# BACKGROUND MODELLING AND PERFORMANCE METRICS FOR VISUAL SURVEILLANCE

LAZAREVIC, N.

PHD

2011

## Abstract

This work deals with the problems of performance evaluation and background modelling for the detection of moving objects in outdoor video surveillance datasets. Such datasets are typically affected by considerable background variations caused by global and partial illumination variations, gradual and sudden lighting condition changes, and non-stationary backgrounds. The large variation of backgrounds in typical outdoor video sequences requires highly adaptable and robust models able to represent the background at any time instance with sufficient accuracy. Furthermore, in real life applications it is often required to detect possible contaminations of the scene in real time or when new observations become available.

A novel adaptive multi-modal algorithm for on-line background modelling is proposed. The proposed algorithm applies the principles of the Gaussian Mixture Model, previously used to model the grey-level (or colour) variations of individual pixels, to the modelling of illumination variations in image regions. The image observations are represented in the eigen-space, where the dimensionality of the data is significantly reduced using the method of the principal components analysis. The projections of image regions in the reduced eigen-space are clustered using K-means into clusters (or modes) of similar backgrounds and are modelled as multivariate Gaussian distributions. Such an approach allows the model to adapts to the changes in the dataset in a timely manner.

This work proposes modifications to a previously published method for incremental update of the uni-modal eigen-models. The modifications are twofold. First, the incremental update is performed on the individual modes of the multi-modal model, and second, the mechanism for adding new dimensions is adapted to handle problems typical for outdoor video surveillance scenes with a wide range of illumination changes.

Finally, a novel, objective, comparative, object-based methodology for performance evaluation of object detection is also developed. The proposed methodology is concerned with the evaluation of object detection in the context of the end-user defined quality of performance in complex video surveillance applications.

# Contents

# Chapter 1

# Introduction

Computer vision applications are concerned with the machine understanding of the content of video scenes without human intervention. Typically, video scene understanding in a visual surveillance context involves a number of tasks such as: preprocessing, foreground detection, object recognition, classification, tracking and event detection. Specifically, these applications are often concerned with the detection of the moving objects in the observed scene, the interaction of these objects with each other and the background. These tasks generally rely on the efficient separation of the foreground from the background, which is usually based on an accurate model of the background scene. This work addresses the problem of performance evaluation of foreground detection algorithms and the background modelling for visual surveillance applications.

## 1.1 Performance evaluation of object detection algorithms

The visual surveillance research field has produced a great number of background modelling algorithms for a variety of applications. Some authors proposed new approaches to the problem while other contributions are useful modifications to existing methods. Implementations of a number of the most popular techniques are now available as off-the-shelf algorithm components, which can be easily integrated into algorithms where a background modelling step is required. However, despite such a large number of available methods, the choice of the most appropriate technique is not always straight forward. Different background modelling techniques promote different features of a visual surveillance algorithm, which in turn has an impact on the algorithm's final performance. Also, the same algorithm may not perform equally in different end-user applications. Furthermore, in real-life applications, a trade-off between desired performance metrics is often inevitable. The final performance of the algorithm requires special consideration when choosing a background modelling method.

Despite considerable effort to develop appropriate methods for the evaluation of background modelling, most evaluation techniques described in the literature do not address the complexity and range of the issues which underpin the design of a good evaluation methodology. Initial contributions largely focused on the provision of suitable datasets and the accompanying ground truthing tools. More recently however, the enormous number of papers on evaluation metrics has exposed the considerable difficulties involved

in establishing an accepted method of ranking competing algorithms - particularly for end-users, technology integrators and governmental agencies. Most contributions have embraced pixel-based methods within a *receiver-operating curve* (ROC) framework. However, such pixel-oriented measures do not necessarily characterise the impact of the motion detection stage on the subsequent processing tasks *e.g.* object tracking. Moreover, the production of pixel-accurate ground truth (necessary for accurate results) is extraordinarily time-consuming and difficult to manually produce. As a consequence, a large number of object-based metrics have been proposed. However, as it is not possible to apply the ROC methodology to object-based metrics, the comparison of algorithms becomes a subjective interpretation of vectors of different metrics.

This work discusses the problems associated with and proposes a novel methodology for the performance evaluation of background modelling and motion detection for visual surveillance applications. It explores the problems associated with both optimising the operating point of motion detection algorithms and the objective performance comparison of competing algorithms. In particular, an object-based approach is developed based on the *F-Measure* - a single-valued ROC-like measure which enables a straight-forward mechanism for both optimising and comparing motion detection algorithms. Despite the advantages over pixel-based ROC approaches, a number of important issues associated with parameterising the evaluation algorithm need to be addressed. The approach is illustrated by a comparison of three motion detection algorithms, including the well-known Stauffer and Grimson algorithm, based on results obtained on two datasets.

## 1.2 Background modelling for fixed cameras and grey-scale video

This work considers fixed cameras and grey-scale surveillance videos in which the background is largely static and features of interest are moving. To detect these moving objects it is necessary to build a model of the static background.

Modelling of video sequence backgrounds can be treated as a high dimensional data space where behaviour of variables representing image pixels is observed in time and space. As a result, efficient background modelling is computationally costly and difficult to perform in real time applications.

In surveillance applications outdoor video sequences are often affected by considerable background variations caused by global and partial illumination variations, gradual and sudden lighting condition changes, and non-stationary backgrounds. The large variation of backgrounds in typical outdoor video sequences requires highly adaptable and robust models able to represent the background at any time instance with sufficient accuracy. Furthermore, in real life applications it is often required to detect possible contaminations of the scene in real time or when new observations become available. Not surprisingly, despite considerable research effort, the problems of accurate and efficient background modelling and motion detection for visual surveillance applications have yet to be solved. This work discusses the possibility of and proposes a method to reduce the complexity of the problem of background modelling by extracting a sufficient amount of information about the behaviour of the available dataset from as little amount of data as possible. The extracted amount of information should be sufficient to ac-

curately model the variability in the dataset. Furthermore, a novel adaptive multi-modal modelling approach is proposed.

The aim of the proposed background modelling method is the classification of observed images as background-only or contaminated with foreground. The problem is addressed by applying eigen-theory and principal components analysis (PCA) on a high-dimensional space of a typical video surveillance recording of an outdoor scene. The PCA is based on the notion that a high-dimensional data vector can be represented by a space of a small number of orthogonal eigen-vectors (or principal components) which form a low-dimensional subspace. As the background structure remains constant over time, and as lighting changes are typically correlated over the image, variations in the pixel variables usually inhabit small subspaces within this very high-dimensional space. The methods explored in this work deal with appropriate choices of the minimum amount of the original data and the lowest number of dimensions required to obtain results which remain within the limits of an acceptable quality loss. Furthermore, considering the nature of the outdoor scenes with changing weather conditions, it is expected that background observation points in the reduced eigen-space would gather in clusters of backgrounds of similar lighting conditions. These clusters represent the *modes* of the background model. A *multi-modal* eigen-model is proposed for improved accuracy of the classification of image observations. Furthermore, both *off-line* and *on-line* modelling are considered. An off-line methodology refers to those background modelling techniques applied to datasets when all of the available data is known and fixed, and all possible variations of the scene background are already contained within the train-

ing set. An on-line method refers to an adaptive technique which allows for merging the information about new observations with the existing knowledge about the video sequence background.

### 1.2.1 Developments to the off-line standard method

The PCA and eigen-models have been previously used to model backgrounds in algorithms reported in the literature. However, the methods for dimensionality reduction of real-life datasets and the multi-modal nature of the outdoor video data have not been sufficiently explored. There are several contributions of the methodology presented in this work. First, in order to ensure the correlation of lighting changes and facilitate a more manageable processing, the original image frame is divided into smaller regions which are individually observed. Second, an informal dimensionality reduction technique previously suggested in the literature is validated on an example of a real-life outdoor surveillance dataset. Third, a subsampling approach is discussed as a possibility for further reduction in the processed amount of data. Fourth, the multi-modal background model is explored and proposed as an improved solution for foreground detection and the classification of image observations.

### 1.2.2 Adaptive multi-modal background modelling

An on-line adaptive method is developed in order to provide an accurate model of the background at any time and adapt to any changes in a timely manner. The proposed adaptive algorithm evolves incrementally with each

new observation. At every time instance it adapts to the new background conditions using the knowledge of the newly acquired data and the accumulated knowledge of the current model. The update strategy is inspired by two previously reported methods: the incremental method of Hall et al. applied to the eigen-space update [Hall et al., 1998], and the improved Gaussian mixture model (GMM) approach [KaewTraKulPong and Bowden, 2001], applied to adaptive learning.

The novelty of the method proposed in this work is as follows. First, the GMM method is generally used to model individual pixel variations by one dimensional Gaussian distributions. This principle is modified and applied to image regions rather than pixels, where image observations are modelled with a mixture of multi-dimensional Gaussian clusters in the eigen-space. Second, the incremental method of Hall et al. was previously used to update the unimodal eigen model which is not suitable for modelling a wide range of varied backgrounds in outdoor surveillance scenes. The incremental principle is modified and applied to a multi-modal model consisting of clustered eigen-subspaces rather than a single unimodal eigen-space. Third, the method of Hall et al. for adding new dimensions to the updated eigen-space is not appropriate for modelling outdoor datasets where the successive observations may significantly vary. A more suitable method, which preserves low dimensionality is proposed.

## 1.3 Summary

This thesis sets out a framework for a new methodology for objective and principled performance evaluation of motion detection algorithms, and proposes a novel algorithm for the adaptive multi-modal eigen-modelling of image backgrounds. The remainder of the work is organised as follows. Chapter 2 offers a review of the published performance evaluation techniques and background modelling algorithms. In Chapter 3 a new objective comparative methodology for performance evaluation is presented. Its application is illustrated with a comparison of three motion detection algorithms. Chapter 4 introduces the problem of modelling in high-dimensional spaces. The PCA by eigen-decomposition is described in detail and used to represent two real-life datasets in the eigen-space of reduced dimensionality. Furthermore, the multi-modal approach is introduced and compared with the unimodal approach when the classification of image observations is performed off-line. Chapter 5 explains the concept of the adaptive incremental multi-modal background modelling. The proposed algorithm is applied to an outdoor video surveillance sequence. The results are discussed and compared with those of an unimodal on-line approach. Finally, Chapter 6 concludes the work offering a critical look at the proposed algorithm with a discussion on its advantages and disadvantages.

# Chapter 2

# Review

## 2.1 Introduction

In many computer vision applications it is essential to separate pixel regions of interest from the rest of the image. Typically, moving objects are segmented from the background scene by subtracting the original image from the background model image and thresholding the difference. The image regions where the pixel difference is above the threshold indicate the presence of foreground objects. The foreground pixels are further processed for object segmentation, localization and tracking.

For an effective object detection it is necessary to provide an accurate model of the background. The background modelling techniques must deal with several challenging issues. Most common challenges, when the fixed camera is used, are gradual and sudden background illumination changes, in both indoor and outdoor scenes, and the non-stationary background, mostly in the outdoor scenes.

There has been an abundance of various background modelling techniques described in literature. However, the problem of accurate background modelling is yet to be solved. The proposed techniques produce variable results depending on the end-application and the type of the dataset. Often, the results depend on the optimization of various parameters for a particular application or dataset case. Furthermore, in many applications there is an obligatory trade-off between different aspects of good performance of the background modelling.

The background modelling approaches reported in the computer vision literature differ significantly. Some deal with different camera set-ups such as multiple cameras [Cai and Aggarwal, 1996; Mittal and Davis, 2001] or moving cameras [Burt et al., 1989; Everts et al., 2007] for object tracking applications. Other approaches treat colour images in different colour spaces [Orwell et al., 1999; Kumar et al., 2002] mainly for suppression of shadows of moving objects. The methods based on multiple camera setting and colour images improve performance for certain applications. However, the cost of improvements is often a significant increase in complexity.

### 2.1.1 Assumptions

Visual surveillance techniques based on the fixed camera assume that the background is largely static and that the features of interest are characterised by motion. Therefore, an accurate modelling of the background is crucial for efficient detection of foreground moving objects. In surveillance applications the fixed cameras are often chosen over pan-tilt-zoom (PTZ) cameras because

of the low cost and low system complexity. The PTZ camera systems do offer additional functionality by allowing operators to control the view point. However, in most applications the low cost solution of a single fixed camera is largely sufficient.

Section 2.2 looks at the background modelling techniques which handle fixed camera and the methodologies for their evaluation.

## 2.2 Review of evaluation approaches

A number of techniques have been proposed to evaluate performance of visual surveillance algorithms. Many of them deal with the evaluation of detection of moving objects [Villegas and Salcedo, 1999; Correia and Pereira, 2000; Erdem and Sankur, 2000; Gao et al., 2000; Stefano et al., 2001; Cavallaro et al., 2002; Mariano et al., 2002; Gelasca et al., 2004; Nascimento and Marques, 2004; Villegas and Marichal, 2004; Aguilera et al., 2005; Hall et al., 2005; Walk et al., 2010], whereas others address evaluation of the tracking of detected objects throughout the video sequence or both [Jaynes et al., 2002; Black et al., 2003; Georis et al., 2003; Erdem et al., 2004; Schlogl et al., 2004; Wu and Zheng, 2004; Thirde et al., 2005; Denman et al., 2009; ?]. Since successful tracking relies heavily on accurate object detection, the evaluation of object detection algorithms within a surveillance systems plays an important part in overall performance analysis of the whole system.

The types of evaluation approaches reported in the literature are illustrated by the graph in Figure 2.1. (Discussion on evaluation metrics is postponed to Section 3.3).

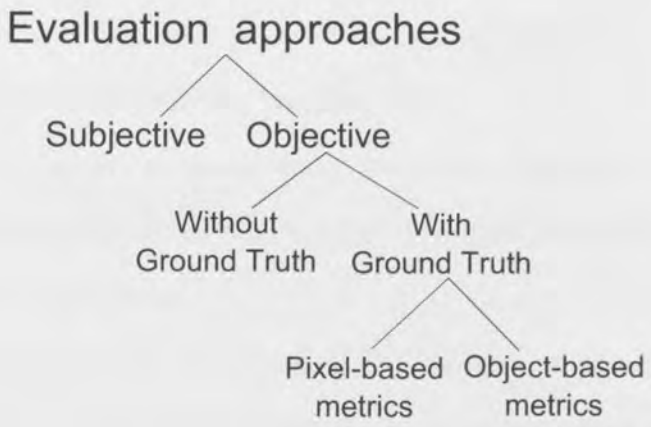



Figure 2.1: Evaluation approaches

Most broadly, object detection evaluation methods can be classified as *subjective* and *objective* methods. Subjective methods evaluate performance of detection relevant to human visual perception. Objective methods use automated tools to quantitatively describe the quality of performance.

Gelasca et al. aim to characterise segmentation errors from a perceptual point of view by subjective evaluation of segmentation degradation as a result of spatial errors such as missing and/or added background and temporal instability perceived as flickering of segmented regions over time [Gelasca et al., 2004]. Villegas and Marichal propose a perceptive weighting of computed objective quantitative measure parameters by looking at the visually desirable properties of a segmentation mask [Villegas and Marichal, 2004; Villegas and Salcedo, 1999]. The weighting reflects the visual relevance of segmentation errors. The main problem with subjective methods is that the evaluation outcome relies on the subjective judgment of a group of human viewers. Furthermore, the representative sample of viewers needs to include a large number of individuals for valid statistical analysis which can be a costly task. The subjective evaluation methods may be suitable for

applications such as film post production or augmented reality but not for surveillance applications such as tracking.

Objective evaluation methods make use of automatic tools for computation of quantitative measures which allow objective comparison of different object detection algorithms.

In the absence of Ground Truth (GT), a number of evaluation methods rely on the expected consistency of an object's appearance and motion throughout an image sequence. By doing so, they often fail to account for sudden changes in scene lighting, motion direction or velocity of the moving objects. Erdem and Sankur propose metrics based on colour and motion consistency along the segmented object boundary throughout the sequence [Erdem and Sankur, 2000]. However, the assumption that pixels which lie inside the object boundary and those outside the boundary vary significantly in colour and motion features may not be true. For example, in outdoor scenes with lighting variations this will cause failure to distinguish between the object and its shadow. Furthermore, the boundary of the object may not be available due to occlusions. Wu and Zheng avoid this problem by looking at bounding boxes rather then an object's contours [Wu and Zheng, 2004]. They propose metrics based on appearance and trajectory consistency, offering a method for self-evaluation and initiating automatic re-initialisation of a tracking algorithm when the tracker's performance falls below some threshold quality.

Evaluation in the absence of GT may provide a tool for self-assessment and feedback of performance measurements in order to improve tracking quality of the algorithm of interest. On the other hand, evaluation based

on GT [Mariano et al., 2002; Black et al., 2003; Erdem et al., 2004; Jaynes et al., 2002; Schlogl et al., 2004] offers a framework for objective comparison of the performance of alternate surveillance algorithms. Such evaluation techniques compare the output of the automatic object detector with the GT obtained manually by drawing bounding boxes around objects, or marking-up the pixel boundary of objects, or labeling objects of interest in the original video sequence. Manual generation of GT is an extraordinarily time-consuming and tedious task and, thus, inevitably error prone even for motivated researchers. (See the work of List et al. for an interesting study on inter-observer variability in this context [List et al., 2005].) Black et al. proposed the use of a semi-synthetic GT where previously segmented people or vehicles are inserted into real video sequences [Black et al., 2003]. The significant properties of synthetic GT objects (size, shape, position, velocity, trajectory) are known precisely to eliminate effect of subjectivity and human error.

Interpretation of evaluation results is obviously based on the type of GT used for comparison, and established standards for GT are emerging. There are several ambiguities involved in the process of GT generation. For example, whether to account only for individual objects or also for groups of objects? Whether to look at the bounding boxes or exact shapes of objects? When is an object considered fully occluded or semi-occluded? Is the GT biased if produced by one person only? And so on. Several GT generation tool are available: ViPER [Mariano et al., 2002], ODViS [Jaynes et al., 2002].

A summary and a comparison of most representative object detection evaluation methods is given in Figure 2.2.

| Approach | Interpretation | GroundTruth | Evaluation metrics |
|---|---|---|---|
| Erdem and Sankur, 2000 | objective | no | colour and motion consistency along the segmented object boundary |
| Correia and Pereira, 2000 | objective | yes | similarity measure (shape, statistical similarity and size) |
| Mariano et al., 2002 | objective | yes | fragmentation and merging |
| Jaynes et al., 2002 | objective | yes | similarity measure (relative position) |
| Cavallaro et al., 2002 | objective | yes | human perception of error in detection (temporal effects of surprise and fatigue) |
| Black et al., 2003 | objective | yes | detection rate |
| Gelasca et al., 2004 | subjective | no | perception of spatial errors and temporal instability |
| Villegas and Marichal, 2004 | subjective | no | perceptive weighting of visually desirable properties |
| Wu et al., 2004 | objective | no | appearance and trajectory consistency |
| Schlogl et al., 2004 | objective | yes | pixel misclassification rate |
| Hall et al., 2005 | objective | yes | overlap threshold, similarity measure (relative position, shape, statistical similarity and size) |
| Nascimento and Marques, 2006 | objective | yes | fragmentation and merging of detected objects |

Figure 2.2: Object detection evaluation methods - a comparison

### 2.2.1 The choice of the dataset for evaluation

Standardisation of datasets has been championed by CAVIAR [Fisher] and PETS[1]. Nationally funded initiatives have also produced datasets including the French *ETISEO* project[2] and the UK Home Office *iLIDS* project[3].

The choice of the dataset inevitably affects the results of the evaluation. The algorithm which performed best on one type of dataset may not be the best choice when a different type of dataset is used. The main characteristics of evaluation datasets are: the density of foreground objects in the scene, outdoor/indoor lighting setting, the choice of the scenarios performed by the foreground objects, the time lapse between frames (full frame rate or longer), the length of the sequence in real time (few minutes, hours, days or longer).

This work focused on the background modelling in outdoor surveillance sequences with a great range of lighting changes due to weather conditions. To fully understand the problems arising form such a dataset it was necessary to observe changes in the data over a long period of time. At the time of writing none of the existing outdoor datasets was suitable being either too short or lacking complex lighting variations. Therefore two new dataset were created, Kingston Carpark and Inrets datasets, as described in Sections 4.5.1 and 4.5.2 respectively. The first dataset contains a variety of lighting changes and covers few minutes recorded at full frame rate. The second dataset was ultimately used because it covers a longer interval of two and a half months with a large range of lighting conditions from bright sunshine to heavy rain.

[1] www.cvg.cs.rdg.ac.uk/slides/pets.html

[2] `www.silogic.fr/etiseo/`

[3] http://scienceandresearch.homeoffice.gov.uk/hosdb/news-events/270405

### 2.2.2 Performance evaluation metrics

Evaluation methods reported in literature have proposed a great number of performance metrics. In this work the focus is on the metrics used to compare the results of motion detection with the ground truth. A brief overview of these metrics is presented below, while a more detailed description is given later in Section 3.3.

Typically, the GT based methods use *true positives* (TP), *false positives* (FP), *false negatives* (FN), and *true negatives* (TN). These metrics are referred to as the *common metrics.* A number of other frequently used metrics are derived from this set of common metrics, such as: *true positive rate* (or *detection rate*) $t_p$, *false positive rate* $f_p$, *false alarm rate* $f_a$ and *specificity* $s_p$. In general, the metrics may be *pixel-based* or *object-based.* In the case of pixel-based evaluation methods, these metrics are calculated on the pixel level, where the ROC technique is often used for the interpretation of the metrics. In the case of object-based evaluation, the true negative objects cannot be defined in a meaningful way. Thus, the ROC interpretation cannot be used.

Some evaluation methods have defined a number of less common, method-specific metrics. An overlap threshold to determine association with the GT [Hall et al., 2005]. A similarity measure between detected and GT objects is also used, such as relative position [Hall et al., 2005; Jaynes et al., 2002] or shape, statistical similarity and size [Correia and Pereira, 2000; Hall et al., 2005]. Some authors used specific metrics to address the problem of the fragmentation and merging of foreground objects [Mariano et al., 2002; Nasci-

mento and Marques, 2004, 2006]. Others have focused on the impact of the human perception error on the result of the evaluation process [Cavallaro et al., 2002; Villegas and Marichal, 2004].

In Chapter 3, a novel, object-based, comparative methodology for an objective performance evaluation of motion detection algorithms is proposed. The proposed methodology uses the *F-measure*, a single-scalar technique which enables a straight forward comparison of object detection algorithms.

## 2.3 Review of background modelling

The detection of moving objects in a surveillance video typically relies on the efficient modelling of the background in the observed scene. In recent years, several surveys of background modelling and foreground detection techniques have been published. Some of them offer a simple overview of the most popular techniques [McIvor, 2000], whereas other propose a classification and an analysis of performance [Cheung and Kamath, 2004; Piccardi, 2004; Benezeth et al., 2008; Mayo and Tapamo, 2009; Bouwmans, 2009]. These surveys are reviewed in turn below.

Cheung and Kamath focus on background modelling techniques with very low initialization requirements and propose classification on *non-recursive* and *recursive* techniques [Cheung and Kamath, 2004]. *Non-recursive* techniques are those that estimate the background using a temporal variation of each pixel within a sliding buffer of a number of most recent frames. The non-recursive techniques are highly-adaptive, but require a large memory to

store a large buffer required to deal with the slow-moving objects. Some of the most-commonly used techniques are in this non-recursive category, such as frame differencing, the median filter [Cutler and Davis, 1998; Lo and Velastin, 2001; Zhou and Aggarwal, 2001; Cucchiara et al., 2003], the linear predictive filter [Cutler and Davis, 1998], and the non-parametric model known as the kernel density estimation (KDE) [Elgammal et al., 2000].

*Recursive* techniques recursively update the model using each incoming frame. These techniques require less memory as they do not need the buffer. On the other hand, they are less adaptive then non-recursive techniques. Some of the most popular recursive methods are the approximated median filter [McFarlane and Schofield, 1995; Remagnino et al., 1997], Kalman filter [Karmann and Brandt, 1990; Wren et al., 1997], and the mixture of Gaussians [Friedman and Russell, 1997; Stauffer and Grimson, 1999; KaewTraKulPong and Bowden, 2001; Power and Schoonees, 2002; Lee et al., 2003].

In their survey, Cheung and Kamath evaluated the reviewed methods on a number of grey-level urban traffic video sequences of variable difficulty (very bright light, fog, snow, and busy traffic). The evaluation is expressed in terms of precision and recall of detected moving objects. In this test the mixture of Gaussians algorithm, after the parameter tuning, performed best.

In a similar manner, Piccardi provides a review of most commonly used methods and a comparison based on speed, memory requirements and accuracy [Piccardi, 2004]. This survey also includes methods that address spatial correlation such as the co-occurrence of image variations [Seki et al., 2003] and eigen-backgrounds [Oliver et al., 2000]. Piccardi concluded that these

methods, compared to other commonly used per-pixel models, provide good accuracy although they require a relatively long training phase.

More recently Benezeth et al. and Mayo and Tapamo published two independent reviews of background modelling algorithms [Benezeth et al., 2008; Mayo and Tapamo, 2009]. Both describe in detail a number of commonly used algorithms and evaluate their performance on a number of colour videos using as the evaluation metrics the precision-recall ROC curves. Benezeth et al. performed the evaluation on a dataset of 29 real, semi-synthetic and synthetic video sequences representing three categories of data: the noise-free, the multi-modal, and the noisy data [Benezeth et al., 2008]. The evaluated algorithms include the following methods: the median filter [Zhou and Aggarwal, 2001], the single Gaussian method (1-G) [Wren et al., 1997], the mixture of Gaussians (GMM) [Stauffer and Grimson, 1999], the kernel density estimation (KDE) [Elgammal et al., 2000], and the so called *MinMax* method [Haritaoglu et al., 2000]. The authors concluded that the choice of the background modelling algorithm will inevitably be motivated by the dataset as no algorithm outperforms the rest in all dataset categories. Nonetheless, the Gaussian based algorithms, notably 1-G, GMM and KDE, proved more reliable for noisy data and the data with background motion. Mayo and Tapamo compared the following techniques: the simple frame differencing, the temporal averaging [Heikkilä and Silvén, 1999], the approximated median algorithm also known as the $\Sigma - \Delta$ method [McFarlane and Schofield, 1995; Manzanera and Richefeu, 2007], the mixture of Gaussians [Stauffer and Grimson, 1999], and the kernel density estimation method [El-

gammal et al., 2000]. Their conclusions, similarly to other reviews, argue that a choice of the most suitable method will depend on the available computational power, the dataset, and the end-user performance requirements.

One of the most comprehensive surveys published classified background modelling techniques into the following categories: *basic background modelling*, *statistical background modelling*, *fuzzy background modelling* and *background estimation* [Bouwmans, 2009].

The basic background modelling techniques use basic mathematical models of the data such as the histogram [Zhang et al., 2008], the running average [Pai et al., 2004] or the approximated median filter [McFarlane and Schofield, 1995]. Although simple to implement and with low computational cost, these techniques fail to successfully model more challenging backgrounds with noise and background motion.

The *statistical background modelling* techniques are further divided into three subcateories: the Gaussian based methods [Wren et al., 1997; Stauffer and Grimson, 1999; Elgammal et al., 2000], those using the support vector machine (SVM) models [Lin et al., 2002], and the subspace learning models using principal components analysis (PCA) and its variants [Oliver et al., 2000; Rymel et al., 2004].

The *fuzzy background modelling* uses the principles of type-2 fuzzy sets applied to Gaussian mixture model to handle the uncertainty in the underlying distributions of the observations due to insufficient and noisy data [Zeng et al., 2008; Baf et al., 2008]. It is argued that the fuzzy method provides a better approximation of the density functions provided that they contain

enough Gaussian mixture components. The *background estimation* models use filters such as Wiener filter [Toyama et al., 1999], Kalman filter [Koller et al., 1994], and Chebytchev filter [Chang et al., 2004].

Bouwmans et al. have also dedicated a separate survey to the background modelling using a mixture of Gaussians [Bouwmans et al., 2008]. The authors review more than 100 publications in this field. The algorithms are categorised according to the specific improvements they bring to the technique such as initialization of the model, the model maintenance and the update of related parameters, the choice of the number of the Gaussians in the mixture and so forth.

## 2.4 Principal component analysis (PCA)

The PCA is arguably the most popular subspace learning technique. It is based on the notion that a high-dimensional data vector can be represented by a space of a small number of orthogonal vectors (or principal components) [Hotelling, 1933]. The orthogonal vectors are in most cases calculated using the eigen-decomposition or the singular value decomposition (SVD). The principal components preserve the underlying variability in the dataset. The low dimensional representation is less costly in terms of computational power and storage space. Researchers in various fields have successfully used PCA to solve high-dimensional problems, for example in the speech processing [DeGroat and Roberts, 1990], chemical engineering [Daszykowski et al., 2007] or industrial process monitoring [Li et al., 2000]. In the field of computer vision, PCA has been used to model a variety of datasets such as edge and line

| Approach | Adaptive | Method | Evaluation metrics |
|---|---|---|---|
| Murakami and Kumar, 1982 | yes | PCA | Images |
| Cootes et al., 1992 | no | PCA | Shapes |
| Chaudhuri et al., 1996 | yes | PCA | Motion |
| Chandrasekaran et al., 1997 | yes | PCA | Images |
| Moghaddam and Pentland, 1997 | no | PCA | Faces |
| Hall et al., 1998 | yes | PCA | Images |
| Oliver et al., 2000 | no | PCA | Backgrounds |
| De la Torre and Black, 2001 | no | RPCA | Faces |
| Zeng et al., 2002 | no | PCA | Edges and lines |
| Liu and Chen, 2002 | yes | PCA | Shot boundary detection |
| Franco et al., 2002 | yes | PCA | Faces, handwritten digits |
| Artac et al., 2002 | yes | PCA | Panoramic images |
| Branson and Agarwal, 2003 | no | SPCA | Faces |
| Li, 2004 | yes | RPCA | Backgrounds, faces |
| Vidal, 2005 | no | GPCA | Motion segmentation |
| Zhang and Zhuang, 2007 | yes | RPCA | Backgrounds |
| Skocaj and Leonardis, 2008 | yes | RPCA | Images |
| Lv et al., 2009 | yes | RPCA | Backgrounds |

Figure 2.3: Examples of computer vision methods using PCA

features [Zeng et al., 2002], shapes [Cootes et al., 1992], faces [Moghaddam and Pentland, 1997; De la Torre and Black, 2001], motion [Chaudhuri et al., 1996], and backgrounds [Oliver et al., 2000; Rymel et al., 2004].

The PCA statistics is traditionally performed on a batch of data. However, in the real world image and particularly video applications, it is often required to process the data in real time. Therefore an adaptive method is needed. In theory, batch methods provide more accurate representation of data than adaptive methods due to the fact that all information about the data is available in advance. Nonetheless, *batch methods* [Oliver et al., 2000; De la Torre and Black, 2001] have opened a range of possibilities for the modelling of image data such as the dimensionality reduction, and hence lower memory requirements, and an on-line learning where new observations are added to the model at every time instance.

Adaptive PCA is an *incremental method* which computes a low dimensional representation of the data by incorporating the new observations when they become available in time. A number of incremental PCA algorithms have been proposed. Early incremental PCA methods included the adaptive update of the low-dimensional space representation, while the mean of the space remained fixed at the mean of the initial training dataset [Murakami and Kumar, 1982; Chaudhuri et al., 1996; Chandrasekaran et al., 1997]. These methods assume that the mean of the original image data is zero, which is obviously not always the case. Hall et al. pointed out that such an assumption causes errors in classification of input images and proposed the first adaptive method with the incremental shifting of the mean [Hall et al., 1998]. Following Hall's findings a number of similar incremental

methods were proposed [Liu and Chen, 2002; Franco et al., 2002; Artac et al., 2002; Li, 2004; Skočaj and Leonardis, 2008]. These algorithms produce the same updated space as Hall et al. and they are similar in terms of accuracy and speed. The main difference is in the way they incrementally update the covariance matrix.

Besides the incremental update, another major improvement of the PCA methodology concerns robustness to outliers in the training data. The traditional PCA relies on the least mean squared error minimization. To reduce the susceptibility to outliers a number of weighting techniques were proposed where the outlier pixels are identified and weighted by a weighting function before being incorporated into the model. The main problem with some of the robust PCA (RPCA) methods is that the weighting and the recalculating of the optimised model are performed in an iterative manner and are therefore computationally costly. Some examples of iterative algorithms are the self- organising algorithms [Xu and Yuille, 1995] and the expectation maximization [De la Torre and Black, 2001; Skočaj and Leonardis, 2008]. Li replaced the iterative optimization by a lookup table for pixel weights to improve the speed of the algorithm [Li, 2004]. Based on the image reconstruction error, Zhang and Zhuang propose a method for weighting the motion regions [Zhang and Zhuang, 2007], while Lv et al. weight the frames with salient motion [Lv et al., 2009], in order to reduce the influence of these frames on the background model.

Other non-adaptive variants of PCA include the following methods: Structured PCA (SPCA) [Branson and Agarwal, 2003], Generalized PCA (GPCA) [Vidal et al., 2005], and Kernel PCA (KPCA) [Schölkopf et al., 1998]. The SPCA

is a linear method which proposes clustering of similar features, where the similarity is measured by the class-conditional chi-squared distance between the distributions of the features, and then applying the classic PCA on the feature clusters. The GPCA is another linear method which defines the subspaces of data points by minimising certain distance function, represents subspaces with a set of polynomials estimated linearly from data, and derives a basis for each subspace by applying standard PCA to the set of derivatives of the polynomials. The KPCA is a non-linear extension of PCA based on embedding the data into a higher-dimensional space using a kernel matrix and then applying the standard PCA on the embedded data.

Figure 2.3 summarises computer vision methods which use PCA modelling in a range of computer vision applications.

### 2.4.1 Other dimensionality reduction methods

The main aims of dimensionality methods are to reduce the requirements for the storage space and to establish more meaningful and efficient representations for better understanding of high-dimensional data.

Linear dimensionality reduction methods represent the data in a space formed by new sets of dimensions derived as linear combinations of the original dimensions. Linear methods which use eigen decomposition are PCA [Hotelling, 1933] (also known as Karhunen-Love transform), Canonical Correlation Analysis (CCA) [Hotelling, 1936], and Linear discriminant analysis (LDA) [Fisher, 1936]. The PCA aims to maximise variance, CCA maximises correlation, where LDA uses maximisation of the interclass vari-

ance as criteria for the selection of a meaningful set of orthogonal axes spanning the new low-dimensional space. Non-negative Matrix Factorization (NMF) [Paatero and Tapper, 1994] and Independent Component Analysis (ICA) [Herault and Jutten, 1986] use more complex criteria for selection of new dimensions. The NMF defines a subspace that minimises reconstruction error for non-negative data, whereas the ICA finds a subspace where components are independent with non-Gaussian distributions. Both NMF and ICA use non-trivial optimisation algorithm. Other linear methods include Singular Value Decomposition (SVD) [Golub and Reinsch, 1970] and Factor Analysis (FA) [Gorsuch, 1983]. The SVD is related to eigenvalue decomposition; the main differences are that it does not require the mean subtraction and that it can be applied to an arbitrary matrix (eigen analysis applies only to certain classes of square matrices). The FA assumes that variables depend on some unknown common factors and estimates how much of the variability is due to these common factors. In fact, PCA is the most common form of FA which results in orthogonal uncorrelated factors.

In many applications linear data transformation may represent a serious constraint. To overcome this problem non-linear methods were developed. One of the most popular techniques is the so called *Kernel trick* which provides a way of finding non-linear subspaces with the techniques normally used for linear data transformation. For example, the Kernel PCA [Schölkopf et al., 1998] first maps the original data into some new high-dimensional space using a non-linear function and then perform a linear PCA in the mapped space. This non-linear function is not computed explicitly but via kernels. However, finding an appropriate kernel for a given problem is not

a trivial task. Other common non-linear methods for dimensionality reduction are: Multidimensional Scaling (MDS) [Sammon, 1969; Niemann, 1980], FastMap [Faloutsos and Lin, 1995], Isomap [Tenenbaum et al., 2000], and Locally Linear Embedding (LLE) [Roweis, 2000]. The MDS projects data onto a low-dimensional space in such a way that the pair-wise distances between all the points in the original dataset are preserved in the new projected space. The FastMap, a fast incremental method, is an example of a MDS technique. The Isomap method assumes that only the pair-wise distances between neighbouring points are known, utilises the Floyd-Warshall algorithm to estimate the the pair-wise distances between all of the other points, and uses MDS to compute the low-dimensional projections of points. The LLE describes the point by weights which represent a linear combination of the nearest neighbours of the point; an optimisation technique based on eigen decomposition is used to compute the low-dimensional embedding of the point while preserving the linear combination of the nearest neighbours.

There are several useful surveys of dimensionality reduction techniques [Jain et al., 2000; Fodor, 2002; van der Maaten et al., 2008; Tsai, 2010].

### 2.4.2 Discussion

In spite of its linear nature, the PCA is selected as a dimensionality reduction method of choice in this work. It is a relatively simple method which provides a compact low-dimensional representation of the data. The outdoor surveillance sequences, which are the main subject of focus in this work, display a large variability due to lighting changes as the main cause of the

non-stationary background. The PCA brings forward these high variability features and by discarding features with low variance provides a low-dimensional model of the dataset. Furthermore, the original image can be reconstructed from its low-dimensional projection and the transformation matrix with minimal reconstruction error. This is a desirable feature since the detected foreground objects are often extracted from the sequence by subtraction from the background model.

