and the simple x, y derivatives can be replaced by spatial differences taken at larger scales, perhaps in several different directions. We will call this family of schemes Internal Motion Histograms (IMH). Figure 6.2 illustrates the differences between the two schemes. IMH takes differences of flow vectors from different cell locations and computes relative flow, whereas MBH computes motion gradients of flow components and thus provides motion boundary orientations. Ideally, the IMH descriptors would be centred on human parts and directly capture the relative movements of different limbs, *e.g.* left *vs.* right leg or legs *vs.* head. However choosing the necessary spatial displacements for differencing would require reliable part detectors, so here we test simple variants based on fixed spatial displacements. The various IMH schemes are as follows:

IMHdiff

IMHdiff is the simplest IMH descriptor. It takes fine-scale derivatives, using (I^x_x, I^y_x) and (I^x_y, I^y_y) to create two relative-flow-direction based oriented histograms, *i.e.*, it captures differences in flow directions across a boundary, not the boundary direction. As with MBH, using separate orientation histograms for the x- and y-derivatives is better than combining them. Variants of IMHdiff use larger (but still central) spatial displacements for differencing – 5 pixels apart ($[1,0,0,0,-1]$ mask), or even 7 – and take spatial differencing steps along several different directions, *e.g.* including diagonal axes. The use of larger spatial displacements reduces the probability that the two flow vectors for which the difference is computed to belong the same image region or person part. This improves performance as it allows the descriptor to capture relative motions between different image regions.

IMHcd

IMHcd uses the blocks-of-cells structure of the HOG descriptors differently. It uses 3×3 blocks of cells, in each of the 8 outer cells computing flow differences for each pixel relative to the

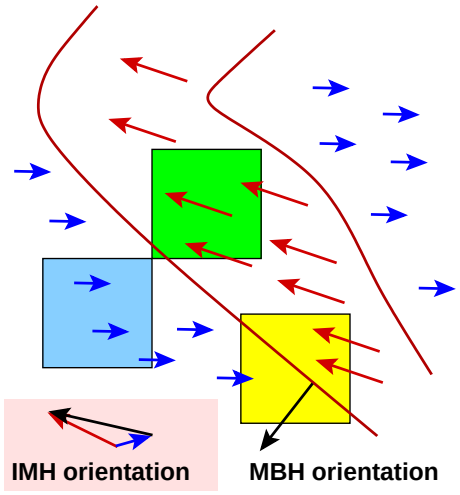


Fig. 6.2. *Differences between the IMH and MBH motion descriptors. The MBH scheme uses x-flow or y-flow gradients to capture motion boundaries (bottom right). Two adjacent cells at the top-left capture the essence of IMH scheme. IMH encodes the motion of a cell relative to its neighbours. This provides relative motion information that is not present in static HOG descriptor. When used alongside static HOG, IMH gives better performance compared to MBH scheme.*

corresponding pixel in the central cell and histogramming to give an orientation histogram[4]. Figure 6.3(a) illustrates the IMHcd encoding. The resulting 8 histograms are normalised as a block. The motivation is that if the person's limb width is approximately the same as the cell size, IMHcd can capture relative displacements of the limbs w.r.t. to the background and nearby limbs. This case is also depicted in Fig. 6.2. If the centred cell of the descriptor block is centred on a human part, the use of multiple overlapping descriptors increases the probability that at least one of the descriptors captures how this part has moved relative to neighbouring regions (including the background and the other human parts). The results in Sect. 6.5 support this hypothesis.

IMHmd

IMHmd is similar to IMHcd, but instead of using the corresponding pixel in the central cell as a reference flow, it uses the average of the corresponding pixels in all 9 cells. The resulting 9 histograms are normalised as a block. Because it averages corresponding pixels over a larger spatial block, IMHmd captures relative motion w.r.t. to general motion in nearby regions, and thus introduces some regularisation in the flow estimates.

IMHwd

IMHwd is also similar to IMHcd but, instead of using non-central differences, it computes the corresponding pixel-wise differences over various Haar wavelet like operators over the block. Figure 6.3(b) illustrates the operators that we have used.

[4] IMHcd uses non-central cell-width spatial differences that access only pixels within the block and evaluates them in 8 different step directions, whereas IMHdiff uses central differences and in the boundary cells it accesses pixels that lie outside the block.

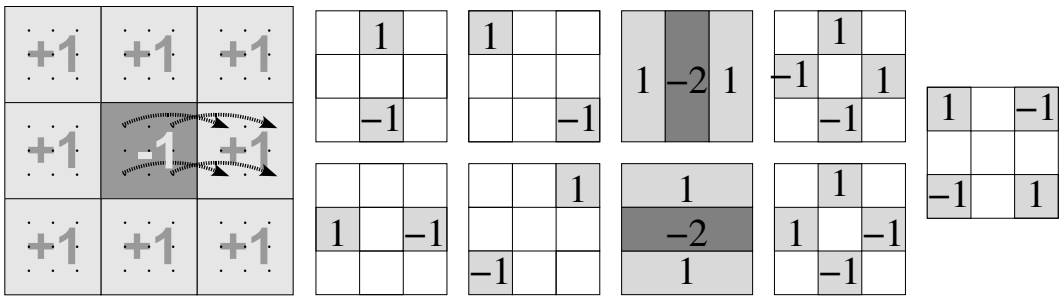


Fig. 6.3. *Different coding schemes for IMH descriptors. (a) One block of IMHcd coding scheme. The block is partitioned into cells. The dots in each cell represent the cell pixels. The arrows emerging from the central cell show the central pixel used to compute differences for the corresponding pixel in the neighbouring cell. Similar differences are computed for each of the 8 neighbouring cells. Values $+1$ and -1 represent the difference weights. (b) The wavelet operators used in the IMHwd motion coding scheme.*

6.2.3 Spatio-Temporal Difference

To compare our motion descriptors with other existing ones, we evaluated a scheme inspired by Viola et al. [2003] based on simple spatiotemporal differencing rather than flow. For each pixel, its 3×3 neighbourhood at the next time step is taken and its image intensity is subtracted from each of these 9 pixels. The absolute values are accumulated over each cell to make a 9 bin histogram for the cell, which then undergoes the usual block normalisation process.

6.3 Optical Flow Estimation

We tried several optical flow methods, ranging from a highly accurate, regularised subpixel-accurate dense algorithm to cell level motion estimation using an MPEG-4 based block coding algorithm. This section provides details.

6.3.1 Regularised Flow Method

The first set of experiments were done with the Otago implementation [Galvin et al. 1998] of the Proesmans et al. [1994] multi-scale nonlinear diffusion based algorithm. This gives high-quality, sub-pixel dense motion estimates, but is computationally expensive taking around 15 seconds per frame. It results in accurate but somewhat over regularised (for this application) flow orientation estimates, typically smoothing the motion boundaries that are critical for human detection. As an example, Fig. 6.4(a,b) shows pairs of input images and Fig. 6.4(c) shows the estimated flow fields. The estimates are globally accurate, correctly estimating the translation in the background, but the motion arising from the person's movement is also smoothed significantly relative to the optimal spatial derivative scale of 1–2 pixels.

6.3.2 Unregularised Multi-Scale Flow Method

The smooth motion estimates produced by the Otago method turn out to be over-regularised for our purpose, and they are also slow to compute. As an alternative we implemented a simple but

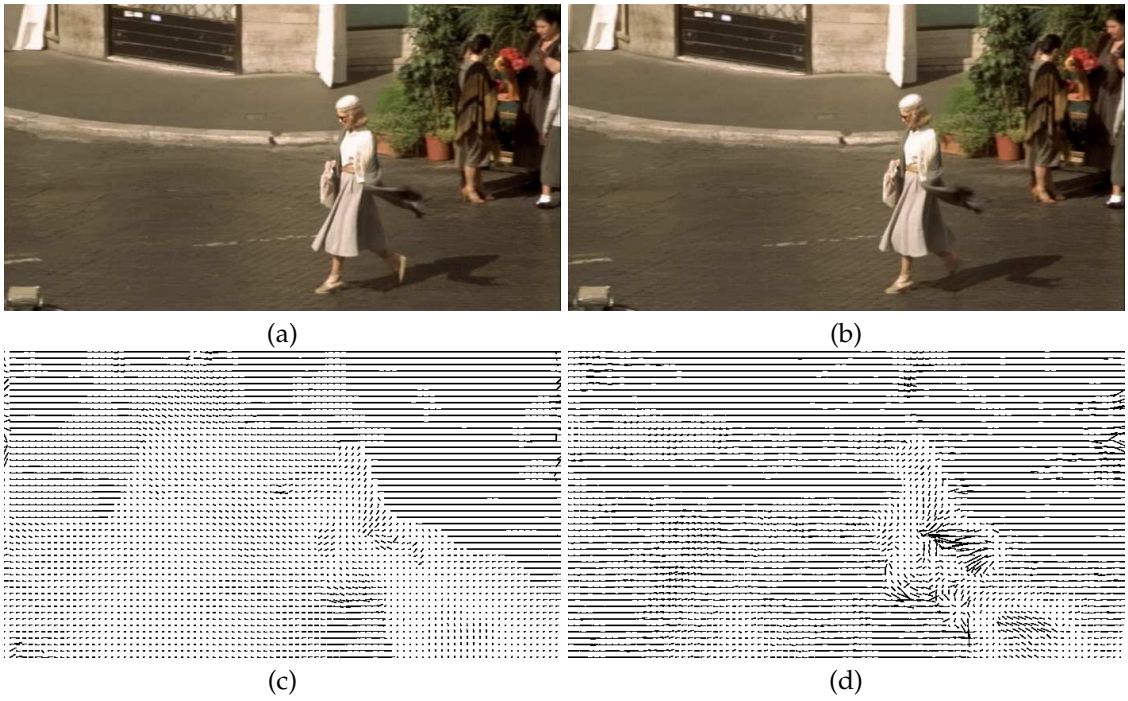


Fig. 6.4. *A pair of consecutive images and the estimated regularised and unregularised flow fields. (a,b) Consecutive images from a video sequence. The scene involves human motion and translation of the camera to the left. (c) Flow fields estimated using Proesmans' flow algorithm [Proesmans et al. 1994]. The method gives accurate but somewhat over-regularised flows that blur the sharp motion boundaries,* e.g. *on the moving woman. (d) Flow fields estimated using our unregularised method described in Sect. 6.3.2. The method estimates flows over* 5×5 *windows for each pixel and propagates the flow fields from coarse to fine scales, but does not perform any regularisation. The sharp motion boundaries on the moving woman are better preserved.*

fast flow method based on the constant brightness assumption [Horn and Schunck 1981]. Flow is found top-down in a multi-scale approach, with initial flow estimates made at a coarse scale propagated downwards and refined in fine scale steps. The infinitesimal motion $\mathbf{w}$ is estimated *independently* at each pixel by solving a damped Linear Least Squares equation

$$\mathbf{w} = -(\mathbf{A}^\top \mathbf{A} + \beta \mathbf{I})^{-1} \mathbf{A}^\top \mathbf{b} \tag{6.1}$$

based on the constant brightness equation $\partial I/\partial t + \mathbf{w}.\nabla I = 0$ over a small $N \times N$ neighbourhood, where $\mathbf{b}$ is an N^2 column vector encoding the temporal image differences $I_{\Delta t} = I_{t+1} - I_t$, $\mathbf{A}$ is an $N^2 \times 2$ matrix of spatial gradients $[I_x, I_y]$ of image I_t, and β is a damping factor introduced to control numerical problems in cases where $\mathbf{A}^\top \mathbf{A}$ becomes singular. The model does not include any explicit spatial regularisation or smoothing and its flow estimates are visibly less smooth than the Otago ones. However the multi-scale estimation framework allows it to estimate rapid movements such as leg motions rather well. Figure 6.4(d) shows a flow field estimated using the algorithm.

