Classifying Tracked Objects in Far-Field Video Surveillance

by

Biswajit Bose

B. Tech. Electrical Engineering
Indian Institute of Technology, Delhi, 2002

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2004

© Massachusetts Institute of Technology 2004. All rights reserved.

Author . . . .
Department of Electrical Engineering and Computer Science
January 8, 2004

Certified by . . . .
W. Eric L. Grimson
Bernard Gordon Professor of Medical Engineering
Thesis Supervisor

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

Classifying Tracked Objects in Far-Field Video Surveillance

by

Biswajit Bose

Submitted to the Department of Electrical Engineering and Computer Science
on January 8, 2004, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Automated visual perception of the real world by computers requires classification of observed physical objects into semantically meaningful categories (such as 'car' or 'person'). We propose a partially-supervised learning framework for classification of moving objects—mostly vehicles and pedestrians—that are detected and tracked in a variety of far-field video sequences, captured by a static, uncalibrated camera. We introduce the use of scene-specific context features (such as image-position of objects) to improve classification performance in any given scene. At the same time, we design a scene-invariant object classifier, along with an algorithm to adapt this classifier to a new scene. Scene-specific context information is extracted through passive observation of unlabelled data. Experimental results are demonstrated in the context of outdoor visual surveillance of a wide variety of scenes.

Thesis Supervisor: W. Eric L. Grimson
Title: Bernard Gordon Professor of Medical Engineering

Acknowledgments

I would like to thank Eric Grimson for his patient guidance and direction through these past 18 months, and for the freedom he has allowed me at the same time in my choice of research topic.

People in the MIT AI Lab (CSAIL?) Vision Research Group have helped create a very friendly work environment. Thanks Gerald, Mario, Chris, Neal, Kevin, Mike, Kinh, ... the list goes on. In particular, without Gerald's help, I would have found navigating the strange world of computer programming much more painful. And Chris has always been very tolerant of my questions/demands regarding his tracking system.

The work described herein would not have been possible without the continued support provided by my father and mother, Tapan and Neera Bose.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer vision (or machine vision) is currently an active area of research, with the goal of developing visual sensing and processing algorithms and hardware that can see and understand the world around them. A central theme in computer vision is the description of an image (or video sequence) in terms of the meaningful objects that comprise it (such as persons, tables, chairs, books, cars, buildings and so on). While the concept of an 'object' comes rather naturally to humans (perhaps because of our constant physical interaction with them), it is very difficult for a computer programme to identify distinct objects in the image of a scene (that is, to tell which pixels in the image correspond to which object). This problem of detection or *segmentation* is an active area of research. Segmentation of a bag of image pixels into meaningful object regions probably requires an intelligent combination of multiple visual cues: colour, shape (as defined by a silhouette), local features (internal edges and corners), spatial continuity and so on. The problem is made challenging by the facts that computers, unlike people, have no *a priori* knowledge of the orientation and zoom of the sensing device (*i.e.*, the camera) and objects are often only partially visible, due to occlusion by other objects which are in the line of sight. Nevertheless, human performance illustrates that these problems are solvable, since they themselves can identify objects in random images shown to them.

Shifting focus from images to video sequences might seem to make matters worse for computer programmes, since this adds an extra temporal dimension to the data.

However, video provides added information that simplifies object detection in that motion in the scene can be used as a cue for separating moving foreground objects from a static background. Object motion can also be used as a feature for distinguishing between different object classes. Further, by tracking objects in video across multiple frames, more information can be obtained about an object's identity.

As the main application of our research is to activity analysis in a scene, we restrict our attention in this thesis to video sequences captured by static cameras and seek to detect and classify objects that move (such as vehicles and pedestrians) in the scene.

1.1 Object Classification

Given a candidate image region in which an object might be present, the goal of object classification is to associate the correct object class label with the region of interest. Object class labels are typically chosen in a semantically meaningful manner, such as 'vehicle', 'pedestrian', 'bird' or 'airplane'. Humans can easily understand events happening around them in terms of interactions between objects. For a computer to reach a similar level of understanding about real-world events, object classification is an important step.

Detection of moving objects in the scene is just the first step towards activity analysis. The output of a motion-based detector is essentially a collection of foreground regions (in every frame of a video sequence) that might correspond to moving objects. Thus, the detection step acts as a filter that focuses our attention on only certain regions of an image. Classification of these regions into different categories of objects is still a huge challenge.

Object classification is often posed as a pattern recognition problem in a supervised learning framework [12]. Under this framework, probabilistic models are used for describing the set of features of each of the N possible object classes, and the class label assigned to a newly detected foreground region corresponds to the object class that was most likely to have produced the set of observed features. Many different object representations (*i.e.* sets of features) have been proposed for the purpose of classification, including 3D models, constellation of parts with characteristic local features, raw pixel values, wavelet-based features and silhouette shape (for example, [15, 13, 31, 32]). However, no one representation has been shown to be universally successful. This is mainly because of the wide range of conditions (including varying position, orientation, scale and illumination) under which an object may need to be classified.

For our purposes, object classification is a process that takes a set of observations of objects (represented using suitable features) as input, and produces as output the probabilities of belonging to different object classes. We require the output to be a probability score instead of a hard decision since this knowledge can be useful for further processing (such as determining when to alert the operator while searching for anomalous activities).

As a supervised learning problem, the time complexity of training for most classification algorithms is quadratic in the number of labelled examples, because of the need to calculate pairwise inner- and/or outer-products in the given feature space. Time complexity for testing is linear in the number of inputs (and perhaps also in the number of training examples). Classifier training is typically performed offline, while testing may be performed either online or offline.

1.1.1 Object Classification Domains

Many visual processes, including object classification, can be approached differently depending on the domain of application: near-, mid- or far-field. These domains are distinguished based on the resolution at which objects are imaged, as this decides the type of processing that can be performed on these objects. In the near-field domain, objects are typically 300 pixels or more in linear dimension. Thus, object sub-parts (such as wheels of a car, parts of a body, or features on a face) are clearly visible. In the far-field domain, objects are typically 10 to 100 pixels in linear dimension. In this situation, the object itself is not clearly visible, and no sub-parts can be detected. Anything in between these two domains can be considered mid-field processing (where the object as a whole is clearly visible, and some sub-parts of an object may be just barely visible).

We address the task of object classification from far-field video. The far-field setting provides a useful test-bed for a number of computer vision algorithms, and is also of practical value. Many of the challenges of near-field vision—dealing with unknown position, orientation and scale of objects, under varying illumination and in the presence of shadows—are characteristics of far-field vision too. Automated far-field object classification is very useful for video surveillance, the objective of which is to monitor an environment and report information about relevant activity. Since activities of interest may occur infrequently, detecting them requires focused observation over prolonged intervals of time. An intelligent activity analysis system can take much of this burden off the user (*i.e.* security personnel). Object classification is often a first step in activity analysis, as illustrated in Section 1.1.2.

In far-field settings, a static or pan-tilt surveillance camera is typically mounted well away from (and typically at some elevation above) the region of interest, such that projected images of objects under surveillance (such as persons and vehicles) range from 10 to 100 pixels in height. Some sample far-field scenes are shown in Figure 1-1.

1.1.2 Applications of Far-Field Object Classification

Far-field object classification is a widely studied problem in the video surveillance research community, since tasks such as automatic pedestrian detection or vehicle detection are useful for activity analysis in a scene [20, 29, 38]. A single round-the-clock far-field surveillance camera, installed near an office complex, car-park or airport, may monitor the activities of thousands of objects—mostly vehicles and persons—every day. In a surveillance system comprising multiple cameras, simply providing the raw video data to security personnel can lead to information overload. It is thus very useful to be able to filter events automatically and provide information about scene activity to a human operator in a structured manner. Classification of moving objects into predefined categories allows the operator to programme the system by specifying events of interest such as ‘send alert message if a person enters building A

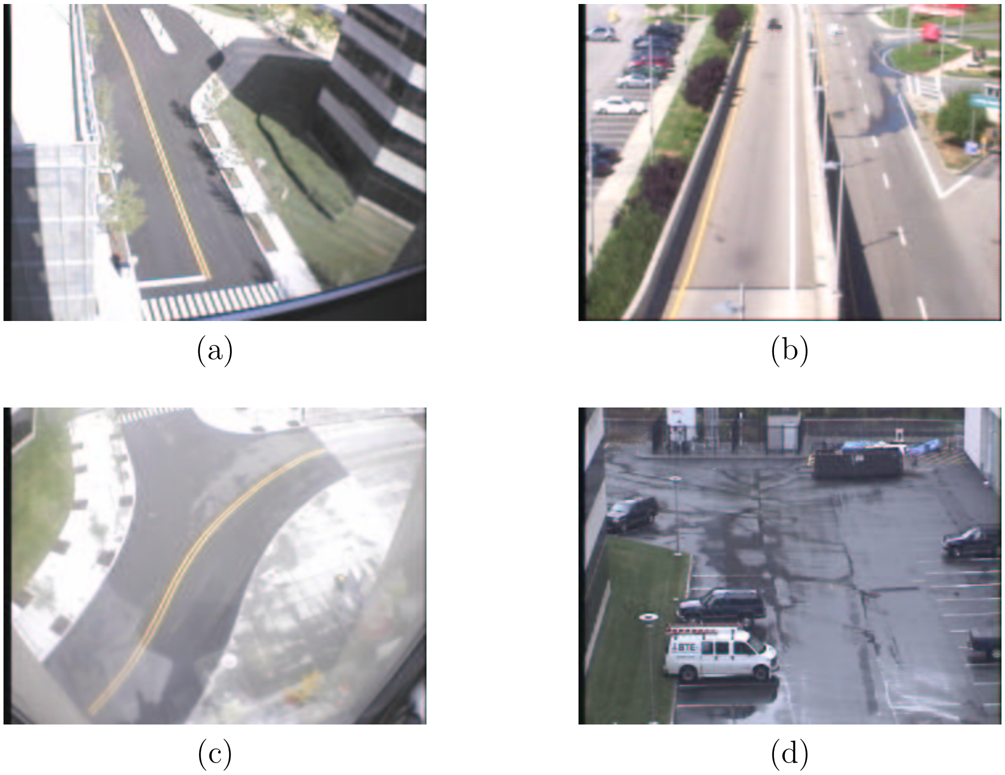

(a) (b) (c) (d)

Figure 1-1: Examples of far-field scenes.

from area B,' or 'track any vehicles leaving area C after 3 p.m.' Classification also provides statistical information about the scene, to answer questions such as finding 'the three most frequent paths followed by vehicles when they leave parking garage D' or 'the number of persons entering building E between 8 a.m. and 11 a.m.'.

1.1.3 Challenging Problems in Far-Field Classification

The far-field domain is particularly challenging because of the low resolution of acquired imagery: objects are generally less than 100 pixels in height, and may be as small as 50 pixels in size. Under these conditions, local intensity-based features (such as body-parts of humans) cannot be reliably extracted. Further, as many far-field cameras are set up to cover large areas, extracted object features may show significant projective distortion—nearby objects appear to be larger in size and to move faster than objects far away.

Most existing vision-based object classification systems can perform well in the restricted settings for which they have been built or trained, but can also fail spectacularly when transferred to novel environments. Sometimes, the scene-specificity is explicitly built into the system in the form of camera calibration. Lack of invariance of the existing methods to some scene-dependent parameter, such as position, orientation, illumination, scale or visibility, is often the limiting factor preventing widespread use of vision systems. This is in stark contrast to the human visual system, which works reasonably well even under dramatic changes in environment (*e.g.* changing from far-field to near-field). What is missing in automated systems is the ability to adapt to changing or novel environments.