The incremental PCA methods reported in the literature enable a real-time adaptive modelling and a significant dimensionality reduction of high-dimensional datasets. Thus, they may be used to model high-dimensional video datasets using the eigen-model with a much smaller number of dimensions than the original image pixel space. However, these methods do not address the complexity and the range of problems related to the background modelling in real-life outdoor video surveillance applications. Typically, these datasets often contain a variety of background changes including gradual and sudden light changes due to weather conditions, background motion such as swaying trees, or stationary objects being left or disappearing from the scene. Considering the nature of the variability in an outdoor scene, it is expected that the background observation points in the reduced eigen-space would gather in clusters of backgrounds of similar lighting conditions. These clusters, also referred to as the *background modes*, can be individually modelled providing an improved low-dimensional representation of the observed backgrounds. Based on this notion of multi-modality and using the low-dimensional adaptive eigen-model, this thesis proposes a novel method for modelling of backgrounds in outdoor surveillance scenes in Chapter 5.

# Chapter 3

# Performance Evaluation of Object Detection Algorithms

## 3.1 Overview

The majority of visual surveillance algorithms rely on effective and accurate motion detection. However, most evaluation techniques described in the literature do not address the complexity and range of the issues which underpin the design of a good evaluation methodology. This chapter explores the problems associated with both optimising the operating point of any motion detection algorithms and the objective performance comparison of competing algorithms. In particular, we develop an object-based approach based on the *F-Measure* - a single-valued ROC-like measure which enables a straight-forward mechanism for both optimising and comparing motion detection algorithms. Despite the advantages over pixel-based ROC approaches, a number of important issues associated with parameterising the

evaluation algorithm need to be addressed. The approach is illustrated by a comparison of three motion detection algorithms including the well-known Stauffer and Grimson algorithm, based on results obtained on two datasets.

## 3.2 Introduction

The development of visual surveillance algorithms has been followed by a large effort to develop appropriate evaluation methods. Initial contributions largely focused on the provision of suitable datasets and the accompanying ground truthing tools. More recently, however, the enormous number of papers on evaluation metrics has exposed the considerable difficulties involved in establishing an accepted method of ranking competing algorithms - particularly for end-users, technology integrators and governmental agencies.

Most contributions have embraced pixel-based methods within a *receiver-operating curve* (ROC) framework. However, such pixel-oriented measures do not necessarily characterise the impact of the motion detection stage on the subsequent processing tasks *e.g.* object tracking. Moreover, it is extraordinarily difficult to manually produce the pixel-accurate ground truth to ensure accurate results. As a consequence, a large number of object-based metrics have been proposed. However, as it is not possible to apply the ROC methodology to object-based metrics, the comparison of algorithms becomes a subjective interpretation of vectors of different metrics.

What is required is a ROC-like methodology capable of generating a meaningful scalar measure of performance. In addition, the methodology should support the optimisation of algorithm parameters. This work pro-

| Output Class | True Class | |
|---|---|---|
| | Foreground | Background |
| Fore | True Positives (TP) | False Positives (FP) |
| Back | False Negatives (FN) | True Negatives (TN) |

Table 3.1: Contingency table

poses the use of the *F-Measure* [van Rijsbergen, 1979] as a means of generating this performance scalar. However, it also becomes necessary to parameterise the evaluation algorithm itself! In Section 3.4 the ROC methodology is reviewed and the F-Measure is introduced. Section 3.5 presents our comparative methodology, illustrating its application with a comparison of a home-spun motion detection algorithm [Renno et al., 2006], recently proposed "motion distillation" algorithm [Surgue and Davies, 2006], and the well-known Stauffer-Grimson algorithm [Stauffer and Grimson, 1999].

The proposed methodology looks at the results of the object detection regardless of which technique was employed by competing algorithms (i.e. analysis of the grey-scale or colour version of the original dataset).

## 3.3 Performance metrics

### 3.3.1 Common performance metrics

Performance evaluation algorithms based on comparison with ground truth can be further classified according to the type of metrics they propose. Typically, ground-truth based metrics are computed from the *true positives* (TP), *false positives* (FP), *false negatives* (FN), and *true negatives* (TN), as represented in the *contingency table*, Table 3.1.

For *pixel-based* metrics FP and FN refer to pixels misclassified as foreground (FP) or background (FN) while TP and TN account for accurately classified pixels [Black et al., 2003; Erdem et al., 2004; Gao et al., 2000; Gelasca et al., 2004; Schlogl et al., 2004; Stefano et al., 2001; Villegas and Salcedo, 1999]. Usually, they are calculated for each frame and an overall evaluation metric is found as their average over the entire video sequence. For *object-based* metrics TP refers to the number of detected objects sufficiently overlapped by GT, FP to the number of detected objects not sufficiently overlapped by the GT, and FN to the number of GT objects not sufficiently covered by any automatically detected objects [Georis et al., 2003; Hall et al., 2005; Nascimento and Marques, 2004]. (Note that this *degree of ovelap* is a parameter of the evaluation process.) Given the nature of the object-based approach, the true negative objects cannot be defined in a meaningful way. Some authors combine both types [Mariano et al., 2002]. Furthermore, a number of methods evaluate individual objects by weighting misclassified pixels according to their impact on the quality of segmented object [Aguilera et al., 2005; Cavallaro et al., 2002; Correia and Pereira, 2000; Erdem and Sankur, 2000; Villegas and Marichal, 2004] - in essence, pixel-based methods.

Typical metrics computed per-frame or per-sequence are *true positive rate* (or *detection rate*) $t_p$, *false positive rate* $f_p$, *false alarm rate* $f_a$ and *specificity* $s_p$. They are defined as

$$t_p = \frac{N_{TP}}{N_{TP} + N_{FN}}, \quad f_p = \frac{N_{FP}}{N_{FP} + N_{TN}}, \tag{3.1}$$

$$f_a = \frac{N_{FP}}{N_{TP} + N_{FP}}, \quad s_p = \frac{N_{TN}}{N_{FP} + N_{TN}} \tag{3.2}$$

where $N_{TP}, N_{FP}, N_{TN}$ and $N_{FN}$ are the number of pixels or objects identified as *true positives*, *false positives*, *true negatives* and *false negatives* respectively.

### 3.3.2 Other performance metrics

All pixel-based methods which evaluate individual object segmentation rely on existence of shape-based ground-truth mask generated by marking individual pixels, a necessary costly process performed in order to avoid errors. In addition to the advantage of avoiding hand labelling individual foreground pixels in every frame, object-based methods only require ground-truth in the form of bounding-boxes [Georis et al., 2003; Hall et al., 2005; Nascimento and Marques, 2004]. The object-level evaluation proposed by Hall et al. plots detection rates and false alarm rates using various values of overlap threshold to determine association with the GT [Hall et al., 2005]. As they do not define true-negatives, false alarm rate is computed as alternative to false positive rate and the area under such curve is used as a measure of performance. Other object-based metrics proposed in the literature are based on the similarity of detected and ground-truth objects i.e. relative position [Hall et al., 2005; Jaynes et al., 2002] or shape, statistical similarity and size [Correia and Pereira, 2000; Hall et al., 2005].

A major problem in motion detection is the fragmentation and merging of foreground objects. While these will impact on pixel-based metrics, a number of explicit metrics have been proposed [Mariano et al., 2002; Nascimento and Marques, 2004, 2006]. Typically these measure the average number of

detected regions overlapping each ground-truth object and average number of ground-truth objects associated with multiple detected regions.

Metrics may also be designed to take account of human perception of error where false positives and false negatives hold different levels of significance by introducing weighting functions for misclassified pixels on an object-by-object basis [Cavallaro et al., 2002; Villegas and Marichal, 2004]. Villegas and Marichal [Villegas and Marichal, 2004] increase the influence of misclassified pixels further from the boundary of ground-truth objects. Cavallaro *et al* [Cavallaro et al., 2002] account for temporal effects of *surprise* and *fatigue* effects where sudden changes in quality of segmentation amplify error perception.

## 3.4 Methodology for interpretation of metrics

The great majority of proposed metrics are restricted to the pixel-level discrepancy between the detected foreground and the ground-truth - namely false positive and false negative pixels. These metrics are useful to assess overall segmentation quality on a frame-by-frame basis but fail to provide an evaluation of individual object segmentation. Often these measures are normalised by image size or the amount of detected change in the mask [Cavallaro et al., 2002], or object *relevance* [Correia and Pereira, 2000]. However, a more principled approach is based on Receiver Operating Curves (ROCs). Evolved to characterise the behaviour of binary classifiers, ROC curves plot

*true positive rate* against *false positive rate* to facilitate the selection of optimal classification parameters and compare alternative classification techniques [Gao et al., 2000; Nascimento and Marques, 2004; Oberti et al., 1999]. An ROC graph is an alternative presentation to plotting metrics for each frame in the sequence which is often difficult to assess by a reader [Aguilera et al., 2005].

Unlike the pixel-level, at the object-level there is an absence of a distinct prior class of negatives in the original dataset which excludes the use of ROC curves as a method for interpretation of classifier's performance. In some applications (*e.g.* facial identification) competing algorithms are presented with images which contain known *clients* and *imposters*. This is essentially a classic binary-classification problem with two prior classes of objects. However, for object-based motion detection evaluation, there is no equivalent prior set of known *imposters*, i.e. false objects in the ground truth! Thus, as it is not possible to identify *true negatives*, the *false positive rate* cannot be computed.

The problem of the absence of a negative class of objects can be avoided by using the alternative evaluation metrics, namely the recall-precision curves [Cleverdon, 1972]. The inverse relationship between the recall and the precision was described by Cleverdon. The interpretation of recall-precision curves and the methodology for the optimisation of the operating point was later developed by van Rijsbergen. This methodology is referred to as the Effectiveness Analysis, or F-measure, [van Rijsbergen, 1979]. This approach looks at the performance on object-level in precision-recall space and does not require calculation of *true negatives*, as described in Section 3.4.2. Both ROC and F-measure methods enable simple assessment of competing algo-

rithm, selection of optimal operating point for an algorithm, and the objective comparison of two or more algorithms.

ROC curves have already been used for evaluation of motion detection algorithms [Gao et al., 2000; Nascimento and Marques, 2004; Oberti et al., 1999]. However, there are few problems in the implementation of these approaches. First they are restricted (as noted earlier) to the pixel-level [Gao et al., 2000; Oberti et al., 1999]. Second they examine total values rather than rates *e.g. false positive rate* which requires the proportion of *negatives* incorrectly classified. Finally they do not always describe a principled method of selecting the optimal operating point [Nascimento and Marques, 2004]. The alternative F-measure methodology has mainly been used in the evaluation of information retrieval systems [van Rijsbergen, 1979]. Although, Martin *et al* proposed the use of F-measure for object boundary detection [Martin et al., 2004], at the time of writing, we are not aware of any work using the F-measure method for the evaluation of object detection algorithms in visual surveillance.

In this work we distinguish between optimisation on a pixel- and object-level, and emphasise the need for the principled selection of the *optimal operating point.* At a pixel-level, we adopt traditional ROC optimisation employing scenario-specific misclassification costs for a given end-user application. At an object-level, we propose the use of F-measure optimisation in the *precision-recall* space, again using relative weighting of precision and recall for given end-user applications. The *optimal operating point* in both cases is determined in a principled manner for two distinct end-application scenarios: *Evidence Gathering* and *Ticket Fraud.*

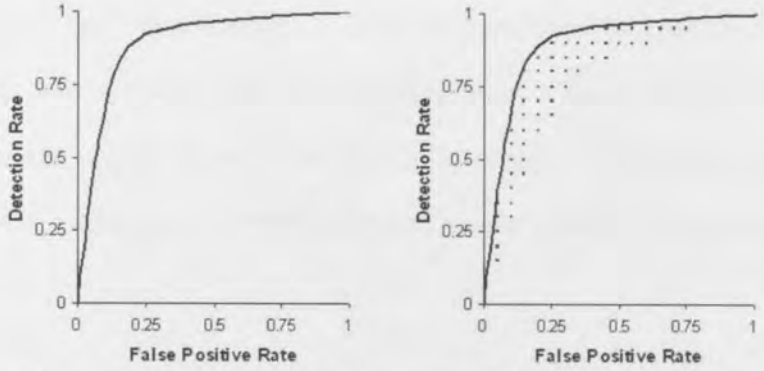
### 3.4.1 ROC-based analysis

Receiver Operating Curves (ROC) are a useful method for interpreting performance of a binary classifier [Provost and Fawcett, 1997]. ROC curves graphically interpret the performance of the decision-making algorithm with regard to the decision parameter by plotting the *true positive rate* ($t_p$) against the *false positive rate* ($f_p$). Each point on the curve is generated for the range of decision parameter values - see Figure 3.1(a). In foreground detection, such decision parameters could be a threshold on the greylevel difference between incoming pixel and reference pixel, or a threshold on the size of any foreground object. When there is more than one classification parameter, a distribution of points representing all parameter value combinations is generated in the ROC space. The required ROC curve is the top-left boundary of the convex hull of this distribution as shown in Figure 3.1(b).

A misclassification cost is associated with each point in ROC space, which depends on the intended application of object detection (*e.g.* tracking, counting people, alarming, detecting a specific person, *etc*) and the ratio of foreground pixels (or objects) to background in the GT. The misclassification cost $C$ at the operating point $(t_p^*, f_p^*)$ is given by

$$C(t_p^*, f_p^*) = (1 - P_T)C_{FP}f_p^* + P_T C_{FN}(1 - t_p^*) \tag{3.3}$$

where $P_T$ is the prior probability of a foreground pixel (or object), and $C_{FN}$ and $C_{FP}$ are the cost of classifying a moving pixel (or object) as stationary and vice versa. Operating points with the same misclassification cost form an iso-performance line. Points on the graph lying above-left of the line have




(a) Single Classification Parameter (b) Multiple Parameters

Figure 3.1: Generating ROC curves

a lower misclassification cost while points below-right of the line have larger costs. In general, the optimal operating point is chosen in the top left quadrant of the ROC space and is defined as the classification parameter value on the iso-performance line with the lowest misclassification cost [Provost and Fawcett, 1997]. The gradient $\lambda$ of this line is defined as

$$\lambda = \frac{(1 - P_T)}{P_T} \frac{C_{FP}}{C_{FN}} \tag{3.4}$$

### Cost scenarios

To explore the effect of the cost ratio, we shall introduce two different scenarios: the *Ticket Fraud* scenario in which the cost of **detaining** an innocent member of the public $C_{FP}$ is defined as double the cost of failing to catch a ticket fraudster $C_{FN}$[1]; and the *Evidence Gathering* scenario in which the cost of **video-ing** an innocent passerby $C_{FP}$ is, say, 10 times smaller than the cost of failing to video a terrorist bomber $C_{FN}$[2]. The relative costs are

[1]Increasingly, the context of prevention of terorism is inverting the cost ratio

[2]This must also capture the storage and search costs

arbitrary, and the relationship of these applications to motion detection is indirect. However, these different cost ratio scenarios ensure we are mindful of the ultimate application in the evaluation stage. (Obviously defining the social costs of violations of *libertarian* and *public safety* concepts is extremely fraught!)

### 3.4.2 Effectiveness analysis (or F-measure)

While ROC analysis offers a solution for the selection of operating points and the definition of scenarios, the use of pixel-based metrics do not accurately measure the generation of detected blobs (and capture their impact on the subsequent tracking and application processes). However, object-based metrics, which are more informative, require more complex machinery to identify the correspondence between detected objects and ground truth. Such machinery is accompanied by an inevitable array of evaluation parameters whose values can have a dramatic effect on the measured performance (see Section 3.5.4). In addition, object-based performance analysis does not provide true negative objects which are crucial to the ROC approach. What is needed is a ROC-like methodology, which avoids the use of true negatives and preserves the idea of different evaluation scenarios. We also propose the F-measure as a tool for optimisation of evaluation parameters in object-based evaluation of object detection algorithms.

The F-measure, or effectiveness measure, characterises the performance of classification in precision-recall space [van Rijsbergen, 1979], and is defined

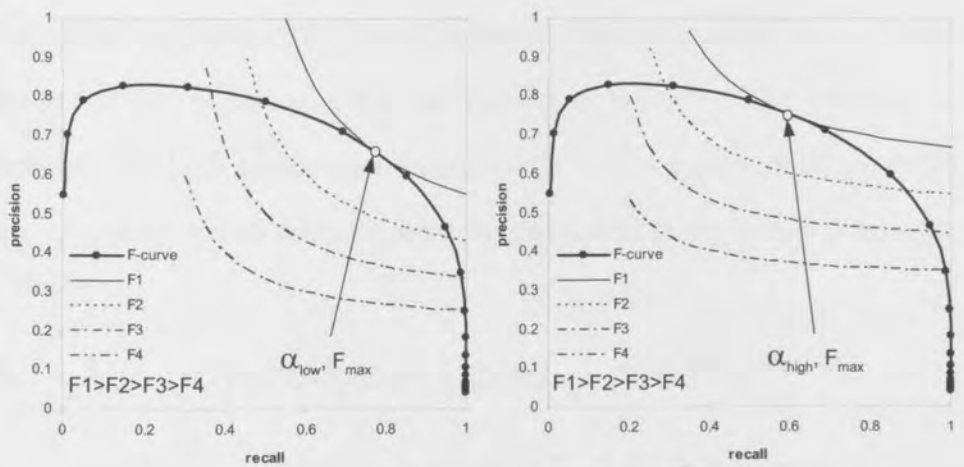
as the weighted harmonic mean of the *precision* (P) and *recall* (R) metrics

$$F = \frac{1}{\alpha * \frac{1}{P} + (1 - \alpha) * \frac{1}{R}} \tag{3.5}$$

where

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \quad P = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{3.6}$$

The parameter $\alpha$ is dependent on the end-user application, and controls the relative importance of P and R (in a similar manner to the cost ratio in Equation 3.4). The goal is to determine the optimal parameters by locating the maximum F-measure for a given $\alpha$.




(a) $\alpha$ low (b) $\alpha$ high

Figure 3.2: Iso-effectiveness lines

Figure 3.2 presents a typical F-curve in precision-recall space generated for the range of possible values of an arbitrary parameter. The figure also shows a set of iso-effectiveness lines along which all points have the same value of F-measure. The shape of these lines is determined by the choice

of $\alpha$. The tangent point between the highest iso-effectiveness line and the F-curve is taken as the optimal operating point.

### Application scenarios

In Section 3.4.1, we introduced the concept of *cost scenarios* in ROC-space in order to evaluate motion detection results in the context of specific end-user applications. However, such an approach to the optimisation of algorithm parameters was restricted to the pixel-level. In a similar fashion, the F-measure approach can be used to optimise both algorithm and evaluation parameters on an object-level. The appropriate choice of the parameter $\alpha$ depends on the end-user application; specifically the application scenarios we identified in Section 3.4.1, i.e *Evidence Gathering* and *Ticket Fraud*. By maximising the F-measure for the chosen $\alpha$ we select the optimal set of parameters for each application scenario. (The discussion on the appropriate choice of $\alpha$ for each of our scenarios is postponed to Section 3.5.4.)

## 3.5 An object-based comparative methodology using the F-Measure

Having introduced both the ROC and F-Measure frameworks we now present and illustrate our proposed object-based comparative methodology. We first start by making remarks on the structure of the evaluation process itself in Section 3.5.1 - in particular the implications of the existence of parameters within the evaluation process itself. While not part of the evaluation

methodology, the datasets used to illustrate the methodology is introduced in Section 3.5.2. The various steps of the methodology itself are presented in Section 3.5.4. In summary these are

- defining variations of the F-Measure metric for specific application scenarios,
- defining a method of associating detected objects with the ground truth objects,
- selecting the optimal values of the evaluation parameters,
- optimising the parameters of each competing algorithm for each scenario, and
- computing the performance of each algorithm.

### 3.5.1 Evaluation architecture

Typical performance evaluation takes the output of some visual surveillance algorithm and compares it with the *ground truth* data - as illustrated in Figure 3.3. The performance of any surveillance algorithm will depend on the choice of a range of parameters. A detection algorithm typically has a number of internal parameters (algorithm parameters) which can be optimised for best performance. However, the evaluation process at object-level also relies on a set of parameters which typically determine the degree of correspondence between detected blobs and ground-truth objects (evaluation parameters). As evaluation on a pixel-level does not involve a "degree of

match" (correspondence between two pixels is established *de facto*), the existence of such evaluation parameters on the object-level is often overlooked.

How would you select appropriate values for these? A naive approach would be to include these evaluation parameters within the ROC methodology to select the optimal algorithm **and** evaluation parameter values. However, the result would be to evaluate each alternative surveillance algorithm with a **different** evaluation algorithm. Hardly an objective comparison! Furthermore, the ROC methodology does not apply at object-level where true negatives are not available. We explore the issue of optimisation at various stages of a typical evaluation system.

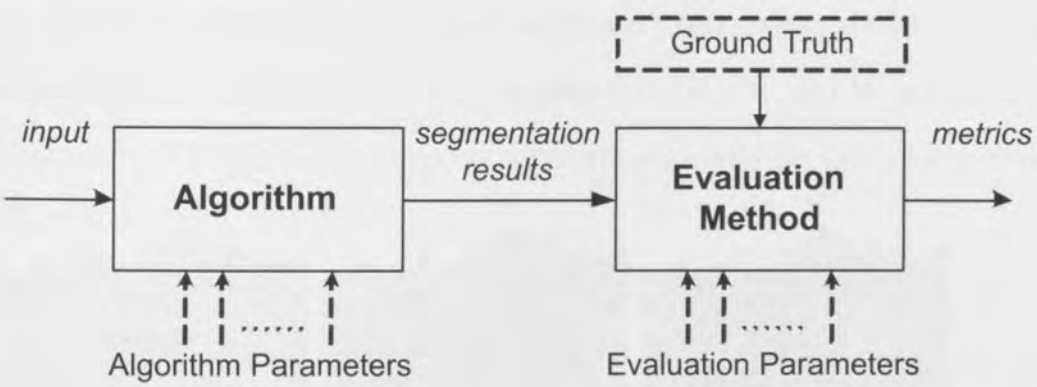



Figure 3.3: Typical performance evaluation system

### 3.5.2 Dataset and ground truth

The evaluation was performed on two datasets: PETS-2001-camera1 dataset and Kingston Carpark dataset. This choice was motivated by the requirement for a standard dataset but also by the need to evaluate competing algorithms on a dataset with a complex background. The PETS dataset is

well known, standard dataset from the PETS series accepted and often used by the research community. It represents a typical outdoor surveillance scene showing a street and a car park in front of an office building with a number of foreground objects, people and vehicles, moving throughout the scene. However, this dataset does not contain much lighting variations. Therefore another less known but more challenging dataset is used, the Kingston Carpark dataset, which contains many interesting features relevant to the problem of background modelling and object detection, such as lighting changes and background motion due to the weather conditions. (Example frames are shown in Figure 3.4.)

The PETS-2001-camera1 dataset consists of 2685 frames with 13 moving objects, people and vehicles, which appear individually and in groups and cross paths. The lighting is fairly stable with only minor background motion.

Figure 3.4: PETS-2001-camera1 dataset

Figure 3.5: Kingston-Carpark dataset

The Kingston-Carpark dataset consists of 8210 frames recording activ-

ities in a car park at full frame-rate covering a period of five and a half minutes. (Example frames shown in Figure 3.5.) The CCTV camera has iris auto-correction and colour switched on. The video sequence includes a total of 24 moving objects, people and vehicles appearing at close, medium and far distances from the camera. There is a variety of both gradual and sudden lighting changes present in the scene due to English weather conditions (bright sunshine interrupted by fast moving clouds, reflections from windows of vehicles and buildings). There are both static and dynamic occlusions present in the scene with moving objects crossing paths and disappearing partly or totally behind static objects. In addition, a strong wind causes swaying trees and bushes to disturb the background.

The ground truth for both datasets is generated manually (by one person) using an in-house semi-automatic tool for drawing bounding boxes for every target within each frame of the video sequence. Ground truth provides the temporal ID of the object, its bounding box enclosing pixels of interest, defines the type of the object whether person or vehicle, and defines the degree of occlusion with other objects *i.e.* unoccluded, semi-occluded or fully-occluded.

### 3.5.3 Detection algorithms

The proposed performance evaluation methodology is applied to three different detection algorithms: the well known Gaussian mixture model, the UV-variation modelling, and the motion distillation algorithm. Note that the proposed evaluation methodology is concerned with the results of the

object detection regardless of which technique was employed by competing algorithms (i.e. analysis of the grey-scale or colour version of the original dataset).

The Gaussian mixture method models each pixel by a number of Gaussian distributions and adapts the model with each new observation [Stauffer and Grimson, 1999]. According to the persistence and the variance of each distribution it is determined which Gaussian distributions correspond to backgrounds. Pixels that do not fit any of the background distributions are marked as foreground until they are included into a newly formed distribution. This is an adaptive per-pixel model.

The UV-variation modelling exploits the observed correlations of U and V components in YUV colour space due to global lighting changes in the scene [Renno et al., 2006]. The proposed method models the possible UV variations using a measure of the global colour content of the frame. It is a unimodal per-pixel model for classification of pixels as foreground, shadow, highlight or background.

The motion distillation method detects moving edges using spatio-temporal wavelet decomposition [Surgue and Davies, 2006]. Rather than modelling the background, this method detects the foreground by the spatial coherence of its motion.

The three methods were chosen for evaluation because of their very different approach in solving the problem of object detection.

### 3.5.4 Methodology

This section outlines the main features of the proposed object-based comparative methodology using the F-measure.

#### Associating ground truth objects to detected objects

In pixel based evaluation the correspondence between the GT and the detected foreground pixels is straight forward. However, where object-based metrics are used gives rise to a correspondence problem. Establishing the correspondence between GT objects and detected objects which can be displaced spatially in the image, or indeed fragmented or merged. The latter is particularly problematic as the density of objects rises. Although the datasets used for the evaluation are of relatively low density, they do provide examples of these problems.

Following a typical approach, we use the degree of overlap between detected objects and GT bounding boxes to establish this correspondence. (This is similar to the few published object-based methods [Hall et al., 2005; Nascimento and Marques, 2004].) In general, this can result in one-to-many and many-to-one relationships. Thus the number of true positives $N_{TP}$ and false negatives $N_{FN}$ could be larger than the number of ground truth objects $N$ *i.e.* $N_{TP} + N_{FN} \geq N$, if the correspondence is not handled correctly. Therefore, the choice of the correspondence algorithm is crucial.

In object-based performance analysis, the label of each object (TP, FP, and FN) is defined as follows. A TP is a detected foreground blob which overlaps a GT bounding box, where the area of overlap is greater than a

proportion $\Omega_b$ of the blob area **and** greater than a proportion $\Omega_g$ of the GT box area. A FP is a detected foreground object which does not overlap a GT bounding box. Finally, a FN is a GT bounding box not overlapped by any detected object. Note there are no definable true negatives for objects.

### Impact of evaluation parameters

Since the association between ground truth and detected objects relies on these evaluation parameters, so we should expect the *precision*, *recall* and hence *F-Measure* values to depend on the overlap values $\Omega_b$ and $\Omega_g$. This dependency can be illustrated in Figures 3.6 and 3.7. A *precision* graph and *recall* graph are presented in each figure for the Stauffer and Grimson algorithm. Each plot records one of these metrics as a function of $\Omega_b$ (the blob overlap threshold) for a fixed $\Omega_g$ (the ground truth overlap threshold). Each graph contains three such plots for three different fixed values $\Omega_g \in \{0.002, 0.05, 0.3\}$. Figures 3.6 and 3.7 represent two different sets of algorithm parameters - the operating points of each of the Ticket Fraud and Evidence Gathering application scenarios. In all cases, these metrics are highly sensitive to the overlap thresholds *i.e.* the evaluation parameters. Having demonstrated this dependency, the selection of the optimal values is described later in this section.

### Parameterising the scenarios

The approach of defining specific end-user applications is utilised here to select suitable parameter values for $\alpha$ (Equation 3.5). The weight $\alpha$ reflects the relative importance of precision and recall for a specific surveillance task.

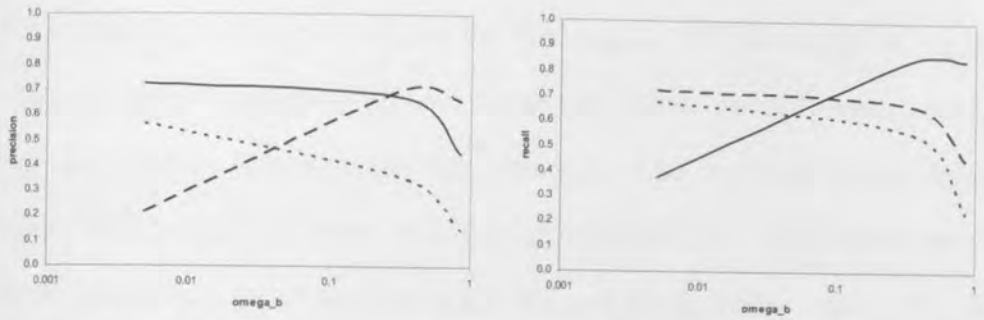



Figure 3.6: Ticket Fraud: varying Blob Overlap Threshold $\Omega_b$ for fixed Ground Truth Overlap Threshold $\Omega_g$ (solid line 0.002, dashed line 0.05, dotted line 0.3)

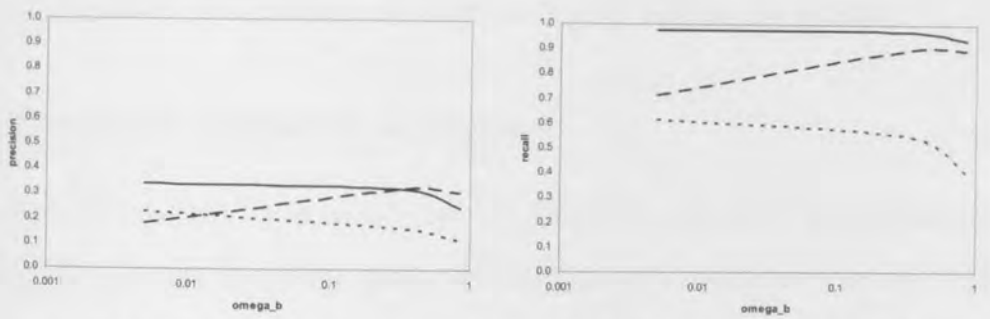



Figure 3.7: Evidence gathering: varying Blob Overlap Threshold $\Omega_b$ for fixed Ground Truth Overlap Threshold $\Omega_g$ (solid line 0.002, dashed line 0.05, dotted line 0.3)

We select values for $\alpha_{TF}$ and $\alpha_{EG}$ associated with the two *Ticket Fraud* and *Evidence Gathering* scenarios defined in Section 3.4.1. In the *Ticket Fraud* scenario, it is essential to keep the number of false positives low due to the relatively high cost of misclassifying negatives as positives, which implies a requirement for high precision and consequently low recall. In the *Evidence Gathering* scenario, a high true positive rate is vital, which implies high recall. Therefore, Ticket Fraud scenario is characterised by relatively high $\alpha_{TF}$ value, whereas Evidence Gathering case requires a low $\alpha_{EG}$.

Despite this relatively principled manner of constraining suitable values

for $\alpha$, the final choice of $\alpha_{TF} = 0.8$ and $\alpha_{EG} = 0.03$ is somewhat arbitrary. As $\alpha$ takes values between 0 and 1, the two values were chosen at either end of this range to demonstrate the approach. The end-user would define the exact values of $\alpha$ for their specific surveillance task. (Similarly, while the ROC methodology of Section 3.4.1 enjoyed an apparent rigour, the actual choice of the cost ratio was also rather arbitrary.) Nonetheless we propose to use and promote these values as part of this standardised object-based optimisation and comparison methodology based on the F-Measure.

**Setting the evaluation parameters**

As illustrated in Figures 3.6 and 3.7, object-based performance metrics are highly dependent on the choice of evaluation parameters and the end-user application. To select the appropriate evaluation parameters we need to perform the optimisation process over the space of all algorithms, all combinations of algorithm parameters, and all combinations of evaluation parameters. Effectively we select the algorithm (and its optimum algorithm parameters) and evaluation algorithm (a specific set of evaluation patameters) that gives the best results. This process is repeated for each application scenario *i.e.* in theory, a different set of evaluation parameters may be selected for the *Ticket Fraud* and *Evidence Gathering* scenario.

Ideally, whenever a new algorithm is evaluated, the whole process of optimising the evaluation parameters over the whole space of all algorithms and parameter combinations should be performed. In reality, we can assume that the objectiveness requirement is satisfied if the evaluation parameter selection is performed once for a few well-known detection algorithms. Once the

evaluation parameters are selected, we can proceed with the optimisation of each algorithm. Finally, we compare the performance of the optimised algorithms for a given application scenario (see Section 3.5.4).

We illustrate this method of selecting the evaluation parameters by generating suitable values for our two area overlap threshold parameters $\Omega_g$ and $\Omega_b$. As illustrated in Figure 3.8 we seek to locate those evaluation parameters generating the maximum F-measure for the chosen scenario *i.e.* a specific $\alpha$. (The F-measure increases as we move toward top right corner in precision-recall space).

A point in the graph represents a specific instance of one of the motion detection algorithms with a specific set of algorithm and evaluation parameters. Such a point is generated for every combination of algorithm, algorithm parameter and evaluation parameter. Also drawn are the two iso-effectiveness lines which include the point that resulted in the largest F-Measure for each scenario. The evaluation parameters associated with each of these ringed points define the optimal evaluation process. For each scenario and each dataset the optimal values of $\Omega_g$ and $\Omega_b$ are listed below in Table 3.2.

| | PETS-2001-camera1 | | Kingston-Carpark | |
|---|---|---|---|---|
| Application Scenario | $\Omega_g$ | $\Omega_b$ | $\Omega_g$ | $\Omega_b$ |
| Evidence Gathering | 0.002 | 0.005 | 0.002 | 0.005 |
| Ticket Fraud | 0.002 | 0.005 | 0.002 | 0.005 |

Table 3.2: Optimal evaluation parameters

Interestingly the values obtained in this example are the same for both datasets which may not always be the case.

### Optimising and comparing detection algorithms

Having generated the optimal set of evaluation parameters, the performance of each algorithm may now be optimised for each scenario by recovering the values of each algorithm's parameters which maximise the F-Measure. Figure 3.9 illustrates this optimisation process for each of our three detection algorithms: Stauffer-Grimson [Stauffer and Grimson, 1999], Renno *et al* [Renno et al., 2006] and Motion Distillation [Surgue and Davies, 2006]. Figures 3.9(a), 3.9(b) and 3.9(c) were generated for the PETS-2001-camera1 sequence, and 3.9(d), 3.9(e) and 3.9(f) for the Kingston-Carpark sequence. Here each point represents a specific set of algorithm parameters. For each algorithm, the previously parameterised evaluation process identifies an optimal combination (the circle in each figure) for each scenario. These optimal algorithm parameters are summarised in Tables 3.3, 3.4 and 3.5 for each algorithm[3] and each dataset.

Ideally detection algorithms would have enough robustness in different datasets requiring little change in their parameter settings. However in practice that is not the case, algorithms display a sensitivity to a specific application. Therefore the optimisation of the algorithm parameters is necessary for each dataset and each application scenario.

### Analysis of the evaluation results

With the evaluation and optimal algorithm parameters determined, the performance of motion detection algorithms can be compared simply by com-

[3]Readers are referred to the relevant papers for a description of the algorithm.

| Algorithm Parameter | PETS-2001-camera1 | | Kingston-Carpark | |
|---|---|---|---|---|
| | Evidence Gathering | Ticket Fraud | Evidence Gathering | Ticket Fraud |
| $\alpha$ | 0.008 | 0.014 | 0.02 | 0.014 |
| $\chi_{RGB}$ | 10 | 12 | 26 | 26 |
| $\tau_b$ | 0.9 | 0.9 | 0.9 | 0.6 |

Table 3.3: Optimal algorithm parameters: Stauffer and Grimson

| Algorithm Parameter | PETS-2001-camera1 | | Kingston-Carpark | |
|---|---|---|---|---|
| | Evidence Gathering | Ticket Fraud | Evidence Gathering | Ticket Fraud |
| $\min_U$ | 0.0001 | 0.0001 | 0.0001 | 0.001 |
| $\min_V$ | 0.0001 | 0.001 | 0.1 | 0.01 |

Table 3.4: Optimal algorithm parameters: Renno *et al*

| Algorithm Parameter | PETS-2001-camera1 | | Kingston-Carpark | |
|---|---|---|---|---|
| | Evidence Gathering | Ticket Fraud | Evidence Gathering | Ticket Fraud |
| Sobel threshold | 20 | 20 | 10 | 30 |
| Image stack | 11 | 9 | 7 | 5 |

Table 3.5: Optimal algorithm parameters: Motion Distillation

paring the equivalent F-Measure metric in each of the application scenarios. Table 3.6 shows the F-measure values corresponding to the optimal operating points of Stauffer-Grimson [Stauffer and Grimson, 1999], Renno *et al* [Renno et al., 2006] and Motion Distillation [Surgue and Davies, 2006] algorithms, for both the Evidence Gathering and Ticket Fraud scenarios, and both PETS-2001-camera1 and Kingston-Carpark datasets. Results illustrate that for both scenarios and both datasets the Stauffer-Grimson algorithm outperforms the other two algorithms. The Motion Distillation algorithm comes very closely behind the Stauffer-Grimson, achieving only slightly lower F-measure values, while the Renno algorithm is in third place. It should be noted, however, that the ranking of the competing algorithms was obtained when the parameters of each algorithm were optimised for each dataset individually. This may hide an underlined sensitivity to the data - a lack of robustness in which the algorithm performance degrades in different datasets. Further experimentation if required over a larger number of datasets to expose such weakness.

| Algorithm | PETS-2001-camera1 | | Kingston-Carpark | |
|---|---|---|---|---|
| | Evidence Gathering | Ticket Fraud | Evidence Gathering | Ticket Fraud |
| Stauffer and Grimson | 0.875 | 0.940 | 0.934 | 0.755 |
| Renno *et al* | 0.850 | 0.807 | 0.725 | 0.583 |
| Motion distillation | 0.867 | 0.937 | 0.887 | 0.720 |

Table 3.6: F-Measure comparison

With the exception of the Renno method, the algorithms achieve a slightly higher F-measure for the *Ticket Fraud* scenario on the PETS-2001-camera1

dataset. On the Kingston-Carpark sequence the situation is reversed *i.e.* the F-measure is higher for the *Evidence Gathering* than for the *Ticket Fraud* scenario for all three algorithms. Given the higher importance of the precision over recall for the *Ticket Fraud* scenario (see Section 3.5.4), this observation reflects the fact that in sequences with generally unchanging backgrounds, relatively few false positives are detected. More changeable backgrounds (*i.e.* in sequences with changing lighting conditions and moving background elements such as trees) cause an increase in the number of falsely detected objects and consequently the loss of precision. This notion becomes interesting, not only as a way of evaluating of the algorithms, but also in the context of defining the level of difficulty of a video sequence in terms of motion detection. In other words, the relationship between achieved levels of the F-measure for the standardised application scenarios can indicate the level of difficulty of a dataset.