Figure 6.5 describes the complete flow algorithm. Our experiments show that flow estimates computed using $N = 5$, a scale refinement step of 1.3, and no smoothing (simple $[-1, 0, 1]$ mask) while computing gradients gives the best detection performance. Hence we set these values as

Input: A pair of consecutive images I_t and I_{t+1}
Output: The estimated dense optical flow fields

– Using a scale step of 1.3, compute the number of steps in a scale-space pyramid such that the image at the top of the pyramid is at least 16×16 pixels
– Starting from the top of the pyramid, for every scale
(a) Resize the images I_t and I_{t+1} to the current scale
(b) Propagate and resample the 2-D flow fields $I^{\mathbf{w}}$ from the previous scale to the current one; if the current scale corresponds to top of the pyramid, initialise the flow fields to zero
(c) Use the flow field to warp the image I_t. Denote the warped image by I'_t
(d) Compute I_x and I_y, the x and y gradients of I'_t, and the difference image $I_{\Delta t} = I_{t+1} - I'_t$
(e) Compute the squared gradient images I_x^2, $I_x I_y$, I_y^2, $I_x I_{\Delta t}$, and $I_y I_{\Delta t}$
(f) If the input images are in colour, for each pixel sum the three colour channels for all of the above gradient images
(g) Compute integral arrays for each of all the gradient images
(h) For each pixel,
(1) Compute

$$(\mathbf{A}^\top \mathbf{A} + \beta \mathbf{I})^{-1} = \begin{bmatrix} \sum_{i,j} I_y^2 + \beta & \sum_{i,j} -I_x I_y \\ \sum_{i,j} -I_x I_y & \sum_{i,j} I_x^2 + \beta \end{bmatrix} \tag{6.2}$$

and

$$\mathbf{A}^\top \mathbf{b} = \begin{bmatrix} \sum_{i,j} I_x I_{\Delta t} & \sum_{i,j} I_y I_{\Delta t} \end{bmatrix}^\top \tag{6.3}$$

using the integral array representation, where $\sum_{i,j}$ denotes the sum over $N \times N$ window centred on the pixel
(2) Estimate the flow vector $\mathbf{w}'$ using (6.1)–(6.3)
(3) Update the flow vector $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{w}'$
(i) If at the finest level, return the estimated flow fields $I^{\mathbf{w}}$

Fig. 6.5. *Our multi-scale dense optical flow computation method. The steps are optimised for rapid flow computation.*

default. To speed up the implementation, we use integral arrays [Viola and Jones 2001] over the gradient images I_x^2, $I_x I_y$, and I_y^2 (used to compute $\mathbf{A}^\top \mathbf{A}$ see eq. (6.2)), and $I_x I_{\Delta t}$, $I_y I_{\Delta t}$ (used to compute $\mathbf{A}^\top \mathbf{b}$, see eq. (6.3)). This allows the matrices $\mathbf{A}^\top \mathbf{A}$ and the vectors $\mathbf{A}^\top \mathbf{b}$ to be rapidly computed with only four additions per element for any window size $N \times N$. The complete multi-scale method is much faster than the Otago one, running in 1 second on DVD resolution 720×405 images[5].

6.3.3 MPEG-4 Block Matching

We also tested motion descriptors based on a MPEG-4 block matcher taken from the `www.xvid.org` codec. No attempt was made to enforce motion continuity between blocks. Even though the matching estimates were visually good, the detection results were not competitive. We think that there are several reasons for this. Firstly, block matching[6] provides only one vote

[5] Usually feature films on movie DVDs are captured at an aspect ratio of 16:9 during production, but stored at a resolution of 720×576 pixels with an aspect ratio of 5:4. Typical movie players resample (zoom in or out) the images to the correct aspect ratio of 16:9 before displaying them. We resample all images to 720×405 pixels.
[6] The MPEG-4 blocks correspond to histogram cells in our notation.

for each cell, whereas with dense optical flow each pixel provides a separate vote into the histogram. Secondly, the block matching flow estimates do not have deep sub-pixel accuracy (usually algorithms in MPEG-4 block matching provide motion estimates up to quarter-pixel accuracy). Experiments on rounding the flow values from the Otago code showed that even $1/10^{th}$ of a pixel of rounding causes the performance to drop significantly. The need for accurate orientation voting is one reason for this. Thirdly, the 8×8 MPEG blocks are too large for optimal results.

6.4 Descriptor Parameters

Before comparing different motion HOG descriptors and how they compare alongside static HOG descriptor, this section presents intra descriptor conclusions and best parameter values for each motion HOG descriptor. This best set of parameters is also used as default parameter values for each descriptor in the experiments presented in Sect. 6.5. For the combined flow and appearance detectors with the optimal cell size of 8×8 pixels, memory constraints limit us to a total of about 81 histogram bins per block. (Increasing the histogram size beyond this is possible, but it reduces the number of hard negatives that can be fitted into memory during re-training to such an extent that performance suffers). In the experiments below, we test: MBH with 6 gradient orientations, 2 separate flow components for I^x and I^y, and 4× block overlap; IMHdiff with 2 displacements (horizontal and vertical $[1, 0, -1]$ masks), 6 flow orientations and 4× block overlap; and IMHcd, IMHwd and IMHmd with eight 8-pixel displacements and 6 flow orientations.

All of the methods use orientation histograms with votes weighted by vector modulus followed by a block-level normalisation – essentially the same scheme as the static HOG descriptor. We tested various different bin sizes, normalisation schemes, *etc*. with similar conclusions to as in Sect. 4.3. For both MBH and IMHdiff, fine (6 bin) orientation coding with 2×2 blocks of 8×8 pixel cells seem to be best. 3×3 blocks of cells (9× block overlap) perform better for the flow-only MBH classifier, but for the combined detectors the performance of this combination drops owing to the increased feature size. Changing the cell size from 8×8 to 6×6 only reduces the performance slightly. Good normalisation of the blocks is critical and for the flow descriptors Lowe's hysteresis-based L2 normalisation seems to do significantly better than L2 or L1-sqrt normalisation. We tried larger displacement masks (3- and 5- pixel displacement) for MBH but found that the performance drops. For the IMHcd/wd/md schemes, 6 and 9 orientation bins give the same performance (we use 6 below), and Lowe's hysteresis based L2 normalisation still works best.

We also evaluated variants that use the least squares image prediction error of the estimated flow as a flow quality metric, down-weighting the histogram vote in proportion to $\exp(-|e|/\sigma)$, where e is the fitting error over the local 5×5 window. This very slightly ($\lesssim 1\%$) improves the performance provided that σ is not set too small. We also tested various motion descriptors that do not use orientation voting (*e.g*. based simply on the modulus of velocity), but the results were significantly worse.

6.5 Experiments and Performance Comparison

For evaluations in this section we use both INRIA static and moving person data set. The first series of experiments uses only the INRIA moving data set. This data set contains a training

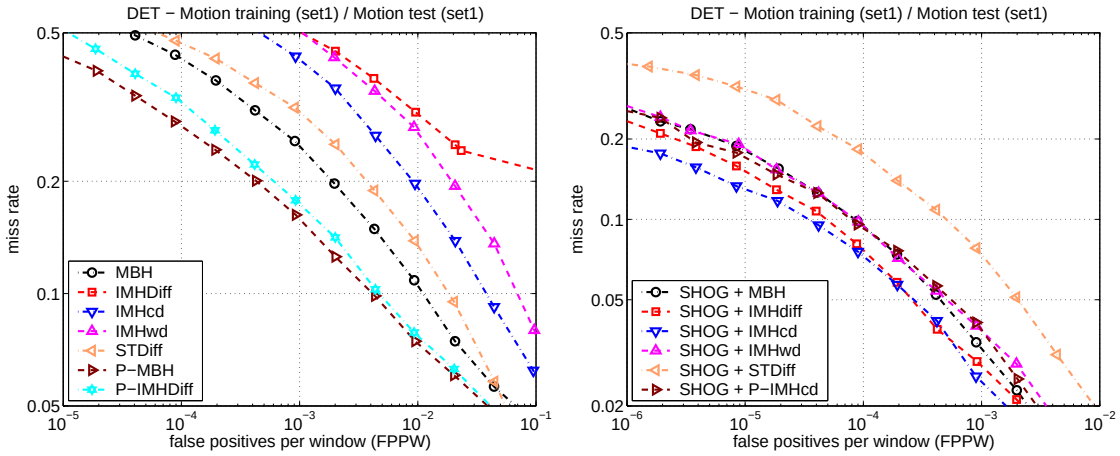


Fig. 6.6. *A comparison of the different motion descriptors, trained on* Motion Training Set 1 *and tested on* Motion Test Set 1*, using (a) the motion feature set alone; and (b) the motion feature set combined with the R-HOG appearance descriptor. The prefix 'P' in the MBH and IMH legends denotes the same methods using Proesmans' flow estimates.*

data set "*Motion Training Set 1*", and two test sets "*Motion Test Set 1* & *2*". *Test Set 1* is similar to *Training Set 1* and is relatively simpler, *Test Set 2* is much harder. For detailed description of the data set see Appendix A. However, in real scenarios, we may be interested in using the same detector for both static images or video sequences. Thus, the second series of experiments form the overall benchmark, where we evaluate both our static and moving person data sets and different combinations of them. We use INRIA static person train and test data sets for these evaluations. In this section, we refer to this set as '*Static Training/Test Sets*'. Even though the *Static Training Set* has no (zero) flow, we find that including it along with *Motion Training Set 1* significantly improves the performance of both the static and the combined detectors (see results below). More precisely, the detector performance on *Motion Test Set 1* improves, without changing that of the static detector on the *Static Test Set*. This is perhaps because the *Motion Test Set 1* images contain many poses that do not appear in the *Static* sets – notably running and other rapid actions. Also as in Sect. 4.3 here we present results using DET curves.

6.5.1 Results on Motion Data Set

We begin by comparing the results of the motion descriptors introduced above, trained and tested on *Motion Training Set 1* and *Test Set 1*. This provides a benchmark on how relevant each motion channel is when used alone. Figure 6.6(a) gives results for detectors learned with the motion descriptors. The oriented histogram of differential flow schemes MBH and IMHdiff with the Proesmans flow method dominate results. In fact for the video test sets (which do contain many frames without much visible movement) these motion features alone are within an order of magnitude of the static HOG detector and significantly better than the static Haar wavelet detector.

When motion and appearance features are combined, neither the Proesmans flow method nor the MBH descriptors perform so well and it is IMHcd and IMHmd computed using our flow method that are the leaders. Below we use R-HOG + IMHmd as the default combined detector, although R-HOG + IMHcd would lead to similar conclusions. Our experiments show

that using the new flow method the combined detector reduces false positives by a factor of more than 4 at 8% miss rate when compared to the same descriptor set using Proesmans flow method. In fact, any regularization aimed at improving the flow smoothness appears to reduce the detector performance.

Figure 6.6 shows that motion-only results are not a good guide to the performance of the combined detector. The reduced spread of the results in the combined case suggests that there is a considerable degree of redundancy between the appearance and motion channels. In particular, IMHdiff and MBH are the schemes with the smallest spatial strides and thus the greatest potential for redundancy with the human boundary cues used by the appearance based descriptors – factors that may explain their reduced performance after combination. Similarly, the strong regularisation of the Proesmans' flow estimates may make them effective cues for motion (and hence occlusion) boundaries, while the unregularised nature of ours means that they capture motion of thin limbs more accurately and hence provide information that is more complementary to the appearance descriptors.

Another point to note is the normalisation method. In the static HOG case we concluded that both L2-Hys and L1-Hys give similar performance for human detection, with L1-Sqrt giving better results for most other object classes, *c.f.* Sect. 4.6. Here for motion HOG we find that L2-Hys sufficiently outperforms L1-Sqrt. The main reason is that our motion encoding scheme uses flow magnitudes, which can vary greatly between shots. Hence large flow vectors should be damped to prevent them from dominating the feature vector. The hysteresis clipping of L2-Hys appears to be effective at achieving this. For unregularised flow the parameter ϵ used to avoid division by zero during normalisation also needs to be increased to provide additional robustness to noise in the flow estimates. $\epsilon = 0.4 \times$ #PixelsInBlock gives good results in our experiments. This renders descriptors relatively robust to errors in flow magnitude while finely capturing the orientations. However, for descriptors using regularised flow method, the results are insensitive to ϵ's value over a large range. Typically smaller values give good results.

6.5.2 Results on Combined Static and Motion Data Set

Figure 6.7 demonstrates the overall performance of a selection of our detectors on several different test sets. Unless otherwise noted, the detectors are trained on the combined *Motion Training Set 1* and *Static Training Set*. The static (appearance based) detectors shown are: R-HOG – our default R-HOG trained on combined data set; R-HOG (static) – R-HOG trained on the *Static Test Set* alone, essentially same as in Chapter 4; and Wavelet – our version of the static Haar wavelet based detector of Papageorgiou and Poggio [2000] described in Sect. 4.2.1. Two combined detectors are also shown: R-HOG + IMHmd – R-HOG combined with the IMHmd flow feature (8-pixel steps in 8-neighbour directions); and R-HOG + ST Diff – R-HOG combined with spatiotemporal differences of Sect. 6.2.3.

Again the good performance of the R-HOG + IMHmd combination is apparent. The absolute results on *Motion Test Set 2* are an order of magnitude worse than on *Motion Test Set 1* owing to the more challenging nature of the images, but the relative rankings of the different methods are remarkably stable. Overall, on video data for which motion estimates are available, the false alarm rates of the best combined detectors are an order of magnitude lower than those for the best static-appearance-based ones.