Our goal in the present work is to overcome the challenge of limited applicability of far-field classification systems. This is made explicit in the problem statement given in the next section.

1.2 Problem Statement

Given a number of video sequences captured by static uncalibrated cameras in different far-field settings (at least some of which are a few hours in duration and contain more than 500 objects), our objectives are:

- To demonstrate a basic object classification system (that uses the output of motion-based detection and tracking steps) based on supervised learning with a small number of labelled examples, to distinguish between vehicles, pedestrians and clutter with reasonably high accuracy.

- To propose a systematic method for selecting object features to be used for classification, so as to achieve high classification performance in any given scene, but also be able to classify objects in a new scene without requiring further supervised training.

- To propose an algorithm for transferring object-classifiers across scenes, and subsequently adapting them to scene-specific characteristics to minimise classification error in any new scene.

1.3 Outline of our Approach

Our work is based on an existing object-detection and tracking system that employs adaptive background subtraction [35]. The regions of motion detected by the tracking system are fed as input to our classification system. After a filtering step in which most meaningless candidate regions are removed (such as regions corresponding to lighting change or clutter), our system performs two-class discrimination: vehicles vs. pedestrians. We restrict our attention to these two classes of objects since they occur most frequently in surveillance video. However, our approach can be generalised to hierarchical multiclass classification.

Our goal is to address the conflicting requirements of achieving high classification performance in any single scene but also being able to transfer classifiers across scenes without manual supervision. We aim to do this without any knowledge of the position, orientation or scale of the objects of interest. To achieve the first goal, we introduce the use of scene-specific local context features (such as image-position of an object or the appearance of the background region occluded by it). We also identify some other commonly used features (such as object size) as context features. These context features can easily be learnt from labelled examples. However, in order to be consistent with our second goal, we use only scene-invariant features to design a baseline classifier, and adapt this classifier to any specific scene by learning context features with the help of unlabelled data. While learning from unlabelled data is a well-studied problem, most methods assume the distributions of labelled and unlabelled data are the same. This is not so in our case, since context features have different distributions in different scenes. Thus, we propose a new algorithm for incorporating features with scene-specific distributions into our classifier using unlabelled data.

We make three key contributions to object classification in far-field video sequences. The first is the choice of suitable features for far-field object classification from a single, static, uncalibrated camera and the design of a principled technique for classifying objects of interest (vehicles and pedestrians in our case). The second is the introduction of local, scene-specific context features for improving classification performance in arbitrary far-field scenes. The third is a composite learning algorithm that not only produces a scene-invariant baseline classifier that can be transferred across scenes, but also adapts this classifier to a specific scene (using context features) by passive observation of unlabelled data.

Results illustrating these contributions are provided for a wide range of far-field scenes, with varying types of object paths, object populations and camera orientation and zoom factors.

1.4 Organisation of the Thesis

Chapter 2 gives an overview of the problem of object detection and classification in far-field video, and a review of previous work done in this area. The basic moving object detection and tracking infrastructure for far-field video on which our object classification system is built is described in Chapter 3. Our choice of features, as well as the feature selection and grouping process, resulting in the separation of features into two categories—scene-dependent and scene-independent—is discussed in Chapter 4. Chapter 5 presents our algorithm for developing a baseline object classifier (which can be transferred to other scenes) and adapting it to new scenes. Experimental results and analysis are presented in Chapter 6. Chapter 7 summarises the contributions of our work and mentions possible applications, as well as future directions for research.

Chapter 2

Far-Field Object Detection and Classification: A Review

This chapter presents an overview of various approaches to object classification in far-field video sequences. We start by analysing the related problems of object detection and object classification, and then go on to review previous work in the areas of far-field object classification and partially-supervised classification.

2.1 Object Localisation in a Video Frame

To define an object, a candidate set of pixels (that might correspond to an object) needs to be identified in the image. Two complementary approaches are commonly used for localisation of candidate objects. These two approaches can be called the motion-based approach and the object-specific (image-based) approach.

Classification systems employing the motion-based approach assume a static camera and background, and use background subtraction, frame-differencing or optical flow to detect moving regions in each video frame [14, 20, 25]. Detected regions are then tracked over time, and an object classification algorithm is applied to the tracked regions to categorise them as people, groups of people, cars, trucks, clutter and so on. The key feature of this approach is that the detection process needs very little knowledge (if any) of the types of objects in the scene. Instead, the burden of categorising a detected motion-region as belonging to a particular object class is left to a separate classification algorithm. Common features for classifying objects after motion-based detection include size, aspect ratio and simple descriptors of shape or motion.

The other popular approach, direct image-based detection of specific object classes, does not rely on object tracking. Instead, each image (or video-frame) is scanned in its entirety, in search of regions which have the characteristic appearance of an object class of interest, such as vehicles [23, 24] or pedestrians [27]. The class-specific appearance models are typically trained on a number of labelled examples of objects belonging to that class. These methods typically use combinations of low-level features such as edges, wavelets, or rectangular filter responses.

The main advantage of motion-based detection systems is that they serve to focus the attention of the classification system on regions of the image (or video-frame) where objects of interest might occur. This reduces the classification problem from discriminating between an object and everything else in the frame (other objects and the background) to only discriminating between objects (belonging to known or even unknown classes). As a result, false detections in background regions having an appearance similar to objects of interest are avoided. Motion-based detection methods can thus lead to improved performance if the number of objects per frame (and the area occupied by them) is relatively small. Detection of regions of motion also automatically provides information about the projected orientation and scale of the objects in the image (assuming the entire object is detected to be in motion, and none of the background regions are detected as foreground).

A disadvantage of object detection methods based on background-subtraction is that they cannot be used if the motion of the camera is unknown or arbitrary. Also, background-subtraction cannot be used for detecting objects in a single image. However, this is not a severe limitation in most situations, because objects of interest will probably move at some point in time (and can be monitored and maintained even while static) and video for long-term analysis of data has to be captured by a static or pan-tilt-zoom camera.

Other disadvantages of background subtraction (relative to image-based detection) include its inability to distinguish moving objects from their shadows, and to separate objects whose projected images overlap each other (*e.g.* images of vehicles in dense, slow-moving traffic). A combination of image-based and motion-based detection (as in [38], for example) will probably work better than either of the two methods in isolation.

2.2 Related Work: Object Classification

Much work has been done recently on far-field object classification in images and video sequences. In this section, we provide an overview of classification techniques for both object specific (image-based) and motion-based detection systems.

2.2.1 Supervised Learning

Object classification can be framed as a supervised learning problem, in which a learning algorithm is presented with a number of labelled positive and negative examples during a training stage. The learning algorithm (or classifier) estimates a decision boundary that is likely to provide lowest possible classification error for unlabelled test examples drawn from the same distribution as the training examples. Use of a learning algorithm thus avoids having to manually specify thresholds on features for deciding whether or not a given object belongs to a particular class.

Various types of learning algorithms have been used for object classification problems. A simple yet effective classifier is based on modelling class-conditional probability densities as multivariate gaussians [22, 25]. Other types of classification algorithms include support vector machines [27, 28, 31], boosting [24, 38], nearest-neighbour classifiers, logistic linear classifiers [10], neural networks [17] and Bayesian classification using mixtures of gaussians.

Another important decision affecting classification performance is the choice of features used for representing objects. Many possible features exist, including entire images [11, 31], wavelet/rectangular filter outputs [28, 38], shape and size [20, 25], morphological features [14], recurrent motion [9, 20] and spatial moments [17].

Labelled training examples are typically tedious to obtain, and are thus often available only in small quantities. Object-specific image-based detection methods require training on large labelled datasets, especially for low-resolution far-field images. Most detection-based methods have severe problems with false positives, since even a false-positive rate as low as 1 in 50,000 can produce one false positive every frame. To get around this problem, Viola et al. [38] have recently proposed a pedestrian detection system that works on pairs of images and combines appearance and motion cues. To achieve desired results, they use 4500 labelled training examples for detecting a single class of objects, and manually fix the scale to be used for detecting pedestrians.

Methods based on background subtraction followed by object tracking suffer much less from the problems of false positives or scale selection, and have been demonstrated to run in real-time [35]. These methods may also be able to track objects robustly in the presence of partial occlusion and clutter.

2.2.2 Role of Context in Classification

Use of contextual knowledge helps humans perform object classification even in complicated situations. For instance, humans can correctly classify occluded objects when these are surrounded by similar objects (such as a person in a crowd) and have no trouble distinguishing a toy-car from a real one, even though they look alike. In many situations, prior knowledge about scene characteristics can greatly help automated interpretation tasks such as object classification and activity analysis. Contextual information (such as approximate scale or likely positions of occurrence of objects) may be manually specified by an operator for a given scene, to help detect certain activities of interest. [6, 26].

Torralba and Sinha [37] have shown that global context can be learnt from examples and used to prime object detection. They propose a probabilistic framework for modelling the relationship between context and object properties, representing global context in terms of the spatial layout of spectral components. They use this framework to demonstrate context driven focus of attention and scale-selection in real world scenes.

2.3 Related Work: Partially Supervised Learning

Obtaining labelled training data for object classifiers is not easy, especially if hundreds of examples are needed for good performance. Methods based on exploiting unlabelled data provide a useful alternative. Many of these were originally developed in the machine learning community for text classification problems [4, 21], and have recently been applied to object detection/classification problems in the machine vision community. Levin *et al.* [24] use a co-training algorithm [4] to help improve vehicle detection using unlabelled data. This algorithm requires the use of two classifiers that work on independent features, an assumption that is hard to satisfy in practice. Wu and Huang [39] propose a new algorithm for partially supervised learning in the context of hand posture recognition. Stauffer [34] makes use of multiple observations of a single object (obtained from tracking data) to propagate labels from labelled examples to the surrounding unlabelled data in the classifier's feature space.

2.4 Where This Thesis Fits In

The problem of developing classifiers which will work well across scenes has not been directly addressed in the machine vision community. Existing systems tend to make scene-specific assumptions to achieve high performance. Our aim is to be able to classify objects across a wide range of positions, orientations and scales.

To the best of our knowledge, no previous work has been done on learning local context features (such as position and direction of motion of objects) from long-term observation, to improve object classification in scenes observed by a static camera.

Our problem also differs from most of the well-studied problems in the machine learning community because a sub-set of the object features that we consider—the scene-specific context features—have different distributions in different scenes. We propose to identify these scene-specific features and initially keep them aside. Later, after training a classifier using the remaining features, the information contained in the scene-specific features is gradually incorporated by retraining with the help of confidence-rated unlabelled data.

A method for solving a problem that is very similar at the abstract level—combining features whose distribution is the same across data sets with other features whose distribution is data set dependent—has been proposed in [3]. Our approach differs from theirs in the classification algorithm used, as well as in our use of mutual information estimates to perform feature selection and grouping.

Chapter 3

Steps in Far-field Video Processing

This chapter presents the basic infrastructure and processing steps needed in going from raw video to object classification and activity analysis. While the emphasis is on the components of the object classification architecture, necessary details of other processing steps such as background subtraction and region-tracking are given. Definitions of important terms and descriptions of standard algorithms are also provided.

3.1 Background Subtraction

Throughout our work, we rely on distinguishing objects of interest from the background in a video sequence based on the fact that the former move at some point in time (though not necessarily in every frame). Given that a particular scene has been observed for long enough by a static camera, and that an object of interest moves by a certain minimum amount, background subtraction is a relatively reliable method for detecting the object.