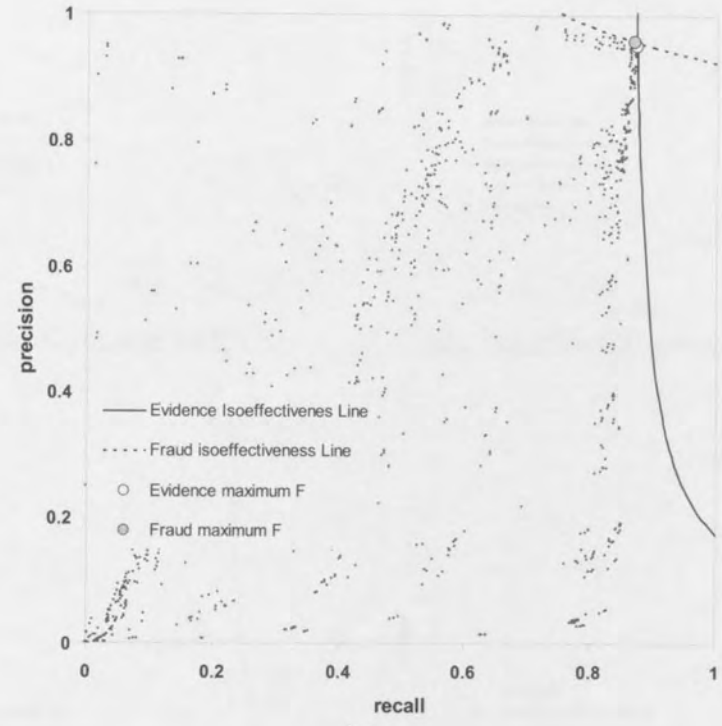
For this illustrative comparison, the Stauffer-Grimson algorithm performed best in these situations. The ranking of the algorithms remains the same for both scenarios and both datasets.

## 3.6 Conclusion

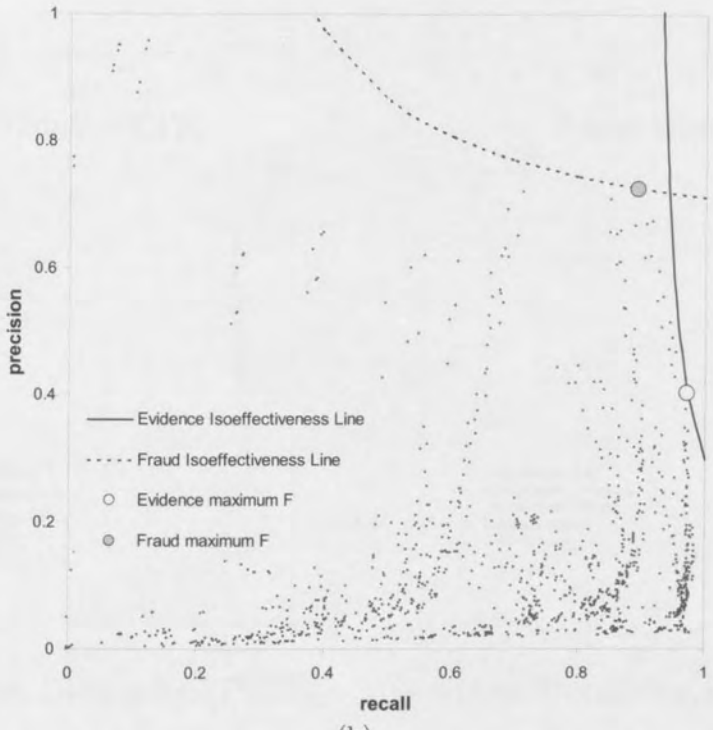
The primary purpose of this chapter was to expose the surprisingly complex issues that arise when designing a well-designed evaluation methodology for comparing motion detection algorithms. Many of these problems arise when focusing on object-based metrics. (Pixel-oriented measures do not adequately capture the impact of the motion detection stage on the subsequent

processing tasks.) Two issues in particular require careful addressing. First, the inevitable existence of evaluation parameters and the need to select appropriate values for these. Second, the absence of true negatives makes it impossible to use the well-known ROC methodology. In addition to these issues, a comparative methodology is required - a methodology that allows the definition of standardised application scenarios which provide context to the comparison.

From these considerations, a new object-based comparative methodology based on the *F-Measure* has been developed. While it provides a single-valued ROC-like measure enabling both optimisation and comparison of motion detection algorithms, there are a number of configuration steps. In summary, these are defining the application scenarios; determining the appropriate $\alpha$ weights for each scenario; defining a method of associating detected objects with the ground truth objects; selecting the optimal values of the evaluation parameters; optimising the parameters of each competing algorithm for each scenario; and finally computing and comparing the performance of each algorithm. The significance of evaluating motion detection within the wider context of a surveillance system has been discussed. However, some of the current weaknesses remain to be addressed: explicit methodology for choosing appropriate $\alpha$ values; standardisation of application scenarios; comparison with more methods reported in the literature; and a bigger range of datasets. Further investigation is needed to explore whether this methodology might be practical further down the evaluation pipe-line, illustrated by the graph in Figure 3.10 *i.e.* measuring the impact of an early visual process on the results of subsequent visual processes.

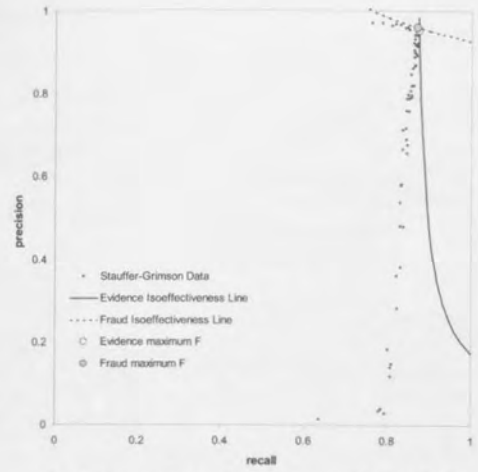



(a)

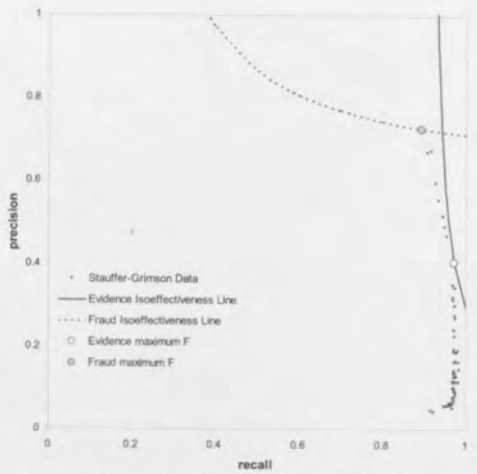



(b)

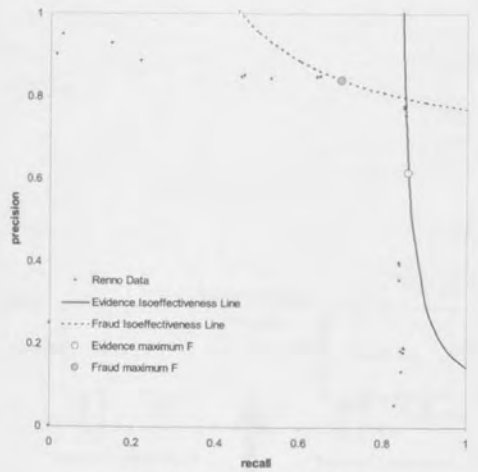
Figure 3.8: Selecting the evaluation parameters (Ticket Fraud and Evidence Gathering scenarios) (a) PETS-2001-camera1 dataset (b) Kingston-Carpark dataset



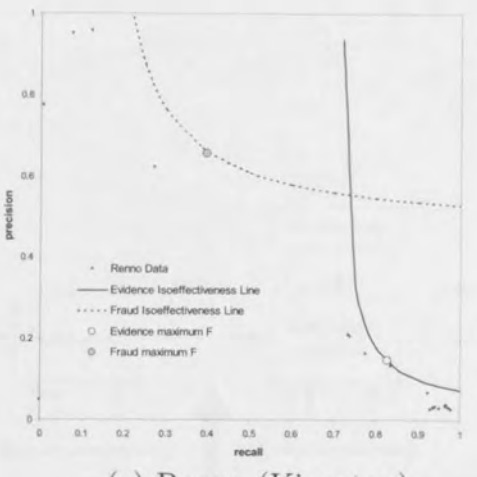

(a) Stauffer-Grimson (PETS)



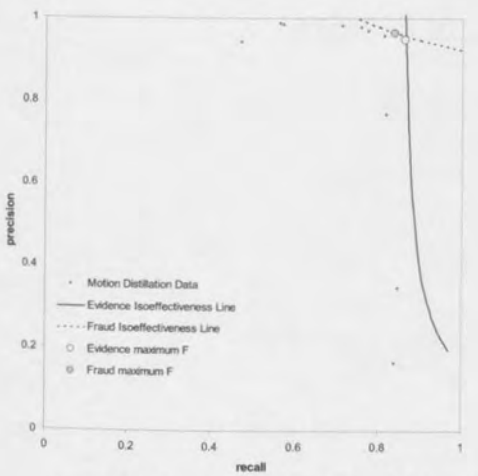

(d) Stauffer-Grimson (Kingston)



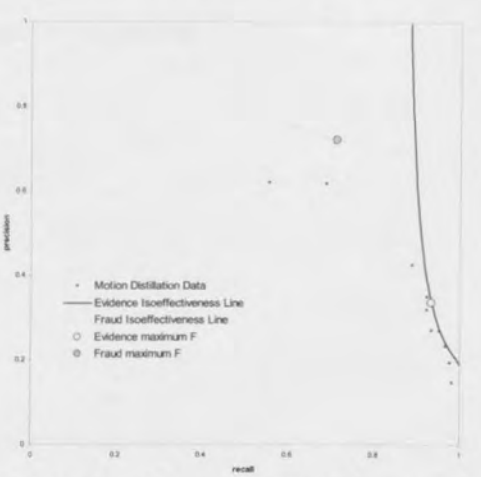

(b) Renno (PETS)




(e) Renno (Kingston)




(c) Motion Distillation (PETS)




(f) Motion Distillation (Kingston)

Figure 3.9: Algorithm optimisation: (a-c) PETS-2001-Camera1 dataset, (d-f) Kingston-Carpark datasets

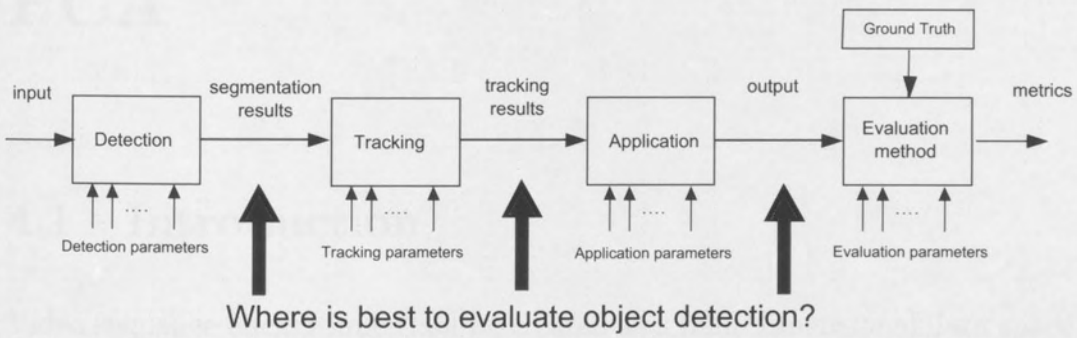



Figure 3.10: Evaluation pipe-line

# Chapter 4

# Background Modelling using PCA

## 4.1 Introduction

Video sequence backgrounds can be treated as a high-dimensional data space where the behaviour of the variability of the grey-levels of the image pixels is observed in time and space. As the background structure remains constant over time, and as lighting changes are typically correlated over the image, variations in these pixel variables typically populate small subspaces within this very high-dimensional space. To illustrate the complexity of the problem, consider that only a few minutes of digital video at full frame rate is equivalent to a dataset of a few thousands of images each of a hundred thousand pixels. Not surprisingly, efficient background modelling is computationally costly and difficult to perform in real time applications. Furthermore, in surveillance applications outdoor video sequences are often affected

by considerable background variations caused by global and partial illumination variations, gradual and sudden lighting condition changes, and non-stationary background. The highly variable nature of background variations in outdoor video sequences requires highly adaptable and robust models able to represent the background at any time instance with sufficient accuracy.

This work discusses the possibility of and proposes a method to reduce the complexity of the problem of background modelling by extracting a sufficient amount of information about the behaviour of the available dataset from as little amount of data as possible. The solution is twofold. The first step is to reduce the dimensionality of the problem by projecting the original data from the image pixel space into some lower dimensional space. Next, it is possible to sub-sample the available dataset in an appropriate manner, in order to reduce the amount of data for processing. The idea is to perform all the analysis and processing on a much smaller number of variables. Once the processing has been done the processed data can be projected back to recreate images in the full dimensionality of the original data space. To make the problem more tractable, the image is first partitioned into smaller subregions. The region size is selected in such a way that the objects moving through the region neither appear too large (therefore occluding the whole region), nor too small.

This chapter explores methods for efficiently choosing the minimum amount of the original data and the lowest number of dimensions required to obtain results which remain within the limits of an acceptable quality loss. This problem is addressed by applying eigen-theory and principal components analysis (PCA) on a video surveillance recording of a typical outdoor scene.

This chapter is concerned with the batch, or off-line, background modelling. An *off-line* methodology refers to those background modelling techniques applied to a prerecorded video sequence, when the amount of available data (duration of the sequence) is known and fixed, and all possible variations of the scene background are already contained within the sequence. An *on-line* method refers to an adaptive technique which allows for merging the information about new observations with the existing knowledge about the video sequence background. The on-line approach is considered later in Chapter 5.

## 4.2 Modelling in high dimensions

### 4.2.1 Handling high–dimensional data spaces

#### Eigen problem definition

The eigen problem is concerned with transforming a regular (non-zero determinant) matrix into a singular (zero determinant) one.

Let $\underline{\underline{\mathbf{I}}}$ be an identity matrix, $\lambda$ a scalar and $\underline{\underline{\mathbf{A}}}$ a square regular matrix. Then matrix $\underline{\underline{\mathbf{Z}}}$ is a scalar matrix.

$$\underline{\underline{\mathbf{Z}}} = \lambda \underline{\underline{\mathbf{I}}} \tag{4.1}$$

Let us also define a new matrix,

$$\underline{\underline{\mathbf{A}}} - \underline{\underline{\mathbf{Z}}} = \underline{\underline{\mathbf{A}}} - \lambda \underline{\underline{\mathbf{I}}} \tag{4.2}$$

with $|\underline{\underline{\mathbf{A}}} - \lambda\underline{\underline{\mathbf{I}}}|$ as its determinant. Furthermore, the *characteristic equation* of $\underline{\underline{\mathbf{A}}}$ is defined as

$$|\underline{\underline{\mathbf{A}}} - \lambda\underline{\underline{\mathbf{I}}}| = 0 \tag{4.3}$$

The solutions $\lambda$ of the characteristic equation of the matrix $\underline{\underline{\mathbf{A}}}$ are called the *eigenvalues* of the matrix $\underline{\underline{\mathbf{A}}}$. The regular matrix $\underline{\underline{\mathbf{A}}}$ can be transformed into a singular matrix $\underline{\underline{\mathbf{A}}}$-$\lambda\underline{\underline{\mathbf{I}}}$ for some specific values of the scalar $\lambda$.

Subtracting $\lambda\underline{\underline{\mathbf{I}}}$ from $\underline{\underline{\mathbf{A}}}$ is equivalent to subtracting scalar $\lambda$ from the elements on the main diagonal of $\underline{\underline{\mathbf{A}}}$. In order to force the determinant of the new matrix to zero, the trace of $\underline{\underline{\mathbf{A}}}$, i.e. the sum of its diagonal elements, must be equal to the sum of these specific values of $\lambda$. The number of different specific values of $\lambda$, denoted as eigenvalues, is equal to the rank of the matrix $\underline{\underline{\mathbf{A}}}$. The rank of a matrix is defined as the maximal number of linearly independent columns (or rows)[1] of that matrix.

$$trace(\underline{\underline{\mathbf{A}}}) = \sum_{i=1}^{r} \lambda_i \tag{4.4}$$

Furthermore, if there is a vector $\underline{\mathbf{e}}_i$, of the same number of rows as $\underline{\underline{\mathbf{A}}}$, such that

$$(\underline{\underline{\mathbf{A}}} - \lambda_i\mathbf{I})\underline{\mathbf{e}}_i = \underline{\underline{\mathbf{0}}} \tag{4.5}$$

then the $\underline{\mathbf{e}}_i$ is an *eigenvector* of the matrix $\underline{\underline{\mathbf{A}}}$ which corresponds to the eigenvalue $\lambda_i$. Eigenvectors that correspond to different eigenvalues are linearly

[1]The column rank and the row rank are always equal. Also, the rank of a non-square $m$x$n$ matrix is at most $\min(m,n)$.

independent. Equation 4.5 is often written as

$$\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{e}}_i = \lambda_i\,\underline{\mathbf{e}}_i. \tag{4.6}$$

**Eigen analysis of a covariance matrix**

Let $\underline{\underline{\mathbf{X}}}$ be a $p$ x $n$ matrix of $n$ observations of $p$ variables $x_i$, $i = 1, \ldots, p$, with observations arranged in columns and variables in rows. Then

$$\underline{\underline{\mathbf{X}}} = \begin{vmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & \ldots & x_{pn} \end{vmatrix}, \tag{4.7}$$

where $x_{ij}$ is the j$^{th}$ observation of the variable $x_i$.

The covariance matrix $\underline{\underline{\mathbf{S}}}$ of $p$ variables is defined as

$$\underline{\underline{\mathbf{S}}} = \begin{vmatrix} \sigma_{11}^2 & \sigma_{12} & \ldots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22}^2 & \ldots & \sigma_{2p} \\ \vdots & \vdots & & \vdots \\ \sigma_{p1} & \sigma_{p2} & \ldots & \sigma_{pp}^2 \end{vmatrix}, \tag{4.8}$$

where $\sigma_{ii}^2$ and $\sigma_{ij}$ are variances and covariances of variables $x_i$, $(i, j = 1, \ldots, p$ and $i \neq j)$.

$$\sigma_{ii}^2 = E[(x_i - \overline{x_i})^2] \tag{4.9}$$

$$\sigma_{ij} = E[(x_i - \overline{x_i})(x_j - \overline{x_j})] \tag{4.10}$$

By definition, the covariance matrix $\underline{\underline{\mathbf{S}}}$ is regular, square and symmetric. Its eigenvalues, denoted as $\lambda_k$, where $k = 1, \ldots, r$, ($r$-rank of the matrix), are real and positive. From Equations 4.4 and 4.8 there follows an interesting property that the sum of variances of variables is equal to the sum of eigenvalues of the covariance matrix.

$$\sum_{i=1}^{p} \sigma_{ii}^2 = \sum_{k=1}^{r} \lambda_k \tag{4.11}$$

### Principal Components Analysis (PCA)

The PCA technique is used to analyse high-dimensional data spaces in order to identify those dimensions which capture most of the variability in the data. It is often used to reduce the complexity of datasets without significant loss of information about their behaviour. The derivation of principal components is explained in great detail in numerous publications. Here, the interpretation of I.T.Jollife has been used, [Jolliffe, 2002].

### Definition

Consider a matrix $\underline{\underline{\mathbf{X}}}$ of $n$ observations of $p$ random variables and its covariance matrix $\underline{\underline{\mathbf{S}}}$. Also, let $\{\underline{\alpha}_1, \underline{\alpha}_2, \ldots, \underline{\alpha}_k, \ldots\}$ be a set of vectors of $p$ elements each.

Look for a linear function $\underline{\alpha}_1^T \underline{\underline{\mathbf{X}}}$ with maximum variance, i.e. chose vector $\underline{\alpha}_1$ in such a way to maximize the variance of $\underline{\alpha}_1^T \underline{\underline{\mathbf{X}}}$. Then look for another linear function $\underline{\alpha}_2^T \underline{\underline{\mathbf{X}}}$, which is uncorrelated with $\underline{\alpha}_1^T \underline{\underline{\mathbf{X}}}$ and maximizes the variance of $\underline{\alpha}_2^T \underline{\underline{\mathbf{X}}}$, and so on. At the k$^{th}$ step, the linear function $\underline{\alpha}_k^T \underline{\underline{\mathbf{X}}}$ has a maximum variance subject to be uncorrelated to $\{\underline{\alpha}_1^T \underline{\underline{\mathbf{X}}}, \underline{\alpha}_2^T \underline{\underline{\mathbf{X}}}, \ldots, \underline{\alpha}_{k-1}^T \underline{\underline{\mathbf{X}}}\}$.

Furthermore vectors $\underline{\alpha}_k$ should be chosen such that they have the unit length, i.e. $\underline{\alpha}_k^T \underline{\alpha}_k = 1$.

It is then said that the new variables $\{\underline{\alpha}_1^T \underline{\underline{\mathbf{X}}}, \underline{\alpha}_2^T \underline{\underline{\mathbf{X}}}, \ldots, \underline{\alpha}_k^T \underline{\underline{\mathbf{X}}}, \ldots\}$ are the *principal components* (PCs) of $\underline{\underline{\mathbf{X}}}$. There can be a maximum of $p$ of these linear functions, but it is hoped that $m << p$ of them will account for most of the variation of $\underline{\underline{\mathbf{X}}}$.

**How do we find PCs ?**

The PCs are derived from two conditions:

- unit vector length

$$\underline{\alpha}_k^T \underline{\alpha}_k = 1 \tag{4.12}$$

- and maximization of variance

$$var[\underline{\alpha}_k^T \underline{\underline{\mathbf{X}}}] = \underline{\alpha}_k^T \underline{\underline{\mathbf{S}}} \underline{\alpha}_k, \tag{4.13}$$

where $k = 1, \ldots, p$ .

The PCs are calculated by combining the two conditions and replacing the linear combination $\underline{\alpha}_k^T \underline{\underline{\mathbf{X}}}$ by a new variable $\underline{\mathbf{z}}_k$, [Jolliffe, 2002, p.4]. Furthermore, using the form of the eigen decomposition as in Equation 4.6, it follows

$$\underline{\underline{\mathbf{S}}} \underline{\mathbf{e}}_k = \lambda_k \underline{\mathbf{e}}_k, \tag{4.14}$$

$$\underline{\mathbf{e}}_k = \underline{\alpha}_k, \tag{4.15}$$

$$\lambda_k = var[\underline{\mathbf{z}}_k], \tag{4.16}$$

$$\underline{\mathbf{z}}_k = \underline{\alpha}_k^T \underline{\underline{\mathbf{X}}}. \tag{4.17}$$

It can be deduced that for $k = 1, ..., p$ the k$^{th}$ PC is equal to $\underline{\mathbf{z}}_k = \underline{\alpha}_k^T \underline{\underline{\mathbf{X}}}$, where $\underline{\alpha}_k$ is an eigenvector of the covariance matrix $\underline{\underline{\mathbf{S}}}$ corresponding to its k$^{th}$ largest eigenvalue $\lambda_k$. As $\underline{\alpha}_k$ is a unit vector, then $\lambda_k$ is equal to the variance of $\underline{\mathbf{z}}_k$, [Jolliffe, 2002, p.3].

In other words, given a *p-dimensional* space $\underline{\underline{\mathbf{X}}}$ with a known covariance matrix $\underline{\underline{\mathbf{S}}}$, it is possible to define a new *m-dimensional* space, $m<<p$, the basis of which is formed by the $m$ eigenvectors of $\underline{\underline{\mathbf{S}}}$, which correspond to the $m$ largest of its eigenvalues. The eigenvalues of $\underline{\underline{\mathbf{S}}}$ represent variations along the basis vectors, i.e. variances of new variables $\underline{\mathbf{z}}_k$ called principal components. This new *m-dimensional* space will capture most of the variation of the original space $\underline{\underline{\mathbf{X}}}$.

### 4.2.2 PCA representation of video data

A grey-level image may be represented as a vector $\underline{\mathbf{x}} = (x_1, ..., x_p)$ where $x_i$ represents the grey-level of the i$^{th}$ pixel in an image containing $p$ pixels. The static background image obtained by a fixed camera may be formulated as an *appearance model* [Oliver et al., 2000] where any image instance may be represented as a linear "deformation" from an average background image $\underline{\hat{\mathbf{x}}}$

$$\underline{\mathbf{x}} = \underline{\hat{\mathbf{x}}} + \delta\underline{\mathbf{x}} \tag{4.18}$$

$$\underline{\mathbf{x}} = \underline{\hat{\mathbf{x}}} + \underline{\underline{\mathbf{P}}}\,\underline{\mathbf{b}} \tag{4.19}$$

$$\underline{\mathbf{b}} = \underline{\underline{\mathbf{P}}}^T \delta\underline{\mathbf{x}} \tag{4.20}$$

where the $pxm$ matrix $\underline{\underline{\mathbf{P}}}$ represents a linear transformation matrix. By its definition $\underline{\underline{\mathbf{P}}}$ is a real orthogonal matrix whose column vectors form an orthonormal basis of the Euclidean space $\mathbf{R}^p$. It is a square matrix whose transpose equals its inverse.

$$\underline{\underline{\mathbf{P}}}\underline{\underline{\mathbf{P}}}^T = \underline{\underline{\mathbf{P}}}^T\underline{\underline{\mathbf{P}}} = \underline{\underline{\mathbf{I}}} \tag{4.21}$$

The transformation matrix $\underline{\underline{\mathbf{P}}}$ can be derived from *principal component analysis* of a large training set $\underline{\underline{\mathbf{X}}}$ of $n$ image observations. In this case, the column vectors of $\underline{\underline{\mathbf{P}}}$ are the eigenvectors of the covariance matrix of $\underline{\underline{\mathbf{X}}}$. Then, the vector $\underline{\mathbf{b}}$ of length $m$ is the projection of image $\underline{\mathbf{x}}$ on the $m$-dimensional eigen-space, where $m << p$ is the number of most significant components of the training dataset.

### 4.2.3 PCA of high-dimensional multivariate Gaussian data

This section describes some simple examples of PCA of high-dimensional multivariate datasets with Gaussian distribution of variables. It demonstrates the relationship between the variability of the data and that of the PCs. Those PCs corresponding to largest eigenvalues of the dataset covariance matrix capture most of the variability of the dataset.

Three types of datasets are simulated. In the first dataset all variables follow Gaussian distribution with standard deviation $\sigma = 2$. In the second dataset half of the variables have $\sigma_1 = 1$ and the other half are with $\sigma_2 = 2$. Finally, in the last dataset each third of the variables has a different standard

deviation $\sigma_1 = 1$, $\sigma_2 = 2$, and $\sigma_3 = 0.5$. In all cases the distribution mean is zero. All three datasets have the same number of variables $p = 10^2$ and same number of observations $n = 10^6$.

Figure 4.1 illustrates the behaviour of eigenvalues for three datasets. The flat line corresponds to the first dataset where all variables follow the same distribution. The observations form a multivariate Gaussian cloud with equal spread along all directions. Therefore, in theory, in the eigen-space all PCs vary equally and the corresponding eigenvalues are of the same magnitude $\sigma^2 = 4$. The slight slope in the graph is a result of the finite number of observations in the dataset and an accidental alignment of observations in some directions; the more observations, the more densely populated Gaussian multivariate cloud and the closer to equal variations along its axes. In the second case, the eigenvalues take two magnitudes, $\sigma_1^2 = 1$ and $\sigma_2^2 = 4$, reflecting the nature of the variability in the dataset. Similarly, in the third dataset, the observed eigenvalue magnitudes are $\sigma_1^2 = 1$, $\sigma_2^2 = 4$, and $\sigma_3^2 = 0.25$.

### 4.2.4 Dataset size

This section discusses the effect of the dataset size to the behaviour of eigenvalues of its covariance matrix. Let a matrix $\underline{\underline{\mathbf{X}}}$ represent a dataset of $n$ observations of $p$ random variables. We consider a special case where the variables take values randomly drawn from a Gaussian distribution with mean $\mu$ and standard deviation $\sigma$. For the purpose of the experiment the number of variables is set to $p = 10^2$, the number of observations takes values

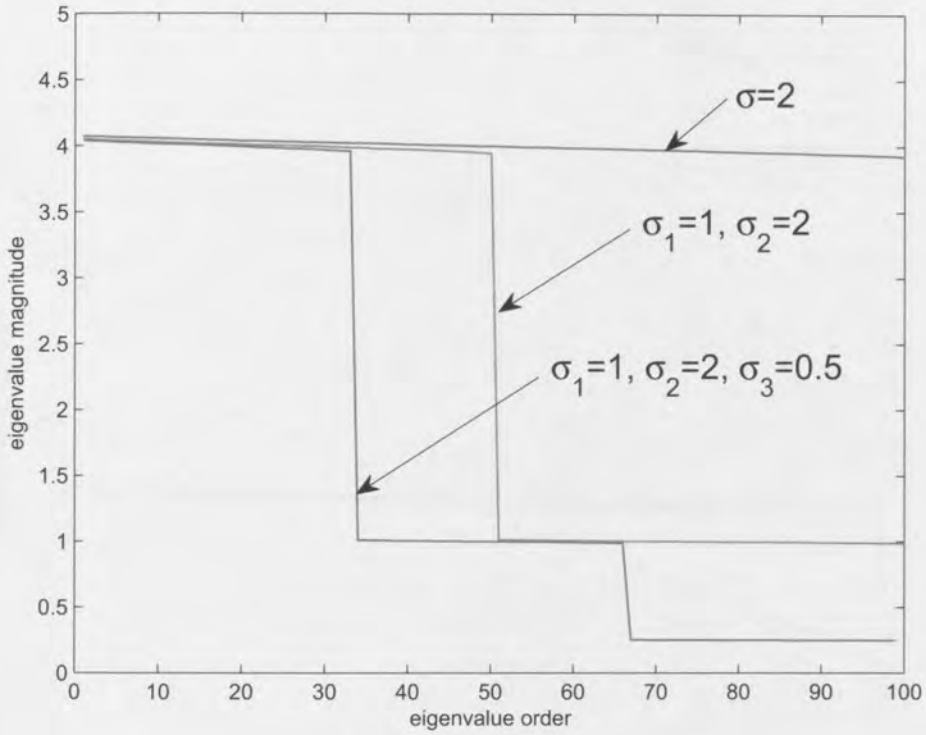



Figure 4.1: Eigenvalue magnitude

$n = \{10^3, 10^4, 10^5, 10^6\}$. The standard deviation is set to $\sigma = 2$ and the mean to $\mu = 0$. In Figure 4.2, we observe magnitudes of eigenvalues as the number of observations decreases.

As all $p$ variables have an identical standard deviation, all eigenvalues are expected to be of equal magnitude. However, for relatively small dataset sizes, compared to the number of variables, the slope of the eigenvalue magnitude is steeper than for larger datasets. The reason for this behaviour is accidental alignment of randomly generated data which is observed in datasets with a small number of observations relative to the number of variables (even though drawn randomly from a Gaussian distribution). This accidental regularity is reflected in the distribution of the eigenvalue magnitudes, where

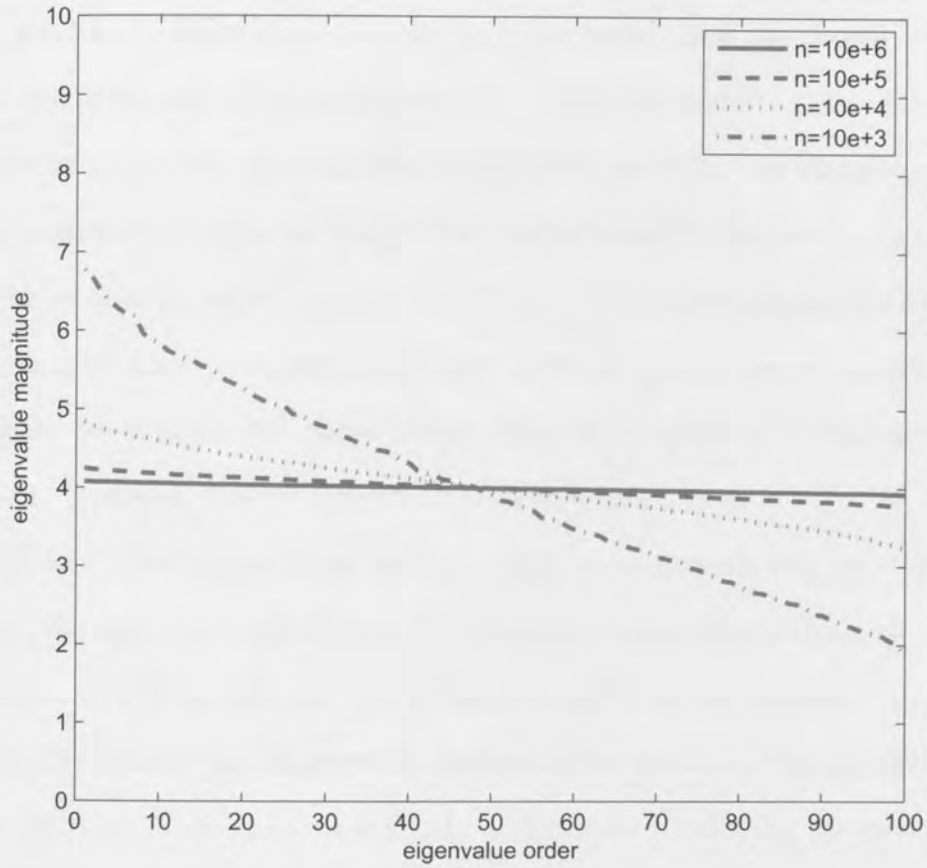



Figure 4.2: Eigenvalues depend on the dataset size

the most significant eigenvalues correspond to modes of variation observed in the data. As the new observations are added and the dataset size grows, any alignment of the data becomes less important. The eigen-model has more available information about the data and learns that all variables vary randomly following the same Gaussian distribution, i.e. there are no dominant modes of variation and eigenvalues become of more similar magnitude. For larger datasets the eigenvalue magnitude plot is flatter, thus maintaining the total sum of magnitudes constant.

The same type of graphs can be drawn for different dataset sizes when the tested ratios between the number of observations and the number of variables are preserved. At the ratio of $10^3$, for a random multivariate Gaussian dataset with a fixed standard deviation of all variables, the eigenvalues are of approximately constant magnitude, dashed line in Figure 4.2. It can be concluded that in order to avoid the effects of accidental regularities and obtain an accurate representation of random data, the number of observations needs to be at least $10^3$ times larger than the number of variables in the dataset. However, this is difficult to achieve in real cases.

For real video scenes in general, in order to adequately capture the variability of data in the eigen-space, the number of observations (frames) should be sufficiently large relative to the number of variables (pixels). Unfortunately, the dataset size is generally limited by the available storage and computation cost. This problem is firstly addressed by reducing the number of variables by dividing the original frame into smaller regions of pixels. Secondly, the error introduced by the insufficient number of observations can be tolerated to a certain extent. Due to the nature of the outdoor sequences, the major variability in the background is largely caused by the global lighting changes due to weather conditions. This variability is expected to be captured by the first few largest eigenvalues, where the rest of the eigenvalues are insignificantly small. As illustrated in the next section, the ratio between the few most significant eigenvalues and the rest may be in excess of few orders of magnitude for a small region of 64 by 64 pixels. The effect caused by having a relatively small number of observations (see Figure 4.2) is not expected to have any significant effect to this already very large ratio. Therefore a

smaller number of observations could be accepted without compromising the accuracy of the eigen representation of the original data variability. Investigations persuaded us that even using as little as twice as many observations as the number of pixels produces acceptable representation. The error of reconstruction did not significantly increase when the number of observations was reduced.

### 4.2.5 Eigenvalue magnitude

The previous section illustrated the behaviour of eigenvalue magnitudes in a special case of multivariate random Gaussian data with identical standard deviation of its variables. Generally, a real dataset is characterised with more diverse variability which will be reflected by the scale of the eigenvalue magnitudes of its covariance matrix. Eigenvalues are characteristic of a dataset and their magnitudes reveal the underlying nature of the variability of the data. This section illustrates the expected scale and the behaviour of eigenvalue magnitudes for real video datasets.