Given that we want methods that can detect people reliably whether or not they are moving, we were concerned that the choice of method might be sensitive to the relative proportion of moving and of static people in the videos. To check this, we tested the detectors not only on the pure video *Test Sets 1* and *2*, but also on the combination of these with the *Static Test Set* (again

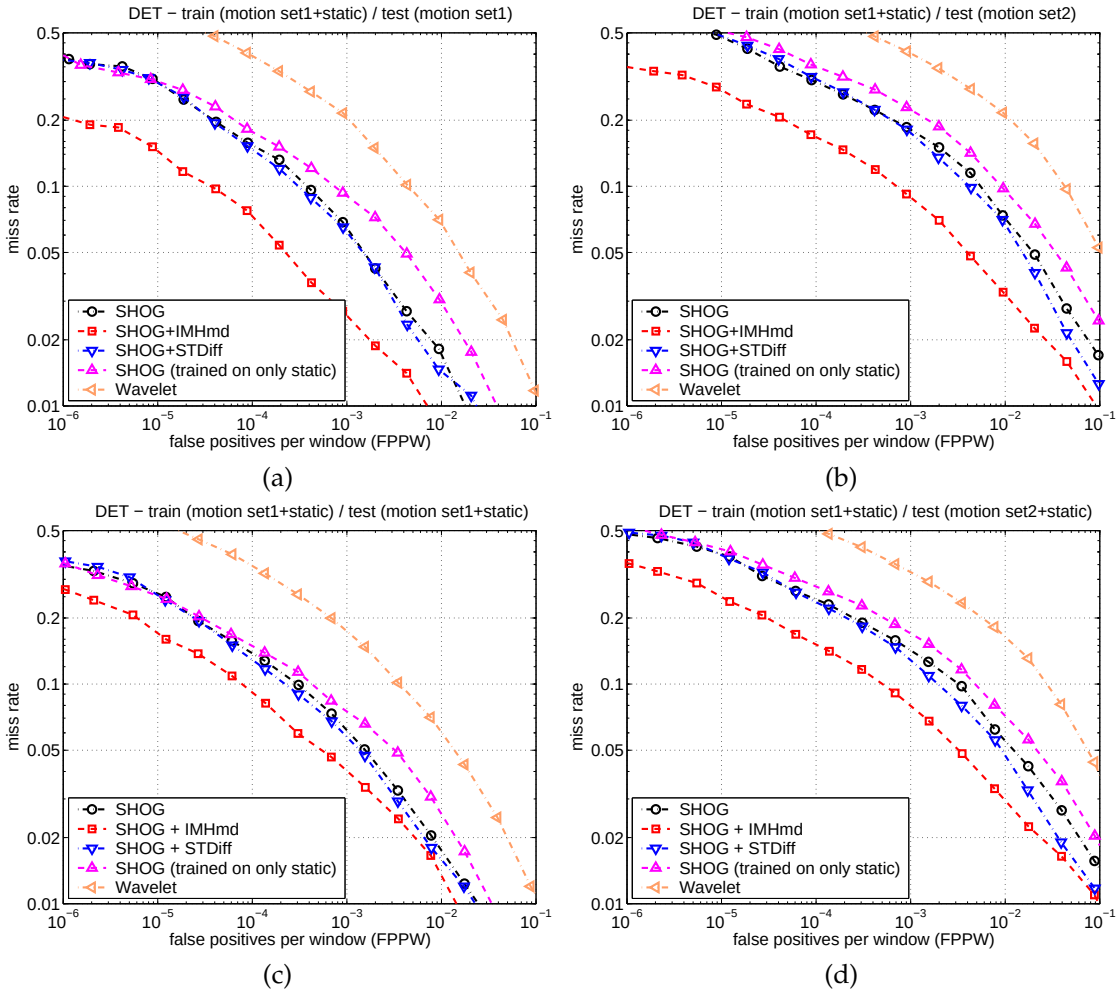


Fig. 6.7. *An overview of the performance of our various motion detectors. All detectors are trained on* Motion Training Set 1 *combined with the* Static Test Set *with flow set to zero. They are tested respectively on: (a) the* Motion Test Set 1*; (b) the* Motion Test Set 2*; (c) the* Motion Test Set 1 *plus the* Static Test Set*; (d) the* Motion Test Set 2 *plus the* Static Test Set.

with static image flow set to zero). The results are shown in Fig. 6.7(c–d). Diluting the fraction of moving examples naturally reduces the advantage of the combined methods relative to the static ones, but the relative ranking of the methods remains unchanged. Somewhat surprisingly, Table 6.1 shows that when used on entirely static images for which there is no flow, the best combined detectors do marginally *better* the best static one. The images here are from the *Static Test Set*, with the detectors trained on *Motion Training Set 1* plus the *Static Training Set* as before.

Figure 6.8 shows some sample detections after non-maximum suppression of the combined detector (R-HOG + IMHmd trained on *Motion Training Set 1 + Static*) on images from *Motion Test Set 2*. *Set 2* contains challenging images taken from different films from the training images. Here there are shots of people in Indian costume, some dance sequences, and people in crowds that are different from anything seen in the training images.

Fig. 6.8. *Sample detections on* Motion Test Set 2 *from the combined R-HOG + IMHmd detector trained on* Set 1 + Static. *Note the variations in pose, appearance, background and lightning. Certain poses, such as people pushing the bus (3th row, 2nd column), fish eye lens views (3th row, 3th column), and views of people from the top (6th row, 2nd column) were not seen in the training set. The detector sometimes has false positives on legs (1st row, 2nd column) and on tyres with relatively oblique views.*

FPPW	10^{-3}	10^{-4}	10^{-5}
R-HOG	6.2%	11.4%	19.8%
R-HOG + IMHmd	5.8%	11.0%	19.8%
R-HOG + ST Diff	5.7%	10.5%	19.7%

Table 6.1. *The miss rates of various detectors trained on* Set 1 + Static *images and tested on purely Static images. Despite the complete lack of flow information, the combined detectors provide slightly better performance than the static one.*

6.5.3 Mixture of Experts

The combined-feature detectors above are *monolithic* – they concatenate the motion and appearance features into a single large feature vector and train a combined classifier on it. We have also tested an alternative *Mixture of Experts* architecture. In this, separate detectors are learned from the appearance features and from the motion features, and a second stage classifier is then trained to combine the (real valued scalar) outputs of these to produce the final detector. In our case the second stage classifier is a linear SVM over a 2D feature space (the appearance score and the motion score), so the final system remains linear in the input features. This approach keeps the feature space dimensions relatively low during training, thus allowing more hard negatives to be included at each stage. (Indeed, for the 2D second stage classifier there can be millions of them). In our experiments these effects mitigate the losses due to separate training and the linear Mixture of Experts classifier actually performs slightly better than the best monolithic detector. For now the differences are marginal (less than 1%), but the Mixture of Experts architecture provides more flexibility and may ultimately be preferable. The component classifiers could also be combined in a more sophisticated way, for example using a rejection cascade Baker and Nayar [1996], Viola and Jones [2001], Sun et al. [2004] to improve the run time.

6.6 Motion HOG Encoding Algorithm

Figure 6.9 presents the complete encoding algorithm for the IMHcd or IMHmd descriptor. The algorithm is similar to the static HOG one, but here we compute flow differences instead of image gradients. We use a hard coded value of 3×3 cells in each block. Other values could also be used, *e.g.* blocks of 2×2 cells for IMHmd or odd numbered 5×5 cells for IMHmd/IMHcd.

6.7 Conclusions

This chapter has developed a family of high-performance detectors for fully visible humans in videos with possibly moving subjects, cameras and backgrounds. The detectors combine gradient based appearance descriptors with differential optical flow based motion descriptors in a linear SVM framework. Both the motion and the appearance channels use oriented histogram voting to achieve a robust descriptor. We studied various different motion coding schemes but found that although there are considerable performance differences between them when motion features alone are used, the differences are greatly reduced when they are used in combination with static appearance descriptors. The best combined schemes used motion descriptors

Input: Two consecutive images of a video sequence and a list of image windows to evaluate
Output: Encoded feature vectors for each supplied window location

Common initial steps:
(a) Optional: Gamma normalise each colour channel of the input image
(b) Compute optical flow using the algorithm in Fig. 6.5

Descriptor computation: For each user supplied image window locations and scales
(a) Rescale the estimated flow fields to the current scale level
(b) Divide the image window into a uniformly sampled dense grids of cells
(c) For each block of 3×3 cells
  (1) For each pixel in the centre cell
    (i) If using IMHcd, compute the flow difference relative to the corresponding pixel in each of the 8 outer cells
    (ii) If using IMHmd, compute the mean of the flow for the corresponding pixels in all 9 cells and use this averaged flow as the reference flow to take flow differences for all 9 cells (as in IMHcd step above)
    (iii) Compute the magnitudes and orientation of the resulting differential flow vector
  (2) Create a spatial and orientation histogram for the block
  (3) For each pixel in the block, use trilinear interpolation to vote into the histogram based on flow magnitude
  (4) Apply *L2-Hys* normalisation independently to each block

Common final steps:
(a) Collect motion HOGs over all of the blocks of the image window into one big vector

Fig. 6.9. *The complete IMHcd, IMHmd encoding algorithm.*

based on oriented histogramming of differences of unregularised multi-scale flow relative to corresponding pixels in adjacent cells (IMHcd) or to local averages of these (IMHmd). As in the static HOG case, we find that over smoothing decreases the performance, again highlighting the importance of fine grained features. However unlike static HOG where fine orientation binning was found to be essential, the motion encoding is relatively insensitive to fine orientation binning. Bins of size less than 20° (9 orientation bins in the histogram) do not help performance or may even damage it. 30°–40° wide bins give good performance for all motion HOGs. We believe that two reasons for this are the accuracy of the flow orientation estimates and the large variations in kinds of motions in the sequence. The overall conclusion that the detector performance is sensitive to the details of feature computation and that robust encoding is essential for good performance is confirmed.

Pablo Picasso, *Les Demoiselles d'Avignon*, Oil on canvas, 1907. ©New York, Museum of Modern Art.

7

Part Based Person Detection

Until now this thesis has focused on "*monolithic*" object detectors that compute a single large feature vector for each normalised object window, learning linear classifiers on these descriptors over a set of training images. However, it is possible that one could build a stronger detector by first detecting individual object parts and then learning to combine co-occurrences of these, as this strategy can explicitly encode geometrical relations between parts and is potentially also more robust to partial occlusions. This chapter extends our monolithic person detection framework to a parts based ones that detect and combine individual body parts to obtain the final detections.

We start in Sect. 7.1 by detailing part detectors for head and shoulders, torso, and legs, describing the key parameters for each of these and evaluating them w.r.t. the monolithic person detector. Treating these part detectors as component detectors, Sect. 7.2 presents our approach to fusing multiple detectors. Using linear SVM as the overall classifier we evaluate different approaches to fusion and show that although the monolithic person detector gives state of the art performance, it can still be improved slightly by using a parts based approach. The best performance is achieved by combining fine-tuned part detectors using a sparse spatial histogram based integration scheme in which for each part only the 3–4 locations with the highest confidence values vote.

7.1 Human Body Part Detectors

We trained three human part detectors – head and shoulders, torso, and legs detectors – in the same way as the monolithic static image person detector described in Chapter 4. The size of the normalised windows for the head and shoulders, torso, and leg detector was respectively 32×32, 48×56, and 48×64 pixels. Each detector was trained and tested on the INRIA static person data set (*c.f.* Sect. A.1), with the normalised windows centred on additional head, torso and leg annotations. Details of how the data set was annotated and how these normalised windows compare to the full person window are presented in Sect. A.3. Here we evaluate the performance of the individual part detectors and compare it to that of the full person detector.

For each part detector we performed two series of experiments. The first used R-HOG feature vectors with the same parameters as the default full person detector in Sect. 4.3. Figure 7.1(a) shows the results. The second series optimised the most sensitive R-HOG parameters – the cell size η, the number of cells ς and the block stride leaving the rest of the parameters the same as in the default person detector. The optimal parameters for the torso detector were found to be same as for the full person detector (*i.e.*, cell size of 8×8 with 2×2 cells in each

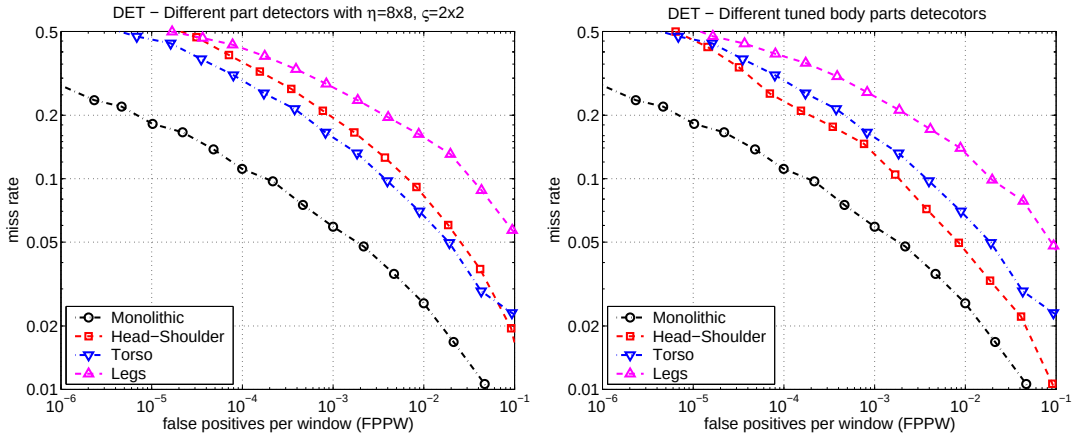


Fig. 7.1. *Performance comparison of head and shoulders, torso, and legs detectors. None of the part detectors perform as well as the full person detector. (a) Results when all of the part detectors use the same R-HOG descriptor parameters as the full person detector. (b) Results when each part detector is tuned to give best performance.*

block and block stride of 8×8 pixels). The head and shoulders detector gave the best results for $\eta = 4 \times 4$, $\varsigma = 2 \times 2$ with a R-HOG block stride of 4 pixels (hence 4-fold coverage of each cell) and cells and blocks half the size of those of the default person detector. The Leg detector gave the best results at $\eta = 4 \times 4$, $\varsigma = 3 \times 3$ and a block stride of 4 pixels with 9-fold coverage for each cell. Figure 7.1(b) compares the performance of these tuned part detectors to that of the full person detector. The head and shoulders detector has improved significantly and now gives the best performance among the three part detectors, but it still generates an order of magnitude more false positives than the full person detector at 10% miss-rate. The torso detector is the second best with 2–3 times more false positives than the head detector. The legs detector is significantly worse giving approximately another order of magnitude more false positives.