Background subtraction consists of two steps: maintaining a model of the background, and subtracting the current frame from this background model to obtain the current foreground. A simple yet robust background model is given by calculating the median intensity value at each pixel over a window of frames. More complex models can adapt to changing backgrounds by modelling the intensity distribution at each pixel as a gaussian (or mixture of gaussians) and updating the model parameters in

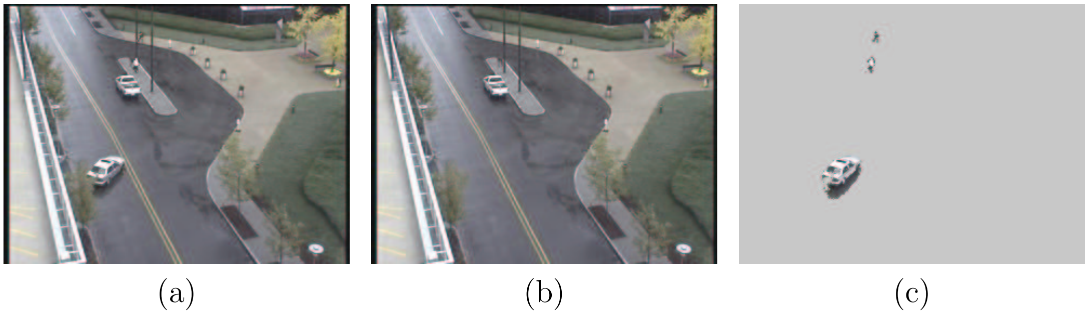


Figure 3-1: An illustration of background subtraction: (a) a video frame, (b) current background model, and (c) pixels identified as belonging to the foreground (shown against a light-grey background for clarity). Note the parked car in (b), which is considered part of the background, and the shadow of the moving car in (c), which is detected as foreground.

an online manner [35]. Adaptive backgrounding is useful for long-term visual surveillance, since lighting conditions can change with time, and vehicles might be parked in the scene (thus changing from foreground to background).

The output of the background subtraction process is a set of foreground pixels, as illustrated in Figure 3-1. At this stage, there is not yet any concept of an object. By applying spatial continuity constraints, connected component regions can be identified as candidate objects. We call each connected-component in a frame a motion-region (or an observation). The information stored for an observation include the centroid location in the frame, the time of observation and the pixel values (colour or grey-scale) for the foreground region within an upright rectangular bounding-box just enclosing the connected-component.

3.2 Region Tracking

The motion-regions identified in each frame by background subtraction need to be associated with one another (across time) so that multiple instances of the same object are available for further processing (such as classification). This is a classic problem of data association and tracking, which has been extensively studied for radar and sonar applications [2]. A simple data association technique uses spatial proximity and similarity in size to assign motion-regions in the current frame with those in the previous frame, while allowing for starting and stopping of tracking sequences if no suitable match is found.

We define a tracking sequence (or a *track*) as a sequence of observations of (supposedly) the same object. The output of the tracking system consists of a set of object tracks. The individual observations that constitute a track are also called instances of a track.

The advantage of performing tracking after background subtraction is that the number of candidate regions for the inter-frame data association problem is greatly reduced. At the same time, many false positives—regions where motion was detected even though there was no moving object present—are eliminated by preserving only those tracks which exceed a certain minimum number of frames in length.

For the purposes of this thesis, background subtraction and tracking were treated as pre-processing steps, and an existing implementation of these steps (developed by Chris Stauffer [35]) was used. We now discuss our main contribution: development of a far-field object classification system.

3.3 Object Classification

Many image-based detection systems implicitly perform classification at the same time as detection, as discussed in Section 2.2. Recently, a few systems have been proposed to use background-subtraction for object detection. In such systems, object classification is a treated as a pattern classification step, involving use of a suitable classification algorithm that is run on the detected and tracked motion-regions with an appropriate set of features. Our system falls into this category.

Pattern classification is a well-studied problem in the field of statistical learning. In this section, we discuss our formulation of object classification from tracking sequences as a supervised pattern classification problem.

3.3.1 Filtering

To demonstrate our algorithms for scene-transfer and scene-adaptation, we consider classification of vehicles and pedestrians. As a pre-processing step, we automatically filter the tracking data to remove irrelevant clutter, thus reducing the classification task to a binary decision problem. Filtering is an important step for long-term surveillance, since (random sampling shows that) more than 80% of detected moving regions are actually spurious objects. This is mainly because lighting changes take place continually and trees are constantly swaying in the wind. Features useful for filtering include minimum and maximum size of foreground region (to filter abrupt changes in lighting), minimum duration (to filter swaying trees), minimum distance moved (to filter shaking trees and fluttering flags) and temporal continuity (since apparent size and position of objects should change smoothly).

Even after filtering out clutter, there are certain classes of objects that are neither vehicles nor pedestrians, such as groups of people and bicycles. Including these classes in the analysis is left for future work.

3.3.2 Video Features

As of December 2003, no single set of visual features has been shown to work for generic object classification or recognition tasks. The choice of features is thus often task-dependent.

There are some common intuitive guidelines, such as use of appearance based features to distinguish objects: a face has a very different appearance from a chair or desk. However, in far-field situations, very few pixels are obtained per object, so local appearance-based features such as parts of a face or body parts of a person cannot be reliably detected. Low resolution data in far-field video prevent us from reliably detecting parts-based features of objects using edge- and corner-descriptors. Instead, we use spatial moment-based features (and their time derivatives) that provide a global description of the projected image of the object, such as size of object silhouette and orientation of its principal axis. The full list of object features we consider is given in Table 3.1; definitions for these features are provided in Section 4.2. Speed, direction of motion and other time derivatives are motion-based features that cannot be used for object classification from a single image.

Though we pick a set of initial features manually, we evaluate them automatically

Feature		Feature
size in pixels ($\mu_{0,0}$)		1st deriv. ϕ_6
norm. 1st deriv. $\mu_{0,0}$		2nd deriv. ϕ_6
norm. 2nd deriv. $\mu_{0,0}$		3rd deriv. ϕ_6
norm. 3rd deriv. $\mu_{0,0}$		4th deriv. ϕ_6
norm. 4th deriv. $\mu_{0,0}$		
		$\eta_{4,0}$ (invariant 4th moment)
x-coordinate		$\eta_{3,1}$ (invariant 4th moment)
y-coordinate		$\eta_{2,2}$ (invariant 4th moment)
velocity magnitude		$\eta_{1,3}$ (invariant 4th moment)
velocity direction		$\eta_{0,4}$ (invariant 4th moment)
acceleration magnitude		1st deriv. $\eta_{4,0}$
acceleration direction		2nd deriv. $\eta_{4,0}$
		1st deriv. $\eta_{3,1}$
principal axis orientation		2nd deriv. $\eta_{3,1}$
1st deriv. orientation		1st deriv. $\eta_{2,2}$
2nd deriv. orientation		2nd deriv. $\eta_{2,2}$
3rd deriv. orientation		1st deriv. $\eta_{1,3}$
ϕ_1 (invariant 2nd moment)		2nd deriv. $\eta_{1,3}$
ϕ_2 (invariant 2nd moment)		1st deriv. $\eta_{0,4}$
1st deriv. ϕ_1		2nd deriv. $\eta_{0,4}$
2nd deriv. ϕ_1		$\eta_{5,0}$ (invariant 5th moment)
3rd deriv. ϕ_1		$\eta_{4,1}$ (invariant 5th moment)
4th deriv. ϕ_1		$\eta_{3,2}$ (invariant 5th moment)
1st deriv. ϕ_2		$\eta_{2,3}$ (invariant 5th moment)
2nd deriv. ϕ_2		$\eta_{1,4}$ (invariant 5th moment)
3rd deriv. ϕ_2		$\eta_{0,5}$ (invariant 5th moment)
4th deriv. ϕ_2		1st deriv. $\eta_{5,0}$
ϕ_3 (invariant 3rd moment)		1st deriv. $\eta_{4,1}$
ϕ_4 (invariant 3rd moment)		1st deriv. $\eta_{3,2}$
ϕ_5 (invariant 3rd moment)		1st deriv. $\eta_{2,3}$
ϕ_6 (invariant 3rd moment)		1st deriv. $\eta_{1,4}$
1st deriv. ϕ_3		1st deriv. $\eta_{0,5}$
2nd deriv. ϕ_3		
3rd deriv. ϕ_3		percentage occupancy
4th deriv. ϕ_3		1st deriv. occupancy
1st deriv. ϕ_4		2nd deriv. occupancy
2nd deriv. ϕ_4		
3rd deriv. ϕ_4		average bg. intensity
4th deriv. ϕ_4		average fg. intensity
1st deriv. ϕ_5		average bg. hue
2nd deriv. ϕ_5		average fg. hue
3rd deriv. ϕ_5		1st deriv. fg. intensity
4th deriv. ϕ_5		1st deriv. fg. hue

Table 3.1: List of object features considered. (bg.=background, fg.=foreground, deriv.=derivative, norm.=normalised.) Definitions and expressions for features are given in Section 4.2.

using mutual information estimates. This is described in Chapter 4. Our reasons for choosing the features mentioned are also discussed there.

Video sequences provide us with two kinds of features, which we call instance features and temporal features. Instance features are those that can be associated with every instance of an object (that is, with each frame of a tracking sequence). The size of an object's silhouette and the position of its centroid are examples of instance features. Temporal features, on the other hand, are features that are associated with an entire track, and cannot be obtained from a single frame. For example, the mean aspect-ratio or the Fourier coefficients of image-size variation are temporal features. Temporal features can provide dynamical information about the object, and can generally only be used after having observed an entire track (or some extended portion thereof). However, temporal features can be converted into instance features by calculating them over a small window of frames in the neighbourhood of a given frame. For example, apparent velocity of the projected object is calculated in this way.

3.3.3 Classifier Architecture

Tracking sequences of objects can be classified in two ways: classifying individual instances in a track separately (using instance features) and combining the instance-labels to produce an object-label, or classifying entire object tracks using temporal features. We chose an instance classifier, for two reasons. Firstly, labelling a single object produces many labelled instances. This helps in learning a more reliable classifier from a small set of labelled objects. Secondly, a single instance feature (*e.g.* position of an object in a frame) often provides more information about the object class than the corresponding temporal feature (*e.g.* mean position of an object).

Each detected object is represented by a sequence of observations, $\mathbf{O} = \{O_i\}, 1 \leq i \leq n$, where n is the number of frames for which the object was tracked. Classification of this object as a vehicle or pedestrian can be posed as a binary hypothesis-testing problem, in which we choose the object class label l_j^* following the maximum-likelihood (ML) rule [12]:

$$l_j^* = \text{argmax } p(\mathbf{O}|l_j) \tag{3.1}$$

(*i.e.* choose l_j corresponding to the higher class-conditional density $p(\mathbf{O}|l_j)$). We use the ML rule instead of the maximum *a posteriori* (MAP) rule because we found the prior probabilities, $p(l_j)$, to be strongly scene-dependent. For instance, while some scenes contain only vehicles, some other contain three times as many pedestrians as vehicles. To develop a scene-invariant classifier, we assume $p(l_1) = p(l_2)$.