Let a matrix $\underline{\underline{\mathbf{X}}}$ represent a dataset of $n$ observations of $p$ random variables. that the largest eigenvalue, denoted as $\lambda_1{=}\lambda_{max}$, is much larger than all the rest, so that $\lambda_k{\approx}0$ for $k = 2, \ldots, p$. (This means that there is one dominant mode of variation in the dataset.) Then Equation 4.11 becomes

$$\sum_{i=1}^{p} \sigma_{ii}^2 = \lambda_{max} \tag{4.22}$$

The maximum magnitude of the dominant eigenvalue when all other eigenvalues are insignificantly small is determined by the number of variables and

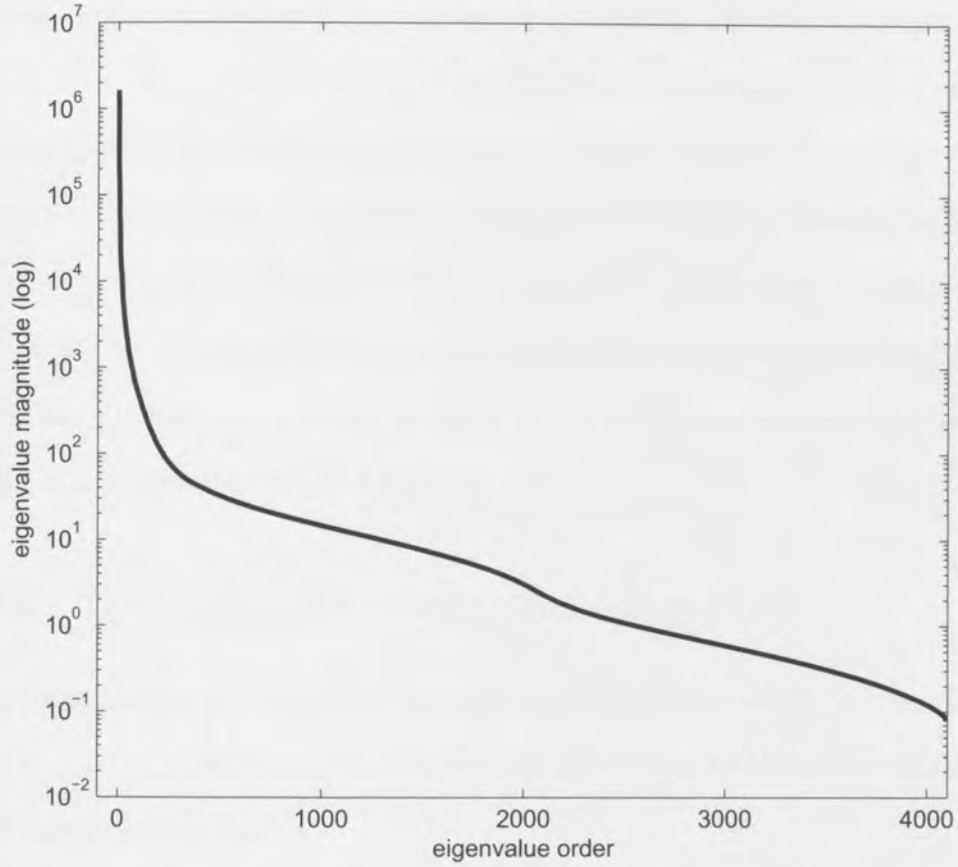



Figure 4.3: Eigenvalues of a real video dataset

their standard deviation.

In other words, if matrix $\underline{\underline{\mathbf{X}}}$ represents a set of $n$ images of $p$ pixels each of which on average has the standard deviation of, for example, five grey-levels, the magnitude of the largest eigenvalue will be of the order of $10^6$ for images of size 64-by-64 pixels.

Real outdoor video surveillance datasets are expected to have several significant eigenvalues larger than all the rest by few orders of magnitude. These significant eigenvalues model the large range of lighting variations typical for such sequences. Experiments have illustrated that such a video sequence may display great differences in the eigenvalues magnitude, the largest one being

typically greater than the following few by as much as an order of magnitude (see the example in Figure 4.33). This feature imposes major constraints on the dimensionality reduction when deciding on the number of principal components to be retained to represent the dataset without significant loss in its variability. Figure 4.3 illustrates an example of a real dataset, described in Section 4.5.1. The graph shows very few significant eigenvalues the largest of which has the magnitude of the order of $10^6$. The eigenvalues were calculated for an image region of 64 by 64 pixels.

### 4.2.6 Dimensionality reduction using PCA

High-dimensional problems are complex and computationally costly. However, it is often possible to discard some of the available data without significant loss in information.

Provided that the variability in data is real and not a product of insufficient dataset size, principal components indicate the variability of the data. The error due to the insufficient amount of available data can be avoided by dividing the original image into smaller regions and analysing the each region individually. Then, the higher the magnitude of the eigenvalue the larger the proportion of the total variability that is contained in the corresponding PC. Dimensionality reduction is based on retaining $m$ number of PCs, which correspond to the first $m$ ordered eigenvalues, where $m$ is significantly smaller than total number of variables $p$, and under the condition that most of the variability in the data is still captured.

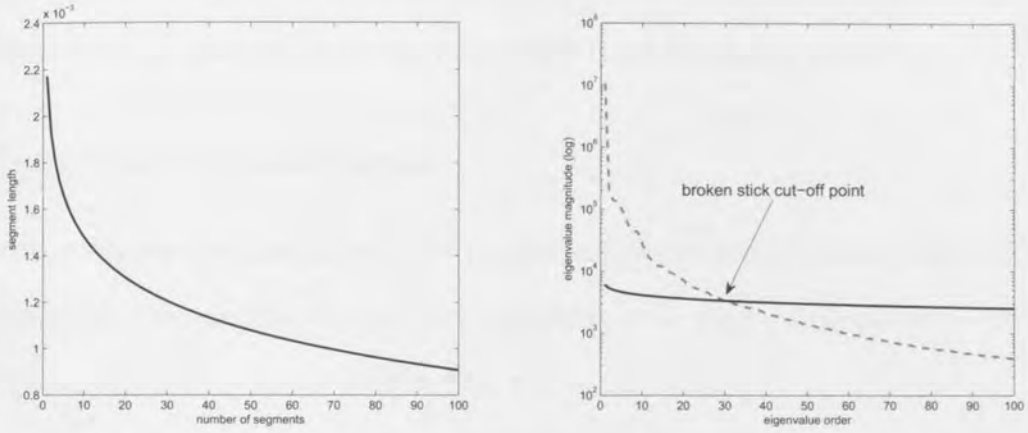
### Cut-off rules

A number of methods, intuitive rather than formal, which determine how small $m << p$ can be, has been described in the literature [Jolliffe, 2002, p.93]. These rules normally retain $m$ dimensions where the sum of the variations of $m$ most significant PCs is greater than some high proportion of the total variance of the data. In general, these rules retain many more PCs than required [Jolliffe, 2002, p.93]. The *broken stick* rule, on the other hand, defines the cut-off dimension considering the relative magnitude of successive ordered eigenvalues.

The *broken stick* rule states that if a stick of unit length is broken randomly in $p$ segments, than the length of the k$^{th}$ longest segment is expected to be $l_k$, as given in Equation 4.23. The rule compares variances with proportions $l_k$ and decides that if the variance of the k$^{th}$ PC is larger than $l_k$ its contribution to the total variation is significant and the PC is retained.

$$l_k = \frac{1}{p} \sum_{j=k}^{p} \frac{1}{j} \tag{4.23}$$

Figure 4.4a shows an example of distribution of segment lengths for the unit-length stick when the number of segments is 4096. Applied to eigenvalues of an image region of a real video dataset (described later in Section 4.5.2) with 4096 pixel variables (that is a 64x64 pixel image) the *broken stick* rule determines the cut-off at the $30^{th}$ largest eigenvalue. In Figure 4.4b the dashed line represents eigenvalue magnitudes of the video data and the full line is the *broken stick* distribution; graphs are plotted on the logarithmic scale for better visibility. In other words, $m = 30$ PCs capture most of the




(a) Unit-length *broken stick* distribution (b) Video data cut-off dimension

Figure 4.4: *broken stick* rule

variability in this particular dataset according to the *broken stick* rule. It can be shown that this corresponds to approximately 95% of total data variance in this case.

Table 4.1 sumarises the choice of a cut-off dimension using various rules for the same video dataset.

| rule | cut-off dimension |
|---|---|
| average eigenvalue | 81 |
| 80% total variance | 5 |
| 90% total variance | 13 |
| 99% total variance | 313 |
| *broken stick* rule | 30 |

Table 4.1: Cut-off number of dimensions

Unfortunately, these cut-off dimension rules are intuitive. The *broken stick* rule is often said to be the most adequate for real data. However, there is no universal answer. Which rule gives the most suitable cut-off point will depend on the nature of the dataset. Having in mind the aim of modelling

variations of a video sequence background, the following section explores an alternative approach to selecting how many dimensions are retained.

### Hyper-sphere of backgrounds

Let us consider a video dataset of $n$ frames and $p$ pixels per frame. The eigen analysis transforms this dataset to a structure of $n$ observation points in the $p$-dimensional eigen-space centered on the mean image.

In real situations, a video sequence will inevitably contain background frames contaminated by various types of noise (intrinsic camera noise, camera movement, objects on the camera lens) or foreground objects present in the scene. In this section we make an assumption that the distribution of the backgrounds in the eigen-space is unimodal, i.e. represented by a single hyper-sphere. In this case the contaminated observations are expected to fall farther away from the centre of the eigen-space, whereas the true background images would be grouped in some way closer to the centre. Of course, in many real cases the eigen-space is more likely to contain multiple modes where observations are grouped in a number of clusters; this multi-modal approach is discussed later in Section 4.3.

All uncontaminated purely background frames define a volume in the $p$-dimensional eigen-space populated exclusively with background observation points. Let us imagine that this volume is in effect a hyper-sphere in the normalised eigen-space, centered on the mean background image. This sphere will then contain all possible variations of the background and we refer to it as the *hyper-sphere of backgrounds*. Consequently, if an observation point falls inside the hyper-sphere of backgrounds it corresponds to a true background

image; otherwise, it is likely that the corresponding image contains data other than background.

The limits of the hyper-sphere of true backgrounds may then be defined as, for example, having the radius three times the standard deviation of the available data, or containing a certain large percentage of total data points. These definitions are arbitrary and tests are needed to determine the most suitable. Therefore, as the dimensionality of the hyper-space increases, the distance, $\chi$, between the centre and a point in the hyper-sphere also increases according to Equation 4.24.

$$\chi^2 = \sum_{i=0}^{i=m} r_i^2, \tag{4.24}$$

where $m = 1, ..., p$, and $r_i$ are projections of the radius on each of $m$-dimensions of the hyper-space. Therefore, in spaces of reduced dimensionality, as the number of retained principal dimensions, $m << p$, increases, the observation points are moving away from the centre of the hyper-sphere.

The aim is to represent the video data in the reduced eigen-space in such a way that, for the smallest possible space dimensionality $m << p$, true background observations remain inside, and all other observations outside the hyper-sphere of backgrounds. The problem of finding the appropriate $m$ is now reduced to finding the threshold hyper-sphere radius in $m$-dimensional space for which this condition is satisfied. The following experiment illustrates the increase of the distance of observation points form the mean when the number of retained dimensions increase. It shows how this effect can be used in classification of observations.

Assuming that the threshold radius of the hyper-sphere of backgrounds contains 95% of the available data[2], we can determine the number of dimensions, $m << p$, for which the hypher-sphere boundary separates the background from the non-background observations. At any of these true background observation points the distance to the mean is smaller than the threshold radius, whereas the distances of contaminated observation points are larger than the threshold radius. In other words, the true background points are inside the hyper-sphere, while the contaminated points are outside.

In Figure 4.5, the blue line on the graph represents the threshold radius of the hyper-sphere of backgrounds which encompasses 95% of all available observations of an image region of a real outdoor basckground sceene (region {7,7} in Figure 4.29 of the dataset described in detail in Section 4.5.2. Three types of classified test observations are considered: a known background (black line), a new unseen background (red line) and an observation contaminated with some foreground data (green line). The graphs show the increase of the Mahalanobis distance between each observation point and the mean background observation in the centre of the hyper-space.

For relatively low space dimensionality, under about 20 in this case (see the inset image), the model, having a very limited knowledge about the dataset, incorrectly places the contaminated observation inside the sphere with backgrounds (green line). As the number of dimensions increases the higher proportion of total variability of the dataset is captured. Therefore, the more PCs are retained, the eigen-model will better describe the data and

[2]The radius of the hyper-sphere is calculated using the chi-squared distribution with $m$ degrees of freedom.

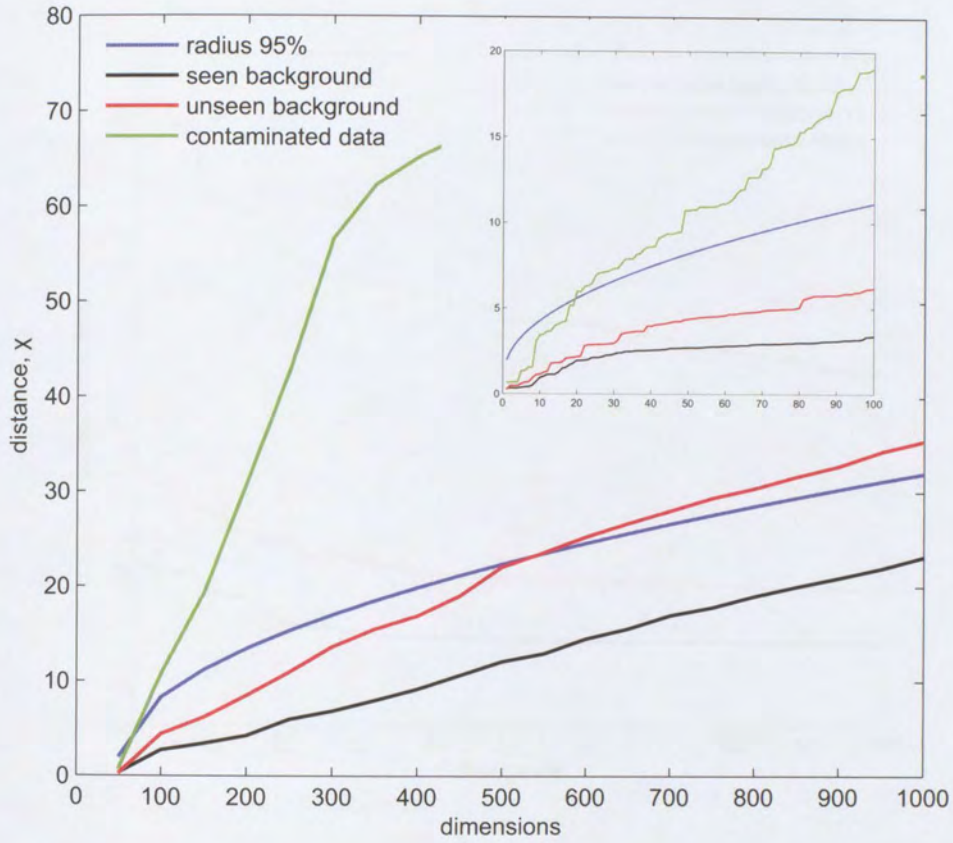



Figure 4.5: Threshold dimension

better distinguish between the true and contaminated backgrounds. The contaminated observations are placed farther away from the hyper-sphere of backgrounds. The cut-off dimension, $m << p$, needs to be large enough to place the contaminated observation outside the hyper-sphere of backgrounds. As it can be seen from the inset graph, the green line crosses the blue line at about 20 dimensions.

The eigen-model was initially created from a limited amount of available, or already seen, true backgrounds. As the model includes more dimensions and captures more information about the training dataset, it is better

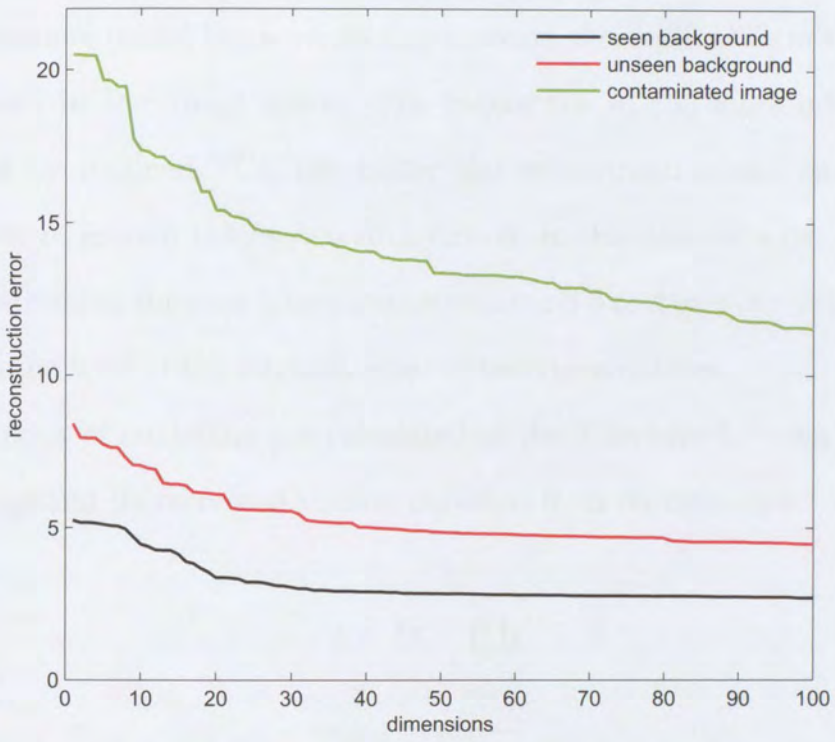



Figure 4.6: Reconstruction error

equipped to pick out new unseen data, even purely background. Eventually, for a certain higher number of dimensions $m$, the true background observation unknown to the model is found beyond the radius of the threshold sphere (red line crosses the blue line at about 550 dimensions). However, the aim is to keep the unseen background observation inside the hyper-sphere of backgrounds. Hence, the cut-off dimension, $m$, must not be too large.

The known observation, a background from the training dataset, remains inside the hyper-sphere of backgrounds for all dimensions (black line).

Figure 4.6 illustrate how the choice of the cut-off dimension affects the ability of the model to reconstruct new data. The graphs represent $e_{rms}$, the

*rms* reconstruction error per pixel, calculated in grey-levels, after a reduced $m$-dimensional model has been used to recreate observations form the eigen-space back in the image space. The higher the $m$ the more information captured by retained PCs, the better the reconstruction and smaller the error. For $m$ around the *broken stick* cut-off, in this case $m = 34$, the error of reconstructing the seen image settles at around 3 to 4 grey-levels per pixel, which is the level of the intrinsic noise of the video dataset.

The error of modelling $\underline{e}$ is calculated as the difference between the original image and its recreated version obtained from its eigen-space representation.

$$\underline{e} = \underline{\delta \mathbf{x}} - \underline{\underline{\mathbf{P}}}\,\underline{\mathbf{b}} \tag{4.25}$$

$$e_{\,rms} = \sqrt{\frac{\underline{e} \cdot \underline{e}}{p}} \tag{4.26}$$

To conclude, the choice of the cut-off dimension is the one of a compromise. The cut-off dimension must be large enough to provide enough knowledge about the data and pick out foreground frames, but not too large so it can still recognise unseen background variations as true backgrounds. The example suggests that the *broken stick* rule provides an adequate choice of the number of dimensions of the reduced space model.

### 4.2.7 Discussion

High-dimensional spaces are complex to model and costly to analyse. By means of eigen analysis it is possible to define a smaller set of dimensions which will capture most of the variation of the original space.

The number of observations in the dataset relative to the number of

retained dimensions affects the accuracy of the eigen-model. In order to avoid the effects of accidental regularities and obtain an accurate representation of random data, the number of observations needs to be at least $10^4$ times larger than the number of variables in the dataset. This is very often difficult to achieve for real video scenes. The dataset size is generally limited by the available storage and computation cost, which may introduce errors in the representation of the data in the eigen-space.

It is generally not obvious how many dimensions should be retained in the reduced eigen-space. The rules which determine the cut-off dimension are intuitive and there is no universal answer as to which rule gives the most suitable cut-off point. The *broken stick* rule is often said to be the most adequate for real data.

The definition of the hyper-sphere of backgrounds, which separates true backgrounds from contaminated observation points in the eigen-space, offers an alternative approach to dimensionality reduction. As the dimensionality of the space increases the distances of the points from the mean also increase. For a particular cut-off dimension all true backgrounds will remain inside the hype-sphere, while the contaminated points will be pushed beyond its limits. The experiment suggests that the *broken stick* rule provides an adequate choice of the cut-off dimension at which the hyper-sphere provides a good separation of contaminated observations. (However, the hyper-sphere can only be calculated on batch data. Therefore, the selection of the cut-off dimension by means of the hyper-sphere is not suitable for an online algorithm.)

## 4.3 Modelling multi-modal background distributions

Previously, it was assumed that a dataset of observations forms a single hyper-sphere when projected to the reduced dimensionality eigen-space (uni-modal model). The number of dimensions of the reduced space was determined by the *broken stick* rule. The distance from the hyper-sphere centre provided the means for classification of a new observation as pure background or contaminated with foreground, where any observation point outside the fixed hyper-sphere radius is labeled as contaminated. However, in many real life cases contaminated observations may fall within the limits of the hyper-sphere and be wrongly labeled as background. This occurs when the contaminated portion of the image is relatively small compared to the image size, or the foreground pixels may be of similar grey-level as the background. Also, unseen global light variations may push a purely background observation point farther away beyond the hyper-sphere limits. Therefore, a more intelligent way of classification is needed.

Given the nature of outdoor scenes, it is expected that background observation points in the reduced eigen-space would gather in clusters of backgrounds of similar lighting conditions. It is possible to determine such clusters and perform the eigen-analysis on each cluster individually. The classification of observations as background/foreground may then be determined by proximity to any of the clusters rather than the general mean. If a point falls within the subspace of one or more clusters it is classified as background; otherwise it is considered as foreground.

### 4.3.1 Algorithm

The method for classification of test images as purely background or contaminated with foreground consists of the following steps: PCA, dimensionality reduction, clustering, PCA of each cluster, dimensionality reduction of each cluster and classification; as shown in Figure 4.7.

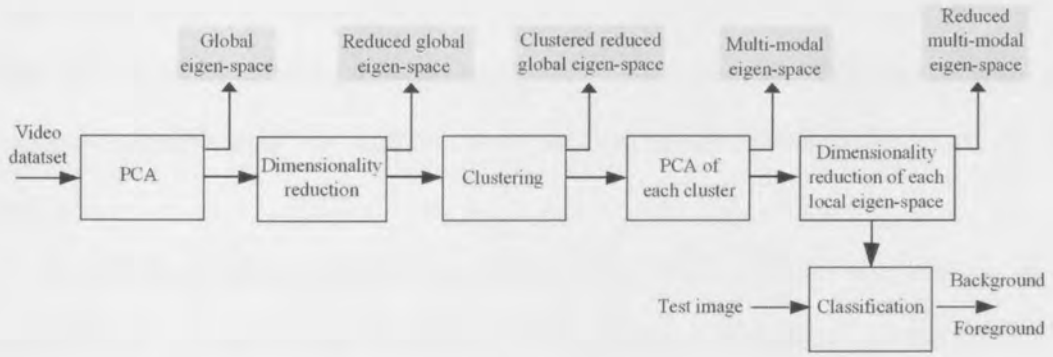



Figure 4.7: Classification algorithm

The training dataset is used to derive the unimodal global eigen-space, reduce the initial dimensionality of the data and to cluster the global eigen-space into a set of multiple modes. (For the purpose of this off-line approach the size of the training dataset is selected as twice as many pixels in the observed image region. For more details on selecting the training dataset refer to Section 4.2.4.)

A *global eigen-space* is derived from the training dataset and assumed to be unimodal. The reduced number of dimensions in this global eigen-space is determined by the *broken stick* rule. The multi-modal nature of the data is captured using the k-means method to cluster observations into subspaces

with similar background conditions. Small clusters, when the number of observations is smaller than a specified threshold, are merged with the nearest large cluster. The threshold is chosen as larger than the number of dimensions (or some multiple of it) to avoid accidental alignment of observations in any of directions. Each cluster is now regarded as an independent subset of observations and a further eigen analysis is performed on each individually. Now each *local eigen-subspace*, or *mode*, is represented with its mean and own set of eigenvalues and eigenvectors. At this point, dimensionality of eigen-subspaces may be further reduced. A multi-modal eigen-space is so created.

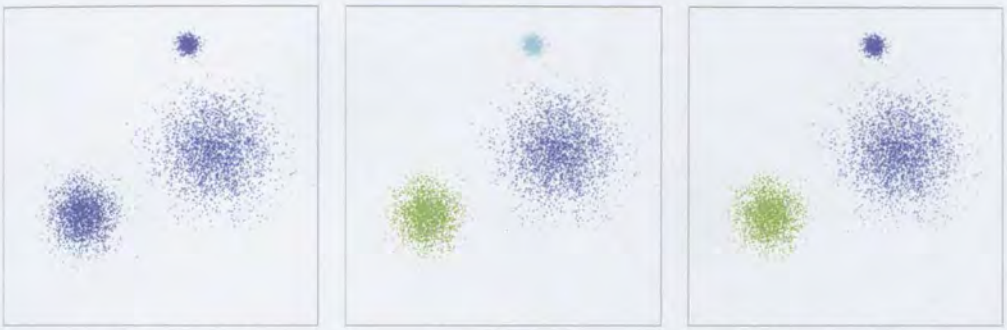
A test observation point is introduced into the clustered eigen-space and classified as a true background scene or contaminated with some foreground. The test point is projected into the locally normalised eigen-subspace of each cluster in turn. If the test point rests within a local hyper-sphere of one or more subspaces it is classified as background, otherwise it is likely to be contaminated by foreground.

### 4.3.2 Classification by subspace clustering

The following experiment illustrates the process of subspace clustering on a set of artificially generated Gaussian data.

Figure 4.8a shows all training observation points (blue) represented around the mean in the global eigen-space. Three clustered subspaces (blue, cyan and green) are identified in this eigen-space, as shown in Figure 4.8b. The smallest cluster (cyan), being below some threshold size, is merged with the

nearest large cluster (blue). After the merger, the updated eigen-space contains two clustered subspaces (blue and green), Figure 4.8c.



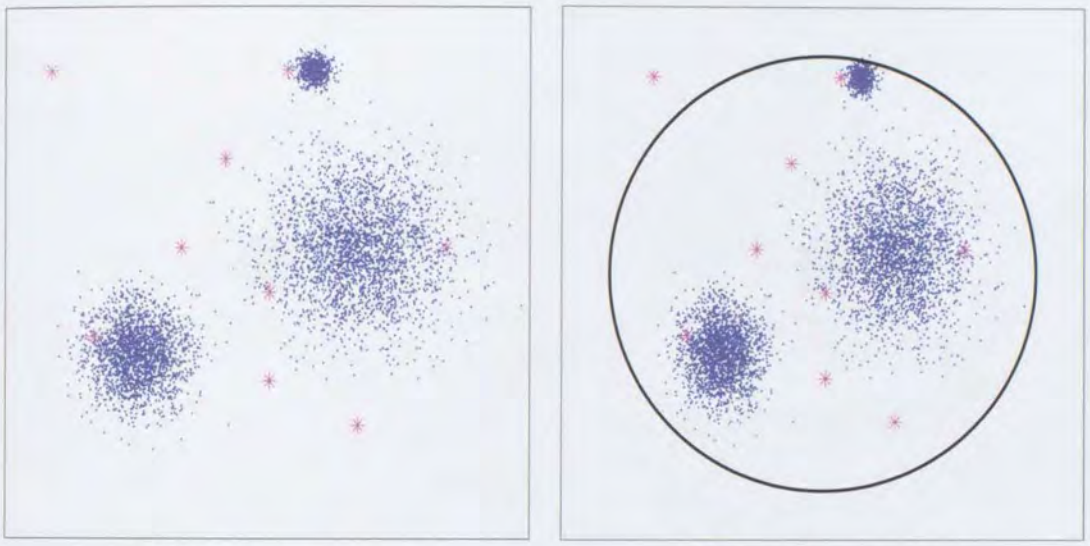
(a) Training set (b) Clusters (c) Adjusted clusters

Figure 4.8: Subspaces

Figure 4.9 shows the principle of the unimodal classification approach. A set of test observation points (magenta) is introduced into the normalised global unclustered eigen-space, Figure 4.9a. The global hyper-sphere, defined as the one which contains 95% of all training data, separates true backgrounds from contaminated observations. In this case, only one observation point is found outside the hyper-sphere and this test observation is classified as contaminated, Figure 4.9b. The unimodal approach results in one contamianted and eight background observations.

Figure 4.10 represents the multi-modal classification approach. As seen in Figure 4.8c, the observations in global eigen-space form two clusters (blue and green). A further eigen-analysis is performed on each cluster. For each cluster it is possible to define a local hyper-sphere which contains 95% of data in that particular cluster. Test observation points (magenta) are then projected onto the each eigen-subspace in turn, Figures 4.10a and 4.10b.

Figure 4.10a shows the eigen-subspace created by eigen-analysis of the



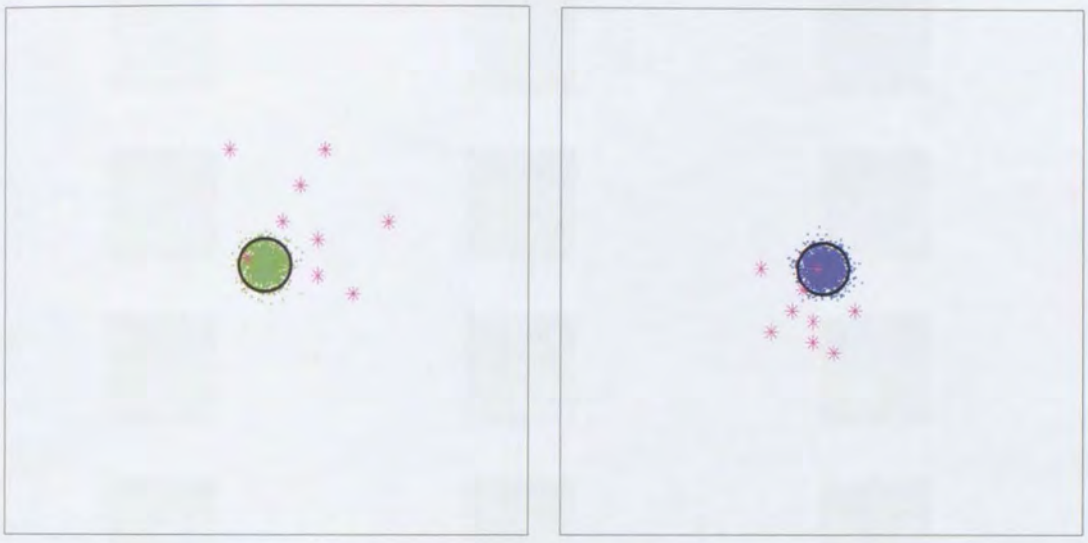
(a) Global eigen-space (b) Global hyper-sphere

Figure 4.9: unimodal classification

green cluster and its local hyper-sphere (black circle). This eigen-space exists independently from the blue cluster. Only test points (magenta) are projected onto this space for the purpose of classification, while the blue cluster points are ignored at this instance. Similarly, Figure 4.10b shows only the test points projections onto the eigen-subspace of the blue cluster.

If a projected test observation point falls within any of the local hyper-spheres it is classified as a pure background observation. Otherwise it is classified as contaminated. In this case, each eigen-subspace classifies only one observation as background. As it is a different point that falls inside the green and blue subspaces, the multi-modal approach results in seven contaminated and two background observations.

The described experiment has illustrated the advantage of classification by subspace clustering when a hyper-sphere is used to separate true backgrounds from contaminated observations. When contaminated proportion of



(a) First eigen-subspace (b) Second eigen-subspace

Figure 4.10: Multi-modal classification

an image is relatively small or similar to background grey-levels the contaminated test image is likely to fall inside the limits of the global hyper-sphere and consequently be wrongly classified as a true background. Clustering partitions the global eigen-space in subspaces which, when normalised individually, transform the global space into a set of local hyper-spheres of observations gathered by similar background conditions. This provides means of better separation between true backgrounds and contaminated points. In the experiment, the global hyper-sphere classified only one out of nine test points as contaminated compared to the subspaces method result of seven out of nine contaminated test points.

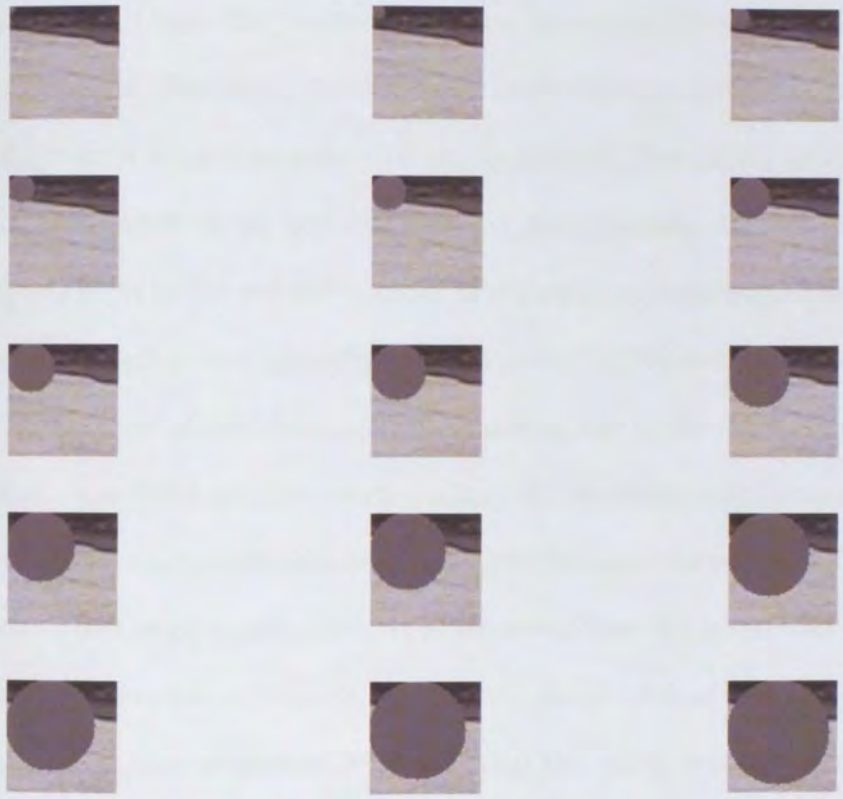


Figure 4.11: Contaminated observations

### 4.3.3 Contaminant size

The following example confirms the advantages of background image classification using eigen-subspaces.

A contaminant disc is introduced in a real life outdoor scene observation, otherwise containing purely background pixels. The contaminant increases in size taking up from 1% to 80% of the image, Figure 4.11.

In Figure 4.12 the blue graph represents the distance of the observation point from the mean in the unimodal global eigen-space. The green graph is the distance of the same observation point from the nearest cluster in the multi-modal eigen-subspaces representation. In both approaches, as the contaminated area grows, the observation point in eigen-space moves away

from the mean. Once the contaminant reaches some critical size the point is pushed beyond the limits of the eigen hyper-sphere which separates the true backgrounds from contaminated image points. The radius of the hyper-sphere is calculated using the chi-squared distribution with $m$ degrees of freedom, where $m$ is the cut-off number of dimensions determined by the *broken stick* rule applied to eigenvalues of the global unimodal eigen-space (blue line) or eigenvalues of the clustered subspaces in the multi-modal eigen-space (green-line). On both graphs, circles mark the contaminated proportion for which the observation is classified as background and crosses mark the observations classified as contaminated. It is observed that the unimodal approach tolerates the contaminant taking up the whole of 10% of the image before it is classified as contaminated, whereas with the multi-modal approach this critical size is only 0.5% of the image.

It can also be noted that up to the contaminant size of about 20% to 25% of the image, the point distance form the nearest cluster mean steadily increases. For larger contaminants we observe an unexpected effect of decreasing distance. This is due to the nature of the image and the contaminant. In this example an artificial disc of a uniform colour equal to the average grey-level of the image was used. When this contaminant covers a larger proportion of the image the observation starts approaching the mean. However, this effect is of no particular interest here as the contaminated point still remains beyond the limits of the hyper-sphere indicating the presence of the foreground object. In other words, the classification will still be accurate.

The experiment suggests that the multi-modal eigen-space approach allows for accurate classification of contaminated observations even when the

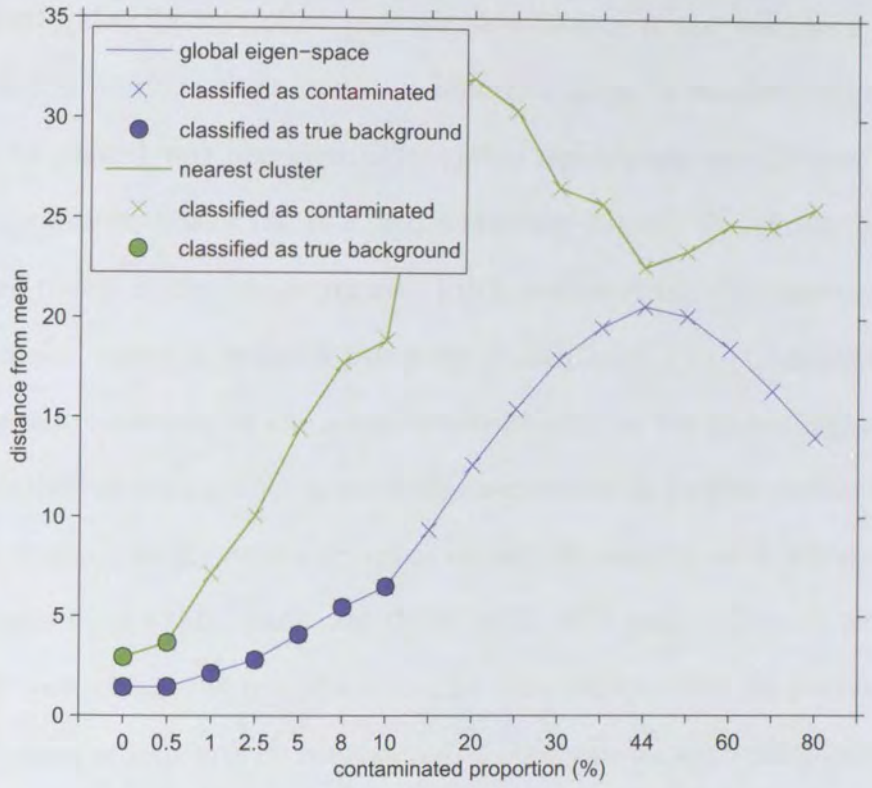



Figure 4.12: Classification by distance from the mean

contaminant is very small relative to the image size. In other words, it is potentially capable of detecting the presence of very small foreground objects of less than 1% of the image size.