The relatively high false positives of the part detectors indicate that it would be difficult to use them as stand-alone object detectors, e.g. to find people whose bodies are not fully visible in the image. For this reason we will not pursue this option further here, but it is an important area for future work. The cues that each part detector is keying on are illustrated in Fig. A.4(b-d), which show the average normalised and centred gradient images over all examples for head and shoulders, torso, and legs, respectively. Note that the head and shoulders detector is not a face detector, its training set contains images of heads and shoulders from all view-points including many side and back views, so its performance can not be compared with that of the standard face detectors. It essentially cues on the round contours of the head, with support from the shoulder silhouettes below it. Figure A.4(c-d) show, respectively, that the torso detector contains the shoulders, the torso and the upper part of the legs, and the legs detector contains the legs and part of the lower torso. The overlap of the different parts detectors is intentional as we found in Sect. 4.3.6 that a small amount of context (*i.e.* a window slightly bigger than the strict object size) helps to improve detector performance. Moreover, for each detector the cells adjacent to the borders have only 2-fold coverage in the feature vector, compared to the remaining (inside) cells which have 4-fold coverage. Thus, although these regions are present in the normalised window, they do not dominate the feature vector. The leg detector gives the worst results mainly because legs vary a lot between images, both in terms of articulation

(such as front and side profile of people walking or standing) and also in terms of their relative position within the normalised windows. They thus prove to be difficult parts to cue on.

7.2 Fusing Multiple Part Detectors

We tried four different approaches to fusing the different part detectors. All used the above part detectors and operate within the window scanning framework that we have used throughout the thesis, densely scanning a fixed resolution window for occurrence of specific parts across the image at multiple scales, testing each location for the presence of the specific parts. However the approaches differ in how the results of the dense scans are fused. One uses only the value of the highest score for each part detector, another uses the actual feature vector corresponding to the highest score, still another combines both the scores and the spatial locations of the detectors, and the last performs Generalised Hough Transform voting and computes the probability of the centroid of the person given the location and the occurrence probability of the different parts. We now describe each approach in detail.

7.2.1 Adaptive Combination of Classifiers (ACC)

This is the simplest architecture for fusing different part detectors. The approach is similar to the ACC presented in Mohan et al. [2001] except that (a) we use linear SVMs instead of kernel SVMs as classifiers, (b) we map the linear SVM scores to probability estimates using the method described in Platt [2000], and (c) our approach had only three parts instead of head and shoulders, left arm, right arm, and legs. ACC creates a feature vector of dimension equal to the number of parts. For each part detector, it performs a dense scan of the window with a stride of 4 pixels and a rectangular scan range from $(16, 16)$ to $(48, 32)$ in (x, y) for the head and shoulders, from $(24, 48)$ to $(40, 56)$ for the torso and from $(24, 80)$ to $(40, 96)$ for the legs. (All positions are relative to the 64×128 window and measured w.r.t. the centre of the respective part window). Figure 7.2 shows these scan ranges overlayed on the average gradient person image. The combined feature vector includes a single score (the highest) for each detector. It is used as input

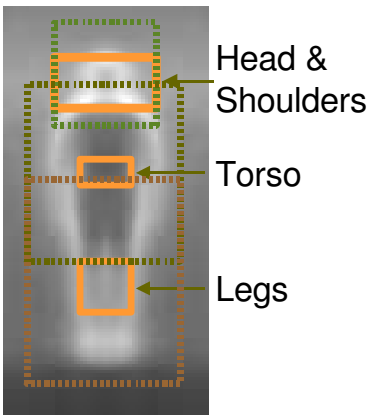


Fig. 7.2. *Scan range for the head and shoulders, the torso and the leg detectors, represented by* solid lines. *The range for all parts is measured w.r.t. the centre of the respective part window.* Dotted lines *in the figure show the extent of each part detector.*

to the second and final stage linear SVM classifier. The rest of the training and test process is similar to that for the full person detector.

7.2.2 Feature Vector Based Adaptive Combination of Classifiers (FVACC)

This approach is similar to ACC presented above, except that it uses the complete feature vector corresponding to the highest score instead of the score. Thus the dimension of the final feature vector is to the sum of the dimensions of the feature vectors for all parts.

7.2.3 Spatial Histograms of Classifiers (SHC)

Section 7.2.5 shows that parts based schemes using ACC or FVACC decrease the performance for our HOG feature vectors when compared to our monolithic detector. This is in contrast to the conclusion in Mohan et al. [2001] that simple combinations of classifiers such as ACC are sufficient to improve the detection performance significantly. As an alternative we developed a histogram based approach that creates a 2-D spatial histogram for each part detector, the extent of the histogram being equal to the scan range of the part detector. There is thus one histogram block in the final combined descriptor vector for each part. The bin bandwidth of the histograms is equal to 4×4 and the scan was performed using the same parameters as in Sect. 7.2.1, *i.e.* a stride of 4 pixels. The approach is similar to the HOG block computation except instead of voting with gradient magnitudes in 3-D spatial-orientation histograms and later normalising, here we vote with detection scores (after transforming them to probabilities using Platt [2000] method) into 2-D spatial histograms and do not normalise them. The scheme implicitly encodes the spatial locations of the parts. The final feature vector is computed by appending all of the spatial histogram entries into one big feature vector. This is the *dense* voting variant, called Dense SHC or DSHC. Another variant performs sparse voting: for each part detector the scores are first sorted in descending order and only the top 3–5 of them are used to vote into their respective locations in the spatial histograms. The rest of the detection scores are ignored, giving an effect similar to non-maximum suppression in feature detection. We call this sparse voting variant SSHC. Section 7.2.5 shows that this is the only scheme that improved the performance compared to the full monolithic person detector, the rest all gave worse or at best similar results. Figure 7.3 illustrates the feature vector computation chain.

7.2.4 Voting in Star Network (VSN)

Recently probabilistic voting in star network based schemes has also been used [Leibe et al. 2004, Leibe and Schiele 2004, Opelt et al. 2006]. In these approaches, during training the classifier estimates the probability distribution of the object centroid given the location of the part. The detection phase then attempts to maximise the probability of the centroid given the detected part location and score. This is achieved by a Generalised Hough Transform [Ballard 1981] where each detector casts a vote for the possible locations of the object centre. Usually a non-maximum suppression stage is used (*e.g.* our mean shift based one) where the detector signals a detection only if the mode value is greater than a threshold. We investigated a similar approach within our window-scan framework with the difference that instead of performing explicit non-maximum suppression, we treated the computed 2-D Hough voting map as a feature vector for our SVM classifier. The rest of the procedure was the same as in the above approaches.

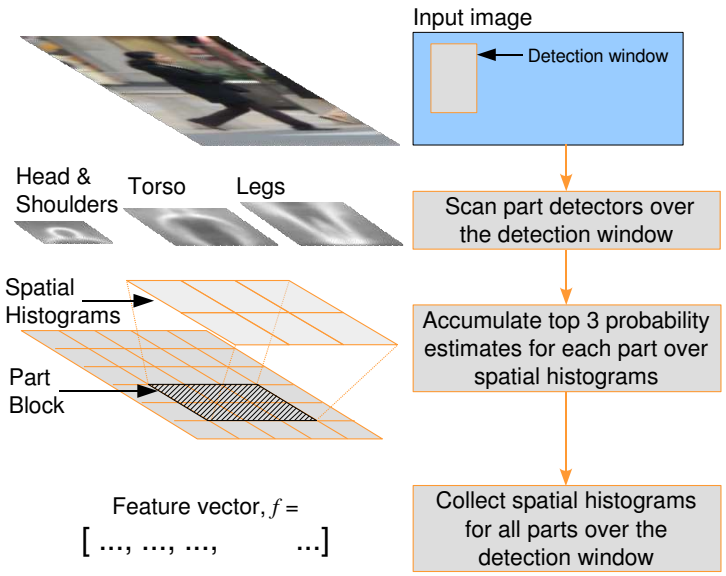


Fig. 7.3. *An overview of feature extraction for spatial histogram of classifiers approach. For each part detector a scan over the detection window is performed and only 3-4 locations with highest probability estimates vote into spatial histograms. In the last step these spatial histograms are collected in one big feature vector.*

7.2.5 Results

Figure 7.4 compares the various approaches for fusing the part detectors. To measure the gain in performance when combining the part detectors, we again performed two series of experiments. The first used part detectors trained using the R-HOG parameters of the default full

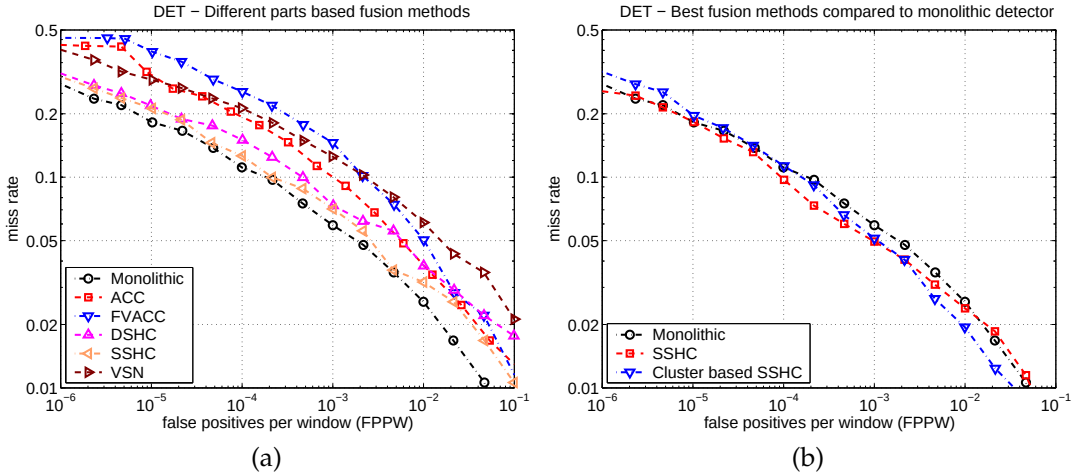


Fig. 7.4. *A comparison of the various part detector fusion methods. (a) Results obtained with part detectors trained using the same parameters as in the default person detector. (b) Results obtained when fine tuned part detectors are used in the SSHC approach. For this particular scenario the detector outperforms the monolithic person detector. See the text for a description of "Cluster based SSHC".*

person detector. These experiments aimed at seeing how much improvement the parts based approach could yield if the same feature vectors were used. Figure 7.4(a) shows the results. None of the fusion based approaches managed to surpass the performance of the monolithic detector. This is perhaps surprising given that other works have observed that by combining different part detectors with geometric information, the performance of the overall detector is improved [Mohan et al. 2001]. Only sparse SHC using the highest 3 scores for each part detector (legend SSHC in the figure) came close to the monolithic detector. One possible explanation for these results is that HOG descriptors already captures the relative geometries of the different parts reasonably well. The proposed part based fusion approaches are manually designed schemes that may tend to lose information during the fusion stage. This seems to be a feature set issue not a classifier one: following Mohan et al. [2001], we also tried kernel SVM for the ACC method but find that it does not produce any gain the performance.

The second series of experiments used the fine tuned parts detectors with the SSHC method (the best fusion approach in Fig. 7.4(a)). The results are shown in Fig. 7.4(b). This time SSHC outperforms the monolithic part detector. The figure also shows the results of a cluster based SSHC method. This is a simple attempt to incorporate an exampler-like structure into the part detectors to allow for greater variability without going to full nonlinear classifiers. For each part detector it computes 10 clusters over positive training examples in the respective HOG descriptor space and measures the posterior assignment probability (normalised exponential of the negative Euclidean distance measure) of each scanned location w.r.t. each cluster centroid. For each location, the similarity results are weighted by the detection confidence score and stored in a 10 dimensional "*similarity*" feature vector. The remaining steps are similar to SSHC and we used the similarity vector for the 3 locations with the highest confidence score to vote into the 2-D spatial histogram. We thus have 2-D histograms where each bin is a 10 dimensional vector. The dimension of the feature vector for each part detector is equal to the number of bins in the 2-D spatial histogram times the number of clusters. Figure 7.4(b) shows that performance of the detector marginally drops compared to the simple SSHC method.

8

Conclusions and Perspectives

This thesis has described a complete framework for the problem of detecting objects in images and videos. The proposed approach builds upon ideas in image processing, computer vision and machine learning to provide a general, easy to use and fast method for object detection. Our main contribution is the development of robust image feature sets for object detection tasks, and we have shown that our proposed features give state-of-the-art performance even when used in conjunction with a simple linear SVM classifier. Besides their direct use in object detection, the proposed features have become quite popular since their publication in 2005. They have been used for several other computer vision tasks, ranging from tracking and activity recognition [Lu and Little 2006], context based object detection [Hoiem et al. 2006] to secure multi-party communication protocols [Avidan and Butman 2006].