The likelihood-ratio test (obtained from Equation 3.1) involves evaluation of $p(O_1, ..., O_n|l_j)$, the joint probability of all the observations conditioned on the class label. For images of a real moving object, this joint distribution depends on many physical and geometric factors such as object dynamics and imaging parameters. A simplifying Markov approximation would be to model the joint probability as a product of terms representing conditional probabilities of each observation given only its recent neighbours in the sequence. However, we choose to avoid estimation of even these conditional probabilities, as their parameters vary with the position of the observation (due to projective distortion). Instead, we search for (approximately) independent observations in the sequence, since the joint probability for independent samples is simply given by $\prod_{i=1}^{n} p(O_i|l_j)$ (*i.e.*, no additional probability distributions are needed to model inter-observation dependences). For every i and j, the probability $p(O_i|l_j)$ can in turn be obtained from the posterior probability of the label given the observation, $p(l_j|O_i)$, by applying Bayes' rule (and cancelling out the marginal observation probabilities upon taking the likelihood ratio):

$$p(O_i|l_j) = \frac{p(l_j|O_i)p(O_i)}{p(l_j)}. \tag{3.2}$$

This means that our classifier can be run separately on each independent observation in a sequence, to produce the corresponding posterior probability of the class label. We approximate independent samples by looking for observations between which the imaged centroid of the object moves a minimum distance. This is useful, for example, to avoid using repeated samples from a stopped object (which is quite common for both vehicles and persons in urban scenes). In our implementation, the minimum distance threshold is equal to the object-length.

3.3.4 Classification with Support Vector Machines

In choosing a suitable classifier, we considered using a generative model (such as a mixture of Gaussians), but decided against it to avoid estimating multi-dimensional densities from a small amount of labelled data. Instead, we chose a discriminative model—support vector machine (SVM) with soft margin and Gaussian kernel—as our instance classifier. (The use of a soft margin is necessary since the training data are non-separable.) In the SVM formulation (for nonseparable data), we look for the maximum-margin separating hyperplane (parameterised by $\mathbf{w}$ and b) for the N training points $\mathbf{x}_i \in \Re^k$ (in a k-dimensional feature space) and corresponding labels $y_i \in \{-1, 1\}$, given the optimisation problem [7]:

$$\text{Minimise} \qquad -\frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum \xi_i \qquad\quad \text{subject to} \quad (3.4) \quad \text{and} \quad \xi \geq 0, \tag{3.3}$$

where we have introduced N nonnegative variables $\xi \;=\; (\xi_1, \xi_2, ..., \xi_N)$ such that

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, ..., N, \tag{3.4}$$

to account for the fact that the data are nonseparable. As in the separable case, this can be transformed into the *dual* problem:

$$\text{Maximise } \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{3.5}$$

subject to

$$\sum y_i \alpha_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, ..., N \tag{3.6}$$

where α_i are the Lagrange multipliers (with associated upper-bound C) and K represents the SVM kernel function. Applying the Kuhn-Tucker conditions, we get

$$\alpha_i(y_i(K(\mathbf{w}, \mathbf{x}_i) + b) - 1 + \xi_i) = 0 \tag{3.7}$$

$$(C - \alpha_i)\xi_i = 0 \tag{3.8}$$

It is useful to distinguish between the support vectors for which $\alpha_i < C$ and those for which $\alpha_i = C$. In the first case, from condition (3.8) it follows that $\xi_i = 0$, and hence, from condition (3.7), these support vector are margin vectors. On the other hand, support vectors for which $\alpha_i = C$ are mostly either misclassified points, or correctly classified points within the margin. The bounds on the Lagrange multipliers help to provide a 'soft margin', whereby some examples can lie within the margin, or even on the wrong side of the classification boundary.

For training our baseline (scene-invariant) object classifier (Section 5.1), we fixed C to a large value (= 1,000). Our scene-adaptation algorithm (Section 5.2), however, uses different values of C for different 'labelled' examples, depending on the confidence of the associated label.

One disadvantage of using SVMs is that the output, d_i, is simply the signed distance of the test instance from the separating hyperplane, and not the posterior probability of the instance belonging to an object class. Posterior probabilities are needed to correctly combine instance labels using the ML rule to obtain an object label. To get around this problem, we retrofit a logistic function $g(d)$ that maps the SVM outputs d_i into probabilities [30]:

$$g(d_i) = \frac{1}{1 + exp(-d_i)}. \tag{3.9}$$

The posterior probability is then given by

$$P(y_i = 1 \,|\, d_i, \lambda) = g(\lambda d_i), \tag{3.10}$$

where the parameter λ is chosen such as to maximise the associated log-likelihood

$$l(\lambda) = \sum_{i=1}^{N} log\, P(y_i \mid d_i, \lambda). \tag{3.11}$$

The posterior probabilities thus obtained are used both for classifying sequences of observations (using Equation 3.1) and for associating confidences with the 'labels' of the classified objects for use in adapting the classifier to a specific scene (Section 5.2). To clarify, test error corresponds to the fractional number of incorrect *object* labels, not instance labels.

3.4 Activity Understanding using Objects

Object classification is typically not an end in itself. Activity analysis in a scene typically relies on object classification. In some sufficiently constrained situations (such as a pedestrian zone), objects from only one class may appear, so classification becomes trivial (since any large moving object must belong to the object class of interest). However, this is not the case in general. Activities of interest might include detecting pedestrians on a street, counting the number of vehicles belonging to different classes on a highway, understanding activities of people in a car-park, ensuring normal movement of airplanes at an airport and tracking vehicles that take unusual paths through a scene. A good object classifier should be able to perform its task in all these situations with the minimum possible task-specific information.

Chapter 4

Informative Features for Classification

It is commonly acknowledged that object representation is key to solving several machine vision tasks. In this chapter, we discuss the set of features we chose to represent objects. Some of the features we consider are not commonly used for object classification; we explain our motivation for using them and demonstrate their utility. We also propose a principled technique to select object features for use in a classification task, which takes into account whether the characteristics of the scene are known *a priori*. This allows us to identify two categories of features: scene-dependent (or context) and scene-independent features. These two categories of features are used in the next chapter to develop object-classifiers that work in a wide range of scenes.

4.1 Features for Representing Objects

The input to our classification system consists of sequences of tracked foreground regions, or observations. After filtering (Section 3.3.1), each observation consists of a 2D silhouette of a vehicle or pedestrian located at a specified position in the image, along with the corresponding pixel-values (in colour or gray-level). As a first approximation, we can ignore the pixel-values (though we will later use this information). This is because we expect vehicles to be painted differently, and pedestrians to wear clothing of a wide variety of colours, so that very little information about the object class is likely to be present in the recorded intensity values. Thus, we are essentially left with a sequence of binary object silhouettes (each observed at a known position and time).

The problem of classifying binary images (*i.e.* silhouettes) has been studied extensively in the image processing and pattern recognition communities (see chapter 9 of [19] for an overview). In order to ensure that the set of features used is sufficiently descriptive, it is a good idea to choose an exhaustive set of features, such as moment-based or Fourier descriptors. However, using the entire set of exhaustive features is neither practical nor necessary, since

1. the obtained silhouettes are corrupted by noise, which renders some of the features useless, and

2. we are interested in discriminating between object classes, not in providing a complete description of the members of either class.

Keeping both of these in mind, we resort to performing feature selection for classification. The information-theoretic concept of mutual information gives us a principled way of doing this selection. First, however, we need to discuss the set of features we consider.

4.2 Moment-Based Silhouette Representation

We have chosen a moment-based representation for the shapes of object silhouettes. The $(p+q)$th-order spatial moment of a silhouette $f(x, y)$ is given by

$$m_{p,q} = \iint f(x, y) x^p y^q \, dx \, dy, \qquad p, q = 0, 1, 2, ... \tag{4.1}$$

where $f(x, y) = 1$ for points inside the silhouette and zero elsewhere. For digital images, these moments can be approximated by replacing the above integrals by summations.

The infinite set of moments $m_{p,q}, p, q = 0, 1, 2, ...$ uniquely determines an arbitrary silhouette. Low-order moments of binary images include the size in pixels (zeroth moment), position of centroid (first moment) and moment of inertia along orthogonal axes (second moment).

4.2.1 Advantage of Moment-Based Representations

Spatial moments are closely related to features that have been used successfully for various perceptual tasks, such as dispersedness or compactness of a silhouette. At the same time, the moments form an exhaustive set, so there is no arbitrariness involved in coming up with features. Neither is there any danger of missing any features.

Another advantage of moments is that they can easily be made invariant to 2D transformations such as translation, rotation, reflection and scaling [18]. This is useful since such transformations are quite common for objects moving on a ground plane and imaged by a surveillance camera from an arbitrary position and orientation. Such moment invariants can also be useful within a single scene, as they cancel some of the effects of projective distortion as an object moves around on the ground plane. To achieve translation-invariance, the moments defined in Equation 4.1 are replaced by the central moments

$$\mu_{p,q} = \iint (x - \bar{x})^p (y - \bar{y})^q f(x, y)\, dx\, dy \qquad (4.2)$$

where $\bar{x} = m_{1,0}/m_{0,0}$ and $\bar{y} = m_{0,1}/m_{0,0}$. To achieve scale-invariance in addition to translation-invariance, the central moments can be converted to normalised moments:

$$\eta_{p,q} = \frac{\mu_{p,q}}{(\mu_{0,0})^\gamma}, \qquad \gamma = (p + q + 2)/2. \qquad (4.3)$$

To achieve rotation- and reflection-invariance in addition to translation-invariance, moment invariants for second and third order moments are given by:

$$\phi_1 = \mu_{2,0} + \mu_{0,2} \qquad (4.4)$$

$$\begin{aligned}
\phi_2 &= (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 && (4.5)\\
\phi_3 &= (\mu_{3,0} - 3\mu_{1,2})^2 + (\mu_{0,3} - 3\mu_{2,1})^2 && (4.6)\\
\phi_4 &= (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{0,3} + \mu_{2,1})^2 && (4.7)\\
\phi_5 &= (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2})[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{0,3} + \mu_{2,1})^2] \\
&\quad + (\mu_{0,3} - 3\mu_{2,1})(\mu_{0,3} + \mu_{2,1})[(\mu_{0,3} + \mu_{2,1})^2 - 3(\mu_{3,0} + \mu_{1,2})^2] && (4.8)\\
\phi_6 &= (\mu_{2,0} - \mu_{0,2})[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{0,3} + \mu_{2,1})^2] \\
&\quad + 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{0,3} + \mu_{2,1}). && (4.9)
\end{aligned}$$

Higher-order moments can be made rotation-invariant by first calculating the principal axis of the object silhouette and rotating the silhouette to align the principal direction with the horizontal axis. The principal axis is obtained as the eigenvector corresponding to the larger eigenvalue of the sample covariance matrix of the 2D silhouette (which in turn can be expressed in terms of the second central moments).

Moment invariants have previously been employed for pattern recognition tasks [18, 36]. In the present work, we shall also consider moments that are not invariant, and show that these features contain useful information in certain circumstances.

For real-world data, moment-based representations are good for distinguishing between objects of different classes (which have gross differences), but not for differentiating objects of the same class based on fine differences (such as between undamaged and damaged machine parts), since the latter only differ in higher order moments that are significantly affected by noise. While there have been efforts directed towards robust use of moments [1, 33], we did not consider these for our classification task as we are able to achieve good results using the standard low-order moments.

4.2.2 Additional Object Features

Object moments provide a complete description of a single observation, at a particular instant of time. To provide a comprehensive description across time, we need to add a descriptor that captures time variation. A simple yet exhaustive set of such descriptors is the set of time derivatives of all orders. The zeroth order derivatives

		Order of time-derivative				
		0	1	2	3	...
	0	size	rate-of-change in size			...
Order of	1	position (of centroid)	velocity	acceleration	jolt/jerk	...
moment	2	orientation and inertia	rate-of-change of inertia			...
	3	skewness				...
	⋮	⋮	⋮	⋮	⋮	⋱

Table 4.1: Exhaustive set of features for representing object silhouettes.

are the moments themselves. Moments of all orders, along with their time derivatives of all orders, contain complete information for characterising any tracking sequence. An illustration of this exhaustive set of features, as well as names for some of the commonly encountered members of this set, is given in Table 4.1.