### 4.3.4 Classification accuracy

The multi-modal classification approach proposes clustering in already reduced space, Figure 4.7. Once the clusters are formed a further dimensionality reduction may be performed on each eigen-subspace individually. The accuracy of the classification in multi-modal eigen-space is largely dependent on the dimensionality of the clustered eigen-subspaces. The following experiment illustrates this dependency.

A real video dataset of an outdoor surveillance scene, described in Section 4.5.2, is modelled by a multi-modal eigen-space. A smaller image region (64 by 64 pixels) was observed. The global eigen-space was derived from a training dataset, where the size of the training dataset was chosen as twice as many pixels in the image region. Prior to clustering, the dimensionality of the global space is reduced using the *broken stick* rule. Clustered eigen-subspaces are initially of the same dimensionality as the global eigen-space. The number of dimensions in each eigen-subspace is further reduced using several dimensionality reduction rules, namely the *broken stick* rule and some percentage of the total variation (95%, 98%, 99% and 100%). A set of 70 real life test images is introduced. The test set contains 40 purely background observations and 30 contaminated observations with contaminants of various sizes, shapes and grey-levels. The classification in the multi-modal eigen-space is performed on this dataset.

Figure 4.13 summarises results of the classification using the multi-modal eigen-space. The variable $k$ in the first column denotes the number of clusters and hence eigen-subspaces. The top table refers to the classification of the purely background test observations, whereas the bottom table refers to the classification of contaminated observations. The number of correctly classified observations per degree of dimensionality of the eigen-subspaces is indicated in columns. Here, the dimensionality of eigen-subspaces is the number of principal components retained in each subspace, determined according to the *broken stick* rule or some percentage of the total variation captured by retained dimensions. The dimensionality reduction rule is indicated at the top of each column.

| k, number of clusters | number of correct classifications out of 40 true background test observations | | | | |
|---|---|---|---|---|---|
| | methods for selection of number of dimensions in clusters | | | | |
| | *broken stick* | 95% | 98% | 99% | 100% |
| 10 | 40 | 40 | 35 | 33 | 15 |
| 15 | 40 | 40 | 32 | 30 | 10 |
| 20 | 40 | 40 | 34 | 29 | 7 |
| 30 | 40 | 40 | 27 | 19 | 2 |
| 45 | 40 | 39 | 32 | 22 | 0 |
| 64 | 40 | 40 | 40 | 33 | 1 |
| 90 | 40 | 40 | 40 | 32 | 0 |

| k, number of clusters | number of correct classifications out of 30 contaminated test observations | | | | |
|---|---|---|---|---|---|
| | methods for selection of number of dimensions in clusters | | | | |
| | *broken stick* | 95% | 98% | 99% | 100% |
| 10 | 5 | 13 | 19 | 22 | 29 |
| 15 | 5 | 15 | 19 | 21 | 29 |
| 20 | 6 | 7 | 18 | 22 | 30 |
| 30 | 5 | 12 | 19 | 24 | 30 |
| 45 | 9 | 13 | 18 | 22 | 30 |
| 64 | 12 | 15 | 16 | 22 | 30 |
| 90 | 0 | 13 | 13 | 17 | 30 |

Figure 4.13: Classification results

It can be seen that the higher the dimensionality of subspaces the worse the classification accuracy of true backgrounds, although the classification of contaminated observations improves. For example, in the case of $k = 30$ eigen-subspaces, when all subspace dimensions are retained only 2 out of 40 true backgrounds are correctly classified, compared with all 30 contaminated test observations being correctly classified. On the other hand, when the subspace dimensionality is reduced using the *broken stick* rule, all 40 true background observations are correctly classified, compared to only 5 out of 30 contaminated observations. The shaded areas in the Figure 4.13 are those with best classification results.

The results suggest that the maximum subspace dimensionality (the column marked as 100%) improves the detection of contamination. At the same time the classification of true background images deteriorates because the high dimensional eigen-model captures small variations caused by the noise, which then become greatly exaggerated. Therefore the model is over-trained causing the test set of true backgrounds to seem very different than the training data. Consequently, the true background observation points fall beyond the limits of the hyper-sphere and are wrongly classified as contaminated. Therefore, the high dimensionality of clustered eigen-subspaces maximises at the same time the true positive and the false alarm rates of classification. On the other hand, lower eigen-subspace dimensionality, i.e. obtained with *broken stick* rule, improves the classification of true backgrounds while the correct classification of contaminanted observations is greatly reduced.

It can also be observed that for larger $k$, the finer partitioning of the space tends to produce better classification results for contaminated observations.

In conclusion, the classification using high dimensional multi-modal clustered eigen-space is suitable for applications in which maximum true positive rate is essential regardless the false alarm rate. For high subspace dimensionality, equal to that of the global space, this method achieves 100% foreground detection compared to the 80% of contaminated images detected in non-clustered unimodal eigen-space. On the other hand, low dimensional clustered eigen-space is suitable for applications where the high true negative rate is preferred.

### Comparison with the unimodal model

If we were to use the unimodal case, where the dimensionality of the global eigen-space was reduced using the *broken stick* rule, the results would be the following - 38 correctly classified backgrounds and 24 correctly classified contaminated observations. While this may seem as a good compromise it is not the best solution in extreme cases when it is crucial to maximise the classification results in terms of the numbers of true positive rate and the false alarm rate. Furthermore, the number of dimensions in such unimodal model is 34 which is much more than in clustered subspaces of the multi-modal model, typically 4 to 7 dimensions. Let us assume that the number of dimensions of the unimodal model is now further reduced to, for example, 5 dimensions. In this case the classification results become very different: all of 40 backgrounds are correctly classified but only 1 of 30 contaminated observations is detected (compared with 5 to 12 in clustered spaces as shown in the bottom table in Figure 4.13). Therefore, the multi-modal model produces better detection of contaminated observations in low-dimensional spaces.

Admittedly, in the case of the off-line modelling, where the complexity and the processing time are not the major issues, the low-dimensionality is not crucial. However, the conclusions obtained by experimenting with the off-line approach are intended to be used for the development of an online version of the multi-modal algorithm. In this case the lower computational cost plays an important role and depends largely on the dimensionality of the model. The complexity of the algorithm is discussed in more detail in Section 5.2.6.

### 4.3.5 Discussion

In many real life cases observations may be wrongly labeled in the global eigen-space. When contaminated proportion of an image is relatively small or similar to background grey-levels the contaminated test image is likely to fall inside the limits of the global hyper-sphere and consequently be wrongly classified as a true background. The multi-modal eigen-space, clustered in subspaces of observations of similar background conditions, provides a more accurate classification.

The experiment suggests that the multi-modal eigen-space approach allows for accurate classification of contaminated observations even when the contaminant is very small, less than 1% of the image size. For comparison, the global unimodal model detects contaminants of the size larger than 10% of the image size.

The dimensionality of the clustered eigen-subspaces may further be reduced to simplify and speed up the processing. However, the dimensionality

reduction affects the classification results. When all eigen-subspace dimensions are retained the detection of contamination is maximised, while at the same time classification of pure backgrounds is poor. Similarly, the low dimensionality of eigen-subspaces improves the classification of pure backgrounds while that of contaminated observations deteriorates. Therefore, the trade-off between the true positive rate and the false alarm rate will determine the choice of the number of subspace dimensions.

## 4.4 Subsampling

Modelling of high-dimensional data is complex and slow to compute. Although the slow computation may not be a problem in off-line solutions, it will certainly be a constraint in an online approach. The results and conclusions obtained for the off-line modelling are envisaged to be extended to an online approach. Therefore, a possible further reduction in dimensionality is discussed here.

Although the dimensionality reduction by means of PCA could provide significant computational savings, further reductions may be achieved by reducing the amount of the processed data by subsampling a relatively small proportion of all the available data in each incoming frame to hypothesise the corresponding background. Subsampling a proportion of pixels in each image observation is expected to result in a lower computational cost.

There are many ways in which a number of pixels may be subsampled from an image. In any case, subsampling inevitably introduces loss of information about the original data. This is reflected in the increased error of the

image reconstruction from its eigen-space representation, when subsampling is performed at a lower subsampling rate.

This section discusses limitations of subsampling and its effect on the classification of background observations in the eigen-space.

### 4.4.1 PCA of subsampled data

Assume that a grey-level image is represented as a vector $\underline{\mathbf{x}} = (x_1, ..., x_p)$, where $x_i$ represents the grey-level of the i$^{th}$ pixel in an image containing $p$ pixels. A subset of $s$ pixels, $\underline{\mathbf{x}}^s = \{x_i; i = 1, s\}$, is subsampled form an image vector $\underline{\mathbf{x}}$. Using the Equation 4.20, the subset of pixels $\underline{\mathbf{x}}^s$ can be represented as a linear deformation from an average image as follows

$$\underline{\mathbf{b}}^s = [\underline{\underline{\mathbf{P}}}^{sT}\underline{\underline{\mathbf{P}}}^s]^{-1}\underline{\underline{\mathbf{P}}}^{sT}(\underline{\mathbf{x}}^s - \underline{\hat{\mathbf{x}}}^s) \tag{4.27}$$

where $\underline{\underline{\mathbf{P}}}^s$, $\underline{\mathbf{x}}^s$ and $\underline{\hat{\mathbf{x}}}^s$ are the subsampled versions of the linear transformation matrix $\underline{\underline{\mathbf{P}}}$, image vector $\underline{\mathbf{x}}$ and the average image $\hat{\mathbf{x}}$. The transformation matrix $\underline{\underline{\mathbf{P}}}$ is square and orthogonal. However, its subsampled version is no longer so. Therefore its transpose is not equal to its inverse and the pseudo-inverse must be used instead as shown in Equation 4.27. Then, $\underline{\mathbf{b}}^s$ is the projection of the subsampled vector $\underline{\mathbf{x}}^s$ onto the eigen-space defined by $\underline{\underline{\mathbf{P}}}^s$. The length of $\underline{\mathbf{b}}^s$ is defined by the number of principal components $m < s < p$.

### 4.4.2 Subsampling background images

One possibility is to randomly subsample pixels from the entire image. The resulting set of pixels is a subsampled representation of the image. If the

process of subsampling is repeated in the same manner, by randomly picking the same number of pixels, another representation of the same image is obtained. Therefore, every randomly subsampled set of pixels, for a fixed subsampling rate, results in a slightly different image representation. When processed in the eigen-space these image representations will never produce same results. In other words, random subsampling causes certain unreliability and the resulting calculations will be subject to undesirable variations. The extent of the variability depends on the subsampling rate: the fewer subsampled data, the larger variability of calculations and the less reliable results.

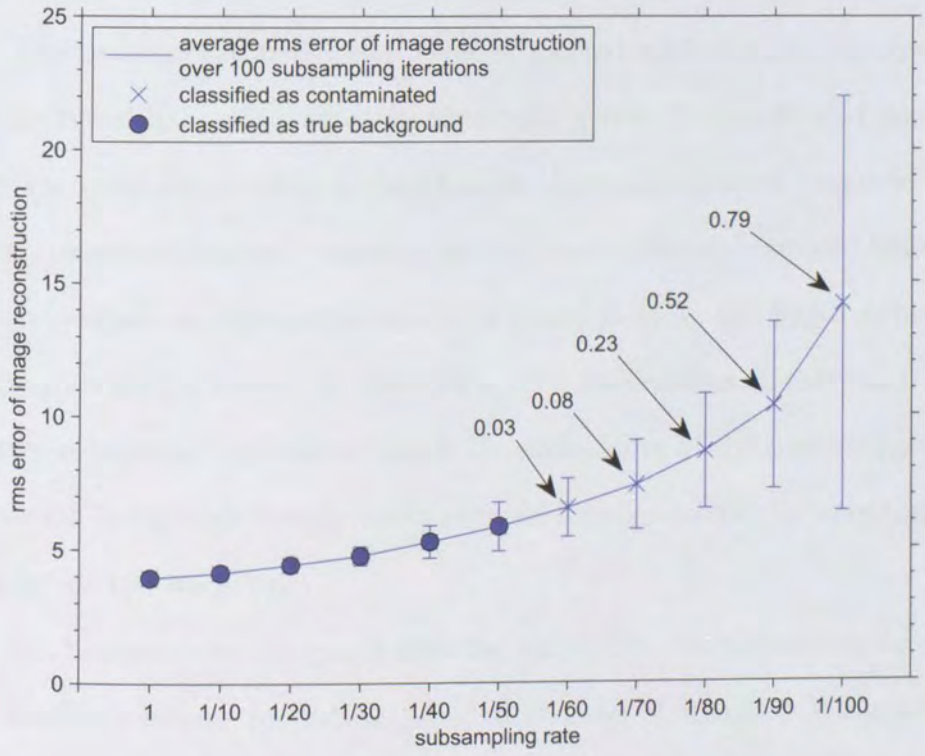



Figure 4.14: Limitations of subsampling

The following experiment illustrates the limitations of random subsam-

pling. A purely background image is subsampled 100 times at a number of different subsampling rates. For each subsampling rate, the reconstruction error (defined as the RMS per pixel difference between the original background image and the reconstructed one) is calculated at every iteration and the average value is plotted on the graph. The subsampled image is projected in the unimodal eigen-space and classified as true background, when it falls inside the hyper-sphere of backgrounds, or contaminated otherwise.

Figure 4.14 shows the error span bar plot of the average reconstruction error over 100 iterations, where a background image is subsampled at a decreasing subsampling rate. Subsampling rate of 1 means that every pixel is taken for processing, whereas 1/100 means that every $100^{th}$ pixel is subsampled. The average reconstruction error is plotted with the associated standard deviation bar. Also, at each subsampling rate, it is indicated whether the subsampled observation is classified as a true background - marked with a circle, or contaminated - marked with a cross. The subsampled observation is classified as contaminated if it is found outside the hyper-sphere of backgrounds during any of the iterations. The cross marks on the graph have numbers associated with them; that is the probability of the true background observation being classified as contaminated for that particular subsampling rate over all 100 iterations.

It can be seen from the graph that the variability due to random subsampling increases largely for subsampling rates lower than 1/40. For example, when the full set of pixels is taken into account (subsampling rate 1 on the x-axis) the reconstruction error is about 4 grey-levels per pixel (which is in the expected range of intrinsic camera noise at full frame rate). For a sub-

sampled set of randomly selected 1% of all pixels, subsampling rate of 1/100, the error increases to 14 grey-levels on average with a standard deviation of 8 grey levels due to random subsampling. Therefore, the reconstruction error and its variability impose constraints on the choice of the subsampling rate.

Assuming that the degradation of the reconstructed image above 50% relative to the one obtained from the full set of pixels, is not acceptable, the subsampling rate should be limited to 1/40. In other words, in this example it is not acceptable to subsample less than 2.5% of available data. At this subsampling rate the error of reconstruction remains below 6 grey-levels per pixel through all random subsampling iterations. Furthermore, due to high variability at lower subsampling rates, it becomes uncertain whether the observation point in global eigen-space remains within the hyper-sphere. For example, at subsampling rate of 1/60 there is 3% probability of the point escaping the hyper-sphere of backgrounds, whereas at the rate of 1/90 it is equally probable that the point resides within or beyond the hyper-sphere boundaries.

The experiment is repeated in the multi-modal eigen-space. In the uni-modal eigen-space, as the subsampling rate decreases and the subsampled proportion becomes smaller, the image representation in the eigen-space deteriorates and consequently moves farther from the region where true background observation are concentrated, beyond the hyper-sphere of backgrounds. Similarly, in the multi-modal space the point moves away from clusters of backgrounds. The multi-modal eigen-space classification is even more sensitive to the changes in the observed image. Therefore, as the subsampling rate decreases, the point will leave the true background region

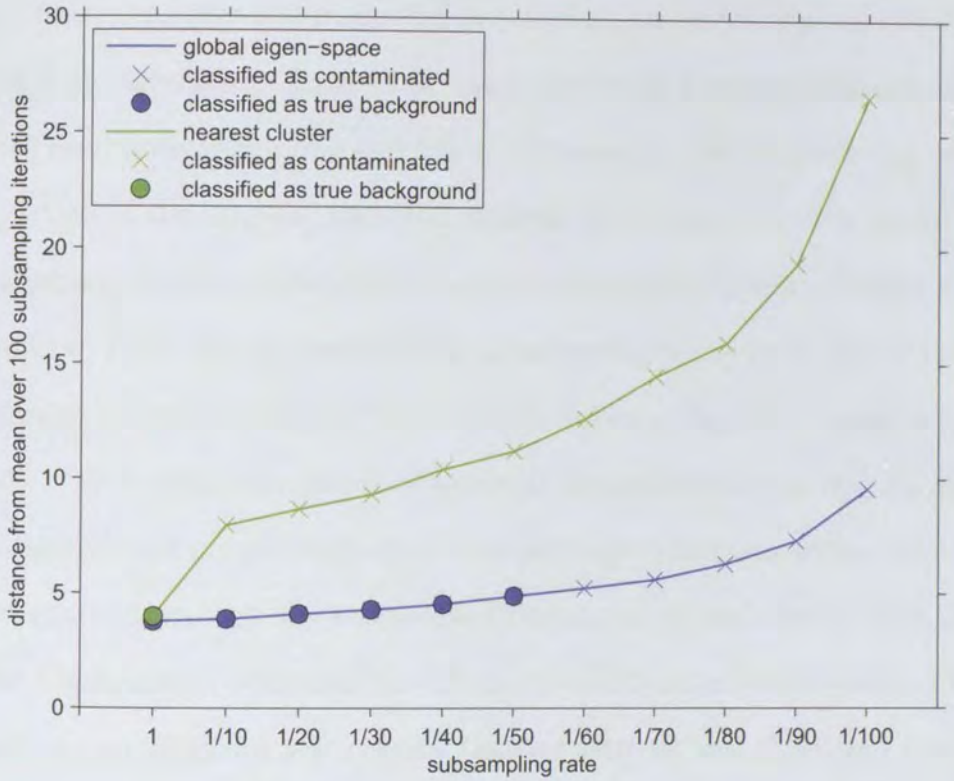



Figure 4.15: Average distance

sooner than in the case of the unimodal classification.

In Figure 4.15, the blue line shows the growing distance of the subsampled observation from the mean in the global eigen-space, whereas the green line represents the distance of the point from the nearest cluster in the multi-modal space. As before, the subsampling is performed over 100 iterations, and at each subsampling rate the average distance is plotted. It can be seen that, in the multi-modal eigen-space, the purely background point, or its subsampled representation, is wrongly classified as contaminated already at the subsampling rate 1/10, compared to 1/60 in the unimodal space. The clustered space is so sensitive to image distortions that the subsampled eigen-representation of the image is no longer recognised as the true background.

Subsampling provides means for great reductions in the processed amount of data. However, the subsampling rate is limited by high variations of results due to random subsampling and loss in information due to discarding of the proportion of the original data. Considering this constraints, the acceptable subsampling rate for unimodal eigen-space classification is determined as no lower than 1/40. When classifying in multi-modal eigen-space, due to higher sensitivity to image changes, the acceptable subsampling rate is much higher, above 1/10 in this case. In other words, it is possible to correctly classify a true background image using its subsampled representation in the unimodal eigen-space given that the subsampled proportion is not smaller than 2.5% of the entire image, compared to 10% in the multi-modal eigen-space. These values are obtained for a particular training dataset and algorithm settings (the number of clusters, the radius of the hyper-sphere of backgrounds) and will differ for different experiment setups. However, it is shown that it is possible to achive significant reduction in the processed amount of data while preserving the accurate classification of true background observations.

### 4.4.3 Subsampling contaminated images

Contaminated images in general contain objects which normally do not belong to the background. The background is modelled in order to extract these objects from the rest of the image.

To simplify and speed up the modelling process it is possible to use a subsampled proportion of the available data, as shown in Section 4.4.2. The background is represented by its subsampled version in the eigen-space. From

this representation and given the background model derived from the training data, it is possible to hypothesize the background area occluded by contaminating objects.

In order to obtain a reliable background representation the subsampled set of data should contain as much as possible information about the background. Therefore, it is desirable to subsample pixels from purely background areas of the image and avoid the contaminated regions. On the other hand, subsampling from the contaminated region is expected to cause image distortion and push the subsampled observation in the eigen-space beyond the hyper-sphere of backgrounds indicating the presence of the contaminant. A more controlled subsampling method is expected to provide means of subsampling from desired regions depending on the goal of processing.

### Subsampling from the entire image

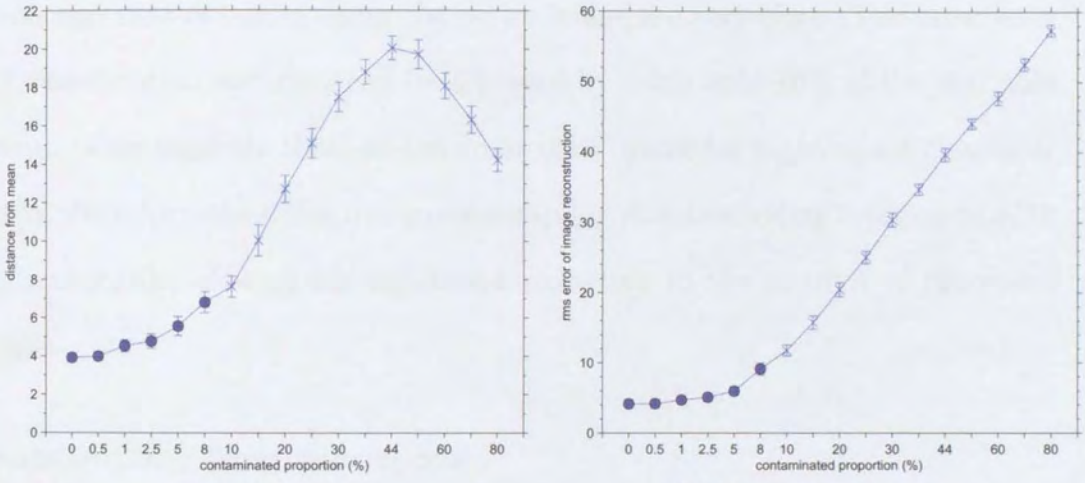
This section discusses a number of subsampling techniques. Random subsampling from the entire image will inevitably include some contaminated pixels which results in distortion of the subsampled background representation. Figure 4.16 illustrates an example of random subsampling from a contaminated image at subsampling rates from 1/10 to 1/40, when the contaminant occludes about 8% of the image.

Figure 4.16: Subsampling from entire image

For larger contaminants, the probability of subsampling from the contaminated region increases causing more significant distortions in the background representation. Figure 4.17 represents results of the unimodal eigen-space classification of a contaminated image randomly subsampled at the rate of 1/10, when the size of the contaminant increases. The average distance from the mean, Figure 4.17a, and the average reconstruction error, Figure 4.17b, are calculated over 100 iterations and plotted with their respective standard deviations. Both graphs increase with the increasing contaminant size. At the given subsampling rate, for contaminants larger than 8% of the image, the subsampled image representation is classified as contaminated as the point in unimodal eige-space is pushed outside the hyper-sphere of backgrounds. At the same time the reconstruction error per pixel, defined as the RMS difference between the true background without the contaminant and the reconstructed background obtained from the subsampled eigen-representation, increases above the expected intrinsic noise level. Similar effects can be observed for different choices of the subsampling rate. The effects of increasing distance and higher reconstruction error are more significant for lower subsampling rates.

Figure 4.18 represents results of the multi-modal eigen-space classification of the same contaminated image randomly subsampled at the rate of 1/10, when the size of the contaminant increases. It can be seen that in this case the subsampled image representation is subject to a high information loss, such that even the true background image with a 0% contamination is wrongly classified as contaminated.

Furthermore, it can be observed that Figure 4.17a is very similar to the




(a) Distance (b) Reconstruction error

Figure 4.17: Subsampled representation in unimodal space

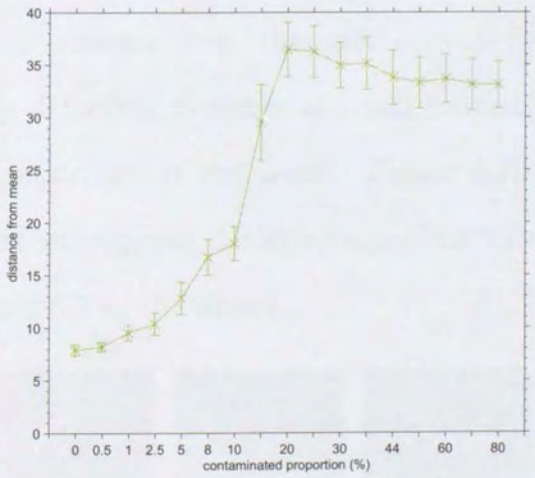



Figure 4.18: Subsampled representation in multi-modal space

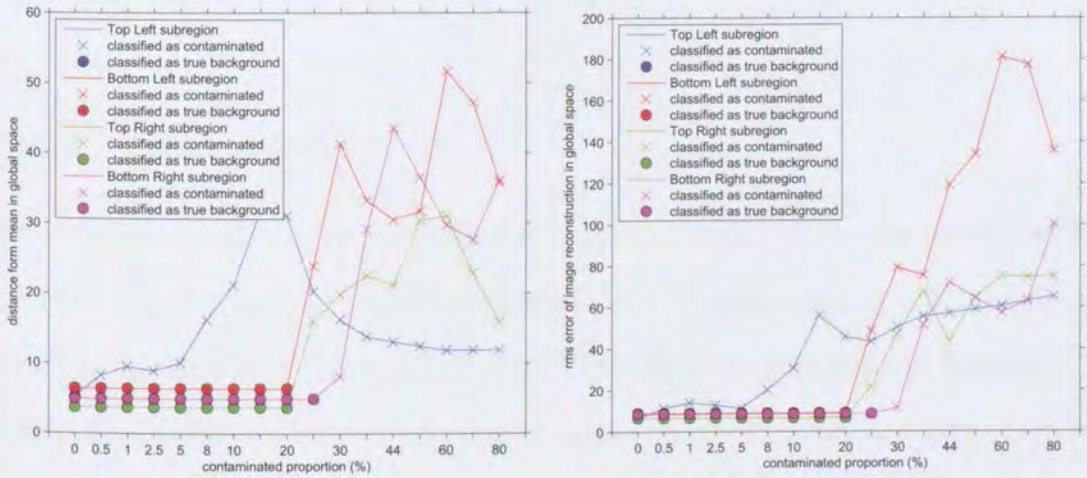
unimodal eigen-space classification plot in Figure 4.12. In other words, the unimodal eigen-space classification result obtained from the subsampled portion and that obtained using the entire image are very close. The same level of classification accuracy can be achieved by using only 10% of the available data. This suggests that, in the context of unimodal eigen-space classification, the information loss due to subsampling at subsampling rates up to 1/10 is acceptable, alowing for significant reduction in the amount of processed data.

### Subsampling from subregions

It was shown that random subsampling from the entire image at the subsampling rate of 1/10 allows for detection of a contaminant of the size of no less than 8% of the image. A more controlled way of subsampling from predefined regions is expected to provide accurate classification of even smaller contaminant. Combined with some knowledge of entry points in the scene and/or expected contaminant sizes, this method can be an useful tool for quickly and reliably detecting presence of small contaminations from a relatively little information about the scene. Figure 4.19 shows an example of subsampling from subregions at the subsampling rate of 1/10 when the contaminant occludes 8% of the image.

Figure 4.19: Subsampling from subregions

Figure 4.20a shows the result of the classification in the unimodal eigen-space of the contaminated image obtained from 10% of the available pixels subsampled from four image subregions, when the contaminant size increases. It can be seen that when subsampling from the top-left region, where the contaminant enters the sceene, the observation is classified as contaminanted for contaminants as small as 0.5% of the image (blue line). The classification obtained form other three subregions labels the image as true background as long as the contaminant is smaller than 20% and remains within limits of the top-left subregion. This suggests that by subsampling from predefined regions it is possible to largely improve the classification accuracy in the unimodal eigen-space even for the contaminants as small as 0.5% of the image.




(a) Distance

(b) Reconstruction error

Figure 4.20: Subsampling from subregions in unimodal space

Figure 4.21 represents results of the multi-modal eigen-space classification of the same contaminated image when the size of the contaminant increases. The image is randomly subsampled from four subregions at the rate of 1/10.

In this case, the subsampled image representation is subject to a high information loss, such that even the three true background subregions with a 0% contamination are wrongly classified as contaminated (red, green and magenta lines).

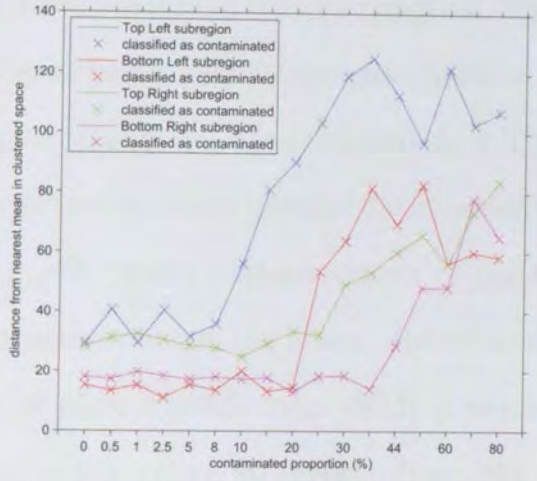



Figure 4.21: Subsampling from subregions in multi-modal space

### 4.4.4 Discussion

After dimensionality reduction a further computational saving may be achieved by reducing the amount of the processed data by subsampling a relatively small proportion of all the available data. The random subsampling, however, has its limitations in terms of high variations of results and loss in information due to discarding of the proportion of the original data. This constraints impose limits on the acceptable subsampling rate. In multi-modal eigen-space, due to higher sensitivity to image changes, the acceptable subsampling rate is much higher than in the unimodal eigen-space.

In the presence of a contaminant the subsampled proportion of the data

will inevitably contain some corrupted samples. For larger contaminants, the probability of subsampling from the contaminated region increases causing more significant distortions in the background representation. In this case, depending on the contaminant size, it is possible to avoid contaminated areas by subsampling from predefined image subregions. This way of controlled subsampling provides accurate unimodal eigen classification of image observations from a very small amount of subsampled data. Combined with some knowledge of entry points in the scene and/or expected contaminant sizes, this method may quickly detect contaminations of the background from a relatively little information about the scene. However, this is not the case with the multi-modal eigen classification which is much more sensitive to image distortions than the unimodal model. Here, the subsampled image representations are subject to an increased information loss and are therefore wrongly classified as contaminated even when there is no contamination in the scene.

## 4.5 Experimental set up

The aim of this work is to analyse and efficiently model the changes in the background of outdoor video sequences which usually contain an abundance of sudden and gradual light changes and non-static background. In order to account for as much of the background variation as possible it is essential to perform the analysis on a video dataset which includes a diversity of background features, such as weather conditions (sunshine, overcast, rain, wind), time of the day (daylight, dawn, dusk, night), length of the daytime, sea-

sons, non-static background motion (swaying trees, bushes, shadows). Furthermore, the preferred dataset should cover a very long period of time and contain very little foreground motion. At the time of writing, in our opinion, none of the existing well known datasets fulfilled such requirements. Therefore, two datasets have been produced which will be analysed and described in detail in this section.

### 4.5.1 Kingston Carpark datasets

#### Dataset description

The Kingston-Carpark dataset is a video sequence recording activities in a car park on a summer day. This is a short video consisting of 8350 frames taken at full frame-rate covering a period of five and a half minutes. Example frames are shown in Figure 4.22. The CCTV camera has iris auto-correction and colour switched on. The video sequence includes a total of 24 moving objects, people and vehicles appearing at close, medium and far distances from the camera. There is a variety of both gradual and sudden lighting changes present in the scene due to English weather conditions (bright sunshine interrupted by fast moving clouds, reflections from windows of vehicles and buildings), and changes in camera settings in response to the light changes. In addition, a strong wind causes swaying trees and bushes to disturb the background. There are both static and dynamic occlusions present in the scene with moving objects crossing paths and disappearing partially or totally behind static objects.

Figure 4.22: Kingston Carpark dataset

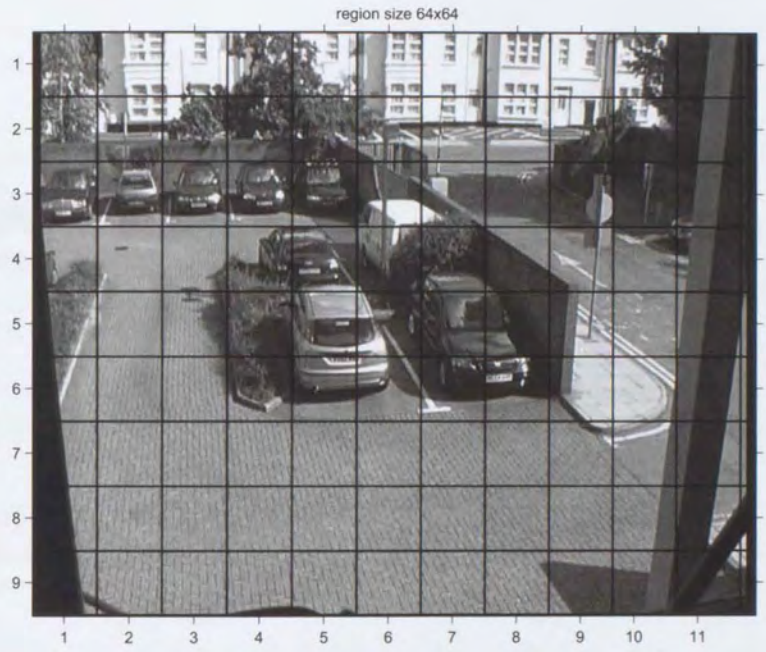
| region | description | colour code |
|---|---|---|
| {1,1} | tree | black |
| {1,5} | part white building, part tree | blue |
| {3,3} | parked dark colour car | green |
| {3,6} | part parked white van, part brick fence | red |
| {5,2} | cobbled parking ground | cyan |
| {5,6} | part silver car, part ground in shadow | magenta |
| {5,10} | tarmac ground | yellow |
| {7,9} | cobbled parking ground | violet |
| {8,3} | cobbled parking ground | brown |

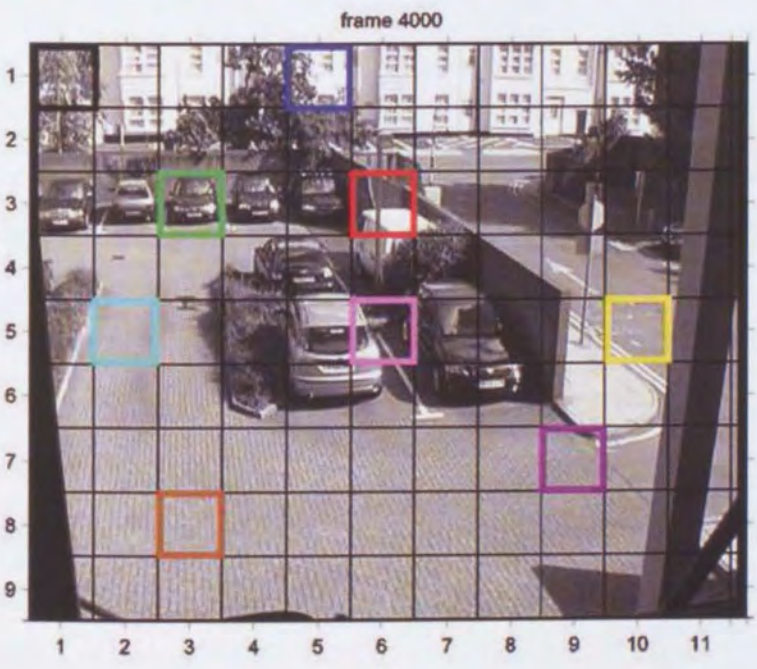
Table 4.2: Selected regions description

**Dataset analysis**

The original video sequence was recorded in colour. We perform all analysis on its grey-scale version. For the purpose of processing and analysis the original frames of size 557-by-720 pixels are divided into smaller regions. An example of 64-by-64 regions is shown in Figure 4.23a. The region size is selected in such a way that the objects moving through the region neither appear too large (therefore occluding the whole region), nor too small. Admittedly, the choice of the region size is rather arbitrary and is derived form the knowledge about the sequence, the perceived distance between the camera and the objects, and the expected size of the objects.

To provide better understanding of the dataset we select and comment on few distinct regions chosen to cover a range of video background content, such as the type of the ground surface (tarmac, cobbles, grass), the amount of saturated white colour surfaces, the presence of non-static background (swaying trees and bushes). The selected regions are marked with rectangles of different colours, Figure 4.23b, and described in Table 4.2.