8.1 Key Contributions

- **Histogram of Oriented Gradient (HOG) feature vectors.** Our first direct contribution is the proposed feature set based on well normalised grids of gradient orientation histograms. These features provide some invariance to shifts in object location and changes in shape, and good resistance to changes in illumination and shadowing, background clutter and camera viewpoint. The overall conclusions are that capturing fine detail with unsmoothed gradients and fine orientation voting, moderately coarse spatial binning, strong normalised and overlapping blocks are all needed for good. The descriptors do not involve any arbitrary thresholding of edges and they are relatively fast to compute.
- **Fusion of overlapping detections.** When exhaustively scanning images for object instances at all scales and locations heuristic methods are usually used to manage overall detections. The overall detector performance can be significantly improved if the fusion algorithm takes into account the number of detections and the confidence score of each detection. We contributed a general algorithm for fusing multiple overlapping detections and concluded that fine sampling of scales and positions is critical for good performance. The proposed algorithm is general and fast, and our experiments show that it gives good performance for other detectors such as the face detection system of Viola and Jones [2001].
- **Oriented histograms of differential motion.** We also generalised our proposed oriented histogram based approach to several motion descriptors based on differential flow fields.

The new schemes allow human motion to be characterised robustly based on noisy optical flow estimates. They adopt a non-parametric approach to estimating relative motion, capturing either motion boundaries or relative motions of nearby image regions. An interesting conclusion is that a simple multi-scale flow method based on brightness constancy without inter-pixel smoothing is sufficient. Regularised optical flow estimates are not needed.

8.2 Limitations of the Approach

The proposed framework has a number of limitations. A few of these are intrinsic shortcomings of the approach, while others relate to extensions that are worth investigating in the near future, and still others are open issues. This section presents the intrinsic limitations and the next section provides a discussion of the future work and open issues.

- Although the proposed approach gives good results for many object classes, it is still basically a 2-D template matching approach that codes object geometry using image position. It requires recurring shape events in the given blocks of the training images to learn a classifier. This requires (a) a geometrically well aligned data set, (b) a sufficiently large number of training images, and (c) relatively structured and rigid object classes. Upright people are such a class, as Fig. 4.10(a) shows. In contrast, consider the "cats" class in the PASCAL 2006 Visual Object Challenge [Everingham et al. 2006a]. Figure 8.1 shows some images. These clearly highlight the huge range of within-class shape variation for cat images. Our approach is unable to provide a good detector for this class. Another problem with this class is the relative lack of shape information. In contrast local feature approaches can cue on the eyes and the fur texture, are insensitive to the lack of shape information, and can therefore provide better performance on this class.

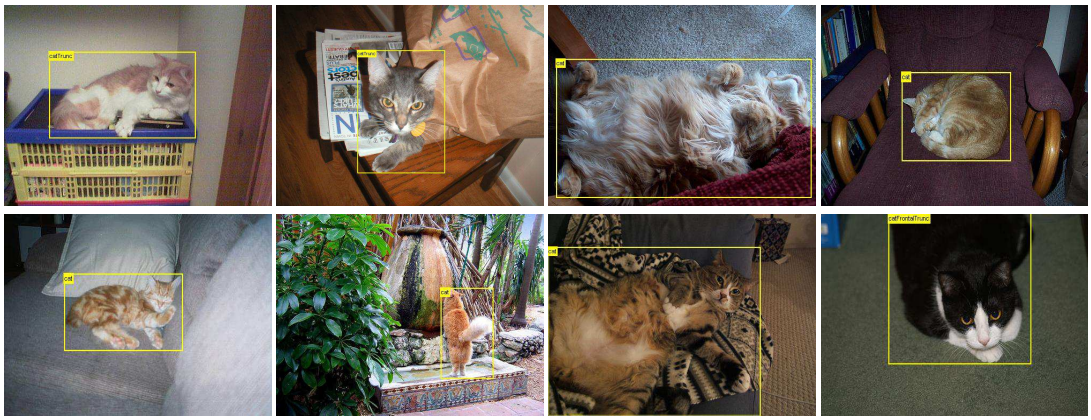


Fig. 8.1. *Some examples of cat images from the PASCAL 2006 Visual Object Challenge. Our template based approach is unable to handle the large amount of within-class shape variation in these images. The images also highlight issues related to the creation of normalised data sets for training: how to best annotate these images and how to normalise them geometrically.*

- The most computationally intensive part of our framework is the estimation of the HOG descriptors. Although the descriptor processing chain is relatively simple and when used

with an efficient detection algorithm such as AdaBoost it allows near real-time implementation [Zhu et al. 2006], the run time is still significantly higher than a similar framework using integral arrays Viola and Jones [2001] to compute wavelet like feature vectors.

- Another disadvantage of our system is the high-dimensionality of its feature vectors. Of course, this results in good discrimination and is thus critical for the overall performance of the detector. However, during learning (with the current batch algorithm) the feature vectors for all training images must be stored in memory. The large size of the vector limits the number of examples (in particular hard negatives) that can be used and thus ultimately limits the detector performance. Hence there is a trade off based on the available training memory, and very large feature vectors tend to be suboptimal.
- The descriptors have a relatively large number of parameters that need to be optimised. Although Chapter 4 pointed out that most of the parameters are remarkably stable across different object classes, some optimisation does need to be performed for each object class and the parameter space is too large to allow every possible combination to be tested.

8.3 Future Work

This section provides some pointers to future research and discusses some open issues in visual object detection.

Detection of moving persons in videos.

As motion HOGs are slower to compute and have different characteristics than static HOGs, an object detector for video sequences could use a rejection chain algorithm to build cascades of detectors. As the appearance channel is typically somewhat more informative than the motion one, the chain would probably learn to reject most image windows using appearance only, using motion HOG descriptors only in cases where motion information is present to further reject false positives. This approach might provide good overall performance while significantly speeding up the run time compared to the current method based on a single monolithic SVM classifier. Also, the current motion HOG descriptors use only two consecutive images of a video sequence, whereas good recognition results in humans require temporal information to be integrated over somewhat longer time periods (at least over 3–4 frames[1]). Thus another future direction could be to use more frames to compute motion HOG descriptors and to study the impact on the detector performance.

The approach to capturing relative dynamics of different body parts that is used in our Internal Motion Histograms is not ideal. It would be interesting to use the part detectors built in Chapter 7 to first detect the various body parts and then try to explicitly encode their relative motions. This raises two issues. Firstly, it seems intuitively that for this approach to perform well, finer grained part detectors such as upper and lower leg detectors and arm detectors may be needed. But given the current state of the art, the reliable detection of small body parts is very challenging. Secondly, our experiments in Chapter 7 show that it is best to incorporate part detector votes from more than one location (typically using the 3–5 locations with the highest confidence values suffices for good results). This implies that if computing relative motions,

[1] [Johansson 1973] experiments show that humans need around a fifth of a second – approximately 3-5 video frames – to recognise the actions of the actor.

one might have to evaluate many possible pairs of combinations. Of these, only those that are exactly on the real body parts would be relevant and the rest would need to be filtered out. It is currently unclear how to achieve this.

Another application in which motion HOGs may prove useful is activity recognition. Here the tasks involve the classification of characteristic movements in videos, and motion HOGs may prove to be useful features owing to their robust motion encoding.

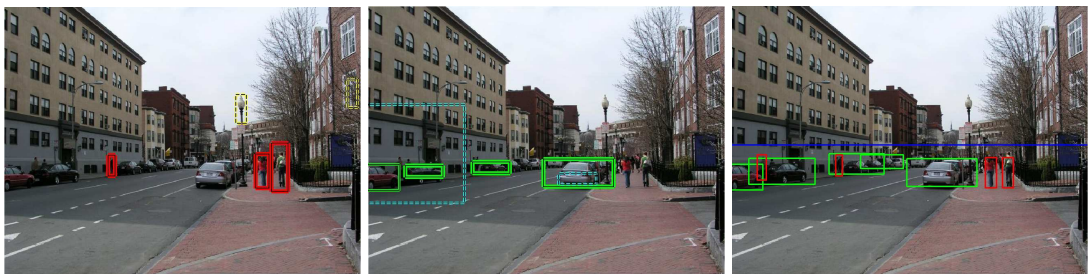
Texture and colour invariant features.

It is also worth investigating texture and colour invariant descriptors or feature spaces. In conjunction with the HOG representation of the visual form, such feature vectors would form a more complete representation that should allow the current approach to be extended to many more object classes. The overall system could use AdaBoost to learn the most relevant features for each object class and to perform all of the stages of optimisations at once. This might also allow us to avoid the extensive parameter tuning of the descriptors. The training time for such a system would be long, but careful implementation should be able to maintain good run time when the system is in use.

Fusion of Bottom-Up and Top-Down Approaches.

It would be interesting to explore the fusion of bottom-up and top-down approaches along the lines of Leibe et al. [2005]. However rather than going from sparse points during the bottom-up stage to dense pixel-wise representations during the top-down stage as in Leibe et al., one could use dense HOG like features to perform bottom-up object or part detections, and then verify these in a sparse top-down approach, such as one that fits potential part detections to a structural model for the object class.

Another challenging issue while relating to top-down information is the exploitation of the general context. Recently several researchers have begun to use context by modelling the relationships between different objects or object classes, surrounding image regions, or scene categories [Kumar and Hebert 2003, Murphy et al. 2003, Sudderth et al. 2005, Kumar and Hebert 2005, Hoiem et al. 2006]. In particular Hoiem et al. [2006] show that by using the interplay of



(a) (b) (c)

Fig. 8.2. *Scene context can help improve the performance of bottom-up low-level object detectors. (a) Results of bottom-up static HOG person detector after multi-scale dense scan. (b) Results of static HOG car detector after similar scan. (c) Results obtained using Hoiem et al. [2006] approach which takes into consideration the interplay between rough 3-D scene geometry, approximate camera position/orientation and low-level person and car detectors.* Images courtesy of Derek Hoiem.

different objects, scene orientation and camera viewpoint, the performance of existing bottom-up object detectors such as those presented in this thesis can be further improved. Figure 8.2 illustrates this. However, current approaches uses manually designed encodings for the contextual information. In the future it would be interesting to expand this to a broader range of background cues and to add higher-level intelligent reasoning to support contextual inferences.

Appendices

A

Data Sets

Any new feature set must be carefully validated on appropriate data sets to show its potential for real-world applications. The data sets should be chosen to be representative for the applications under consideration. It is also crucial that they should not contain selection biases that will perturb the results. This thesis presents feature sets for both static images and video sequences. Our primary goal is person detection so we proposed two challenging data sets reflecting this application: a static person data set and a moving person data set. This appendix describes these and the other data sets used for our evaluation, and also explains how annotations were performed.

A.1 Static Person Data Set

We used two different person data sets for evaluating our static image encoding schemes. The first one is the well-established MIT pedestrian data set [Papageorgiou et al. 1998, Papageorgiou and Poggio 2000, Mohan et al. 2001] and the second one is the new INRIA static person data set introduced by us.

A.1.1 MIT Pedestrian Data set

This data set contains 709 pedestrian images taken in city streets. Each image is 64×128 pixels and contains either a front or a back view of a centred, standing person. The range of poses is relatively limited, and the people are normalised to have approximately the same size in each image. Figure A.1 shows some sample images. We separated the MIT data set into 509 training

Fig. A.1. *Some sample images from the MIT pedestrian data set. The subjects are always upright and standing. The data set contains only front or back views of pedestrians in city scenes.*

Fig. A.2. *Some normalised image windows from the new INRIA static person detection data set introduced by us. Note the variations in pose. Most of images contain people standing or walking. Some images have people running, going downhill, bicycling, or playing.*

and 200 test images (plus their left-right reflections). The MIT data set does not contain negative images, so we used the negative images from the new INRIA static data set.

A.1.2 INRIA Static Person Data Set

Section 4.4 showed that our detectors give essentially perfect results on the MIT data set. Due to its range of poses and restricted viewing conditions, the MIT set is not suitable for detailed evaluation and comparison of state of the art methods, so we produced a new and significantly more challenging 'INRIA static' person data set. We collected 498 images taken with a personal digital camera over a period of 8 months in a wide range of places and under different weather conditions, and added 497 images from the Graz-01 data set [Opelt et al. 2004, Opelt and Pinz 2004]. The people in the images are usually standing, but may appear in any pose or orientation and against a wide variety of backgrounds including crowds. Many are bystanders taken from the image backgrounds, so there is no particular bias on their poses. Figure 1.1 shows some images from the data set.

As images of people are highly variable, to learn an effective classifier, the positive training examples need to be properly normalised and centred to minimise the variance among them. For this we manually annotated all upright people in the original images. Sect. A.4 gives details of how the annotations are performed. Figure A.2 shows some samples of image windows. The data set is split into a training and a testing set. The positive training set contains 1208 image windows and the test set contains 566. The positive training and test windows are reflected left-right, effectively doubling the number of positive examples.