It should be noted that since second and higher order moments have been normalised for scale-invariance (as discussed above), their time derivatives will also be normalised. Derivatives of silhouette size (the zeroth order moment) can also be normalised by dividing by the silhouette size.

So far we have ignored the information present in the object's appearance (*i.e.* pixel-values within the foreground region). We now add a few descriptors such as average foreground and background intensity and hue, and their time derivatives; the complete list is given in Table 4.2, where most of these features are shown to contribute little to the classification task.

In practice, it is not possible to accurately calculate very high order spatial moments from image data. To approximate the contribution of high order moments, we introduce an extra feature: percentage occupancy. This is defined as the percentage of pixels corresponding to the object silhouette within the smallest principal-axis-aligned bounding rectangle around the silhouette.

4.3 Using Mutual Information Estimates for Feature Selection

Having decided our feature space for object representation, we need to perform feature selection. Feature selection is necessary not only for computational reasons (to pick a finite number of features from the infinite feature set defined above) but also to remove non-informative features before training an object classifier, and thus avoid overfitting when training on a small labelled data set.

It might seem that the right thing to do for the purpose of feature selection is simply to calculate test errors (or cross-validation errors) for different groups of features, and pick the group giving lowest error. However, this would make the feature selection process classifier dependent. Our aim is not to suggest that one classifier is better than another, but rather to provide a framework for feature selection that can be used with any classifier. Thus, we perform both feature selection and feature grouping by calculating mutual information estimates.

The mutual information (MI), $I(\mathbf{X};Y)$, between a continuous random vector $\mathbf{X}$ and a discrete random variable Y is given by [8]:

$$I(\mathbf{X};Y) = H(\mathbf{X}) - H(\mathbf{X}|Y), \tag{4.10}$$

where $H(\mathbf{X})$ and $H(\mathbf{X}|Y)$ are the entropy of $\mathbf{X}$ and the conditional entropy of $\mathbf{X}$ given Y respectively. $I(\mathbf{X};Y)$ is a non-negative, symmetric function of $\mathbf{X}$ and Y.

For the purpose of feature selection, $\mathbf{X}$ represents the set of features and Y represents the object class label. The required entropies can then be estimated from data using kernel-based non-parametric estimates of marginal and conditional densities, and plugged into Equation 4.10 to obtain an MI estimate.

Intuitively speaking, the mutual information between two variables is a measure of how much of the uncertainty in the value of one variable is reduced by knowing the value of the other. Thus, features having high MI with class labels are likely to be very useful in correctly predicting the object class. It can be proved that among a set of features, the single feature having the highest MI with a label is also the one which, if used alone, will give the lowest classification error.

Most classification tasks make use of more than one feature. Unfortunately, simply calculating MI scores between individual features and labels and choosing the features with the highest scores is not guaranteed to give the most informative set of features for classification. In order to be (theoretically) optimal, MI scores need to be calculated between all possible sets of features and labels.For instance, if a road cuts diagonally across the scene and people walk on footpaths beside it, the individual MI scores of x- and y- image coordinates with labels might be low, but the two considered jointly may accurately classify the object. In principle, this argument holds for all possible sets of features. However, is in generally computationally infeasible to calculate mutual information between all labels and all possible sets of features. In practice, it is unlikely that three or more features calculated from real-world scenes will conspire to give significantly better classification results than pairs of features acting together. Therefore, we repeat our MI calculations for all possible pairs of features.

The mutual information $I(\mathbf{X_1}, \mathbf{X_2}; Y)$ between a pair of features $(\mathbf{X_1}, \mathbf{X_2})$ and the label Y can be expanded as follows [8]:

$$I(\mathbf{X_1}, \mathbf{X_2}; Y) = I(\mathbf{X_1}; Y) + I(\mathbf{X_2}; Y|\mathbf{X_1}). \tag{4.11}$$

Since MI is non-negative, the above equation shows that the mutual information for any pair of features is never less than the MI for either of the individual features. In the limiting case, if one of the features (say $\mathbf{X_1}$) perfectly predicts the class label, or there exists a deterministic relation connecting the two features being considered, the second term in Equation 4.11 is zero. On the other hand, this second term can be large for features which are individually poor in helping to determine the object class, but jointly provide significant information about the class label.

In the next two sections, we use MI estimates for our features to perform feature selection and feature grouping.

4.4 Feature Selection for a Single Scene

In this section, we assume that all the objects whose features are being considered were taken from a single scene. We will generalise this to the case of multiple scenes in Section 4.5, where we consider feature grouping in addition to feature selection.

4.4.1 Information Content of Individual Features

Mutual information (MI) scores between class labels and a number of low-order moments, time-derivatives of moments, and intensity/colour-based features were estimated. Moments up to fifth order and time derivatives up to fourth order were calculated. In general, high order time derivatives tended to be less significant (*i.e.* have lower MI scores) than low order ones, due to the effects of noise. The MI scores for the more significant features—those whose scores are greater than 0.05 bits—are shown in Table 4.2.

From the MI scores, it is clear that time derivatives play an important role in differentiating vehicles from persons. This is because vehicles are rigid objects with a fixed silhouette shape, while the shape of a pedestrian's silhouette changes throughout the walking cycle. In fact, changes in the pedestrian silhouette are approximately periodic, so both low-order and high-order time derivatives provide information about the object class.

After looking at the distribution of MI scores for our set of features, a threshold of 0.25 bits was chosen as the minimum required MI for an informative feature. Features below this threshold were considered individually irrelevant to the classification task (but were not discarded yet, as they might be relevant in a pair with another feature). Features above this threshold were considered individually relevant for classification (but were not guaranteed to be selected, as two or more of these features might provide essentially the same information, and hence be redundant). The final feature selection decision was made after estimating the information content of pairs of features, as discussed next.

Feature	MI	y/n		Feature	MI	y/n
size in pixels ($\mu_{0,0}$)	0.71	y		1st deriv. ϕ_6	0.60	n(p)
norm. 1st deriv. $\mu_{0,0}$	0.43	y		2nd deriv. ϕ_6	0.60	n(p)
norm. 2nd deriv. $\mu_{0,0}$	0.55	y		3rd deriv. ϕ_6	0.62	n(p)
norm. 3rd deriv. $\mu_{0,0}$	0.39	y		4th deriv. ϕ_6	0.59	n(p)
norm. 4th deriv. $\mu_{0,0}$	0.32	n(p)				
				$\eta_{4,0}$ (invariant 4th moment)	0.71	n(p)
x-coordinate	0.22	y(p)		$\eta_{3,1}$ (invariant 4th moment)	0.46	y
y-coordinate	0.17	y(p)		$\eta_{2,2}$ (invariant 4th moment)	0.32	y
velocity magnitude	0.39	y		$\eta_{1,3}$ (invariant 4th moment)	0.26	y
velocity direction	0.13	y(p)		$\eta_{0,4}$ (invariant 4th moment)	0.31	y
acceleration magnitude	0.16	n		1st deriv. $\eta_{4,0}$	0.44	y
acceleration direction	0.08	n		2nd deriv. $\eta_{4,0}$	0.37	y
				1st deriv. $\eta_{3,1}$	0.21	n
principal axis orientation	0.41	y		2nd deriv. $\eta_{3,1}$	0.14	n
1st deriv. orientation	0.19	n		1st deriv. $\eta_{2,2}$	0.27	y
2nd deriv. orientation	0.27	y		2nd deriv. $\eta_{2,2}$	0.18	n
3rd deriv. orientation	0.20	n		1st deriv. $\eta_{1,3}$	0.44	y
ϕ_1 (invariant 2nd moment)	0.41	y		2nd deriv. $\eta_{1,3}$	0.41	y
ϕ_2 (invariant 2nd moment)	0.40	y		1st deriv. $\eta_{0,4}$	0.32	y
1st deriv. ϕ_1	0.66	y		2nd deriv. $\eta_{0,4}$	0.22	n
2nd deriv. ϕ_1	0.57	y		$\eta_{5,0}$ (invariant 5th moment)	0.41	y
3rd deriv. ϕ_1	0.51	n(p)		$\eta_{4,1}$ (invariant 5th moment)	0.35	y
4th deriv. ϕ_1	0.46	n(p)		$\eta_{3,2}$ (invariant 5th moment)	0.29	y
1st deriv. ϕ_2	0.78	y		$\eta_{2,3}$ (invariant 5th moment)	0.18	n
2nd deriv. ϕ_2	0.74	y		$\eta_{1,4}$ (invariant 5th moment)	0.24	n
3rd deriv. ϕ_2	0.71	n(p)		$\eta_{0,5}$ (invariant 5th moment)	0.20	n
4th deriv. ϕ_2	0.64	n(p)		1st deriv. $\eta_{5,0}$	0.44	y
ϕ_3 (invariant 3rd moment)	0.50	y		1st deriv. $\eta_{4,1}$	0.35	y
ϕ_4 (invariant 3rd moment)	0.55	y		1st deriv. $\eta_{3,2}$	0.26	y
ϕ_5 (invariant 3rd moment)	0.51	y		1st deriv. $\eta_{2,3}$	0.24	n
ϕ_6 (invariant 3rd moment)	0.38	y		1st deriv. $\eta_{1,4}$	0.25	n
1st deriv. ϕ_3	0.68	y		1st deriv. $\eta_{0,5}$	0.29	y
2nd deriv. ϕ_3	0.68	y				
3rd deriv. ϕ_3	0.65	n(p)		percentage occupancy	0.32	n(p)
4th deriv. ϕ_3	0.64	n(p)		1st deriv. occupancy	0.24	n
1st deriv. ϕ_4	0.75	n(p)		2nd deriv. occupancy	0.17	n
2nd deriv. ϕ_4	0.77	n(p)				
3rd deriv. ϕ_4	0.76	n(p)		average bg. intensity	0.63	y
4th deriv. ϕ_4	0.74	n(p)		average fg. intensity	0.16	n
1st deriv. ϕ_5	0.77	y		average bg. hue	0.21	n
2nd deriv. ϕ_5	0.73	y		average fg. hue	0.24	n
3rd deriv. ϕ_5	0.76	n(p)		1st deriv. fg. intensity	0.04	n
4th deriv. ϕ_5	0.72	n(p)		1st deriv. fg. hue	0.23	n

Table 4.2: Average mutual information (M.I.) scores between object features and labels in a single scene (averaged over seven scenes), along with decision to select (y) or reject (n) features. M.I. is measured in bits (maximum possible score = 1.0). (p) means the decision was made after considering M.I. scores for pairs of features. bg.=background, fg.=foreground, deriv.=derivative, norm.=normalised.

Feature 1	Feature 2	MI	Category
x-coordinate	y-coordinate	0.51	A
velocity direction	orientation	0.66	A
norm. fourth deriv. $\mu_{0,0}$	ϕ_5	0.53	B
third deriv. ϕ_1	ϕ_5	0.54	B
fourth deriv. ϕ_1	ϕ_5	0.54	B
third deriv. ϕ_2	first deriv. ϕ_5	0.79	B
fourth deriv. ϕ_2	first deriv. ϕ_5	0.79	B
third deriv. ϕ_3	first deriv. ϕ_5	0.81	B
fourth deriv. ϕ_3	second deriv. ϕ_5	0.77	B
first deriv. ϕ_4	first deriv. ϕ_5	0.81	B
second deriv. ϕ_4	first deriv. ϕ_5	0.80	B
third deriv. ϕ_4	first deriv. ϕ_5	0.80	B
fourth deriv. ϕ_4	first deriv. ϕ_5	0.80	B
third deriv. ϕ_5	first deriv. ϕ_2	0.81	B
fourth deriv. ϕ_5	first deriv. ϕ_2	0.79	B
first deriv. ϕ_6	first deriv. ϕ_5	0.79	B
second deriv. ϕ_6	first deriv. ϕ_5	0.78	B
third deriv. ϕ_6	first deriv. ϕ_5	0.79	B
fourth deriv. ϕ_6	first deriv. ϕ_5	0.81	B
$\eta_{4,0}$	first deriv. ϕ_5	0.81	B
percentage occupancy	$\eta_{4,1}$	0.39	B

Table 4.3: Mutual information (MI) scores of note between pairs of object features and labels in a single scene, along with corresponding decision to select both the features (A) or reject the first feature (B) in the pair. MI scores are given in bits. See Section 4.4.2 for an explanation of categories A and B.