(b)

Figure 4.23: Regions

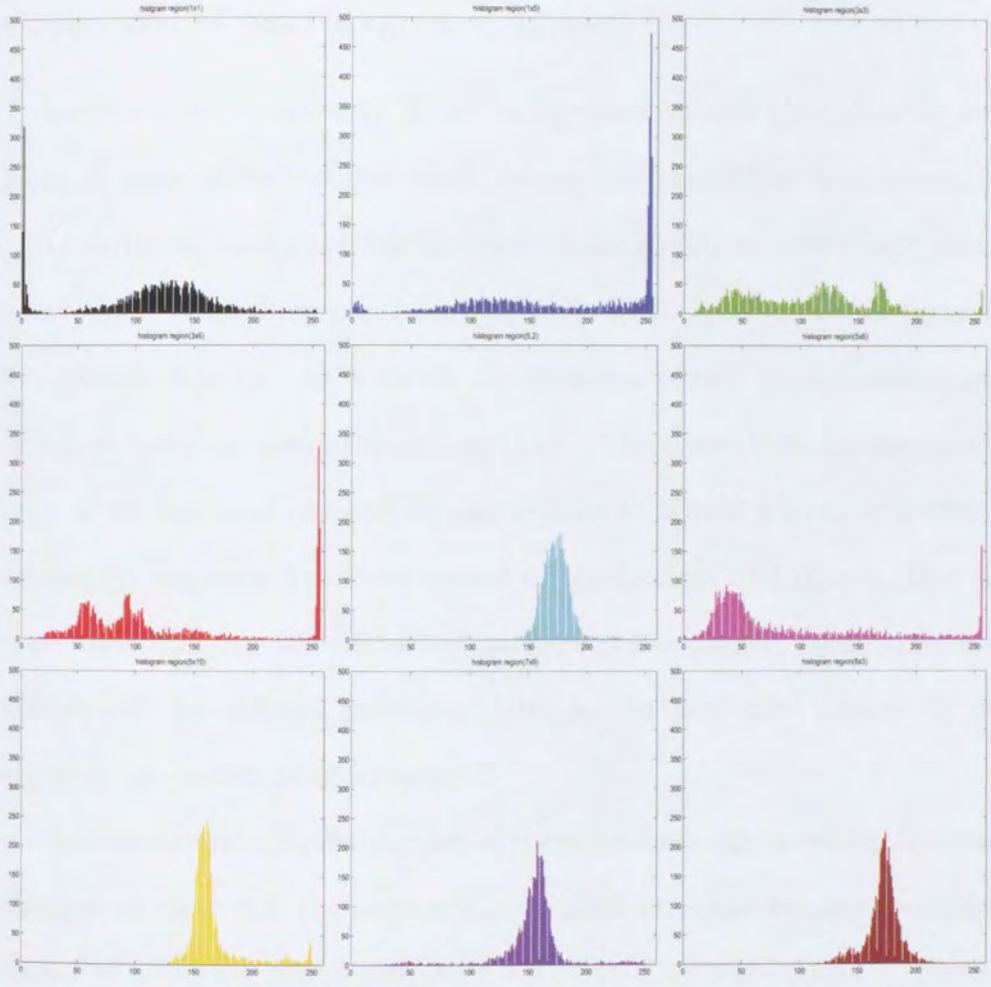


Figure 4.24: Histogram of grey-level pixel values for selected regions

Figure 4.24 shows grey-level histograms of chosen regions calculated for a typical frame. The histograms reflect the diversity of chosen background regions; some are distributed over the whole grey-level range (black, blue, green, red and magenta) whereas others are more concentrated around mid-range grey-levels (cyan, yellow, violet and brown).

### Complexity of the background dataset

To illustrate the complexity of the background dataset the following analysis looks at some of its features, their nature and variability in time and space.

In order to model the background of the scene, we select only frames in which there is no foreground motion while leaving out all other frames with foreground objects. As a result we obtain a purely background sequence, which is however discontinuous in time. There are 1500 frames consisting only of background images[3] in our sequence. These frames will then form a training sequence for development of the background model. Due to this time-discontinuous nature of the background sequence, some time-variable features of the dataset presented later in the text may appear to change abruptly at certain time instances[4].

To demonstrate the behaviour of these features due to global illumination changes of the scene, the sequence is divided into time windows which correspond to following frame ranges: 1–550, 551–750, and 750–1450, divided by dotted lines on the graphs.

Grey-level pixel values taken at consecutive time instances inevitably vary due to the intrinsic noise introduced by camera, the movement of the camera, and the illumination changes present in the scene. Figure 4.25 shows the time variation of the average RMS grey-level difference between successive frames for selected regions. How significant this variation will be depends on the characteristics of the background. Regions which include background motion

[3]frame ranges selected from the original video sequence: 1050-1600, 3600-4000, 5300-5600, 6600-6800, 8200-8250

[4]frames 550, 950, 1250 and 1450

caused by swaying trees, such as regions $\{1,1\}$ (black) and $\{1,5\}$ (blue), exhibit more significant difference between successive time instances. Regions covering still backgrounds, tarmac and cobbled floor, such as regions $\{5,2\}$ (cyan) and $\{8,3\}$ (brown), are subject mainly to the intrinsic camera noise. Regions which include more reflective surfaces, such as stationary vehicles in regions $\{3,6\}$ (red plot) and $\{5,6\}$ (magenta), display a similar low, noisy, and slightly less stable difference variability. The intrinsic noise level of the dataset, estimated as the RMS difference between the successive frames, is typically 3.2 grey-levels per pixel. This value is obtained for a region with no reflective surfaces nor non-stationary background such as region $\{8,3\}$.

Figures 4.26 and 4.27 show variations of the mean and the standard deviation of pixel grey-levels of selected regions, as they vary from frame to frame over the whole background sequence.

The time-variability of the mean value and the standard deviation of pixel grey-levels illustrates the effects of illumination changes in the scene. In the first and the third time window of the sequence, the global illumination of the scene is fairly stable. Significant variations of the mean pixel grey-level are visible only for the regions $\{1,1\}$ (black) and $\{1,5\}$ (blue); these variations are caused by erratic movement of the swaying trees rather than global illumination changes. In same time intervals, the pixel grey-level standard deviation is stable for all regions but those with moving trees. This variation is caused by the fairly dark tree leaves which move and reveal parts of bright white buildings behind. It is this large difference in grey-levels between two background surfaces and the time-variable proportion of the image taken by each of them that causes the variation of the standard deviation.

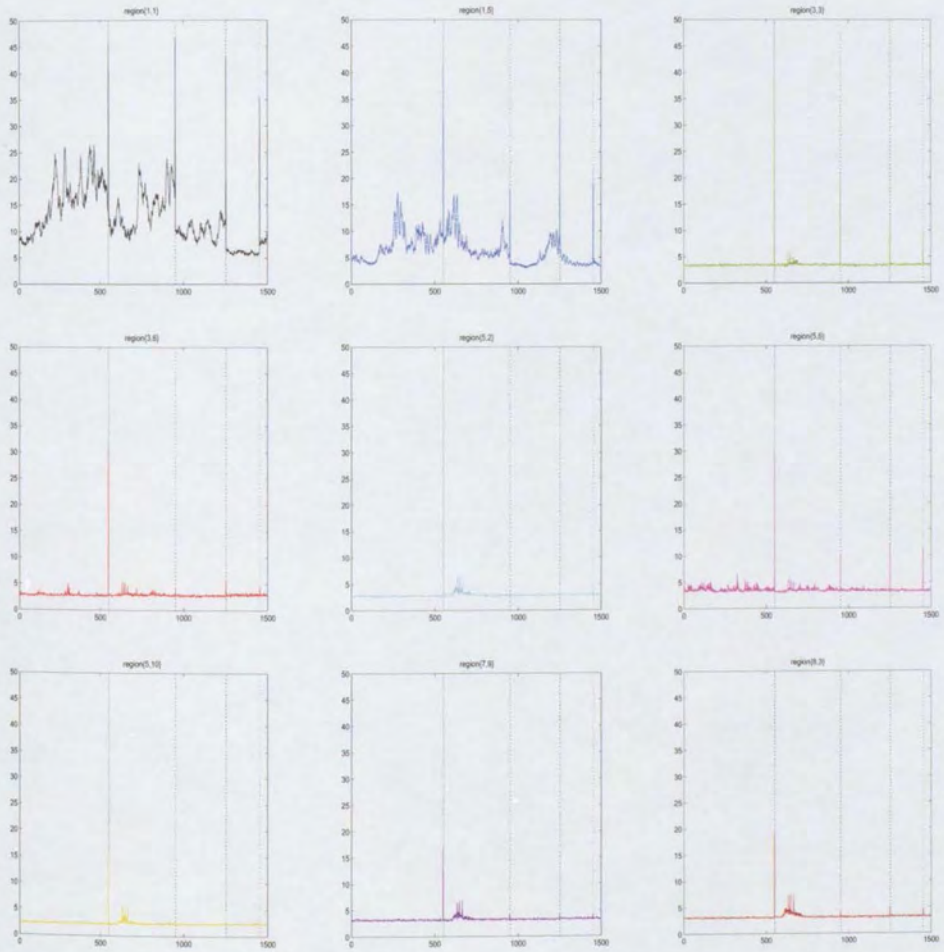


Figure 4.25: Difference between successive frames for selected regions

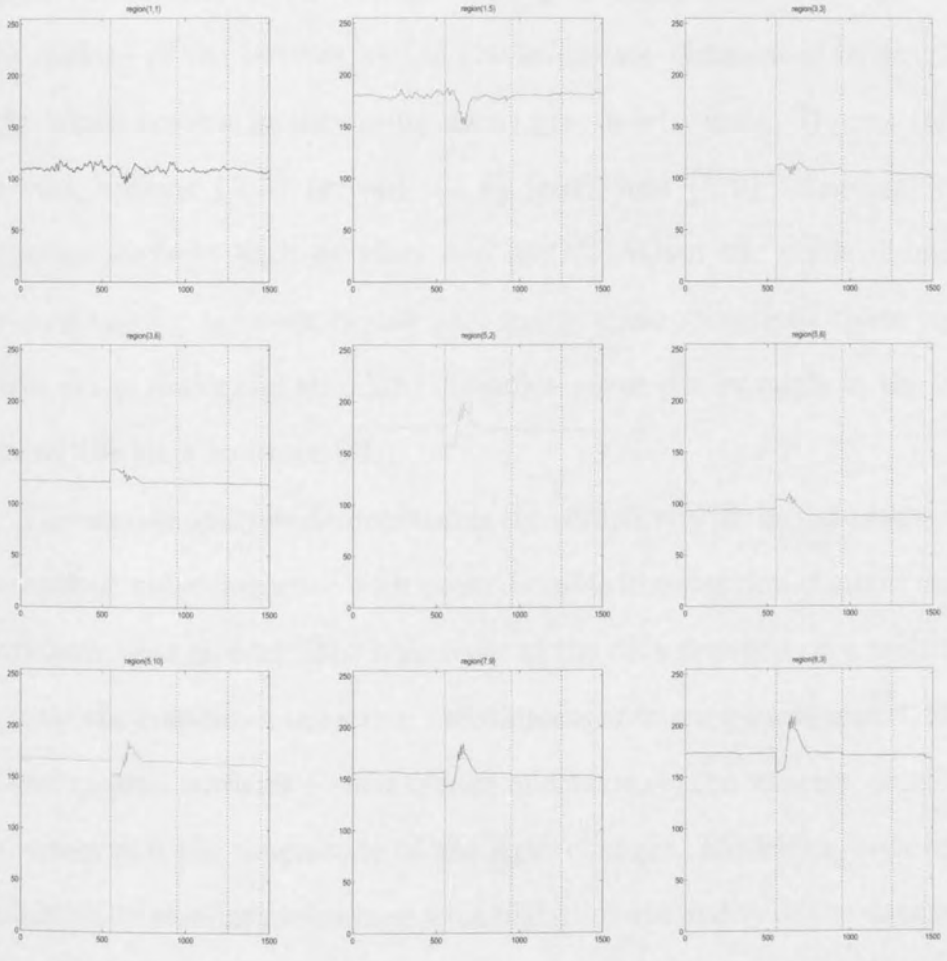


Figure 4.26: Mean pixel grey-level for selected regions

In the second time window the illumination varies due to changing weather conditions, irregularly alternating clouds and sunshine. When the change occurs, around the frame 600, the regions in the top half of the scene, $\{1,1\}$ (black), $\{1,5\}$ (blue), $\{3,3\}$ (green) and $\{3,6\}$ (red), are in the shade during this time interval. Hence the corresponding grey-level means decrease. The regions of the bottom half of the image are illuminated by bright sunlight which results in increasing mean grey-level values. During this time interval, regions $\{3,3\}$ (green), $\{3,6\}$ (red), and $\{5,6\}$ (magenta) contain reflective surfaces such as glass and metal. When the scene illumination changes rapidly between cloudy and sunny these reflections cause irregular peaks in the mean and standard deviation curve (for example in the interval around the time instance 650).

This simple analysis demonstrates the complexity of the data representing an outdoor video sequence with unpredictable illumination changes and non-stationary background. The behaviour of the data depends on a multitude of factors: the content of the scene – motionless or moving background, the type of background surfaces – their colour and texture, the weather conditions – the speed and the magnitude of the light changes. Modelling such complex variability of the data combined with high dimensionality of the data space is very difficult. Statistical analysis of multivariate data, such as PCA, should provide better understanding of complex dataset.

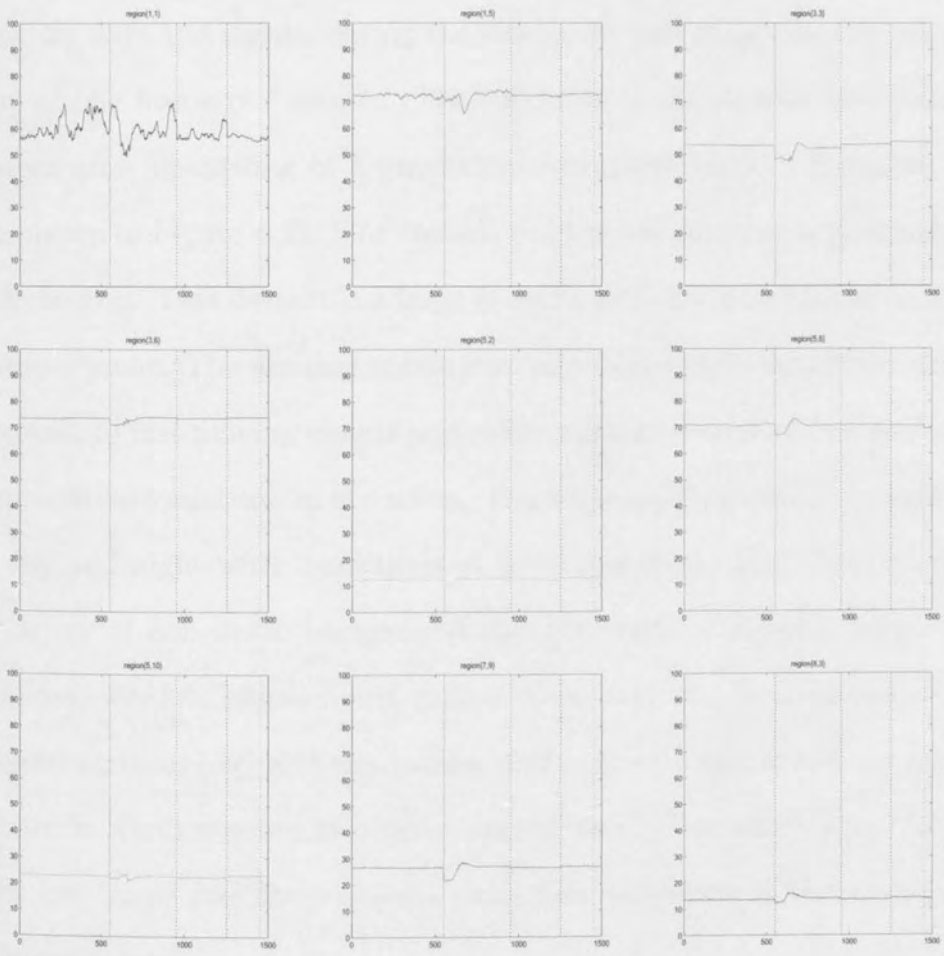


Figure 4.27: Standard deviation of pixel grey-levels for selected regions

### 4.5.2 Inrets dataset

#### Dataset description

The Inrets dataset is a record of activities in a public courtyard in front of a small office building. It was generated over a period of two and a half months, days and nights, during the late spring and early summer, at a frame rate of one frame per minute. The total size of the dataset is about 40000 frames after discarding of a number of corrupted frames. Example frames are shown in Figure 4.28. The camera has the iris auto-correction and colour switched on. This dataset is a large collection of changing backgrounds in an outdoor scene. The weather conditions vary from bright sunshine, uniformly overcast to fast moving clouds and rain, causing a range of fast and gradual illumination variations in the scene. The scene lighting cycles through stages of day and night with transitions of dawn and dusk. Furthermore, there is a variety of non-static background changes, such as swaying trees, parked vehicles, window blinds being pulled down and up, changing state of the ground surfaces (dry and wet tarmac and concrete, grown and cut grass). A relatively small number of objects appear throughout the video. Given the very low frame rate these objects move fast remaining in the scene only for a frame or two.

#### Dataset analysis

The original video sequence was recorded in colour. We perform all analysis on its grey-scale version. For the purpose of processing and analysis the original frames are divided into smaller regions of 64-by-64 pixels. To provide

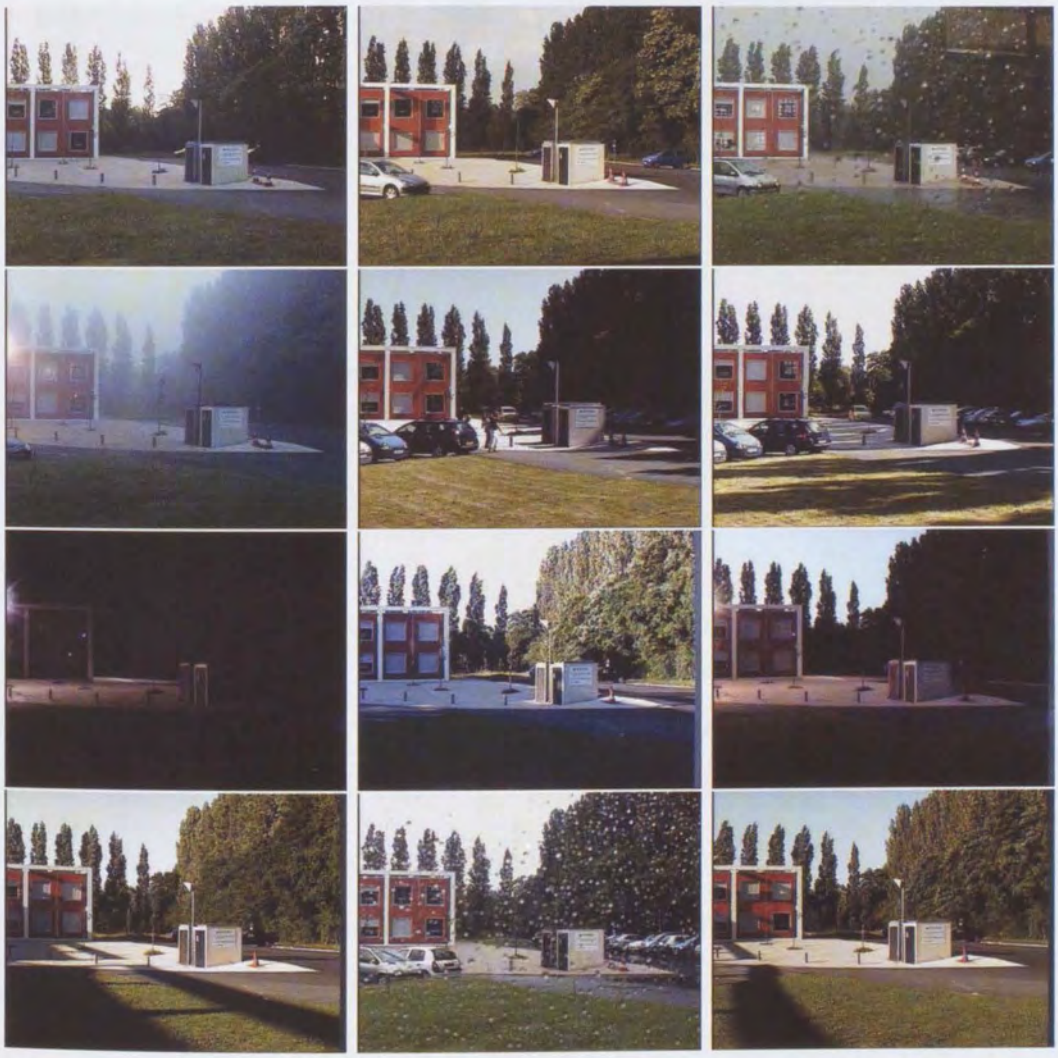


Figure 4.28: Inrets dataset

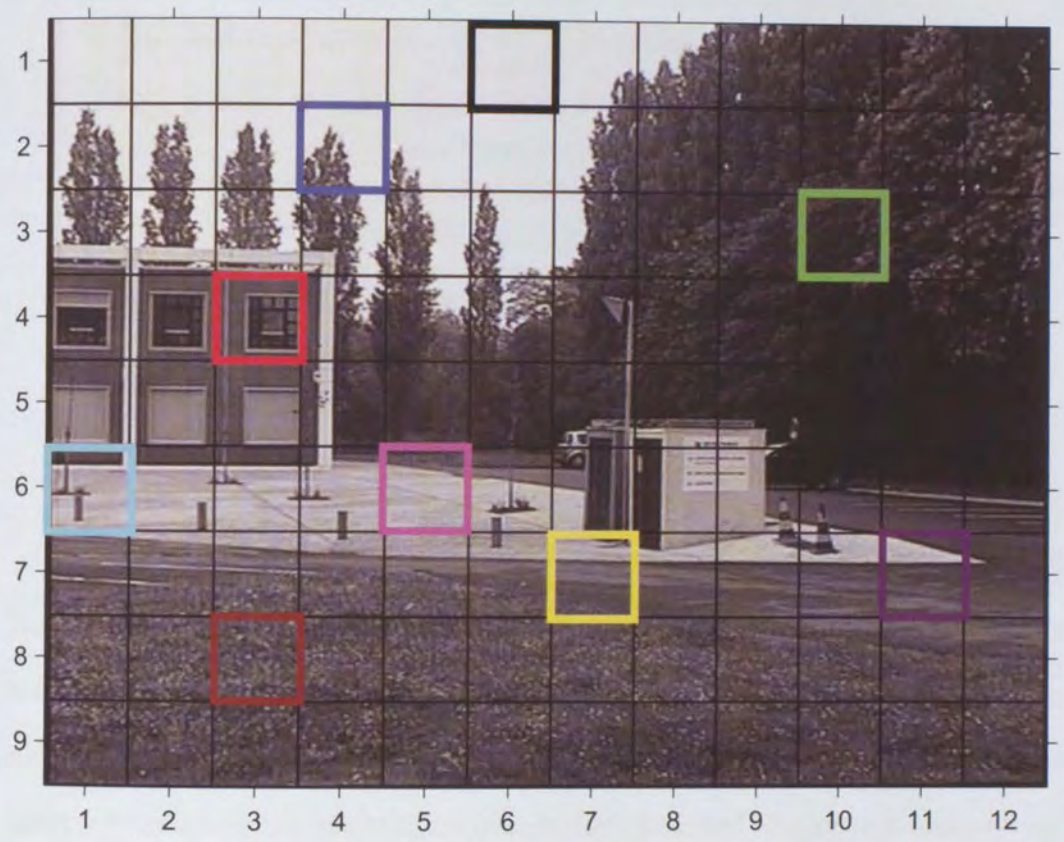



Figure 4.29: Grid of selected regions: Inrets dataset

better understanding of the dataset we select and observe a few distinct regions chosen to cover a range of video background content, such as the type of the ground surface (concrete, tarmac, grass), the presence of non-static background (swaying trees fully or partially covering the region), open sky horizon with fast and slow passing clouds, and reflective glass surfaces. The selected regions are shown in Figure 4.29 and described in Table 4.3.

### Complexity of the background dataset

To illustrate the complexity of the background dataset the following analysis looks at some of its features, their nature and variability in time and space.

| region | description | colour code |
|---|---|---|
| {1,6} | open sky | black |
| {2,4} | swaying tip of a tree | blue |
| {3,10} | thick leafy tree | green |
| {4,3} | glass window | red |
| {6,1} | concrete ground close to the building | cyan |
| {6,5} | concrete ground away from the building | magenta |
| {7,7} | part concrete, part tarmac ground | yellow |
| {7,11} | part concrete, part tarmac ground | violet |
| {8,3} | grass | brown |

Table 4.3: Selected regions description

In order to model the background of the scene we select only daylight frames when the street lights are switched off. Objects rarely appear in the sequence after which they either remain stationary in the scene or quickly disappear after a frame or two due to the low frame rate of recording. As a result of omitting the night time frames the obtained sequence is discontinuous in time. The sequence covers about 20 days with a great variety of light changes. The time intervals corresponding to different days are marked by dotted lines in the following graphs.

Figure 4.30 shows temporal variation of the grey-level RMS difference between pixels of successive frames for the selected regions over a time interval of 3000 frames taken at frame rate of one frame per minute. The variability observed for the region $\{1, 6\}$ (black), covers only the sky on the horizon, is mostly due to global light changes where the sunny and cloudy weather conditions frequently alternate. The sequence is so rich with light changes that the average pixel difference between two sky images, recorded one minute apart, reaches up to 100 grey-levels. In the late evening and early morning, when the rest of the scene is fairly dark, the sky appears comparatively very

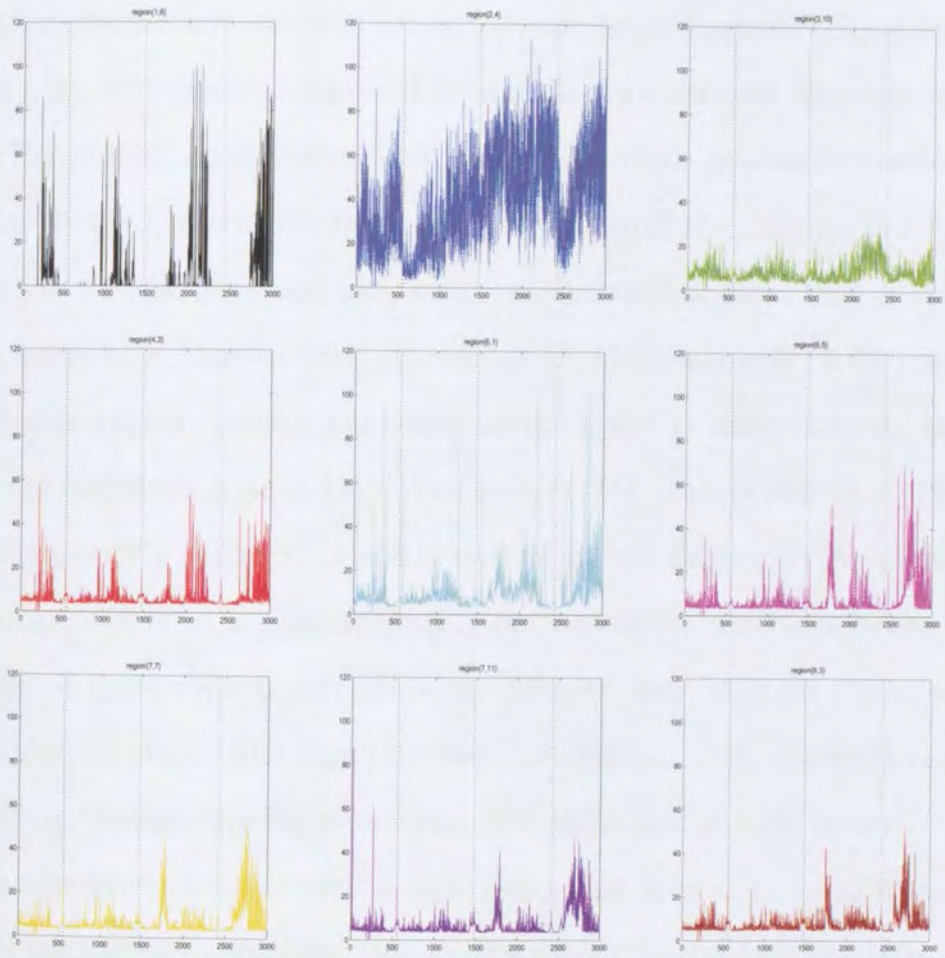


Figure 4.30: RMS difference between successive frames for selected regions

bright and reaches the white saturation grey-level. Therefore, around the end and the beginning of the time intervals (days) marked by dotted lines, there is no difference between successive time instances for this region. The region $\{2, 4\}$ (blue) displays very dynamic changes between successive time instances. Here, the tip of a tree shifts from one side of the region to another covering and uncovering the sky behind. It is this change in the proportion of the bright sky pixels and dark tree pixels in the region that creates such a dramatic difference between successive frames. The region $\{3, 10\}$ (green)

contains dense tree leaves of similar colours, hence the recordings of this area vary less from frame to frame. The next three graphs, red, cyan and magenta in Figure 4.30, display occasional high peaks which are due to changes other than weather conditions. In the region $\{4,3\}$ (red) they are due to reflections on the window glass and movements of the window blind that is pulled up or down from time to time, in regions $\{6,1\}$ (cyan) and $\{6,5\}$ (magenta) vehicles appear, remain stationary in the scene or disappear. In the case of the last three regions $\{7,7\}$ (yellow), $\{7,11\}$ (violet) and $\{8,3\}$ (brown), the variability observed in the graphs is caused exclusively by global light changes. Here, it is also interesting to observe the difference between portions of graphs which correspond to different time intervals (days) covered by the sequence. The first two days are overcast with not much sunshine getting through the dense clouds. The third day is with sunny intervals, whereas the last day is very sunny. Peaks that appear in the graphs during this last day are caused by long shadows from the surrounding trees, which move constantly across the scene.

The intrinsic noise of this dataset, estimated as the average RMS difference between successive frames, is 6.67 grey-levels per pixel. This value is obtained for a region with no reflective surfaces nor non-stationary background such as region $\{8,3\}$. The estimated intrinsic noise of the Inrets dataset is higher that that of the Kingston Carpark dataset.

Figures 4.31 and 4.32 show time variations of the mean and standard deviation of pixel grey-levels for selected regions. The profile of variations depends on the type of the background covered by the selected region and the type of environmental conditions during the time interval. The region of

the sky reaches white level saturation at the beginning and the end of the day when it appears very bright compared to the rest of the scene; in these time intervals both mean and the standard deviation for this region are saturated and appear as flat lines on graphs.

An interesting observation can be made regarding the last time interval which covers a very bright sunny day. At the beginning around the time instance 2420, as the day grew brighter the mean grey-level in all regions (except the saturated region of the sky) started to increase. Very shortly after, it started to drop rapidly. This firstly occurred in the leftmost parts of the scene in the region $\{6, 1\}$ (cyan), then progressively in the mid regions and finally, around the frame 2500, in the far right regions $\{7, 11\}$ (violet) and $\{8, 3\}$ (brown). This trend of decreasing mean continued until about time instance 2750. During this time interval, shortly after the day began, a number of long dark shadows, caused by surrounding trees, created a stripy pattern of dark and light that was constantly moving over the scene. The shadows firstly covered the entire width of the scene and then gradually withdrew from left to right regions, finally disappearing at about the time instance 2770. As it was getting generally brighter in the scene the camera continuously adjusted the iris and the amount of light passing through, which caused the mean of the regions in the shadow to progressively decrease.

This short analysis of basic statistical characteristics of the dataset illustrated the challenging nature of modelling outdoor video sequences with non-stationary backgrounds and significant changes in lighting levels due to weather conditions. It is very difficult to describe the behaviour of such data in a meaningful way if only the mean and the standard deviation are

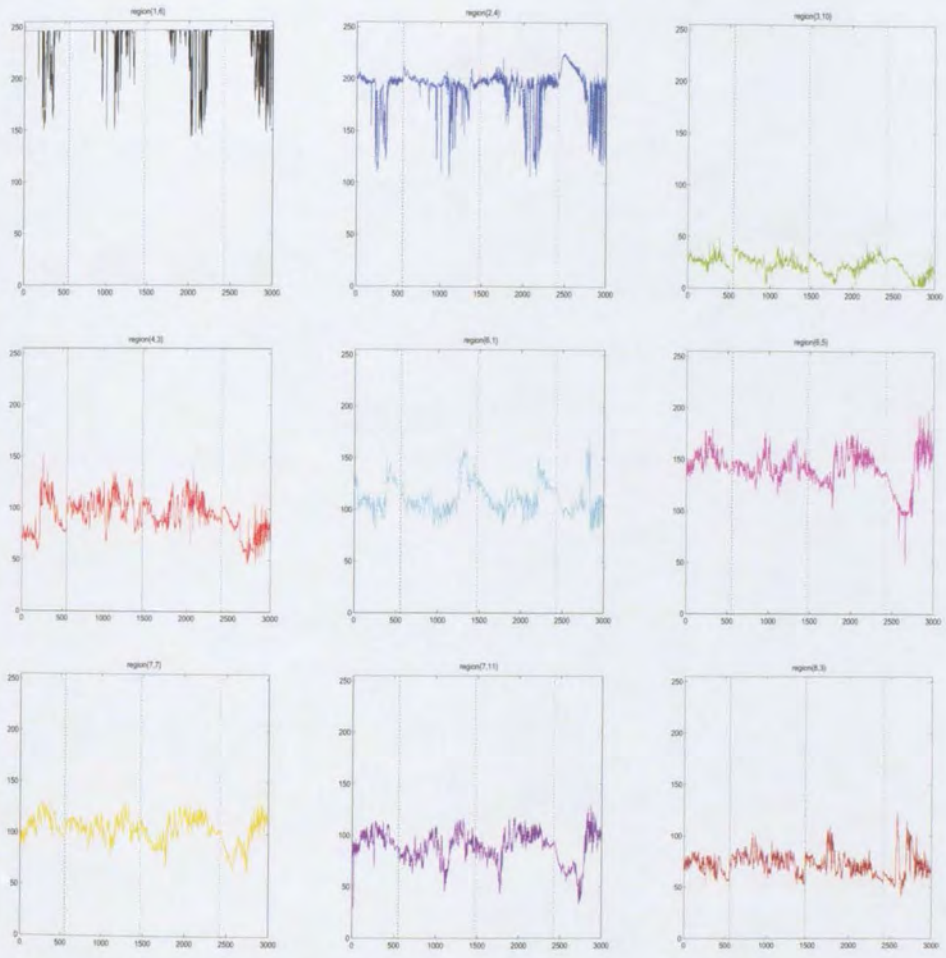


Figure 4.31: Mean pixel grey-level for selected regions

observed. A method, such as PCA, which provides means for unveiling the underlying variability in multivariate data, is expected to offer a more meaningfully explanation of the behaviour of described datasets.

## 4.6 Results

This section describes the results of the eigen-analysis of the video data described in Section 4.5.2. The amount of variability in the data captured by

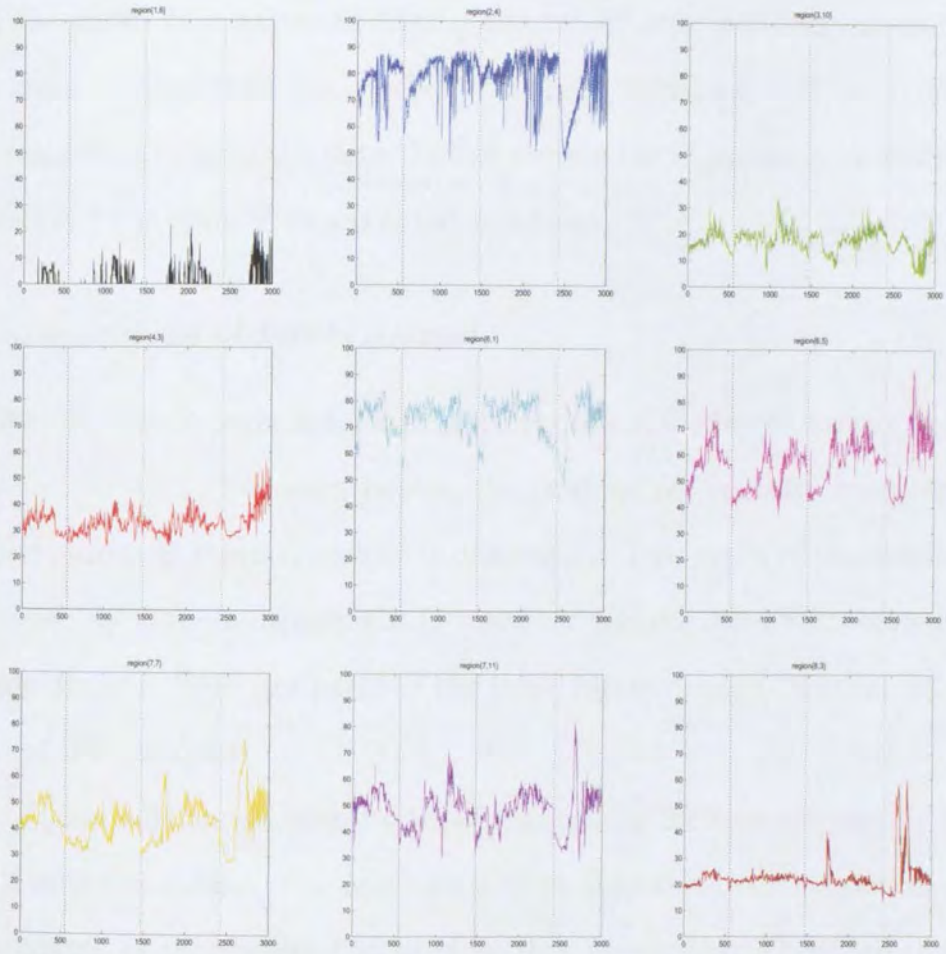


Figure 4.32: Standard deviation of pixel grey-level for selected regions

the principal components is analysed. The results of different dimensionality reduction methods are discussed. Finally, the reduced dimensionality eigen-space representation is presented and discussed.