The negative training set contains 1218 images and the negative test set 453 images. The negative images were collected from the same set of images as the positives, except that they do not contain any people. They include indoors, outdoor, city, mountain, and beach scenes. Some images also focus on cars, bicycles, motorbikes, furniture, utensils, etc. The details of how the negative windows were sampled from these images were presented in Sect. 3.4. The database is publicly available for research purposes from `http://lear.inrialpes.fr/data`.

Fig. A.3. *Some sample images from INRIA moving database, which contains moving people with significant variation in appearance, pose, clothing, background, illumination, coupled with moving cameras and backgrounds. Each pair shows two consecutive frames.*

A.2 INRIA Moving Person Data Set

Chapter 6 proposed appearance and motion based human detectors for video sequences. To train and validate our detectors, we created the INRIA moving person data set. Shots were selected from various movie DVDs and personal digital camera video sequences. We tried to ensure that the selected shots formed a representative sample of such data, with the only constraint being that the images contain upright persons. The data set contains people in all sorts of motion, including standing, walking, running, dancing and performing every day activities, from a range of different viewpoints. Figure 1.3 shows some image frames.

We annotated the humans in all shots. Our main training set, "*Motion Training Set 1*", was obtained from 5 different DVDs and 16 personal video sequences of 1–2 minutes duration. It contains a total of 182 shots with 2781 human examples (5562 examples after left-right reflections). We created two test sets. "*Motion Test Set 1*" contains 50 shots and 1704 human examples from unseen shots of the same DVDs used in *Motion Training Set 1*. This provides a somewhat easier test set whose content is similar to the training set. To ensure that there is no particular bias in our results, we also created the more challenging "*Motion Test Set 2*" containing 2700 human examples from 128 shots from 6 new DVDs. This set is considerably harder than *Test Set 1*. It contains shots of people in Indian costume, some dance sequences, and people in crowds that are different from anything seen in the training images. Approximately 80% of the shots in *Test Set 1* and 68% in *Test Set 2* contain either a moving camera, significant background motion or both.

A.3 Person Part Data Set

We also added part annotations constituting of head, torso and legs to the new INRIA static database. This data set was used in Chapter 7. Normalised head and shoulders, torso, and legs consisted, respectively, of 32×32, 48×56, 48×64 pixel windows. Figure A.4(a) shows an example 64×128 pixel person image window and the average gradient image over all such windows, and Figs. A.4(b)-(d) show, respectively, an example and the average gradient for each of

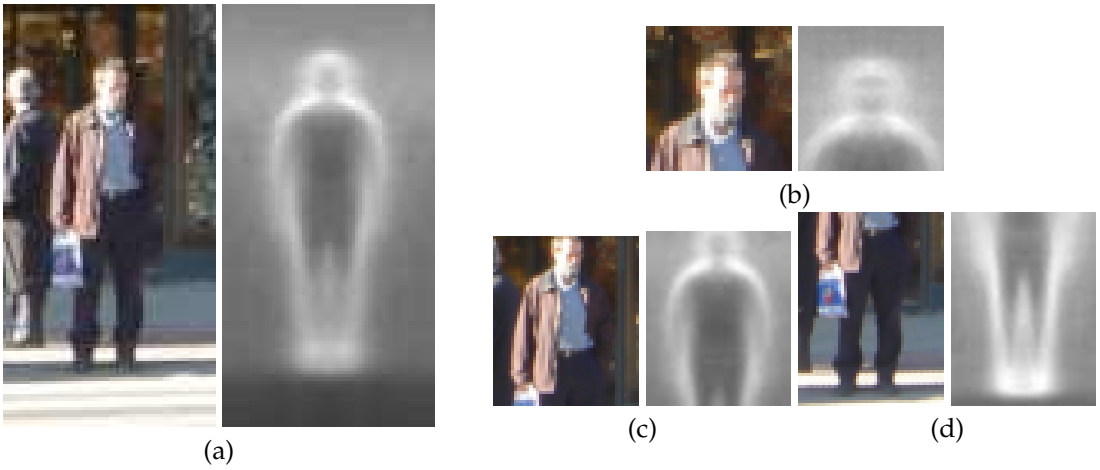


Fig. A.4. *Example images from the head and shoulders, torso and legs data sets and the corresponding average gradient images over all examples. (a)* 64×128 *pixel full person image. (b,c,d) Different normalised and centred head and shoulders, torso and legs windows, respectively.*

the corresponding normalised head and shoulders, torso and leg windows. The average gradient image for the individual parts is significantly sharper than the corresponding parts in Fig. A.4(a). This is to be expected, as the part annotations were independently centred on the associated body part.

A.4 Annotation Methodology

For generality all annotations were made on the original images. Each annotation consisted of a centre point and a bounding box surrounding the object. Typically the bounding boxes contain the object, in this case the persons head, torso and legs; except for cases where arms or legs extend sideways the bounding boxes were centred on only the torso and contained only this body part. Figure A.5 shows an image and three annotations marked on it. The image regions

Fig. A.5. *Example images from the INRIA static person data set with corresponding original annotations marked on it. The marked bounding boxes are illustrated with darkened regions. Note the annotations contain significant variations in people pose.*

belonging to the annotations were cropped and rescaled to 64×128 pixel image windows. On average the subjects height is 96 pixels in these normalised windows to allow for an approximately 16 pixel margin on each side. In practise we leave a further 16 pixel margin around each side of the image window to ensure that flow and gradients can be computed without boundary effects. The margins were added by appropriately expanding the annotations on each side before cropping the image regions.

B

Evaluation Methodology

For the purposes of comparison, we need to quantify the different detectors' performance. In our framework to object detection, there are two approaches to the evaluation: window level detectors use Detection Error Tradeoff (DET) or Receiver Operating Characteristics (ROC) curves to quantify the binary classifier performance, and overall object detectors use Recall-Precision (RP) curves to measure the accuracy of the object detection and localisation. In both cases the detectors output a *confidence score* for each detection, where larger values indicate higher confidence that the object is present at the tested location. For all detectors, both evaluation methods start from the lowest possible score, evaluate the respective parameters such as number of false positives, recall rate or precision rate, and then progressively increase the threshold till they reach the highest possible score. Each tested threshold provides a point on the curve.

B.1 Detection Error Tradeoff (DET) Curves

Detection Error Tradeoff (DET) curves form natural criterion for our binary classification tasks as they measure the proportion of true detections against the proportion of false positives. They plot miss rate versus false positives (here False Positives Per Window tested or FPPW) on a log-log scale. Miss rate is defined as

$$\text{MissRate} = 1 - \text{Recall} = \frac{\#\text{FalseNegatives}}{\#\text{TruePositives} + \#\text{FalseNegatives}} \tag{B.1}$$

We plot FPPW along x-axis and miss rate along y-axis. Lower values denote better classifier performance. DET plots are used extensively in speech and in NIST evaluations. They present the same information as Receiver Operating Characteristic (ROC) curves but allow small probabilities to be distinguished more easily. We often use a false positive rate of 10^{-4} FPPW as a reference point for results. This may seem arbitrary but it is no more so than, *e.g.* Area Under ROC. Small FPPW are necessary for detector to be useful in practice owing to the number of windows tested, typically of the order of 1000–10,000 windows. In a multi-scale detector (before non-maximum suppression) 10^{-4} corresponds to a raw error rate of about 0.8 false positives per 640×480 image tested. At these FPPW the DET curves are usually very shallow so even very small improvements in miss rate are equivalent to large gains in FPPW at constant miss rate. For example, for our default static image person detector (Sect. 4.4) at 10^{-4} FPPW, every 1% absolute (9% relative) reduction in miss rate is equivalent to reducing the FPPW at constant miss rate by a factor of 1.57 – a 5% absolute reduction in miss rate is equivalent to a 10 times reduction in false positives.

B.2 Recall-Precision (RP) Curves

Recall-Precision curves measure how a detector performs in practice, where it has to detect and localise all instances of an object class in an image. For each value of the detection confidence, it computes recall and precision as follows

$$\text{Recall} = \frac{\#\text{TruePositiveDetections}}{\#\text{TotalPositives}}; \qquad \text{Precision} = \frac{\#\text{TruePositives}}{\#\text{TruePositives} + \#\text{FalsePositives}} \tag{B.2}$$

We opt not to use RP curves for comparison of window classifiers as the definition of precision in RP curves uses the number of false positives, which can be significantly high for a window level classifier as it often tests millions of negative examples compared against few thousand positive examples. The other disadvantage is that localisation results can be defined in multiple ways, *e.g.* how much overlapping between the ground-truth and a prediction should be defined as correct detection, and if overlapping predictions are made should all predictions be considered correct or only one prediction as true positive and rest as false positives. These issues bring additional parameters in the evaluation. However, for the purpose of comparison of object localisation performance, RP curves provide a standard scale for evaluations which is independent of the different approaches used, for example if the localisation was performed using a window scanning approach or by other means. Hence we use RP curves to provide final comparisons of the overall detectors and choose DET curves when comparing the performance of different feature extraction algorithms.

We use the definition of Everingham et al. [2006b]. Each prediction is given by its centre location $\mathbf{c} = [x, y]$ and scale s. As the aspect ratio of the object window is fixed, the scale parameter s determines the area of the object window. A detection is considered true if the area of overlap a_o between the predicted region b_p and the ground truth region b_{gt}

$$a_o = \frac{\text{area}\left(b_p \bigcap b_{gt}\right)}{\text{area}\left(b_p \bigcup b_{gt}\right)}$$

exceeds 50%. At most one prediction per ground truth object is considered as true and any remaining predictions are considered to be false. Thus multiple overlapping detections are penalised.

Often we use the integrated Average Precision (AP) to summarise the overall performance, except that instead of the area under the Recall-Precision curve (which gives the mean precision over a particular recall interval), we use the interpolated precision. For a recall interval $r \in [r_s, r_e]$, this is defined as the maximum precision for any recall rate in the interval $[r_s, r_e]$

$$p_i(r) = \max_{\{\tilde{r}:\tilde{r} > r \in [r_s, r_e]\}} p(\tilde{r})$$

where $p(\tilde{r})$ is the precision at recall $\tilde{r}$. This is same as the Average Precision measure from the Text Retrieval Conference (TREC) evaluation, which measures the interpolated precision at a set of 11 equally spaced recall levels over the full range $[0, 1]$, except that instead of 11 bins we use a large number of smaller bins. The interpolated precision reduces the effect of the characteristic "sawtooth" pattern of false detections that is typical of RP curves, and the use of area under the RP curve penalises methods that have missing values at high recall (*i.e.* that are unable to detect some objects altogether).

C

Other Approaches to Person Detection

During the course of this thesis we tried a number of other feature sets and detection frameworks, but their performance did not compare with that of the current HOG descriptors. This appendix gives a very brief summary of these approaches and discusses our motivations in trying them, their potential advantages and why we did not purse the work in these directions.

C.1 Key Point Based Person Detector

During the initial months of this thesis, we tried to develop key point based approaches to human detection but we were disappointed by the poor repeatability of the features for this task. We evaluated Harris [Harris and Stephens 1988], LoG [Lindeberg 1998], Harris-Laplace [Mikolajczyk and Schmid 2004] feature sets. Our experiments showed that even for two close by frames in a video sequence, these detectors did not fire consistently on the same scene elements. They were unable to cope with the sheer variation in human clothing, appearance and articulation. Figure C.1 illustrates this. A super-pixel [Mori et al. 2004] based approach was also evaluated but resulted in similar conclusions. This led us to try silhouette or shape fragment based approaches as an alternative.

(a) (b)

Fig. C.1. *Inconsistent behaviour of key point detectors on humans. (a) Detected key points by the Harris-Laplace detector [Mikolajczyk and Schmid 2004] for one frame of a video sequence. (b) Detected key points for the next frame of the sequence. Note that although the detector fires reliably for the same points in the background, it fails to perform consistently on the similar locations for the person.*

C.2 Contour Fragment Based Person Detector

Given the considerable variability introduced by occlusions, pose variations, clothing, shape and texture, etc, it seems appropriate to build detectors based on characteristic local fragments of body contours, as only these are stable under all of these above changes.

Our initial approach to this began by performing edge detection on the image and approximating the detected edges by straight line segments for efficiency. Any set of nearby (not necessarily neighbouring) segments can form a "*contour fragment*" part. The framework tests and selects the most characteristic contour fragments to act as crude part candidate detectors. They are created by randomly sampling nearby segments from the set of all segments in the positive training images. To match a reference contour fragment to an image, we perform exhaustive search by aligning any line segment from the reference fragment to each segment in the image. Segments are matched using a projection cost metric that computed the amount of overlap after projection of all of the image segments within some ϵ-tube region to the reference segment. Multiple votes onto a pixel in the reference segment are counted as one. The overall cost of matching a contour fragment is the ratio of sum of overlap for all segments in the reference fragment to the total length of all segments in the same fragment. The projection cost metric uses hard matching constraints: it considers only lines within an ϵ-tube region with $\epsilon = 3$ pixels. The matched segment can optionally be weighted by the amount of rotation the reference fragment has undergone to be classified as a match.