4.4.2 Information Content of Feature Pairs

Mutual information scores for pairs of features were estimated and studied with the purpose of identifying features belonging to either of two categories:

- category A: features which are individually irrelevant, but relevant when considered jointly with other features, and
- category B: features which are individually relevant, but redundant when considered jointly with other features.

As stated earlier, both these decisions can be made conclusively only after considering all possible groups of features. Here we assume that only pairs of features are jointly relevant (and that groups of three or more features will not provide significantly more information than any pair of features). We now present the algorithms used to decide the category membership of features.

An individually irrelevant feature $\mathbf{X_1}$ (identified in Section 4.4.1) belongs to category A if there exists some other feature $\mathbf{X_2}$ such that $I(\mathbf{X_1}, \mathbf{X_2}; Y) - I(\mathbf{X_1}; Y) > i_1$ and $I(\mathbf{X_1}, \mathbf{X_2}; Y) - I(\mathbf{X_2}; Y) > i_1$, for a chosen threshold i_1. This means that the information provided by the two features considered jointly is substantially greater than that provided by each individually, so there is merit in selecting these features. For our experiments, we set $i_1 = 0.10$ after observing the distribution of difference in MI between pairs of variables and individual variables.

For the purposes of testing category B membership, all category A members are considered individually relevant. Then, an individually relevant feature $\mathbf{X_3}$ (identified in Section 4.4.1 or in category A) belongs to category B if there exists another feature $\mathbf{X_4}$ such that all the following four conditions are satisfied:

1. $I(\mathbf{X_4}; Y) > \mathbf{X_3}; Y)$,
2. $I(\mathbf{X_4}, \mathbf{X_3}; Y) - I(\mathbf{X_4}; Y) < i_2$,
3. for all other features $\mathbf{X_5}$, $I(\mathbf{X_5}, \mathbf{X_3}; Y) - I(\mathbf{X_4}, \mathbf{X_3}; Y) < i_2$ and

4. for each feature $\mathbf{X_5}$ such that $I(\mathbf{X_5}, \mathbf{X_3}; Y) - I(\mathbf{X_3}; Y) > i_3$ and $I(\mathbf{X_5}, \mathbf{X_3}; Y) - I(\mathbf{X_5}; Y) > i_3$, $I(\mathbf{X_5}, \mathbf{X_4}, \mathbf{X_3}; Y) - I(\mathbf{X_4}; Y) < i_3$,

for suitable thresholds i_2 and i_3. The basic idea behind these four conditions is to find an individually relevant feature $\mathbf{X_4}$ which can provide (almost) all the information provided by feature $\mathbf{X_3}$, thus rendering the latter redundant. Conditions 1 and 2 ensure that the two features jointly do not provide much more information than $\mathbf{X_4}$ alone. Condition 3 checks that $\mathbf{X_3}$ considered jointly with any other feature does not provide much more information than $\mathbf{X_4}$ considered jointly with that other feature. Condition 4 further checks that for those features $\mathbf{X_5}$ which provide substantial information when considered jointly with $\mathbf{X_3}$, the information contained in $\mathbf{X_3}$, $\mathbf{X_4}$ and $\mathbf{X_5}$, all three considered jointly, is not substantially more than the information in $\mathbf{X_4}$ alone. Though this last condition subsumes the other three, it is tested last as it is computationally more expensive to calculate 3-variable MI scores. For our experiments, the thresholds chosen (by observing the distribution of MI scores) were $i_2 = 0.05$ and $i_3 = 0.10$.

The most relevant mutual information results (corresponding to category A or B membership) between class labels and pairs of features are given in Table 4.3. In this table, the column labelled 'Feature 1' corresponds to $\mathbf{X_1}$ or $\mathbf{X_3}$, while the column labelled 'Feature 2' corresponds to $\mathbf{X_2}$ or $\mathbf{X_4}$.

The final feature selection (for any single scene) is as follows:

- features that are individually irrelevant and not in category A are rejected,
- features that are in category B are rejected, and
- all remaining features are selected.

These results are given in Table 4.2, in the column containing 'y' (selected) or 'n' (rejected); the list of selected features is also repeated in Table 4.4.

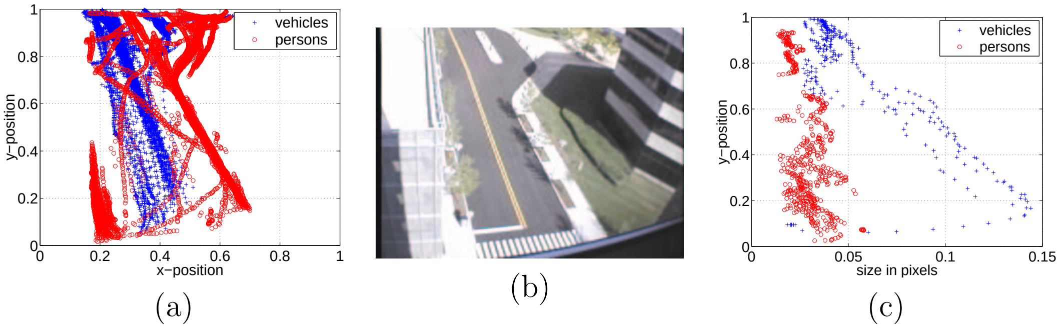
It is important that while the features rejected above are unlikely to be useful for further processing, the features selected above are not always guaranteed to be useful. These features should give good classification performance when both training and testing a classifier on data from any single scene. However, we are interested in the more general case of training in one scene and testing in another scene. Among the features selected above, there are some that will not be useful in this general case. We study these next.

4.4.3 Context Features

We define context features as features that are useful for classification (training and testing) in any single scene, but not for training in one scene and testing in another scene. Context features are thus scene-dependent/specific. They do not transfer across scenes, because they have different distributions in different scenes.

For some features, such as image position of an object, it is clear that the feature is scene-specific. However, for others such as aspect ratio or orientation, it is not obvious whether the feature is scene-dependent. Therefore, we estimate mutual information scores between features and labels for data drawn from different scenes in Section 4.5. However, before describing these calculations, let us gain some insight into how context features help in classification in a single scene.

We incorporate some elementary contextual knowledge—that which can be learnt from prolonged observation of a scene—in our classification framework in the form of scene-dependent context features. To demonstrate the role played by scene-dependent context features, we performed a pair of experiments with and without these features. Position and direction of motion were used as context features, since their distribution is clearly scene-dependent. We chose two scenes (scenes (a) and (c) in Figure 1-1) having a total of 500 tracked objects and randomly selected 30 labelled objects from each scene as the training sets T_a and T_c for the respective scenes. We trained two SVM classifiers for each scene. Classifiers C_a^s and C_c^s were trained on T_a and T_c respectively using size (in pixels) and speed (*i.e.* magnitude of velocity), but without position or direction of motion, as features, and then tested on other objects from the respective scenes, giving test errors of 9.4% and 3.2%. Classifiers C_a^a and C_c^a were trained on T_a and T_c respectively after including position and direction of motion in the feature-space. The test errors obtained in this case were 0.7% and



(a)

(b)

(c)

Figure 4-1: (a) Scatter plot illustrating spatial distribution of vehicles and persons in scene (a) of Figure 1-1 (which is shown again here in (b) for convenience), in which significant projective foreshortening is evident. (c) Using the y-coordinate as a normalising feature for bounding-box size can greatly improve performance, as demonstrated by the fact that vehicles and pedestrians are clearly distinguishable in the 2D feature space.

0.8% respectively. To show that this trend does not depend on the particular classifier chosen, we repeated the experiments with a generative model-based classifier. The two class-conditional densities were modelled as multivariate gaussians with unknown mean and variance. Inclusion of scene-specific features for scenes (a) and (c) led to reduction of test error by 7% and 2% respectively.

In both cases, the addition of context features to the classifier's feature space led to significant improvement in test performance. There are two reasons for this improvement. Firstly, the chosen context features (position and direction of motion) capture the inherent regularities in structured scenes. For instance, the different spatial distributions of object classes in an urban scene are a result of the scene structure, *i.e.* roads and footpaths. Detecting roads and footpaths automatically and reliably is a hard problem; it is much easier to learn the spatial distribution of object classes from labelled data (as shown in Figure 4-1(a)), and use this for enhancing object classification. In the absence of structural regularities (in an open field, for example), context features would not provide extra information. Fortunately, most urban/highway scenes do exhibit some degree of structure. Thus, even in scenes such as car-parks, where vehicles and pedestrians can in principle occur at the same locations, they tend to have different preferred paths of motion in practice. The second reason has to do with the projective distortion introduced by the camera, as a result of which size and speed of objects are affected by object position. Normalisation of image measurements by correcting for the distortion will help to classify objects reliably. However, normalisation with a single camera is a difficult problem unless some assumptions are made about the scene or camera. Using image position as a feature is a non-parametric way of performing normalisation. This is clearly demonstrated in Figure 4-1(c), where by simply using y-position in the image along with size of bounding-box as object features, and a linear SVM kernel, test error of 3% was obtained for scene (a) considered above.

4.5 Feature Grouping for Multiple Scenes

Feature grouping is necessary to separate scene-specific context features from scene-independent features, so that only the latter are used for training a scene-invariant baseline classifier. Mutual information is once again used as a tool to perform this categorisation. This time, however, the data set used for MI calculations consists of objects taken from a group of different scenes (and not just from a single scene). As a result, one can expect a wide variation in the values of scene-specific features in this data set. This should lead to lowered MI scores for these features. On the other hand, scene-independent features calculated for objects from different scenes should be similar, so that their MI scores will remain (almost) unchanged. In practice, we differentiate between these two groups of features by setting a threshold d_0 on the change in MI between the single-scene data set and the multiple-scene data set. If the change in MI exceeds this threshold, the feature is considered to be a scene-specific context feature. Otherwise, it is considered to be a scene-independent feature.