### 4.6.1 Eigen-analysis of video datasets

One way of modelling the variability of high dimensional datasets, such as video data, is to perform eigen-analysis of the dataset and determine principal components which capture some significant proportion of the total variability

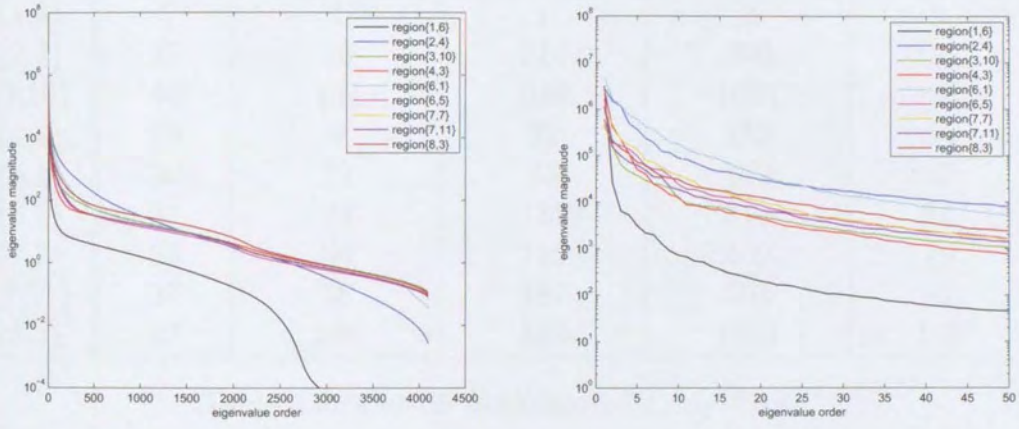
in the data. It is expected that a number of principal components, which is much smaller than the number of original variables, will be sufficient to explain behaviour of the data. In this section the eigen-analysis described in Section 4.2 is applied to a real video dataset.

### Eigen-analysis of Inrets dataset

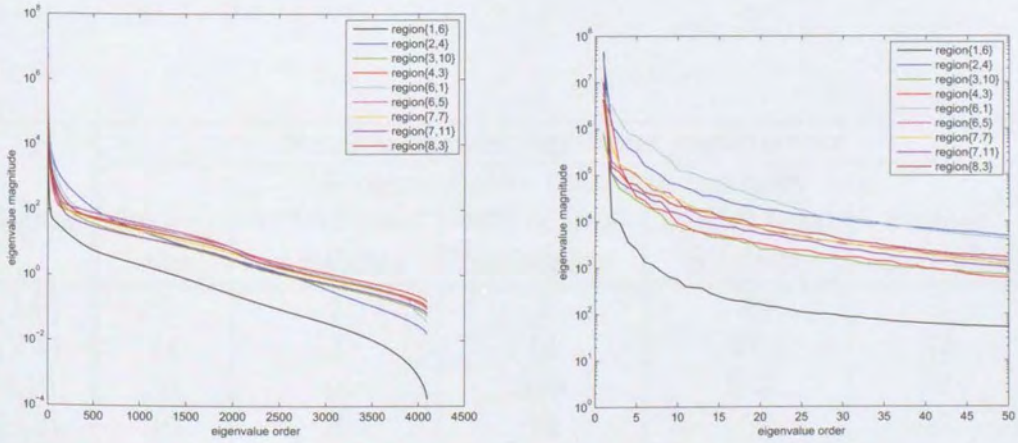
Eigen-analysis is performed on selected regions of the Inrets dataset described in Section 4.5.2. For each region, the training set contains twice as many observations as there are pixels in the region. Two types of training sets are chosen, one containing daylight frames only and the other with both day and night frames. Both are parts of the large Inrets dataset, Section 4.5.2, and are of the same size.

Figure 4.33 shows magnitudes of eigenvalues for selected regions for two sets of training data. The magnitude of an eigenvalue reflects the amount of variability of the dataset captured by that eigenvalue. Therefore the shape of the eigenvalue plot depends on the nature of the data. Most significant eigenvalues or principal components (PCs), those of low orders, capture most of the variability in the dataset. Eigenvalues of low magnitude, those of high orders, are assumed to capture only the noisy variations in the dataset.

Two plots in Figure 4.33a present eigenvalues for the training set which contains only frames recorded during the daylight. The left graph spans over all eigen-components accounting for the variability along all directions of the eigen-space. The right graph shows the variance captured by the fifty most significant PCs. Similarly, the plots in Figure 4.33b present eigenvalues for




(a) day data




(b) day and night data

Figure 4.33: Eigenvalues magnitude for selected regions (log)

| Region | Number of most significant eigenvectors | | | | |
|---|---|---|---|---|---|
| | Dimensionality reduction methods | | | | |
| | Broken stick | 95% of total variability | 98% of total variability | 99% of total variability | average eigenvalue |
| {1,6} | 5 | 1 | 1 | 2 | 9 |
| {2,4} | 45 | 76 | 214 | 366 | 122 |
| {3,10} | 46 | 197 | 648 | 1081 | 133 |
| {4,3} | 19 | 9 | 39 | 172 | 44 |
| {6,1} | 30 | 22 | 55 | 112 | 67 |
| {6,5} | 34 | 34 | 111 | 313 | 81 |
| {7,7} | 35 | 34 | 115 | 409 | 79 |
| {7,11} | 37 | 48 | 187 | 604 | 91 |
| {8,3} | 37 | 158 | 583 | 1033 | 142 |

Table 4.4: Cut-off dimension for day data

| Region | Number of most significant eigenvectors | | | | |
|---|---|---|---|---|---|
| | Dimensionality reduction methods | | | | |
| | Broken stick | 95% of total variability | 98% of total variability | 99% of total variability | average eigenvalue |
| {1,6} | 2 | 1 | 1 | 1 | 2 |
| {2,4} | 11 | 3 | 14 | 47 | 32 |
| {3,10} | 30 | 46 | 399 | 803 | 87 |
| {4,3} | 10 | 3 | 15 | 102 | 27 |
| {6,1} | 26 | 16 | 40 | 96 | 54 |
| {6,5} | 12 | 4 | 19 | 76 | 33 |
| {7,7,} | 14 | 5 | 16 | 44 | 33 |
| {7,11} | 11 | 2 | 12 | 33 | 31 |
| {8,3} | 23 | 17 | 78 | 267 | 56 |

Table 4.5: Cut-off dimension for day and night data

the training set which covers both day and night frames. It can be observed that magnitudes of the few most significant eigenvalues obtained for the day and night data are in general higher than those obtained for the day dataset, after which they drop faster than those in the case of day data. Alternations between the day and the night time result in a pattern of global illumination change which dominates all other variations that may be observed in the day and night data. The dominant pattern of variation is captured by the very first few eigenvalues of the highest magnitudes.

These few most significant PCs capture most of the variability in the day and night data; for example, in the case of region {6,5} only four of the most significant PCs contain 95% of the total variability in the day and night data compared to thirty four in the day data (see Table 4.4 and Table 4.5). At the same time, total variance contained in the day and night data is approximately three times that in the day and night data.In terms of standard deviation that is on average 36 grey-levels per pixel for the day and night data, compared to 21 for only day data. This is expected because the inclusion of night observations adds more variability to pixel grey-levels.

### 4.6.2 Dimensionality reduction

Dimensionality reduction is based on retaining $m$ number of PCs, which correspond to the first $m$ ordered eigenvalues, where $m$ is significantly smaller than total number of variables $p$, and under the condition that most of the variability in the data is still captured. In our example, the number of variables $p$ is equal to the number of observed pixels in a selected region; that

is 4096 variables in the 64-by-64 pixel region. This represents the number of PCs in the eigen-space.

The higher the magnitude of the eigenvalue the larger is the proportion of the total variability that is contained in the corresponding PC. Table 4.4 illustrates the proportion of total variability contained in PCs and summarizes the results of dimensionality reduction for each selected region in day data. Equally, Table 4.5 corresponds to the day and night data. For each selected region the number of principal components is obtained by several dimensionality reduction methods and displayed in corresponding columns. The reduction methods used in this example are the *broken stick* rule, a proportion of total variability and the average eigenvalue. These dimensionality reduction methods were discussed in detail in Section 4.2.6.

The results of dimensionality reduction confirm that due to the presence of a dominant global illumination change in the day and night data, fewer PCs are required to capture the same proportion of variability as in the day–only dataset. This is true for all dimensionality reduction methods. Furthermore, it is suggested that the *broken-stick* rule provides most acceptable dimensionality reduction for both sets of video data.

### 4.6.3 Eigen-space representation

This section illustrates the data distribution in the eigen-space. The eigen-representation reveals and models the underlying variability patterns in data that are otherwise obscured or difficult to describe.

Figures 4.34, 4.35 and 4.36 show a two dimensional eigen-representation

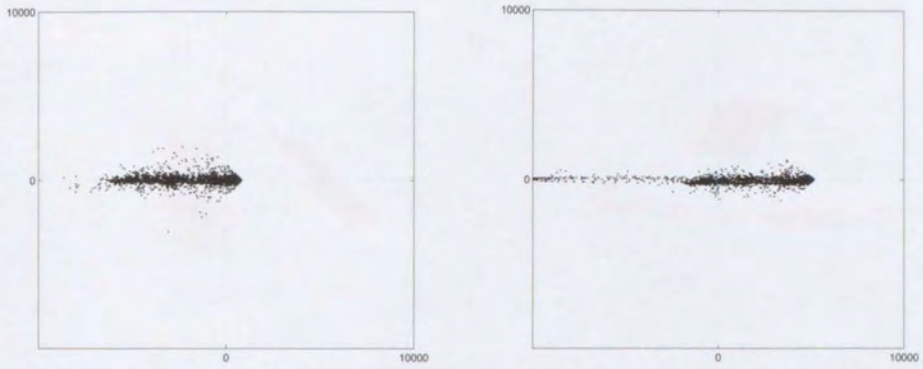
of selected regions, where the two chosen eigen-space dimensions are the two most significant principal components (PCs). The first most significant PC is plotted along the x-axis; the second most significant PC along the y-axis. For each region there is a pair of graphs displayed. The graph on the left side corresponds to the day only data, the one on the right to the day and night dataset.

It is observed that regions {1,6} (black) and {2,4} (blue) are most affected by the alternation of day and night time. Those regions cover, entirely or partially, open sky pixels the grey-levels of which vary greatly from extremely bright during the day to extremely dark during the night. This is reflected by extended variability span of the first most significant PC on the x-axis of the right graph, Figures 4.34a and 4.34b.

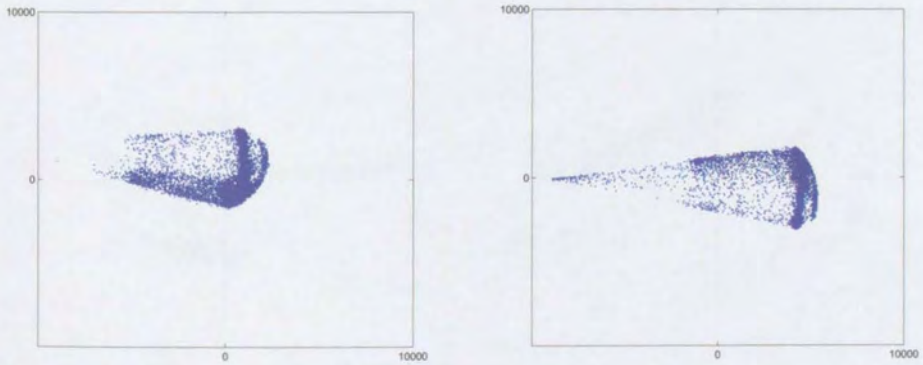
Region {3,10} (green) covers dark thick tree leaves which are of similar grey-levels during the day and the night time. The variability of pixels in this region is not dominated by the day/night light change. Hence there is not much difference between the two graphs in Figure 4.34c.

Region {4,3} (red) includes a window of an office building where a bright colour blind is regularly pulled up or down creating two states of the image region. The two states place the corresponding observations in one of two distinct clusters in the eigen-space, as shown in graphs in Figure 4.35a. The graph on the right hand side includes the night time observations adding a new dominant PC to the eigen-space.

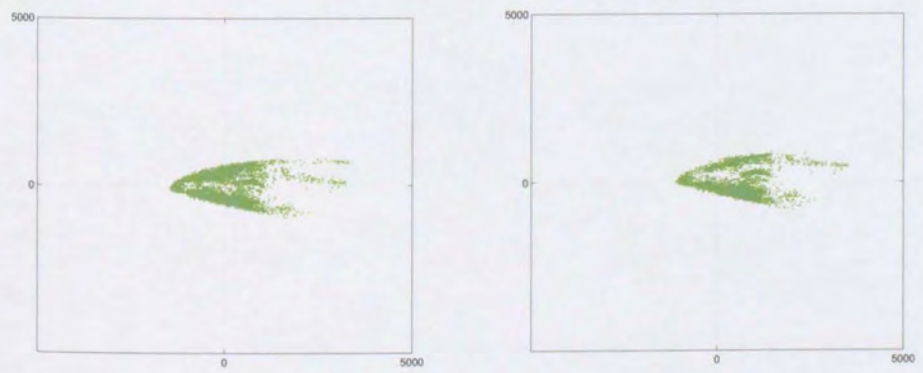
Region region {6,1} (cyan) forms a complex structure in the eigen-space due to transformations of the region caused by vehicles being parked and staying for periods of time or leaving the scene. Observations in the eigen-




(a) region{1,6}

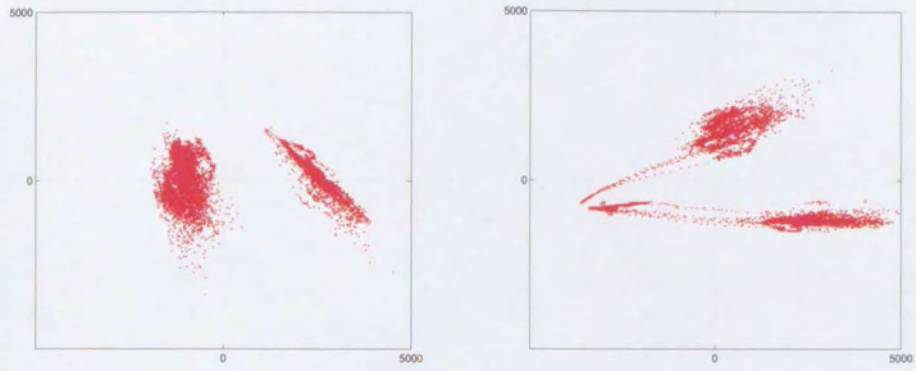



(b) region{2,4}

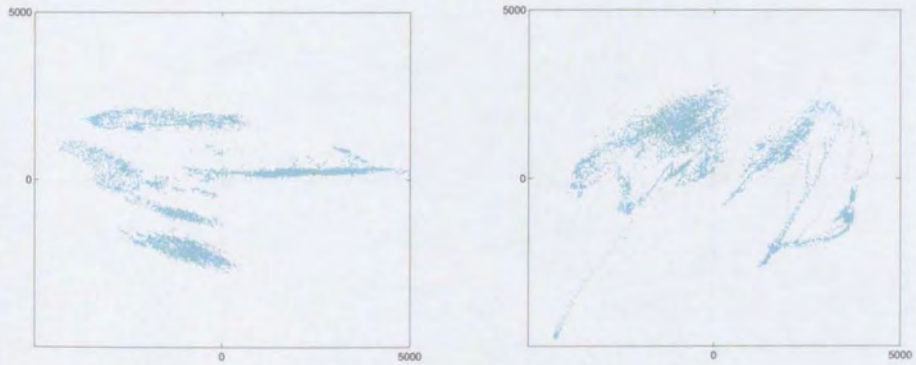



(c) region{3,10}

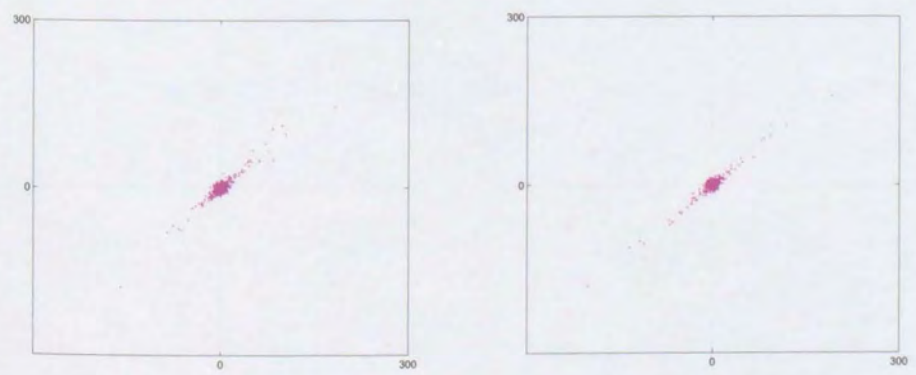
Figure 4.34: Eigen-representation in two dimensional space (part 1)




(a) region{4,3}

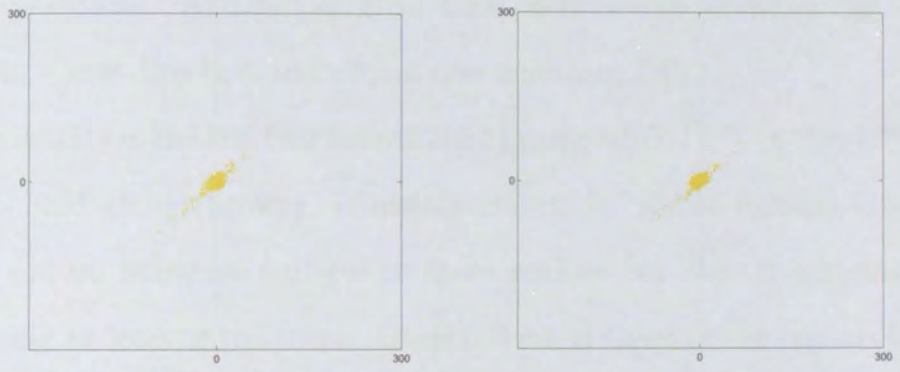



(b) region{6,1}

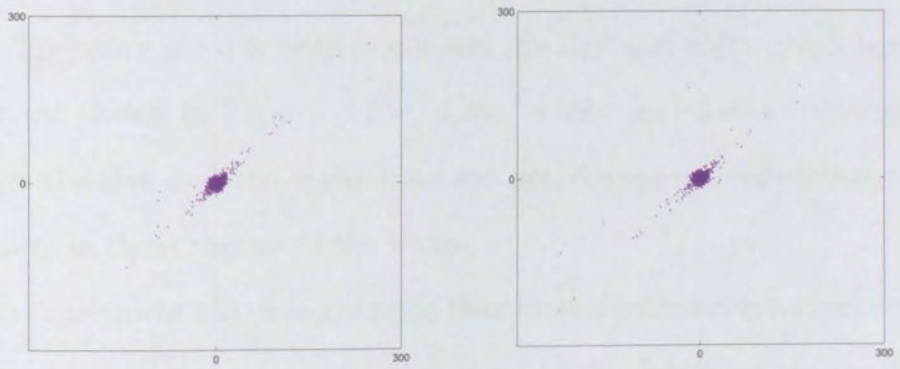



(c) region{6,5}

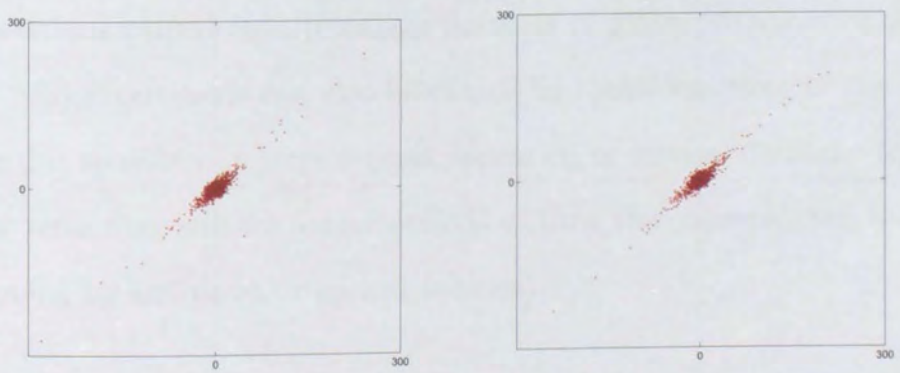
Figure 4.35: Eigen-representation in two dimensional space (part 2)




(a) region{7,7}




(b) region{7,11}




(c) region{8,3}

Figure 4.36: Eigen-representation in two dimensional space (part 3)

space have a form of clusters, Figure 4.35b, which correspond to particular scene conditions. Addition of night time observations stretches the eigen-space in a new direction and adds a new dominant PC.

Variability in the last four regions, {6,5} (magenta), {7,7} (yellow), {7,11} (violet) and {8,3} (brown), is mainly caused by global lighting changes. There are no reflective surfaces in those regions nor objects entering and remaining or leaving the scene. Observations of these image regions in the eigen-space form noisy clouds slightly stretched in the direction of global lighting variation. During the night time a street lamp illuminates these areas. Therefore the day only graph and the day and night graph are very similar, as shown in Figures 4.35c, 4.36a, 4.36b, and 4.36c. Alternations between the day and the night time are not dominant contributor to the variability in these regions of the image.

The experiment has demonstrated that most significant principal components in the eigen- representation indeed model the dominant changes in the data, those that affect grey-levels of a majority of pixels in the scene. Such are variations caused by alternating patterns of global illumination in the scene. Major variations can also be caused by transformations of the scene setting due to relatively large objects appearing or leaving the scene repeatedly or remaining still for longer periods of time (for example, the window blind going up and down or parked vehicles).

## 4.7 Conclusion

This chapter looked at the possibility of modelling the background in video data using an off-line PCA method. Although the computation cost may not be a problem in off-line solutions, it will certainly be a constraint in an online approach. The results and conclusions obtained for the off-line modelling are envisaged to be extended to an online approach. The details of the methodology were explained and the results illustrated with an example of a video surveillance dataset which included a variety of background changes. It was demonstrated that it is possible, by means of eigen analysis, to define a smaller set of dimensions which will capture most of the variation of the original high-dimensional dataset. In addition, the concepts of unimodal and multi-modal eigen-spaces were explored. However, several problems were noted. These are summarised below.

### 4.7.1 Low-dimensional data representation

The reliability of the low-dimensional eigen representation of the data is influenced by the number of observations in the dataset relative to the number of retained dimensions. The number of observations should be, if possible, at least $10^4$ times larger than the number of variables in the dataset. Otherwise, the accidental regularities in the random data may be modelled as significant. The desired accuracy is generally difficult to achieve for real video datasets. The dataset size is typically limited by the available storage and computation costs. The limited amount of data may introduce errors in the representation of the data in the eigen-space.

### 4.7.2 Selecting the number of PCs

The methods for selecting the number of PCs were also investigated in Section 4.2.6. It is generally not obvious a priori how many dimensions should be retained in the reduced eigen-space. The rules which determine the cut-off dimension are intuitive and there is no universal answer as to which rule gives the most suitable number of retained dimensions. The *broken stick* rule is often said to be the most adequate for real data. However, a more principled alternative was explored using a training dataset which contained both background-only and contaminated images.

An alternative dimensionality reduction approach defined a concept of the *hyper-sphere of backgrounds* in the eigen-space of the training data. The hyper-sphere of backgrounds is defined as a multi-dimensional volume containing a high percentage (e.g. 95%) of all background-only training points in the eigen-space. New observations, both unseen backgrounds and contaminated with foreground, are projected onto this eigen-space. Ideally, the boundary of the hyper-sphere will separate the background and the foreground observations. All background-only observations will be found inside the spher, while all the observations contaminated with foreground will be outside the sphere. This can be achieved by choosing an appropriate dimensionality of the space. In general, a low dimensional model has a very limited information about the data, only the most significant changes are modelled while smaller variations are ignored. Thus, all (or a great majority) of new points are placed closer to the centre inside the hyper-sphere. When new dimensions are added to the model, its ability to model finer variations

and distiguish between background and foreground increase. For a certain critical (or cut-off) number of dimensions the background and foreground observation points are well separated by the boundary of the hyper-sphere. Interestingly, this critical dimensionality corresponds to the one defined by the *broken stick* rule.

However, the dimensionality reduction using the hyper-sphere of backgrounds can only be performed for a training dataset where all observations are available. Nonetheless, this experiment has validated the *broken stick* rule as an appropriate choice for dimensionality reduction for modelling real outdoor video data. The hyper-sphere of backgrounds in the reduced eigen-space can then be used for the classification of observations as background-only (inside the hyper-sphere) or contaminated with foreground (outside the hyper-sphere).

### 4.7.3 Multi-modal modelling

The hyper-sphere of backgrounds was defined in the unimodal eigen-space where the background observations are modelled as one multivariate Gaussian distribution. However, in many real life cases observations may be wrongly classified in this global eigen-space, as shown in Figure 4.9.

When the contaminated proportion of an image is relatively small or similar to background grey-levels, the contaminated test image is likely to fall inside the limits of the global hyper-sphere and consequently be wrongly classified as a true background. It was proved that the multi-modal eigen-space, clustered in subspaces of observations of similar background conditions, pro-

vides more accurate classification results. In this context, the problem of dimensionality reduction for the cluster subspaces was discussed. It was shown that the full dimensionality of clusters maximizes the detection of contaminations, while at the same time classification of true backgrounds is poor. Therefore, the trade-off between the true positive rate and the false alarm rate is needed to determine the appropriate choice of the number of subspace dimensions.

### 4.7.4 Subsampling

In an attempt to further reduce the amount of processed data, the possibility and the limitations of subsampling a relatively small proportion of all the available data were also explored in Section 4.4. The lowest possible required amount of data was discussed in the case of both the unimodal and the multi-modal eigen-model. It was shown that for the multi-modal method, the lowest acceptable subsampling rate is generally higher than for the unimodal eigen-space. If the amount of subsampled data is too low, the multi-modal method becomes too sensitive to image changes, which causes wrong classification of background-only observations as contaminated.

However, any random subsampling from an image may also include contaminated pixels. This may produce an inaccurate background model. Therefore, a more selective approach to subsampling is needed. For example, depending on the contaminant size, it is possible to avoid subsampling from contaminated areas by subsampling from predefined image subregions. This way of controlled subsampling would provide a more accurate classification of

image observations from a very small amount of subsampled data. Combined with some knowledge of the scene, such as entry and exit points, and/or expected contaminant sizes, this method may quickly detect contaminations of the background from a relatively little information about the scene.

### 4.7.5 Results and discussion

The proposed method aims to model background changes in outdoor video sequences over long periods of time and to provide better understanding of the nature of its variability. For that purpose two datasets were created. One dataset was recorded at full frame rate over a short period few minutes and the second at one frame per minute over a long period of two and a half months offering a valuable collection of long days with a variety of weather conditions. The video frames were divided into smaller regions to facilitate the analysis of the data. In Section 4.5, a set of representative regions was selected for both datasets. The complexity of the datasets was demonstrated in terms of the difference between the successive frames, the mean pixel grey-level variations, and the standard deviation of pixel grey-level. However, although simple to compute, the mean and the standard deviation do not provide a meaningful description of the behaviour of the dataset. As an alternative, a more suitable approach using the modelling in the eigen-space was proposed.

It was demonstrated that the eigen-analysis approach provides better understanding of the nature of background variations in video datasets. Experiments showed that the most significant principal components in the eigen-

space indeed model the global lighting changes in the scene. The most significant components have variances equal to the largest eigenvalues of the data covariance matrix. An example in Section 4.6.3 illustrated that only two most significant dimensions provide a good representation of the global lighting changes in the scene.

This chapter addressed an issue of off-line modelling of video backgrounds when all the data are available for analysis. However, in real-life applications it is often required to analyse data in real-time, on-line rather than off-line. Relying on principles drawn from the results of the multi-modal off-line approach, a new adaptive multi-modal background modelling algorithm is proposed in the next chapter.

# Chapter 5

# Adaptive Multi-modal Background Modelling

## 5.1 Introduction

Real life outdoor video surveillance sequences often contain a variety of background changes including gradual and sudden light changes due to weather conditions, background motion such as swaying trees, or stationary objects being left or disappearing from the scene. Efficient modelling must provide a reliable model of the background pixels at any time instance to enable the correct classification of image observations as either true background or contaminated by foreground pixels. Due to constant changes in background, this is a challenging task.

In the previous chapter, it has been demonstrated by the example of an off-line batch algorithm that the multi-modal eigen-model approach provides a promising tool for foreground detection and observation classification. The

presence of the objects smaller than 1% of the observed image region was correctly detected. This is a great improvement compared to the uni-modal eigen-model approach, which only detected the presence of the objects larger than 10% of the image region. However, in real life applications it is often required to detect possible contaminations of the scene in real time.

In real-time applications the challenge is to perform both the classification and the update of the background model at each time instance. For this a different type of algorithm is proposed in this chapter; one that takes advantage of multi-modal eigen-modelling approach combined with an ability to develop and evolve in time in order to adapt to any changes in the background in a timely manner.

The proposed adaptive algorithm evolves incrementally with each new observation. At every time instance it adapts to the new background conditions using the knowledge of the newly acquired data and the accumulated knowledge of the current model. The update strategy is inspired by two previously reported methods: (a) the incremental eigen model update method [Hall et al., 1998], and (b) the improved mixture model approach [KaewTraKulPong and Bowden, 2001] applied to adaptive learning.

The novelty of the method proposed in this work is threefold. First, the mixture model, described by KaewTraKulPong and Bowden, refers to modelling of individual pixel variations by one dimensional Gaussian distributions in the grey-scale or color image space. The novelty of our approach is such that this principle is applied to image regions rather than pixels, where image observations are modelled with a mixture of multi-dimensional Gaussian

clusters in the eigen-space. Second, the incremental method of Hall et al. was developed for uni-modal eigen model which is not suitable for modelling a wide range of varied backgrounds in outdoor surveillance scenes. Therefore, the incremental update principle is modified and applied to a multi-modal model consisting of clustered eigen-subspaces rather than a single uni-modal eigen-space. Third, the method of Hall et al. imposes a very rigorous rule as a condition for adding new dimensions to the updated eige-space. However, such rule is not appropriate for modelling outdoor datasets where the successive observations may significantly vary. This would cause adding a new dimension for each new image, which contradicts the principle of the proposed low-dimensional modelling. Therefore, a more suitable method for adding new dimensions is proposed.

This chapter sets out a framework for the adaptive multi-modal eigen-modelling of image backgrounds and is organised as follows. Section 5.2 discusses the issues relevant to the on-line modelling, describes the principles of multi-modality and incremental adaptability. Section 5.3 describes the proposed algorithm, detailing the mathematical and statistical principles, the algorithm steps and parameters. In particular, distinct stages of the algorithm are explained: the model initialisation stage, observation classification stage and model update stage. The choice of algorithm parameters is also explained. Section 5.4 describes the results obtained by applying the proposed algorithm to a real video surveillance dataset. It describes the dataset and particular challenges it poses. The obtained results are discussed and compared to those obtained by the non-adaptive off-line algorithm and the

uni-modal approach. Finally, Section 5.5 concludes the chapter offering a critical look at the proposed adaptive multi-modal method with a discussion on its advantages and disadvantages.

## 5.2 Adaptive multi-modal modelling of backgrounds

### 5.2.1 Overview

This section addresses particular issues related to the on-line modelling of the background in outdoor surveillance scenes. An efficient model is expected to accurately represent the background at any time in order to enable an accurate classification of incoming observations as background-only or contaminated with foreground. The on-line modelling requires an adaptable model which evolves in time taking into account the changing background conditions.

The proposed algorithm performs modelling of backgrounds using clustered eigen-subspaces of reduced dimensionality. Clusters of points in the eigen-space represent observations with similar background conditions. The model is initialised with a training set of the first $N_t$ observations in the dataset. New observations are processed using the multi-modal model and classified either as a background or contaminated with foreground pixels. Using the information about the new observation and the accumulated knowledge about the dataset, the model adapts to the new background conditions incrementally.

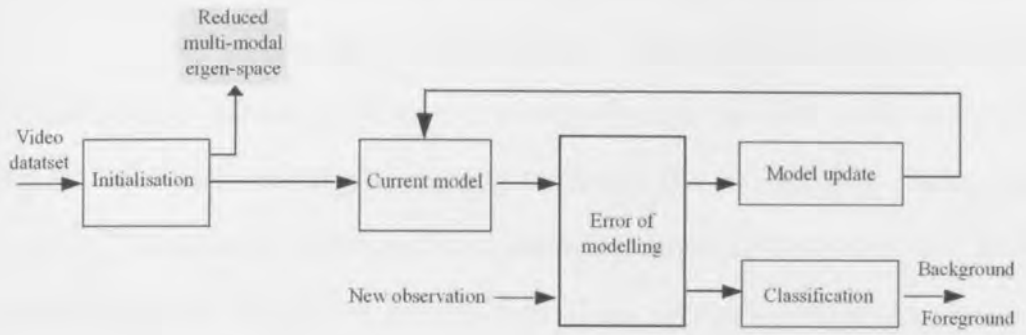



Figure 5.1: Adaptive multi-modal algorithm for classification

The initialisation phase creates an initial clustered model from the first $N_t$ available observations, which is used as a starting point in the model development. The model is initialised by applying an off-line PCA to the training set of observations to obtain an eigen representation of the training data. Next, the dimensionality of the eigen-space is reduced. This reduced eigen-space is referred to as the *global eigen-space*. The training data projected into the global eigen space is then clustered into a set of clusters, or *modes*, which gather observations with similar lighting conditions. A second PCA is performed on each cluster and a new eigen-space for each cluster is obtained. The dimensionality of each cluster eigen-space is further reduced. These clustered eigen-spaces are referred to as the *local eigen-spaces*. The structure of the multi-modal model is described in Section 5.2.2.

The algorithm for adaptive classification is illustrated by the diagram in Figure 5.1. After the initialisation stage new observations are acquired in real-time or when they become available. The new observation may contain either one of the *previously observed background* conditions, a *new background*

lighting condition or *foreground* objects. The model aims to classify the new observation as background or foreground. The classification is performed by means of a test designed to determine whether the new observation may be matched with one of the existing modes of the multi-modal background model. The model is incrementally updated for each new observation. Every new observation brings new information about the background changes. The model incorporates this information in order to adapt to new background conditions. The update is based on the information about the new observation and the error of modelling introduced by the inaccurate outdated model. The evolving background conditions may require a more or less significant model update depending on the nature and the extent of the observed background changes. In some cases a simplified update of one local eigen-space is sufficient, whereas in other cases a new dimension or indeed a new cluster (or mode) is required.

Section 5.2.2 describes the multi-modal model structure. Section 5.2.3 explains the principles of the incremental update of a uni-modal eigen model. Section 5.2.4 proposes a novel method for incremental update of a multi-modal eigen model. Section 5.2.5 outlines the application of the proposed update strategy.

### 5.2.2 Multi-modal model structure

The multi-modal idea is inspired by the mixture model approach of KaewTraKulPong and Bowden where the variations of individual pixels in time are modelled by a mixture of one-dimensional Gaussian distributions. Here,

the mixture model approach is modified and applied to image regions rather than pixels. It is assumed that the vector of image region grey-levels can be represented by multi-dimensional Gaussian clusters in the eigen-space. A new image acquired at a time instance $t$ is either modelled by the one of the existing clusters of the model or a new cluster is created using the information about the new image observation. Using principles similar to those of the Gaussian mixture model (GMM) [KaewTraKulPong and Bowden, 2001], the eigen multi-modal model evolves in time to accommodate new incoming observations.

The multi-modal model is a set of clustered observation points in the global eigen-space, where each cluster is defined by its own local eigen-space. Clusters represent previously learned types of background conditions. The local eigen-spaces model the variability of the backgrounds in each cluster. The model consists of $K$ clusters, or modes, with the dimensionality significantly reduced compared to the original dataset of image observations.

The dimensionality of the original image space is $p$, which is the number of pixel variables in an image. Using PCA the dimensionality can be significantly reduced to $m << p$ dimensions such that the most significant information about the data is preserved.

The global eigen-space model $\mathcal{M}_g$ is defined as

$$\mathcal{M}_g = \left\{ \underline{\underline{\mathbf{P}}}_g, \underline{\mathbf{\Lambda}}_g, \underline{\mu}_g, m \right\} \tag{5.1}$$

where $\underline{\underline{\mathbf{P}}}_g$ is a $p$ x $m$ rotation matrix, $\underline{\mathbf{\Lambda}}_g$ is a vector of $m$ eigenvalues, $\underline{\mu}_g$ is $m$-dimensional mean, and $m$ is the dimensionality of the global eigen-space.

A local eigen-space model $\mathcal{M}_i^{(t)}$ for the i$^{th}$ cluster with $m_i \leq m$ dimensions at a time instance $t$ is defined as

$$\mathcal{M}_i^{(t)} = \left\{ \underline{\underline{\mathbf{P}}}_i^{(t)}, \underline{\mathbf{\Lambda}}_i^{(t)}, \underline{\mu}_i^{(t)}, \omega_i^{(t)}, m_i^{(t)} \right\}, \tag{5.2}$$

where $\underline{\underline{\mathbf{P}}}_i^{(t)}$ is a $p$ x $m_i$ rotation matrix, $\underline{\mathbf{\Lambda}}_i^{(t)}$ is a vectors of $m_i$ eigenvalues, $\underline{\mu}_i^{(t)}$ is the $m_i$-dimensional cluster mean, $\omega_i^{(t)}$ is the weighting coefficient at time instance $t$, and $m_i^{(t)}$ is the dimensionality of the cluster at time $t$. The modes may be of the same dimensionality as the global-eigen space or further reduced to $m_i \leq m$ number of dimensions. Furthermore, the modes may be all of the same or variable number of dimensions, depending on the adopted dimensionality reduction approach. To support underlying aim of real-time implementation we reduce the dimensionality of the model as much as possible while the significant proportion of variability in the data is still preserved. The role of the weight $\omega_i^{(t)}$ is similar to that of the weights in the GMM method, i.e. it describes the portion of the data accounted for by the Gaussian.