Given this matching cost, a set of highly relevant contour fragments is selected using the Information Bottleneck (IB) principle of Tishby et al. [1999]. This tries to minimise the Mutual Information (MI) between the training images and the set of fragments (compressing the information required to express the images) while at same time maximising the MI between the

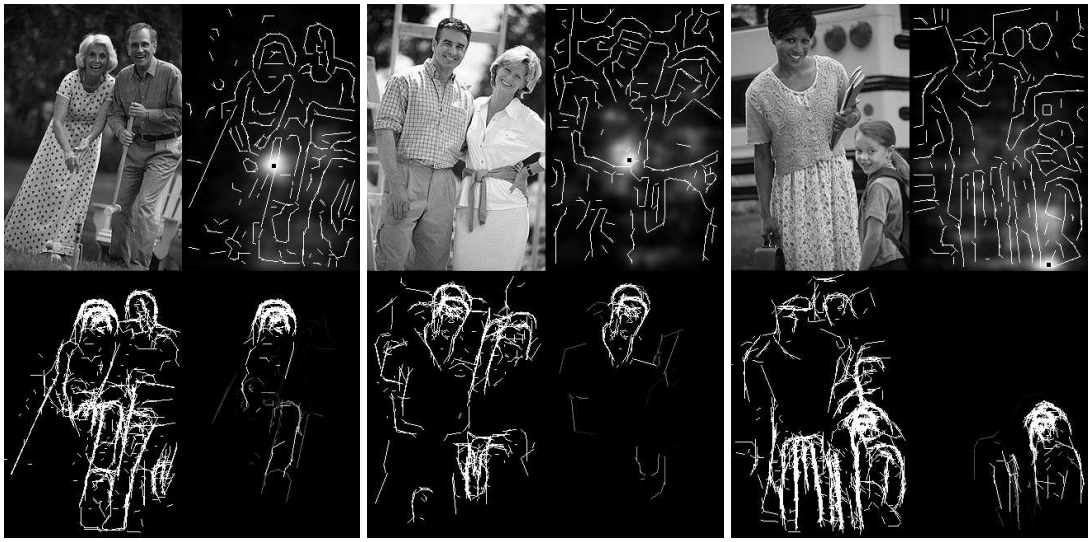


Fig. C.2. *Some detection results for the contour fragment based framework. In each panel, reading clockwise from top-left: the input image, the approximation of its edges by line segments with the estimated probability mask $p(\text{Centroid}|\text{All Fragments})$ and the final estimated centroid location, overlayed the set of line segments used in the detection weighted by their relevance to the overall detection, and the set of all line segment matches that the system used while estimating the probability mask.*

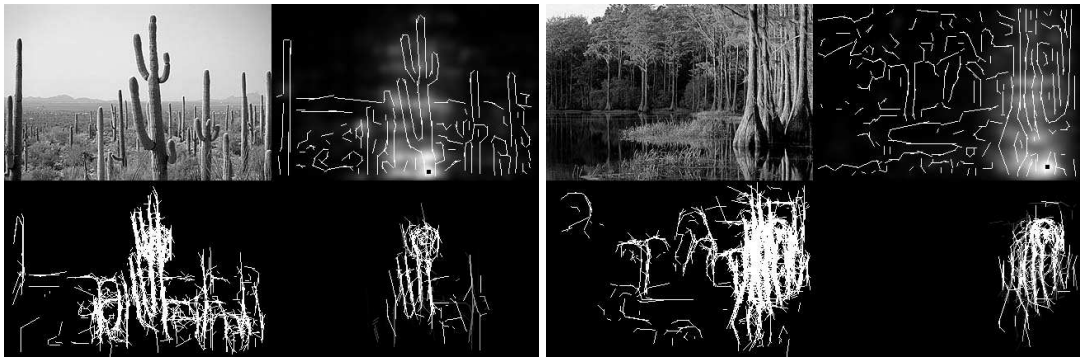


Fig. C.3. *Some false detection results for the contour fragment based framework. The line detector approach was sensitive to clutter in images and usually resulted in false positive under these scenarios.*

fragments and the detection task (maximising the relevance of the fragments in predicting the class). If it happens that the contour fragments are the most compact way to encode the relevant information[1], the method reduces to maximising the MI between the detection task and the fragments. The most relevant fragments are selected using an approach similar to Ullman et al. [2001], Vidal-Naquet and Ullman [2003]. Next, for each relevant fragment, the approach computes the distribution of the fragment location w.r.t. the centroid of the object on the training data. This provides $p(\text{Centroid}|\text{Fragment})$. The overall detection framework uses mean shift along the lines of the method presented in Sect. 7.2.4, except that like Leibe et al. [2004], Leibe and Schiele [2004] we return all modes with values higher than a user supplied threshold as final detections. Figure C.2 shows some positive test images, the relevant contour fragments used for the predictions, and the detection results, and Fig. C.3 shows some negative images containing clutter with detected false positive centre locations.

A key advantage of the approach is that it is a window-free framework. The parts can be any irregularly shaped contour fragment that is sufficiently characteristic and that occurs frequently enough to be useful. The matching function is computationally resistant to the surrounding clutter as it only registers support from the lines within the ϵ-tube regions. However the approach also proved to have several disadvantages: (a) it was sensitive to the edge detection method and the threshold (whose performance varies from image to image depending upon the contrast and clutter in the images), (b) it required positive training images with clean background for training – a restrictive scenario, (c) it was not very resistant to local illumination changes, and (d) the matching function and the selection of the relevant contour fragments was slow owing to the need for exhaustive matching and fragment selection. Although the approach did give reasonable results on an earlier test set with relatively clean images, it did not perform well for natural images under more realistic conditions. Recently edge based approaches have again become more popular for object detection tasks [Opelt et al. 2006, Ferrari et al. 2006], but it remains to be seen whether above limitations can be overcome[2]. Hence we again changed our direction and this time developing dense HOG representation.

[1] Compared to other approaches that use either image gradients, (*e.g.* HOG), or edges, the approximation of edges using straight lines minimises the number of bits required to represent the image.

[2] The contour fragment approach gave particularly poor results for natural images that contained a lot of clutter. Similarly Leibe [2006] report that their detector does not perform well for natural images. In contrast the proposed HOG descriptors give very good results for natural images and rarely result in false positives on natural clutter.

D

Trilinear Interpolation for Voting into 3-D Spatial Orientation Histograms

The histogram computation step in Figs. 4.12 and 6.9 involves distributing the weight of the orientation or optical flow gradient magnitude for every pixel in the cell into the corresponding orientation bins. A naive distribution scheme such as voting the nearest orientation bin would result in aliasing effects. Similarly, pixels near the cell boundaries would produce aliasing along spatial dimensions. Such aliasing effects can cause sudden changes in the computed feature vector. For example, if a strong edge pixel is at the boundary of a cell in one image and due to slight change in imaging conditions it falls into the neighbouring cell in the next, the naive voting scheme assign the pixel's weight to different histograms bins in the two cases. To avoid this, we use 3-D linear interpolation of the pixel weight into the spatial orientation histogram. In the 3-D space the linear interpolation algorithm is know as *trilinear* interpolation. The process is simple but we have recorded several questions about this so, the section provides the details.

We first describe linear interpolation in a one dimension space and then extend it to 3-D. Let $\mathbf{h}$ be a histogram with inter-bin distance (*bandwidth*) b. $\mathbf{h}(x)$ denotes the value of the histogram for the bin centred at x. Assume that we want to interpolate a weight w at point x into the histogram. Let x_1 and x_2 be the two nearest neighbouring bins of the point x such that $x_1 \leq x < x_2$. Linear interpolation distributes the weight w into two nearest neighbours as follows

$$\mathbf{h}(x_1) \leftarrow \mathbf{h}(x_1) + w \cdot \left(1 - \frac{x - x_1}{b}\right)$$
$$\mathbf{h}(x_2) \leftarrow \mathbf{h}(x_1) + w \cdot \left(\frac{x - x_1}{b}\right)$$

The extension to 3-D is simple. Let w at the 3-D point $\mathbf{x} = [x, y, z]$ be the weight to be interpolated. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be the two corner vectors of the histogram cube containing $\mathbf{x}$, where in each component $\mathbf{x}_1 \leq \mathbf{x} < \mathbf{x}_2$. Assume that the bandwidth of the histogram along the x, y and z axis is given by $\mathbf{b} = [b_x, b_y, b_z]$. Trilinear interpolation distributes the weight w to the 8 surrounding bin centres as follows

$$
\begin{aligned}
\mathbf{h}(x_1, y_1, z_1) &\leftarrow \mathbf{h}(x_1, y_1, z_1) + w \left(1 - \tfrac{x-x_1}{b_x}\right) \left(1 - \tfrac{y-y_1}{b_y}\right) \left(1 - \tfrac{z-z_1}{b_z}\right) \\
\mathbf{h}(x_1, y_1, z_2) &\leftarrow \mathbf{h}(x_1, y_1, z_2) + w \left(1 - \tfrac{x-x_1}{b_x}\right) \left(1 - \tfrac{y-y_1}{b_y}\right) \left(\tfrac{z-z_1}{b_z}\right) \\
\mathbf{h}(x_1, y_2, z_1) &\leftarrow \mathbf{h}(x_1, y_2, z_1) + w \left(1 - \tfrac{x-x_1}{b_x}\right) \left(\tfrac{y-y_1}{b_y}\right) \left(1 - \tfrac{z-z_1}{b_z}\right) \\
\mathbf{h}(x_2, y_1, z_1) &\leftarrow \mathbf{h}(x_2, y_1, z_1) + w \left(\tfrac{x-x_1}{b_x}\right) \left(1 - \tfrac{y-y_1}{b_y}\right) \left(1 - \tfrac{z-z_1}{b_z}\right) \\
\mathbf{h}(x_1, y_2, z_2) &\leftarrow \mathbf{h}(x_1, y_2, z_2) + w \left(1 - \tfrac{x-x_1}{b_x}\right) \left(\tfrac{y-y_1}{b_y}\right) \left(\tfrac{z-z_1}{b_z}\right) \\
\mathbf{h}(x_2, y_1, z_2) &\leftarrow \mathbf{h}(x_2, y_1, z_2) + w \left(\tfrac{x-x_1}{b_x}\right) \left(1 - \tfrac{y-y_1}{b_y}\right) \left(\tfrac{z-z_1}{b_z}\right) \\
\mathbf{h}(x_2, y_2, z_1) &\leftarrow \mathbf{h}(x_2, y_2, z_1) + w \left(\tfrac{x-x_1}{b_x}\right) \left(\tfrac{y-y_1}{b_y}\right) \left(1 - \tfrac{z-z_1}{b_z}\right) \\
\mathbf{h}(x_2, y_2, z_2) &\leftarrow \mathbf{h}(x_2, y_2, z_2) + w \left(\tfrac{x-x_1}{b_x}\right) \left(\tfrac{y-y_1}{b_y}\right) \left(\tfrac{z-z_1}{b_z}\right)
\end{aligned}
\tag{D.1}
$$

In our case, when histogramming along the orientation dimension, the orientation bins are also wrapped around.

E

Publications and Other Scientific Activities

Publications

Refereed Journals

- Navneet Dalal, Bill Triggs and Cordelia Schmid. Object Detection Using Histograms of Oriented Gradients. In preparation for submission to Pattern Analysis and Machine Learning.
- Tomás Rodríguez, Ian Reid, Radu Horaud, Navneet Dalal and Marcelo Goetz. Image interpolation for virtual sports scenarios. In *Journal of Machine Vision and Applications*, June 2005.
- Adrien Bartoli, Navneet Dalal and Radu Horaud. Motion Panoramas. In *Journal of Computer Animation and Virtual Worlds*, December 2004. 15(5): 501-517. (Also INRIA Research Report – 4771, 2003).

Refereed Conferences and Workshops

- Navneet Dalal, Bill Triggs and Cordelia Schmid. Human Detection Using Oriented Histograms of Flow and Appearance. In *Proceedings of the European Conference on Computer Vision*, Graz, Austria, May 2006. Vol. II, pp. 428-441.
- Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, USA, June 2005. Vol. II, pp. 886-893.
- Kazunori Okada, Dorin Comaniciu, Navneet Dalal and Arun Krishnan. A Robust Algorithm for Characterizing Anisotropic Local Structures. In *Proceedings of the Eighth European Conference on Computer Vision*, Prague, Czech Republic, May 2004. Vol. I, pp. 549-561.
- Navneet Dalal and Radu Horaud. Indexing Key Positions between Multiple Videos. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, Orlando, Florida, USA, December 2002, pp. 65-71.
- Adrien Bartoli, Navneet Dalal, Biswajit Bose and Radu Horaud. From Video Sequences to Motion Panoramas. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, Orlando, Florida, USA, December 2002, pp. 201-207.

Book Chapter

- Mark Everingham et al. The 2005 PASCAL Visual Object Classes Challenge. In F. d'Alche-Buc, I. Dagan, J. Quinonero (eds), *Selected Proceedings of the first PASCAL Challenges Workshop*, LNAI, Springer, (in press) 2006.

Patents

Navneet Dalal and Dorin Comaniciu. *Multi-scale Detection of Local Image Structures*. US Serial No. 2003P15288US01, filed September 2004.