Mutual information scores between individual features and labels for a multiple-scene data set created from 7 scenes are given in Table 4.4. The corresponding decisions for scene-dependent ('d') or scene-independent ('i') features are also indicated. The threshold used, $d_0 = 0.20$, was chosen after observing the distribution of MI scores. Once again, MI scores for pairs of variables (not shown here) played a role for some features (as indicated by a 'p' in parentheses). The scenes were chosen so as to adequately represent common variations in height, viewing angle and zoom of

Feature	MI	i/d		Feature	MI	i/d
size in pixels ($\mu_{0,0}$)	0.47	d		2nd deriv. ϕ_3	0.61	i
norm. 1st deriv. $\mu_{0,0}$	0.14	d		1st deriv. ϕ_5	0.75	i
norm. 2nd deriv. $\mu_{0,0}$	0.17	d		2nd deriv. ϕ_5	0.74	i
norm. 3rd deriv. $\mu_{0,0}$	0.13	d		$\eta_{3,1}$ (invariant 4th moment)	0.33	i
x-coordinate	0.02	d		$\eta_{2,2}$ (invariant 4th moment)	0.31	i
y-coordinate	0.02	d		$\eta_{1,3}$ (invariant 4th moment)	0.23	i
velocity magnitude	0.35	i		$\eta_{0,4}$ (invariant 4th moment)	0.24	i
velocity direction	0.04	i(p)		1st deriv. $\eta_{4,0}$	0.41	i
principal axis orientation	0.23	i(p)		2nd deriv. $\eta_{4,0}$	0.32	i
2nd deriv. orientation	0.24	i		1st deriv. $\eta_{2,2}$	0.26	i
ϕ_1 (invariant 2nd moment)	0.41	i		1st deriv. $\eta_{1,3}$	0.47	i
ϕ_2 (invariant 2nd moment)	0.20	d		1st deriv. $\eta_{0,4}$	0.32	i
1st deriv. ϕ_1	0.48	d		$\eta_{5,0}$ (invariant 5th moment)	0.42	i
2nd deriv. ϕ_1	0.36	d		$\eta_{4,1}$ (invariant 5th moment)	0.38	i
1st deriv. ϕ_2	0.47	d		$\eta_{3,2}$ (invariant 5th moment)	0.34	i
2nd deriv. ϕ_2	0.40	d		1st deriv. $\eta_{5,0}$	0.41	i
ϕ_3 (invariant 3rd moment)	0.44	i		1st deriv. $\eta_{4,1}$	0.31	i
ϕ_4 (invariant 3rd moment)	0.51	i		1st deriv. $\eta_{3,2}$	0.25	i
ϕ_5 (invariant 3rd moment)	0.60	i		1st deriv. $\eta_{0,5}$	0.23	i
ϕ_6 (invariant 3rd moment)	0.47	i		average bg. intensity	0.21	d
1st deriv. ϕ_3	0.65	i				

Table 4.4: Mutual information (M.I.) scores between object features and labels across multiple scenes, along with decision regarding whether feature is scene-dependent (d) or scene-independent (i). M.I. is measured in bits (maximum possible score = 1.0).

camera. Equal numbers of vehicle and pedestrian observations from each scene were represented in the data sets, thus giving all scenes equal importance and assuming equal populations of the two object classes.

A few of the results deserve special mention. While most of the moment invariants do indeed turn out to be scene-independent, some of the second-moment invariants and their derivatives have been classified as scene-specific. This is because the invariance in question is only with respect to translation, rotation, reflection and scaling. A typical scene change also involves some non-isotropic scaling (*i.e.* affine transformation) due to change in elevation of the camera above the ground plane. Further, most objects have cast shadows, whose effect in a given scene depends on the lighting direction characteristic of that scene. These two factors are fixed for a given scene, but may differ between scenes.

An alternative method for determining the appropriate grouping of features is to calculate mutual information between features and scenes, for a given object class. If a feature has high MI with scenes, it takes on distinct values in different scenes and is thus scene-specific. On the other hand, a feature having low MI with scenes has approximately the same distribution in different scenes and is thus scene-independent. While this method is technically applicable, it requires many more data samples from each scene than the method used above in order to estimate scene-specific distributions needed for MI calculation.

Based on the above grouping of features, we propose to develop transferable classifiers by training them on only the scene-independent features. Most classification methods do not transfer to novel scenes mainly because they use object size, aspect ratio or some feature closely related to these—features that we identify here as scene-specific—in their classifier feature space. Note that even though size has a reasonable score across a group of scenes, the fact that this score is much lower than the score in a single scene implies that it is a rather poor feature for transfer to at least some type of scenes (hence we consider it scene-dependent).

We have shown that scene-specific context features can be used for reducing classification error when training and testing in the same scene. While this in itself is useful for many surveillance applications, we really would like to be able to transfer classifiers across scenes, while still enjoying the benefits of using context features. We propose a scene transfer and adaptation algorithm to do exactly this next.

Chapter 5

Transferring Classifiers across Scenes

Having identified scene-dependent and scene-independent features in the previous chapter, we now describe the main learning algorithm for achieving both scene-transfer and scene-adaptation. Scene transfer can be defined as the process of developing a scene-invariant classifier using training from one (or a few) scenes. The essential requirement is that classification performance should be reasonably high in any scene, irrespective of the position, orientation and scale of objects. Scene adaptation can be defined as the process of improving a baseline scene-invariant classifier in a specific scene by using scene-specific context features. This can be done with the help of unlabelled data.

5.1 Scene Transfer

Scene-dependent features are of no utility in designing a scene-invariant classifier. Thus, the design of a scene-invariant classifier is perhaps obvious: train a classifier using only scene-independent features on a small labelled set of examples from 2 or 3 scenes (or even a single scene). As described in the next section, the classification accuracy of this baseline classifier is around 80%. More importantly, the average posterior probability of the label given an observation is expected to be higher for correctly classified test examples than for incorrectly classified ones in a new scene. Our scene-adaptation algorithm is now employed to improve classification performance in a novel scene by using unlabelled data to incorporate information about scene-dependent features.

5.2 Scene Adaptation

We propose the following novel decision-directed learning [12] algorithm for scene-adaptation:

1. Apply the baseline classifier, C_{base}, to the new scene, to find 'labels', L_1, for unlabelled examples along with associated posterior probabilities.

2. Convert posterior probabilities to confidences by shifting and scaling. Probabilities of 0.0 and 1.0 are mapped to confidences of -100% and 100% respectively. (The 'label' of a single unlabelled instance, i.e., the sign of the posterior probability of that observation given a class, can be different from the overall 'label' for the unlabelled object, as indicated by a negative confidence value.)

3. Train a scene-specific classifier using only the scene-independent features on both the original labelled examples L_0 and the 'labels' L_1 generated for unlabelled data, after making two changes:

   - For each unlabelled object, the 5% least confident instances are removed from further consideration (to afford some robustness to gross outliers). Each remaining instance is then assigned the same confidence value, equal to the mean confidence of these instances.

   - The bound on the Lagrange multiplier (in the SVM formulation: see Equation 3.6) for each training instance from an unlabelled object is set as $C_i = 1000 \times$confidence value.

   This step produces a partially-adapted classifier, C_{part}, which does not yet use scene-specific context features.

4. Apply C_{part} to the unlabelled data to generate a second set of 'labels', L_2, and associated confidences for unlabelled data.

5. Repeat step 3, using both scene-dependent and scene-independent features, but only the 'labels' in L_2, to obtain a fully-adapted classifier, C_{full}.

In step 3, the 5% cutoff was chosen by observing the distribution of object confidences and the corresponding object silhouettes.

Since, by definition, true labels are not available for unlabelled data, there is a trade-off between using examples that are classified with high confidence (*i.e.* far from the decision boundary, and thus not very informative) and using ones that are labelled with low confidence (*i.e.* close to the boundary and thus more informative, but also more likely to be incorrect). A meaningful balance is obtained by varying the Lagrange multipliers for 'labelled' instances in proportion to the corresponding confidences. A large Lagrange multiplier heavily penalises an incorrect classification of the corresponding training example. Thus, our algorithm is able to allow points near the classification boundary (of the baseline classifier) to modify the adapted solution slightly, without letting incorrect (but low-confidence) 'labels' significantly disrupt the training process. The equalisation of confidences within a tracking sequence is done to avoid using incorrect instance labels with high confidence for retraining. The underlying assumption is that incorrectly classified objects for the baseline classifier will have lower confidence on average than correctly classified ones.

The classifier needs to be adapted in two steps (3 and 5) because the distribution of scene-dependent features in the labelled and unlabelled data can be completely unrelated. Also, a two-step process gradually removes the information provided by true labels (from training scenes) and increases reliance on the (uncertain) information provided by the new scene. In the next section, we give experimental results to illustrate the effectiveness of this approach.

Chapter 6

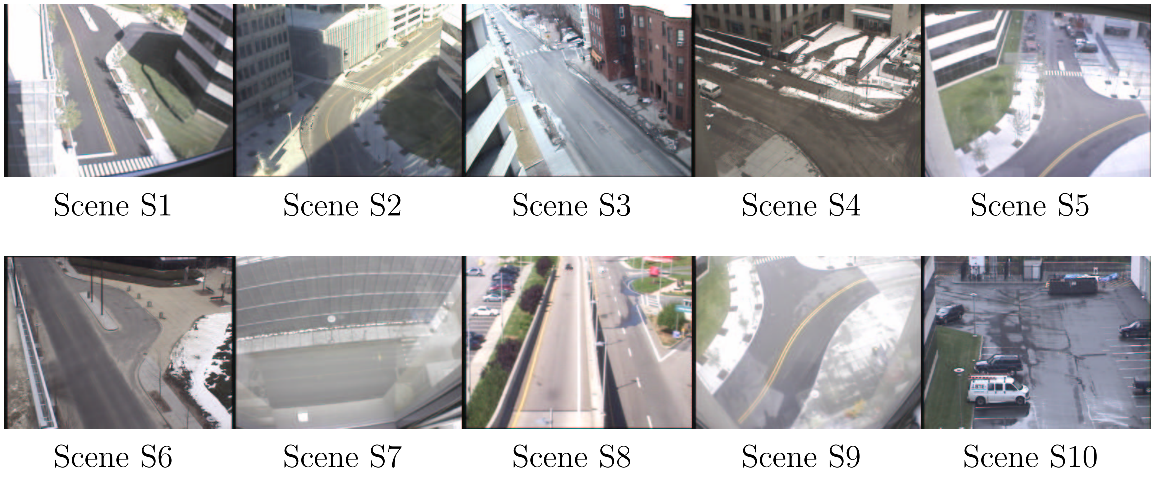
Experimental Results

In this chapter, we describe the series of experiments performed for testing our classification algorithms, both within a single scene and across multiple scenes. This is accompanied by an analysis of the results obtained.

6.1 Selection of Experimental Data

We used a data set of more than 1500 object tracks from ten different scenes (shown in Figure 6-1) for testing our algorithms. The scenes used for our experiments are shown in Figure 6-1. These scenes were chosen after studying a large number of far-field surveillance videos available on the Internet, to try and accurately represent the range of camera positions, orientations and zoom factors that are found in real situations. The video sequences from the chosen scenes were captured during daylight hours (in order to achieve good tracking) at different times of the year. Object class distributions range from highway scenes containing only cars to scenes containing three times as many pedestrians as cars. Cast shadows and reflections are present in some scenes. Video was captured and processed at the rate of about 8 frames per second at a resolution of 320×240 pixels. The typical size of a pedestrian was 10×25 pixels, while that of a vehicle was 30×60 pixels.

The object tracks were obtained from real-time tracking data, and corresponded to over 5 hours of tracking (spread across 10 scenes). After filtering out clutter, we were



Scene S1 Scene S2 Scene S3 Scene S4 Scene S5

Scene S6 Scene S7 Scene S8 Scene S9 Scene S10

Figure 6-1: The full set of scenes used in our experiments

left with 1737 object tracks. 26 of these were actually clutter that was not filtered out; these were removed from the database manually. 194 objects corresponding to groups of pedestrians, bicyclists or gross tracking errors (such as two objects tracked as one over many frames) were also removed manually. However, many tracks containing less severe errors (such as two objects temporarily merging, or an object temporarily merging with the background) were left in the database.

The database of tracks was then split up as follows. Model selection to fix the bandwidth of the SVM Gaussian kernel was carried out on a set of 50 objects. The mutual information calculations described in Chapter 4 were performed on a separate set of 80 objects from 7 scenes. The remaining object tracks were used for training and testing.