Book

Coauthoring a book *"Nonparametric Analysis of Visual Data: the Mean Shift Paradigm"* with Dorin Comaniciu, Siemens Corporate Research, and Peter Meer, Rutgers University, NJ. Publisher: Springer. Tentative publishing date: late 2006.

Projects

- Scientific collaborator in **aceMedia**, an European Union sixth frame work project. Led cross-functional team on "Person Detection and Identification" distributed across INRIA, France Télécom R&D and Philips Research.
- Collaborated in European Union project **EVENTS** (INRIA, Oxford University, Siemens, Eptron) developing image interpolation algorithms for rendering novel views.

Other Activities

- Peer reviewer for IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Peer reviewer for International Conference on Computer Vision, IEEE Conference on Computer Vision and Pattern Recognition, European Conference on Computer Vision.

List of Figures

List of Tables

References

S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume IV, pages 113–127, 2002.

S. Avidan and M. Butman. Blind vision. In *Proceedings of the 8th European Conference on Computer Vision, Graz, Austria*, 2006.

S. Baker and S. Nayar. Pattern rejection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Francisco, California, USA*, 1996.

D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

E. Barth, C. Zetzsche, and I. Rentschler. Intrinsic two-dimensional features as textons. *Journal of the Optical Society of America – Optics, Image Science, and Vision*, 15(7):1723–1732, 1998.

P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs fisherfaces: Recognition using class specific linear projection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 1997.

S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, pages 454–461, 2001.

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

M. Bichsel. *Strategies of robust object recognition for the automatic identification of human faces*. Ph.d. thesis, ETH Zurich, 1991.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

D. Comaniciu. Nonparametric information fusion for motion estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume I, pages 59–66, 2003a.

D. Comaniciu. An algorithm for data-driven bandwidth selection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 25(2):281–288, 2003b.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, volume 1, pages 438–445, July 2001.

N. Cristianini and J. Shawe-Taylor. *Support Vector Machines*. Cambridge University Press, 2000.

C.D. Barclay J.E. Cutting and L.T. Kozlowski. Temporal and spatial factors in gait perception that influence gender recognition. *Perception & Psychophysics*, 23(2):142–152, 1978.

J.E. Cutting and L.T. Kozlowski. Recognizing friends by their walk: Gait perception without familiarity cues. *Bull. of the Psychonomic Society*, 9(5):353–356, 1977.

N. W. Daw. The psychology and physiology of colour vision. *Trends in Neurosciences*, 1:330–335, 1984.

V. de Poortere, J. Cant, B. Van den Bosch, J. de Prins, F. Fransens, and L. Van Gool. Efficient pedestrian detection: a test case for SVM based categorization. In *Workshop on Cognitive Vision*, 2002. Available online: http://www.vision.ethz.ch/cogvis02/.

G. Dorkó and C. Schmid. Selection of scale-invariant parts for object class recognition. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, volume 1, pages 634–640, 2003.

A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, pages II:726–733, 2003.

M. Everingham, L. van Gool, C. Williams, and A. Zisserman. Pascal visual object classes challenge results. `http://www.pascal-network.org/challenges/VOC/voc/`, 2006a.

M. Everingham, A. Zisserman, C. K. I. Williams, L. van Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, Gy. Dorkó, S. Duffner, J. Eichhorn, J. D. R. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, T. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, and J. Zhang. The 2005 pascal visual object classes challenge. In F. d'Alche Buc, I. Dagan, and J. Quinonero, editors, *Selected Proceedings of the first PASCAL Challenges Workshop*. LNAI, Springer, 2006b.

P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, pages 66–75, 2000.

R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume II, pages 264–271, 2003.

V. Ferrari, T. Tuytelaars, and L. V. Gool. Object detection by contour segment networks. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, volume II. Springer-Verlag Berlin Heidelberg, 2006.

B. Fischer. Overlap of receptive field centers and representation of the visual field in the cat's optic tract. *Vision Research*, 13:2113–2120, 1973.

D. Fleet and A. Jepson. Stability of phase information. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1253–1268, 1993.

F. Fleuret and D. Geman. Coarse-to-fine face detection. *International Journal of Computer Vision*, 41(1-2):85–107, 2001. ISSN 0920-5691.

W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland*, pages 281–305, June 1987.

W. Förstner and A. Pertl. Photogrammetric standard methods and digital image matching techniques for high precision surface measurements. In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice II*, pages 57–72. Elsevier Science Publishers B.V., 1986.

D.A. Forsyth and M.M. Fleck. Body plans. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 678–683, 1997.

W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *Intl. Workshop on Automatic Face- and Gesture- Recognition, IEEE Computer Society, Zurich, Switzerland*, pages 296–301, June 1995.

W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. Computer vision for computer games. In *2nd International Conference on Automatic Face and Gesture Recognition, Killington, VT, USA*, pages 100–105, October 1996.

W.T. Freeman and E.H. Adelson. The design and use of steerable filters. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine learning, Bari, Italy*, pages 148–156, July 1996a.

Y. Freund and R. E. Schapire. Game theory, on-line prediction and boosting. In *COLT '96: Proceedings of the 9th Annual Conference on Computational Learning Theory*, pages 325–332, New York, NY, USA, 1996b. ACM Press. ISBN 0-89791-811-8.

B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *Proceedings of the ninth British Machine Vision Conference, Southampton, England*, 1998. URL `http://www.cs.otago.ac.nz/research/vision`.

D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA*, pages 87–93, 1999.

D. M. Gavrila, J. Giebel, and S. Munder. Vision-based pedestrian detection: the PROTECTOR+ System. In *Proc. of the IEEE Intelligent Vehicles Symposium, Parma, Italy*, 2004.

I. Haritaoglu, D. Harwood, and L. Davis. W4: Real-time surveillance of people and their activities. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, NY, USA*, 2006. Accepted for publication.

K.P. Horn and G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

D. Hubel. *Eye, Brain, and Vision*. W.H. Freeman & Company, 1995.

D.H. Hubel and T.N. Wiesel. Uniformity of monkey striate cortex: a parallel relationship between field size, scatter, and magnification factor. *J. of Computational Neurology*, 158(3):295–305, 1974.

D. Ingle. The goldfish as a retinex animal. *Science*, 227:651–654, 1985.

Intel OpenCV. Open source computer vision library, 2006. URL `http://www.intel.com/technology/computing/opencv/index.htm`.

S. Ioffe and D. Forsyth. Mixture of trees for object recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, volume II, pages 180–185, 2001a.

S. Ioffe and D. Forsyth. Finding people by sampling. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 1092–1097, 1999.

S. Ioffe and D. A. Forsyth. Probabilistic methods for finding people. *International Journal of Computer Vision*, 43(1):45–68, 2001b.

T. Joachims. Making large-scale svm learning practical. In B. Schlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. The MIT Press, Cambridge, MA, USA, 1999.

G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

T. Kadir and M. Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Washington, DC, USA*, pages 66–75, 2004.

L.T. Kozlowski and J.E. Cutting. Recognizing the sex of a walker from a dynamic point-light display. *Perception & Psychophysics*, 21(6):575–580, 1977.

L.T. Kozlowski and J.E. Cutting. Recognizing the gender of walkers from point-lights mounted on ankles: Some second thoughts. *Perception & Psychophysics*, 23(5):459, 1978.

S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, page 1150, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1950-4.

S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *Proceedings of the 10th International Conference on Computer Vision, Bejing, China*, pages 1284–1291, 2005.

E. H. Land. Color vision and the natural image. part i. *Proceedings of the National Academy of Sciences of the USA*, 45:115–129, 1959a.

E. H. Land. Color vision and the natural image. part ii. *Proceedings of the National Academy of Sciences of the USA*, 45:651–654, 1959b.

S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume 2, pages 319–324, 2003.

S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, August 2005.

B. Leibe. Informative features for profile-view pedestrians. Personal Communication, March 2006.

B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. In *Proceedings of the DAGM'04 Annual Pattern Recognition Symposium, Tuebingen, Germany*, volume 3175, pages 145–153. Springer LNCS, August 2004.

B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proceedings of the ECCV Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic*, pages 17–32, May 2004.

B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Diego, California, USA*, pages 876–885, June 2005.

T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

D. G. Lowe. Local feature view clustering for 3D object recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, pages 682–688, December 2001.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 1150–1157, 1999.

W. L. Lu and J. J. Little. Simultaneous tracking and action recognition using the PCA-HOG descriptor. In *The Third Canadian Conference on Computer and Robot Vision (CRV-06)*, Quebec, Canada, June 2006.

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

D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, May 2004.

R. K. McConnell. Method of and apparatus for pattern recognition, January 1986. U.S. Patent No. 4,567,610.

K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume I, pages 128–142, May 2002.

K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. ISSN 0162-8828.

K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, volume I, pages 69–81, 2004.

K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *Proceedings of the 10th International Conference on Computer Vision, Bejing, China*, volume 2, pages 1792–1799, 2005.

A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, April 2001.

G. Mori and J. Malik. Estimating human body configurations using shape context matching. *Workshop on Models versus Exemplars in Computer Vision, CVPR*, 2001.

G. Mori and J. Malik. Recognizing objects in adversarial clutter: Breaking a visual captcha. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume 1, pages 134–141, 2003.

G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Washington, DC, USA*, volume 2, pages 326–333, 2004.

K. P. Murphy, A. B. Torralba, and W. T. Freeman. Graphical model for recognizing scenes and objects. In *Proceedings of the Neural Information and Processing Systems, Vancouver, Canada*, 2003.

A. Opelt and A. Pinz. Graz 01 data set. On Web, 2004. URL `http://www.emt.tugraz.at/~pinz/data/GRAZ_01/`.

A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, volume II, pages 71–84, 2004.

A. Opelt, A. Pinzl, and A. Zisserman. A boundary-fragment-model for object detection. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, volume II, pages 575–588. Springer-Verlag Berlin Heidelberg, 2006.

C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the 6th International Conference on Computer Vision, Bombay, India*, pages 555–562, 1998.

J. Platt. Probabilities for SV machines. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74, Cambridge, MA, 2000. MIT Press.

F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Diego, California, USA*, volume 1, pages 829–836, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2.

M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, volume 2, pages 295–304, 1994.

D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume II, pages 467–474, 2003.

P. Reinagel and A. M. Zador. Natural scene statistics at the centre of gaze. *Network: Computation in Neural Systems*, 10:341–350, 1999.

R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume IV, pages 700–714, 2002.

H. A. Rowley, S. Baluja, and T. Kanade. Neural networks based face detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 20(1):22–38, 1998.

R. E. Schapire. The boosting approach to machine learning, an overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.

B. Schiele and J.L. Crowley. Object recognition using multidimensional receptive field histograms. In *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, pages 610–619, 1996a.

B. Schiele and J.L. Crowley. Probabilistic object recognition using multidimensional receptive field histogram. In *Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria*, pages 50–54, 1996b.

B. Schiele and J.L. Crowley. Where to look next and what to look for. In *4th Int. Symposium on Intelligent Robotic Systems*, pages 139–146, July 1996c.

C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, May 1997.

C. Schmid, G. Dorkó, S. Lazebnik, K. Mikolajczyk, and J. Ponce. Pattern recognition with local invariant features. In C.H. Chen and S.P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, pages 71–92. World Scientific Pub., 2005. ISBN 981-256-105-6.

H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, volume I, pages 746–751, 2000.

H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, 2004.

B. Schölkopf and A. Smola. *Learning with Kernels*. The MIT Press, Cambridge, MA, USA, 2002.

E. L. Schwartz. Spatial mapping in the primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, 25(4):181–194, 1977.

A. Shashua, Y. Gdalyahu, and G. Hayun. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *International Symposium on Intelligent Vehicles*, pages 1–6, 2004.

L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propogation. In *Proceedings of the Neural Information and Processing Systems, Vancouver, Canada*, 2003.

L. Sirovitch and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America*, 2:586–591, 1987.

C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, 2001.

Y. Song, L. Goncalves, E. D. Bernardo, and P. Perona. Monocular perception of biological motion - detection and labeling. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 805 – 812, 1999.

E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proceedings of the 10th International Conference on Computer Vision, Bejing, China*, pages 1331–1338, 2005.

J. Sun, J.M. Rehg, and A. Bobick. Automatic cascade training with perturbation bias. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Washington, DC, USA*, pages II:276–283, 2004.

M. Tipping. Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machile Learning Research*, 1:211–244, 2001.

N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, USA*, pages 586–591, 1991.

S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. In *4th International Workshop on Visual Form, Capri, Italy*, May 2001.

V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995.

M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, pages 281–288, 2003.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, volume I, pages 511–518, 2001.

P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. ISSN 0920-5691.

P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, volume 1, pages 734–741, 2003.

M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, pages 18–32, 2000.

C. Zetzsche, K. Schill, H. Deubel, G. Krieger, E. Umkehrer, and S. Beinlich. Investigation of a sensorimotor system for saccadic scene analysis: an integrated approach. In *From animals to animats, Proc. of the Fifth Int. Conf. on Simulation of Adaptive Behavior*, volume 5, pages 120–126, 1998.

Q. Zhu, S. Avidan, M. Ye, and K.T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, NY, USA*, 2006. Accepted for publication.