6.2 Classification Performance

To test our algorithms, we performed two types of classification experiments:

- Without scene transfer: training on 30 objects, testing on 150 objects in the same scene.

- With scene transfer/adaptation: training on 30 objects from 2-3 scenes, testing on 150 objects in a new scene.

For each training set considered, object features were first calculated for all (independent) observations in each track, giving rise to n_1 positive and n_2 negative samples. Then, an equal number n_3 of samples from each class were randomly chosen from these samples (where $n_3 < n_1$ and $n_3 < n_2$, and was typically around 300). For each test set considered, object features for all (independent) observations were used.

Average classification errors for the above two types of experiments (averaged over 5 trials) are as follows:

- Without scene transfer, using only scene-independent features: 5.9%
- Without scene transfer, using both scene-dependent and scene independent features: 0.3%
- With scene transfer, using only scene-dependent features (baseline classifier): 11.8%
- With scene transfer and scene adaptation (using both types of features): 5.6%

The average reduction in error due to adaptation with the help of scene-specific context features is thus 6.2%.

For the sake of comparison, we also tried training classifiers on both scene-dependent and scene-independent features from one scene and then testing using these features in a different scene. The average classification error in this case was 26%. Thus, as expected, the classifier was mislead by the scene-dependent features in its training set.

We present a detailed analysis of one scene-transfer/adaptation experiment. The labelled set T_L, used for training the baseline classifier C_{base}, consisted of 30 objects (17 vehicles and 13 persons—300 instances from each class) from scenes S4 and S5 shown in Figure 6-1. This baseline classifier was applied to a novel scene, S1. Of the 150 test objects in this new scene, the assigned labels $\hat{L}_1$ for 133 objects were correct. Thus, test error (after scene transfer, but without adaptation) was 11.3%. This is comparable to some existing classification systems which are trained and tested on the same scene. The average confidence for vehicle labels was 47%, while that for persons

	a	b	c	d	e
Labelled training set scene	$(S1)$	$(S1)$	$(S4/S5)$	$(S4/S5)$	None
Unlabelled 'training set' scene	None	None	None	$(S1')$	$(S1'')$
Test set scene	$(S1)$	$(S1)$	$(S1)$	$(S1)$	$(S1)$
Transfer ?	No	No	Yes	Yes	Yes
Adaptation ?	No	No	No	Yes (Partial)	Yes (Full)
Type of features	S.I.	Both	S.I.	S.I.	Both
% Test error	8.4	0.4	11.3	9.3	6.2

Table 6.1: Performance evaluation for scene $S1$ (Figure 6-1): test errors using various classifiers and features. 'S.I.' = scene-independent features. 'Both' = scene-dependent + scene-independent features. Labels for $S1'$ are produced in step c, and those for $S1''$ in step d.

was 37% (note that 0% represents no confidence, as it corresponds to a posterior probability of 0.5). Average confidence for correct labels was 51%, while that for incorrect labels was 24%. This difference in confidences is because the range of feature variation among persons is much less than the corresponding range among vehicles. In the scene adaptation process, bounds on the Lagrange multipliers were varied according to the average object confidences, as described in Section 5.2. After partial adaptation, test error (using only scene-independent features) decreased to 9.3%. After full adaptation, the error further decreased to 6.2%. Thus, our bootstrapping technique resulted in a performance boost of about 5% for this particular scene.

The above results are summarised in Table 6.1. For comparison, the results of training scene-independent and scene-specific classifiers on a labelled set T_1 taken from scene S1 itself are also repeated here. As expected, best classification results are obtained by training on T_1, and using both scene-dependent and scene-independent features. The fully-adapted classifier, working with both types of features, demonstrates a significant improvement over the baseline classifier, and even the partially-adapted classifier. This is because of the significant projective distortion evident in this scene, as well as the characteristic spatial distribution of vehicles and pedestrians. The resulting classification performance is better than simply using scene-independent features for training in scene S1 itself.

6.3 Performance Analysis

It is useful to analyse some aspects of performance improvement from the use of scene-specific context features. In this section we seek to address the following issues:

- How much of the improvement upon adding position as a feature comes from its role in normalising projectively distorted features (such as size and velocity) and how much from the inherent spatial regularity present in the scene?

- How would performance be affected by changing the thresholds used in selecting and grouping features?

- What are the failure cases of the current algorithm?

6.3.1 Role of Position as a Context Feature

Position is a useful context feature both because of spatial regularities in real world scenes (such as cars staying on roads and pedestrians on footpaths) and because of the implicit normalisation of projectively distorted features achieved by including position in the feature space. To estimate the relative importance of these two factors, we performed three tests using a hand-picked subset of features. In all cases, training and testing were performed in the same scene. In the first test, size and speed were the only features used, leading to a test error of 9%. In the second test, size, speed and position (x- and y-coordinates) were used together, leading to test error of 0.5%. Before performing the third test, projective distortion was corrected for in the test scene. This was done by manually specifying a number of point-coordinates for rectifying the ground plane up to a similarity transformation from the real world plane [16]. Finally, for the third test, size, speed and position were calculated in the rectified scene and then used for classification, resulting in a test error of 1.4%. Thus, we conclude that 7.6% of the improvement in performance came from the effect of normalisation, while 0.9% was due to the spatial regularity of object paths in the scene.

It is interesting to note that in some scenes, such as one in which objects recede from the camera along parallel lines, one of the two position coordinates only provides normalisation information, while the other only provides class-specific spatial information.

6.3.2 Choice of Thresholds for Feature Selection and Grouping

We varied the thresholds used for performing feature selection and grouping (Sections 4.4 and 4.5) by 25% in either direction and studied the effects on choice of features and eventually classification performance. The two extreme cases involved selection of 12 more and 10 fewer features. Average classification errors, both without and with scene transfer, varied by a maximum of about 1%.

6.3.3 Failure Cases

In general, cases where the original scene-specific classifier (before transfer) fails include occlusion (as the object leaves the scene), objects that are consistently far away from the camera (and hence have sizes of around 10×5 pixels, producing very noisy features), objects whose silhouettes were not complete (due to similarity in reflectance from foreground and background) and objects that were merged with one another while tracking. Additional cases where the transferred and adapted classifier fails (but the original classifier works) include objects that show large feature variations as they move through the scene, or scenes with prominent shadows.

Chapter 7

Conclusion and Discussion

We have proposed a system for far-field object classification from video sequences that addresses some of the significant challenges in the field. Use of a discriminative (SVM) instance-classifier on simple object descriptors, along with a probabilistic method for combining instance confidences into object labels, allows for very low classification error (less than 1%) using only a small number of objects. The concept of scene-specific features, as well as some new features (position/direction of motion) are introduced and shown to benefit classification. Scene-independent and scene-dependent features are identified using mutual information estimates in order to design scene-invariant classifiers. At the same time, a decision-directed learning algorithm has been proposed to adapt classifiers to scene-specific characteristics by carefully using unlabelled data. Our scene-invariant classifier has over 85% accuracy; an further improvement of about 6% is obtained by using our scene-adaptation algorithm.

Though our scene-transfer algorithm has been developed using a support vector machine classifier, the concepts of using scene-specific context features and learning from unlabelled data to boost classification performance are independent of the choice of classifier.

It should be noted that selection of context features is task-dependent. For instance, if all the scenes considered have approximately the same scale, it makes sense to use object size as a scene-independent feature for training the baseline classifier. In fact, this was the case when we first experimented with tracking data from a smaller set of scenes [5]. Our intention here is to provide a principled mechanism to be able to decide on which features are scene-dependent. The key to correctly applying this mechanism to a specific problem lies in selecting a representative data set (that suitably models the variation across scenes) for mutual information calculation.

In future, we would like to extend the classification framework to other object classes (*e.g.* groups of people) or sub-classes (*e.g.* cars, vans and trucks). A hierarchical approach is probably best for this purpose, wherein groups of people are classified as people at the first level, and later identified as a separate class.

Classification will certainly benefit from improvements in tracking (such as fewer holes in objects). Best possible results will probably be obtained by integrating object tracking and classification, whereby knowledge of object class is fed back into the tracking system to help locate the object in the next frame.

Bibliography

[1] I. Balslev, K. Doring, and R.D. Eriksen. Weighted central moments in pattern recognition. *Pattern Recognition Letters*, 21:381–384, 2000.

[2] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, Inc., 1988.

[3] D. Blei, J. Bagnell, and A. McCallum. Learning with scope; with application to information extraction and classification. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2002.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. Conf. on Computational Learning Theory*, 1998.

[5] B. Bose and E. Grimson. Learning to use scene context for object classification in surveillance. In *Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003.

[6] F. Bremond and M. Thonnat. Issues in representing context illustrated by scene interpretation applications. In *Proc. of the International and Interdisciplinary Conference on Modeling and Using Context (CONTEX'97)*, February 1997.

[7] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

[8] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley and Sons, 1991.

[9] Ross Cutler and Larry Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781 –796, 2000.

[10] C.P. Diehl. *Towards Efficient Collaborative Classification for Distributed Video Surveillance.* PhD thesis, Carnegie Mellon University, 2000.

[11] C.P. Diehl and J.B. Hampshire II. Real-time object classification and novelty detection for collaborative video surveillance. In *Proceedings of the International Joint Conference on Neural Networks*, volume 3, pages 2620–2625, 2002.

[12] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* Wiley-Interscience, 2001.

[13] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[14] G.L. Foresti. Object recognition and tracking for remote video surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(7):1045–1062, October 1999.

[15] W.E.L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* The MIT Press, 1990.

[16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2001.

[17] J.E.L. Hollis, D.J. Brown, I.C. Luckraft, and C.R. Gent. Feature vectors for road vehicle scene classification. *Neural Networks*, 9(2):337–344, 1996.

[18] M.K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8(2):179–187, 1962.

[19] A.K. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, N.J., 1989.

[20] O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *Proceedings European Conf. on Computer Vision*, 2002.

[21] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. 16th Intl. Conf. on Machine Learning*, 1999.

[22] Takeo Kanade, Robert Collins, Alan Lipton, Peter Burt, and Lambert Wixson. Advances in cooperative multi-sensor video surveillance. In *Darpa Image Understanding Workshop*, pages 3–24. Morgan Kaufmann, November 1998.

[23] Z. Kim and J. Malik. Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In *Proceedings of the International Conference on Computer Vision*, 2003.

[24] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *Proceedings of the International Conference on Computer Vision*, 2003.

[25] A.J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real-time video. In *IEEE Workshop on Applications of Computer Vision*, 1998.

[26] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):873–889, August 2001.

[27] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 193–199, 1997.

[28] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1), 2000.

[29] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, and S. Harp. Urban surveillance systems: From the laboratory to the commercial world. *Proceedings of the IEEE*, 89(10):1478–1497, 2001.

[30] J.C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Smola et al., editor, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[31] M. Pontil and A. Verri. Support vector machines for 3d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.

[32] H. Schneiderman and T. Kanade. A statistical model for 3d object detection applied to faces and cars. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[33] A. Sluzek. Identification and inspection of 2-d objects using new moment-based shape descriptors. *Pattern Recognition Letters*, 16:687–697, 1995.

[34] C. Stauffer. Minimally-supervised classification using multiple observation sets. In *Proceedings of the International Conference on Computer Vision*, 2003.

[35] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

[36] M.R. Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70:920–930, 1980.

[37] A. Torralba and P. Sinha. Statistical context priming for object detection. In *Proceedings of the International Conference on Computer Vision*, pages 763–770, 2001.

[38] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proceedings of the International Conference on Computer Vision*, 2003.

[39] Y. Wu and T.S. Huang. View-independent recognition of hand postures. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.