Finally, the multi-modal model structure $\mathcal{M}^{(t)}$ of $K$ modes at a time $t$ is defined by the set of clustered local eigen-spaces $\mathcal{M}_i^{(t)}$ where $i = 1, ..., K$.

$$\mathcal{M}^{(t)} = \left\{ \mathcal{M}_1^{(t)}, \mathcal{M}_2^{(t)}, \ldots, \mathcal{M}_K^{(t)} \right\} \tag{5.3}$$

For clarity of mathematical expressions the subscript $(t)$ will be omitted in the text from now on.

### 5.2.3 Principles of incremental eigen-space update

Hall et al. proposed a method for incremental update of the uni-modal eigen-space. There are two aspects of this method [Hall et al., 1998]. First, at any time instance the model is estimated from the new observation and the model at the previous time instance. Second, the update method includes an ability to add new dimensions to the space. The incremental method of Hall et al. is outlined in this section.

Let us consider a $p$-dimensional dataset with a covariance matrix $\underline{\underline{\mathbf{S}}}$ and its reduced dimensionality uni-modal eigen-space $\mathcal{M}_u$ defined as

$$\mathcal{M}_u = \{\underline{\underline{\mathbf{P}}}, \underline{\mathbf{\Lambda}}, \underline{\mu}, m\} \tag{5.4}$$

where $\underline{\underline{\mathbf{P}}}$ is a $p$ x $m$ rotation matrix, $\underline{\mathbf{\Lambda}}$ is a vector of $m$ eigenvalues, and $\underline{\mu}$ is $m$-dimensional mean of the eigen-space. At each time instance a new $p$-dimensional observation $\underline{\mathbf{x}}$ is acquired and projected onto the model $\mathcal{M}_u$. The projection $\underline{\mathbf{b}}$ of the vector $\underline{\mathrm{x}}$ onto $\mathcal{M}_u$ is obtained as

$$\delta\underline{\mathbf{x}} = \underline{\mathbf{x}} - \underline{\mu} \tag{5.5}$$

$$\underline{\mathbf{b}} = \underline{\underline{\mathbf{P}}}^T \delta\underline{\mathbf{x}} \tag{5.6}$$

The *error of modelling* is represented by the *residue vector* $\underline{\mathbf{h}}$, which is calculated as the difference between the new $p$-dimensional observation $\underline{\mathbf{x}}$ and its recreated version obtained from the reduced $m$-dimensional eigen-space

[Hall et al., 1998]. The residue vector is defined as

$$\underline{\mathbf{h}} = \delta\underline{\mathbf{x}} - \underline{\underline{\mathbf{P}}}\,\underline{\mathbf{b}} \tag{5.7}$$

or by substituting $\underline{\mathbf{b}}$

$$\underline{\mathbf{h}} = (\mathrm{I} - \underline{\underline{\mathbf{P}}}\,\underline{\underline{\mathbf{P}}}^T)\,\delta\underline{\mathbf{x}} \tag{5.8}$$

The model is updated each time using only the residue vector $\underline{\mathbf{h}}$ and the knowledge of the current estimate of the covariance matrix, $\underline{\underline{\mathbf{S}}}$.

The updated space rotation matrix $\underline{\underline{\mathbf{P}}}'$ can be derived from the eigen decomposition of the updated covariance matrix $\underline{\underline{\mathbf{S}}}'$.

$$\underline{\underline{\mathbf{S}}}'\,\underline{\underline{\mathbf{P}}}' = \underline{\underline{\mathbf{P}}}'\,\underline{\mathbf{\Lambda}}' \tag{5.9}$$

The new covariance matrix is defined as

$$\underline{\underline{\mathbf{S}}}' = \alpha\,\underline{\underline{\mathbf{S}}} + (1-\alpha)\,\delta\underline{\mathbf{b}}\delta\underline{\mathbf{b}}^T \tag{5.10}$$

where $\alpha$ is a weighting function and $\delta\underline{\mathbf{b}}$ is the new projected observation.

The new rotation matrix $\underline{\underline{\mathbf{P}}}'$ is updated by inclusion of a new orthogonal unit vector. The new orthogonal vector of choice is the residue unit vector $\widehat{\underline{\mathbf{h}}}$, defined as follows

$$\widehat{\underline{\mathbf{h}}} = \begin{cases} \dfrac{\underline{\mathbf{h}}}{\|\underline{\mathbf{h}}\|} & \text{if } \|\underline{\mathbf{h}}\| \neq 0 \\ \\ \underline{\mathbf{0}} & \text{otherwise} \end{cases} \tag{5.11}$$

Essentially, Equation 5.11 represents a test for adding new dimensions to the updated eigen-space. However, it can be seen that the a new dimension is added each time, except when the residue vector $\underline{\mathbf{h}}$ is zero. In other words, a new dimension is created whenever a new observation does not lie exactly within the current eigen-subspace. In real-life applications, however, any new observation is unlikely to lie exactly within the subspace. Although possibly small, the residue $\underline{\mathbf{h}}$ is unlikely to be zero. Thus, new dimensions are continually added. What is need is a method for detecting if this small non-zero residue $\underline{\mathbf{h}}$ is significantly large to add a new dimension. We propose a solution to this problem in Section 5.2.4.

The new updated rotation matrix $\underline{\underline{\mathbf{P}}}'$ is calculated using the residue vector as

$$\underline{\underline{\mathbf{P}}}' = [\underline{\underline{\mathbf{P}}}, \widehat{\underline{\mathbf{h}}}]\, \underline{\underline{\mathbf{R}}} \tag{5.12}$$

Substituting Equations 5.10 and 5.12 in 5.9 yields another eigen decomposition

$$\underline{\underline{\mathbf{D}}}\, \underline{\underline{\mathbf{R}}} = \underline{\underline{\mathbf{R}}}\, \underline{\underline{\mathbf{\Lambda}}}' \tag{5.13}$$

where

$$\underline{\underline{\mathbf{D}}} = \alpha \begin{bmatrix} \underline{\underline{\mathbf{S}}} & \underline{\mathbf{0}} \\ \\ \underline{\mathbf{0}}^T & 0 \end{bmatrix} + (1-\alpha) \begin{bmatrix} \underline{\mathbf{b}}\, \underline{\mathbf{b}}^T & \underline{\gamma}\, \underline{\mathbf{b}} \\ \\ \underline{\gamma}\, \underline{\mathbf{b}}_c^T & \underline{\gamma}^2 \end{bmatrix} \tag{5.14}$$

and $\underline{\gamma} = \widehat{\underline{\mathbf{h}}}^T \delta \underline{x}$.

Matrix $\underline{\underline{\mathbf{R}}}$ is therefore found from the eigen decomposition given by Equation 5.13. Specifically, the set of updated subspace eigenvalues $\underline{\mathbf{\Lambda}}'$ is directly calculated as eigenvalues of $\underline{\underline{\mathbf{D}}}$. Matrix $\underline{\underline{\mathbf{R}}}$ is then computed as a set of eigen-

vectors of $\underline{\underline{\mathbf{D}}}$ and by substitution in Equation 5.12 the updated subspace rotation matrix $\underline{\underline{\mathbf{P}}}'$ is obtained. It is assumed that the updated space contains one additional dimension.

### 5.2.4 Incremental update of multi-modal model

The incremental method of Hall et al. described in Section 5.2.3 suffers from a number of limitations. In this section we propose appropriate modifications.

There are two main limitations of the method of Hall et al.. First limitation is that it was derived for the uni-modal eigen-space which is too general for the purpose of modelling real-life outdoor surveillance scenes. The uni-modal model does not provide enough flexibility to accommodate a large range of background data. Second limitation is that the additional dimension is added each time when the new observation does not lie exactly within the current eigen-space, i.e. when the residue vector $\underline{\mathbf{h}}$ is non-zero. However, in real-life datasets, any new observation is unlikely to lie exactly within the subspace. Thus, where this non-zero $\underline{\mathbf{h}}$ is small, which occurs frequently, new dimensions are continually added. What is needed is a method for detecting if $\underline{\mathbf{h}}$ is significantly large to add a new dimension. Therefore, the approach of Hall et al. is not suitable for modelling of a variable background with diverse illumination changes often found in real-life outdoor surveillance scenes. Thus, an adaptive multi-modal model is proposed as a more appropriate solution.

The proposed adaptive multi-modal approach suggests two modifications for the incremental update. First, the principle of the incremental update of

Hall et al. is applied to the multi-modal eigen-model $\mathcal{M}$, where the eigen-subspaces of individual modes $\mathcal{M}_i$ are updated. Second, the requirement to add a new dimension for all non-zero $\underline{\mathbf{h}}$ vectors, as suggested by Hall et al., is relaxed by inclusion of a preset threshold $\Phi$ on the magnitude of $\underline{\mathbf{h}}$, which provides a more robust model update. This is expected to reduce the number of times a new dimension is required. (The additional eigen-vector is orthogonal to the existing set of eigen-vectors and is determined by the direction to the new observation point. Once the new observation is added the eigen-vectors are recalculated as described in Section 5.2.3.)

We introduce the notion of the *external modelling error*, $\Delta\underline{\mathbf{h}}$, which represents the component of the residue vector $\underline{\mathbf{h}}$ in dimensions other than those of the $m_i$-dimensional eigen-subspace $\mathcal{M}_i$. The external modelling error $\Delta\underline{\mathbf{h}}$ is defined as

$$\Delta\underline{\mathbf{h}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \underline{\mathbf{h}}_{m_i+1} \\ \underline{\mathbf{h}}_{m_i+2} \\ \vdots \\ \underline{\mathbf{h}}_{p} \end{bmatrix} \tag{5.15}$$

where $\underline{\mathbf{h}}_j$ is the j$^{th}$ component of the $p$-dimensional vector $\underline{\mathbf{h}}$. (Note that components $\underline{\mathbf{h}}_j$, where $j = 1, .., m$, are negligible.)

The test for adding new dimensions, which was defined in Equation 5.11, can now be modified. A new dimension is added to the cluster subspace $\mathcal{M}_i$

only when the magnitude $\|\Delta\underline{\mathbf{h}}\|$ exceeds the threshold $\Phi$.

$$\widehat{\underline{\mathbf{h}}} = \begin{cases} \dfrac{\underline{\mathbf{h}}}{\|\underline{\mathbf{h}}\|} & \text{if} \quad \|\Delta\underline{\mathbf{h}}\| > \Phi \\ \\ \underline{\mathbf{0}} & \text{else} \end{cases} \tag{5.16}$$

In other words, when the external error of modelling by the $m_i$-dimensional cluster $\mathcal{M}_i$ is above the threshold $\Phi$, the new orthogonal vector $\widehat{\underline{\mathbf{h}}}$ is added to the updated eigen-subspace $\mathcal{M}_i^{'}$ and therefore its dimensionality is incremented. On the other hand, when the error of modelling is below the threshold $\Phi$, the new orthogonal vector $\widehat{\underline{\mathbf{h}}}$ is reduced to $\underline{\mathbf{0}}$ and therefore a new dimension is **not** added to the updated space.

The growing dimensionality $m_i$ of the cluster's subspace $\mathcal{M}_i$ is limited by a post-processing step of eliminating the smallest dimensions. After each update, the eigenvalues of the cluster's subspace are checked. All dimensions corresponding to eigenvalues smaller than the estimated noise are eliminated straight away. However, the dimensionality of the subspace will inevitably grow as new observations are added to it. This problem can be solved in the following manner. It is possible to define some critical number of dimensions beyond which the cluster's dimensionality $m_i$ should not increase; for example a proportion of the global space dimensionality $m$. At this point, when $m_i$ reaches this critical value, the *broken stick* rule may be applied to the array of the cluster's eigenvalues to reduce $m_i$.

The details of the update method, including how the threshold $\Phi$ is calculated, are described later on in Section 5.3.7.

### 5.2.5 Update strategy

This section outlines the proposed update strategy. The detailed description is given in Section 5.3.7. The nature of the new observation will influence the choice of the update strategy. Three cases can be identified according to the semantics of the new observation:

- Case 1: The new observation represents a *previously observed* background. It can be *matched* to one of the clusters of the model and is referred to as the *mode-matched observation.* The matching cluster is called the *mode-matched cluster.* Its model is designated by $\mathcal{M}_c$ and defined similarly to other modes as

$$\mathcal{M}_c = \left\{ \underline{\underline{\mathbf{P}}}_c, \underline{\mathbf{\Lambda}}_c, \underline{\mu}_c, \omega_c, m_c \right\}. \tag{5.17}$$

  In this case, one of two update sub-strategies is possible, which is determined by the magnitude of the external modelling error, $\|\Delta\underline{\mathbf{h}}\|$, observed in the global eigen-space. The possible update sub-strategies are:

  i) The magnitude $\|\Delta\underline{\mathbf{h}}\|$ is smaller than the threshold $\Phi$.

  In this case, it is assumed that the new observation is very similar to other points in the matched cluster. Essentially it is assumed that the cluster has some noise contribution in all the dimensions not within the cluster's subspace. The cluster's mean and the covariance are updated only, without adding new dimensions to the cluster's local eigen-subspace.

ii) The magnitude $\|\Delta\underline{\mathbf{h}}\|$ is larger than the threshold $\Phi$.

In this case, it is assumed that the new observation contains significant components in dimensions other than those of the matched mode. The mean and the covariance of the matched cluster are updated and a new dimension is added.

- Case 2: The new observation represents a new *unknown* background. It can not be matched to any of the clusters of the model. In this case, the multi-modal model is updated by adding a new mode with initially high variance and low weight, and with the mean in the new observation point. In keeping with the GMM approach, one of the old clusters is chosen as the least relevant and removed.

- Case 3: The new observation represents an image *contaminated* with foreground objects. Ideally, we don't want to use the contaminated observation to update the background model. However, the algorithm cannot tell at this stage whether this observation is a new unknown background (described in case 2) or contaminated with foreground. Therefore, the multi-modal model is updated by adding a new mode with initially high variance and low weight, and with the mean in the new observation point. Assuming that a foreground object is moving in the scene, any newly created cluster will remain in the model only for a short time before being replaced, as the least relevant, by a new cluster. Indeed, it might be a foreground object which remains in the scene and becomes a part of background. In this case, the new cluster becomes a new mode of the model.

In all cases, the notation $\mathcal{M}'$ will be used to identify the updated multi-modal model. Similarly, $\mathcal{M}_c'$ designates the updated model of the mode-matched cluster.

### 5.2.6 Complexity of the algorithm

At every time instance when a new observation is acquired the model needs to adapt to reflect the new conditions. There are two main computational steps that need to be performed each time: finding the new covariance matrix and recalculating the new eigen-space from the new covariance matrix.

If the batch method is performed at every time instance that involves recalculation of the covariance matrix for all observations obtained in the past plus the new observation. Even if not all but only a certain number of past observations was considered, that would still involve keeping track of a large number of observations (for details on the required dataset size refer to Section 4.2.4). The incremental method, on the other hand, allows for a relatively simple estimation of the new covariance matrix using only the information about the old covariance matrix and the new observation, Equation 5.14.

At the core of any PCA algorithm is the eigen decomposition of the covariance matrix of the data. The complexity of the eigen decomposition is $\mathcal{O}(m^3)$ where $m$ is the number of dimensions. Therefore, in order to reduce the computational cost, it is essential to keep the number of dimensions as low as possible. In the case of the adaptive multi-modal model the eigen decomposition, Equation 5.13, is performed when the mode-matched cluster

$\mathcal{M}_c$ is updated. The complexity of the multi-modal algorithm is therefore reduced to $\mathcal{O}(m_c^3)$ where $m_c$ is the dimensionality of the mode-matched cluster. (It will be shown later in Section 5.4.3 that $m_c$ may be reduced to as few as 4 to 7 dimensions.)

Section 4.3.4 illustrated the advantages of using the multi-modal model over the unimodal for detection of contaminated observations when the dimensionality of the data is greatly reduced (to only few dimensions). Having this advantage in mind an adaptive multi-modal algorithm is proposed.

## 5.3 Adaptive multi-modal algorithm

### 5.3.1 Overview

The proposed algorithm summarised in Figure 5.2, performs adaptive modelling of backgrounds using clustered eigen-subspaces of reduced dimensionality. Points in the eigen-space are grouped into clusters with similar background conditions. The model is initialised with a training set of observations. The training set consists of $N_t$ first observations in the dataset. On the basis of the position of the new observation in the local eigen-subspace and the modelling error, the classification and the model update are performed following the steps in Figure 5.2.

### 5.3.2 Algorithm

The flow chart in Figure 5.2 represents the structure of the proposed algorithm. The algorithm consists of a number steps.

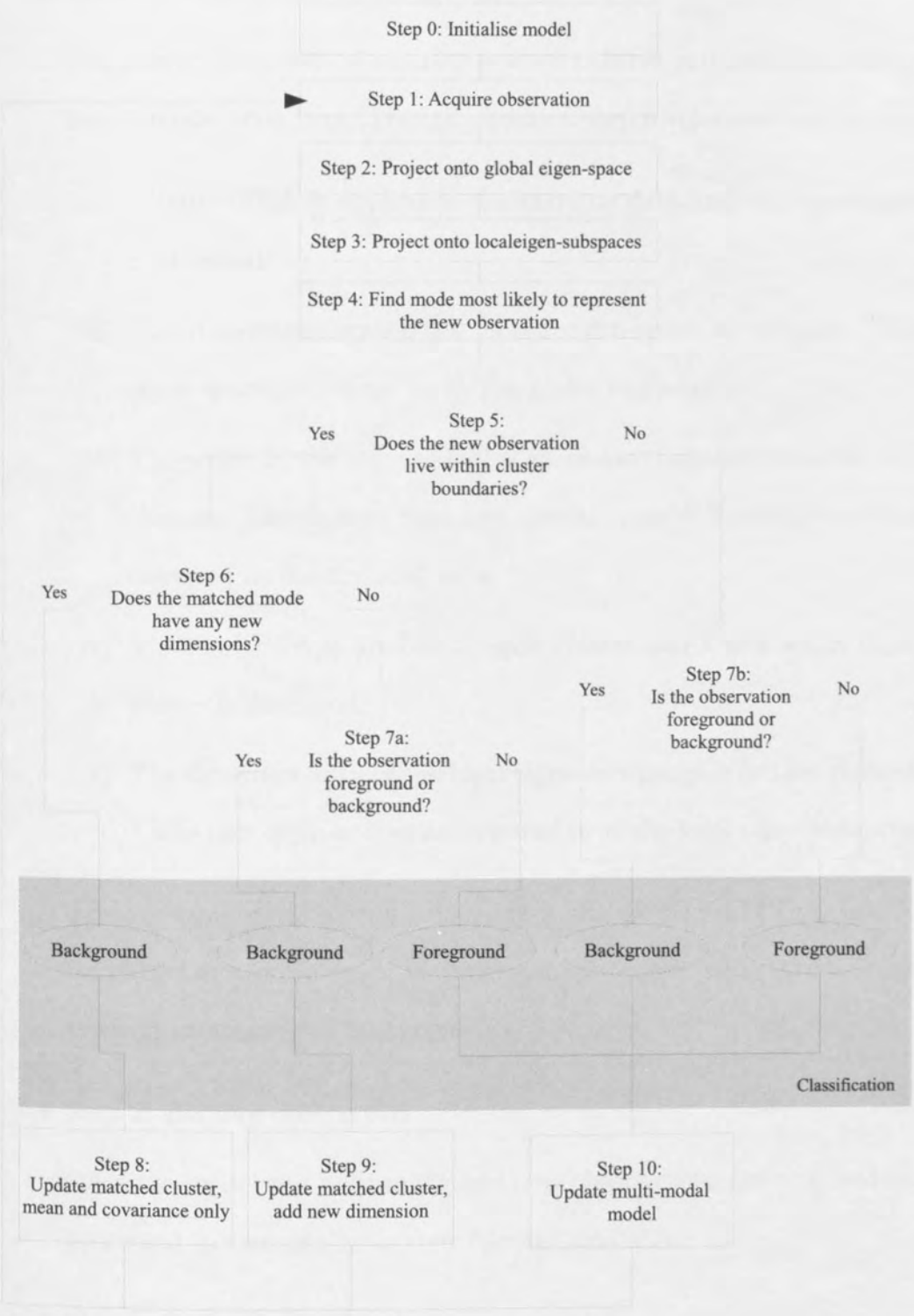



Figure 5.2: Adaptive multi-modal algorithm

- Step 0: Initialise model

  The role of the initialisation step is to establish a suitable initial multi-modal model structure. The initialisation step involves several actions.

  i) A batch PCA is applied to the training data and the eigen-space is obtained.

  ii) The dimensionality of the global eigen-space is reduced. This eigen-space is referred to as the *global eigen-space.*

  iii) The points in the reduced global space are clustered into a set of $K$ clusters. The clusters represent similar types of lighting conditions observed in the training data.

  iv) A second PCA is applied to each cluster and a new set of eigen-spaces is produced.

  v) The dimensionality of the local eigen-subspaces is further reduced. These new eigen-spaces are referred to as the *local eigen-subspaces.*

  The outcome of the initialisation step is the model $\mathcal{M}$ of Equation 5.3 composed of a set of reduced dimensionality eigen clusters, which correspond to clusters of backgrounds.

- Step 2: Acquire observation

  Once the model has been initialised new observations are acquired and processed incrementally as they become available.

- Step 3: Project onto global eigen-space

  A newly acquired observation is projected onto the global eigen-space.

- Step 4: Project onto local eigen-subspaces

  The new global eigen-space projection point is projected onto the local eigen-subspace of each cluster in turn.

- Step 5: New observation mode-matching test

  This step investigates whether the new observation is recognised by the current model and can be modelled by one of the existing low-dimensional cluster modes. This step is discussed in detail in Section 5.3.5.

- Step 6: Any other significant dimensions?

  This step investigates whether the new observation contains significant components in dimensions other than the few dimensions of the matched cluster. The method is detailed in Section 5.3.6.

- Step 7a-b: Foreground/background classification

  The classification is based on the error of modelling in the image space which is compared to a threshold estimated from the intrinsic noise of the data set. The details of the classification step are given in Section 5.3.6.

- Step 8: Update the mode-matched cluster

  The update of the mode-matched cluster is performed. The mean and the eigen-space of the cluster are updated without adding any new dimensions, as described in Section 5.3.7.

- Step 9: Update the mode-matched cluster and add a dimension

  The update of the matched cluster is performed. The mean and the eigen-space of the cluster are updated and a new dimension is added to the space, as described in Section 5.3.7.

- Step 10: Update the multi-modal model

  A new cluster is created with the new observation as its mean, high variants and low weight, as explained in Section 5.3.8. The existing clusters are ordered by the relevance and the last one is replaced by the new cluster.

The following sections describe the details of the algorithm steps.

### 5.3.3 Model Initialisation

The model is initialised by a training set of $N_t$ observations. A batch PCA is performed on the training set to obtain the global eigen-space. The global eigen-space is a high-dimensional multivariate space with as many dimensions as there are variables in the training dataset; that is $p$ dimensions, where $p$ is the number of pixels in a frame of the surveillance video dataset. The dimensionality of the global eigen-space may be reduced to $m << p$ dimensions, while the significant variability in the data is preserved, as discussed in Section 4.2.6. The dimensionality reduction method of choice is the *broken stick* rule, [Jolliffe, 2002, p.93].

The reduced $m$-dimensional global eigen-space is clustered using $K$-means method into $K$ clusters. The clusters, or modes, are expected to model observations with similar illumination conditions of the scene. Another PCA

is performed on each mode in turn to compute its local eigen-space. Initially, the local eigen-spaces are of the same dimensionality as the global eigen-space. However, the dimensionality of each eigen-subspace may be individually further reduced to $m_i \leq m$. The number of retained dimensions for each initial cluster $m_i$ is calculated individually by applying the *broken stick* rule of Jolliffe to the set of eigenvalues of each cluster.

The initial multi-modal model structure $\mathcal{M}^{(0)}$ of $K$ modes, Equation 5.3, is defined by the set of clustered local eigen-spaces as:

$$\mathcal{M}^{(0)} = \left\{ \mathcal{M}_1^{(0)}, \mathcal{M}_2^{(0)}, \ldots, \mathcal{M}_K^{(0)} \right\} \tag{5.18}$$

where

$$\mathcal{M}_i^{(0)} = \left\{ \underline{\underline{\mathbf{P}}}_i^{(0)}, \underline{\mathbf{\Lambda}}_i^{(0)}, \underline{\mu}_i^{(0)}, \omega_i^{(0)}, m_i^{(0)} \right\}, \tag{5.19}$$

and $i = 1, ..., K$.

The initial model is a multi-modal reduced dimensionality eigen representation of the training set. It provides a suitable structure for an adaptive modelling of multi-modal multi-variate high-dimensional data.

### Weighting coefficients

The initial set of coefficients $\omega_i^{(0)}$ $(i = 1, ..., K)$ is associated with the initial set of clusters. Weights capture the perceived relevance of clusters. At the initialisation stage, the largest cluster is perceived as the most relevant. The initial coefficients are derived from the probability of a training observation point belonging to an initial cluster. In Equation (5.20), $\omega_i^{(0)}$ is the initial weight of the $i^{th}$ cluster, $N_i$ is the number of points belonging to the $i^{th}$

cluster and $N_t$ is the size of the training dataset. All weights add up to 1. The weights will be updated at each new time instance.

$$\omega_i^{(0)} = \frac{N_i}{N_t} \tag{5.20}$$

### 5.3.4 Choosing the initialisation parameters

An initial model is constructed from a training set composed of the first $N_t$ image observations. A batch PCA is performed on the entire training dataset and the global-eigen space is obtained. The dimensionality of the global eigen-space is then reduced using the *broken stick* rule. The observation points in the reduced space are the clustered in $K$ using the $K$-means clustering technique. A second PCA is performed on each cluster individually to obtain a set of $K$ local eigen-spaces. Finally, the dimensionality of each local eigen-space is further reduced using *broken stick* rule. As a result, the initial model is composed of a set of low-dimensional eigen-subspaces which model the modes of background variations in the training dataset. The choice of initialisation parameters is dependent on the nature of the dataset. The choice of the training data, the number of modes $K$, and the dimensionality of modes are discussed in this section.

#### Training data

The approach for choosing the training data is illustrated by the example of the dataset described in Section 4.5.2. This dataset represents a continuous video recording of an outdoor scene with significant light variations due to

changing weather conditions. Furthermore, the dataset is of a rather low frame rate of one frame per minute. It is assumed that the on-line adaptive multi-modal algorithm will be deployed to model this scene over a long period of time of many days. Therefore, considering the nature of the dataset cycling through days and the low frame rate of few hundred frames per day cycle, it is expected that one day of data will provide a sensible choice of the training dataset. The model may be inialised with the first $N_t$ frames of the sequence covering one day of data.

### Choosing the initial number of clusters

The initial number of eigen-subspace clusters is estimated by a simple analysis of the training data, which was chosen to include a time interval of one day. The number of clusters should roughly correspond to the number of different types of backgrounds appearing throughout a day of data. Too many clusters may cause unnecessary fragmentation of the data points, which may result in wrong classification of new backgrounds unable to fit in any of the small clusters of known backgrounds. Too few clusters risk to result in wrong classification of images contaminated with small foreground objects, which may go undetected in a large cluster of observations with similar background conditions.

The estimation of the initial number of modes is illustrated by the example of the dataset described in Section 4.5.2. A number of different days is analysed. Each day of data is clustered in 5, 10, 15, 20, 25 and 30 clusters. The number of observations assigned to each cluster is observed. The goal is to determine the number of clusters which is most likely to correspond to

the number of different types of backgrounds appearing throughout a day of data. The test is based on the assumption that clusters which contain fewer than 10% of the total number of observations result from over-fragmetation of the data. These points are more likely to belong to one of the larger clusters. Therefore, clusters with more than 10% of total number of observations are considered as valid. The number of valid clusters for each day is shown in Table 5.1.

| | | K-Means Clustering | | | | | |
|---|---|---|---|---|---|---|---|
| Day | Observations | K=5 | K=10 | K=15 | K=20 | K=25 | K=30 |
| 1 | 918 | 5 | 5 | 2 | 1 | 0 | 0 |
| 2 | 949 | 4 | 4 | 4 | 1 | 0 | 0 |
| 3 | 911 | 3 | 5 | 4 | 3 | 3 | 1 |
| 4 | 1175 | 3 | 5 | 5 | 3 | 2 | 0 |
| 5 | 960 | 3 | 6 | 4 | 1 | 0 | 0 |
| 6 | 970 | 4 | 4 | 4 | 4 | 0 | 0 |
| 7 | 949 | 4 | 5 | 4 | 3 | 0 | 0 |
| 8 | 951 | 5 | 6 | 5 | 2 | 3 | 0 |
| 9 | 880 | 4 | 6 | 3 | 1 | 1 | 0 |
| average | 962 | 3.5 | 5.1 | 4.3 | 2.2 | 1 | 0.1 |

Table 5.1: Number of clusters per day

The maximum average number of valid clusters obtained in this example is 5, when $K$-means clustering with $K = 10$ is used. The goal is to aggregate rather than to fragment the data. When the space is fragmented into small clusters the mode-matching of the new background images is more likely to fail. As a result, new clusters may be unnecessarily created while the existing still relevant clusters are prematurely replaced. This may cause an increased risk of misclassification of background images. Therefore, $K$=10 is adopted as a suitable number of initial clusters for this dataset.

#### Dimensionality of modes

The modes of the training set represent clusters of observation points with similar lighting conditions. These form a set of subspaces in the global eigen-space of the dataset. For each cluster a local eigen-model is computed. The data space now consists of multiple local eigen-subspaces modelling lighting conditions in the scene. These local eigen-subspaces will initially share the same dimensionality as the global eigen-space. However, the number of dimensions in each local subspace may be further reduced. The dimensionality reduction method of choice is the *broken stick*, as discussed in Section 4.2.6.

### 5.3.5 Locating the most representative mode

After the initialisation stage new observations are acquired when they become available. The goal of the algorithm is to classify the new image as a true background or contaminated with foreground. A new observation image may contain a background similar to one of the already observed background conditions modelled by the clusters,it may contain an unknown background, or it may contain a foreground object. The classification and the model update strategy depend on the nature of the new observation as presented in Section 5.2.4.

The *mode-matching* test is designed to determine whether the new observation belongs to any of the known backgrounds. The known backgrounds are represented by the clusters of the current model and their local eigen-subspaces. The test shows whether the new observation may be *mode-matched* with any of the clusters.

**Mode-matching**

A new observation vector $\underline{\mathbf{x}}$ is acquired and the global mean $\underline{\mu}_g$ is subtracted from it. The new observation difference vector $\delta\underline{\mathbf{x}}$ is projected onto the reduced $m$-dimensional global eigen-space using the rotation matrix $\underline{\underline{\mathbf{P}}}_g$.

$$\delta\underline{\mathbf{x}} = \underline{\mathbf{x}} - \underline{\mu}_g \tag{5.21}$$

$$\underline{\mathbf{b}}_g = \underline{\underline{\mathbf{P}}}_g^T \delta\underline{\mathbf{x}} \tag{5.22}$$

The projection of the new observation is tested against all existing clusters in turn in an attempt to find a matching cluster. The mode-matching test consists of the three following steps:

i) The point $\underline{\mathbf{b}}_g$ is subtracted from cluster's mean and projected onto the $m_i$-dimensional local eigen-subspace of the cluster, where $m_i \leq m$.

$$\delta\underline{\mathbf{b}}_g = \underline{\mathbf{b}}_g - \underline{\mu}_i \tag{5.23}$$

$$\underline{\mathbf{b}}_i = \underline{\underline{\mathbf{P}}}_i^T \delta\underline{\mathbf{b}}_g \tag{5.24}$$

ii) The mode-matching is based on the Mahalanobis distance $\chi_i$ between the new projection $\underline{\mathbf{b}}_i$ and the cluster $\mathcal{M}_i$. The Mahalanobis distance $\chi_i$ is calculated as

$$\chi_i(\mathcal{M}_i, \underline{\mathbf{b}}_i) = \underline{\mathbf{b}}_i^T \Sigma_i^{-1} \underline{\mathbf{b}}_i \tag{5.25}$$

where $\Sigma_i$ is the covariance matrix of the i$^{th}$ cluster.

The candidate cluster $\mathcal{M}_c$ is defined as the most probable match of all possibly matching clusters $\mathcal{M}_i$. The conditional probability that the observation $\underline{\mathbf{b}}_i$ belongs to the cluster $\mathcal{M}_i$ is defined as

$$p(\mathcal{M}_i, \underline{\mathbf{b}}_i) = \frac{1}{\sqrt{(2\pi)^{m_i}}} e^{\frac{\chi_i(\mathcal{M}_i, \underline{\mathbf{b}}_i)}{2}} \tag{5.26}$$

Thus the most probable cluster $\mathcal{M}_c$ is defined as

$$c = \arg\max_i \{p(\mathcal{M}_i, \underline{\mathbf{b}}_i)\} \tag{5.27}$$

The candidate cluster $\mathcal{M}_c$ is defined by its $p$ x $m_c$ rotation matrix $\underline{\underline{\mathbf{P}}}_c$, the vector of $m_c$ eigenvalues $\underline{\mathbf{\Lambda}}_c$, the $m_c$-dimensional mean vector $\underline{\mu}_c$, the weighting coefficient $\omega_c$, and the dimensionality $m_c$:

$$\mathcal{M}_c = \left\{\underline{\underline{\mathbf{P}}}_c, \underline{\mathbf{\Lambda}}_c, \underline{\mu}_c, \omega_c, m_c\right\} \tag{5.28}$$

iii) Although the most probable, the candidate cluster $\mathcal{M}_c$ may not actually be matched to the observation. The new observation's projection may in fact lie too far away from the candidate cluster, i.e. the conditional probability is too small.

$$R = cdf^{-1}(0.95|m_c) \tag{5.29}$$

It is assumed that the space of the candidate cluster is limited by the boundary of the $m_c$-dimensional hyper-sphere which contains a large proportion, let us say 95%, of the points belonging to the cluster. The

radius $R$ of such a boundary is calculated as the inverse cumulative distribution function ($cdf$) of $\chi^2$-distribution for a given probability of 0.95 and $m_c$ degrees of freedom, Equation refeq:R.

The candidate cluster $\mathcal{M}_c$ is matched to the new observation only if the new observation projection falls within the radius of the cluster's bounding hyper-sphere. In other words, the Mahalanobis distance between the candidate cluster and the new observation projection is no larger than the cluster's bounding radius.

$$\chi_c\left(\mathcal{M}_c, \underline{\mathbf{b}}_c\right) \leq R \tag{5.30}$$

If the condition is satisfied the candidate cluster $\mathcal{M}_c$ is a match. Otherwise, a match is not found for the new observation.

The mode-matching test is performed on all $K$ clusters. If a mode-match exists this indicates that the projection of the new observation belongs to the mode-matched cluster within the reduced dimensionality subspace of only $m_c$ dimensions. It should be noted that outside this $m_c$-dimensional local subspace the new observation may contain significant components in other dimensions of the global space. Therefore, the classification of the new observation is not straightforward and additional tests are needed. The classification strategy is described in the following Section 5.3.6.

### 5.3.6 Classification of observations

The main aim of the adaptive multi-modal model is to provide means to classify the new image observation as background or foreground. The previous mode-matching test provided some information about the new observation, namely its position relative to the existing clusters within the reduced subspace of each. However, outside these low-dimensional subspaces the new observation may contain significant components in other dimensions of the global eigen-space. Therefore, additional tests are needed to complete the classification. The choice of the appropriate classification strategy depends on whether a mode-match has been found for the new observation, or not.

#### Classification of unmatched observations

If a matching cluster cannot be found this means that the new observation is not recognised by the existing model. In other words, the new image does not contain any of the previously observed background conditions known to the model. The unmatched observation may either represent a new type of background or an image contaminated with foreground objects. (This situation corresponds to the semantic cases 2 and 3 described in Section 5.2.5.) In this case, the classification decision is based on the error of modelling calculated in the full dimensional image space.

The error of modelling $\underline{e}$ in the image space is calculated as the difference between the original observation $\underline{\delta \mathbf{x}}$ in the image space and its recreated version obtained from its eigen-space representation.

$$\underline{e} = \underline{\delta \mathbf{x}} - \underline{\underline{\mathbf{P}}}_g \underline{\mathbf{b}}_g \tag{5.31}$$

$$e_{rms} = \sqrt{\frac{\underline{e} \cdot \underline{e}}{p}} \tag{5.32}$$

The *rms* error, $e_{rms}$, is calculated and compared to the threshold $\varepsilon$ determined by the noise level in the image space. If the *rms* error of modelling is lower than the threshold $\varepsilon$ the observation is classified as background. Otherwise, the observation is classified as contaminated with foreground objects. Thus, if $\beta$ is the classification label, then

$$\beta = \begin{cases} \text{background} ; & e_{rms} \leq \varepsilon \\ \text{foreground} ; & \text{else} \end{cases} \tag{5.33}$$

### Classification of mode-matched observations

An observation was mode-matched to a cluster if its projection falls within the 95%-boundary of the reduced $m_c$-dimensional local eigen-subspace of the matched cluster. In this low-dimensional subspace the observation belongs to the cluster. However, in reality the same observation may also occupy other dimensions of the global eigen-space. Viewed in the global space the same observation may be in effect far away from the boundaries of the matched cluster. The magnitude of the error between an observation and its projection within the low-dimensional cluster subspace reveals whether the observation contains additional significant components in other dimensions.

Let us represent this projection error by a $m$-dimensional residue vector $\underline{\mathbf{h}}$. The residue vector $\underline{\mathbf{h}}$ is calculated as the difference between the new observation in the $m$-dimensional global eigen-space $\underline{\mathbf{b}}_g$ and its recreated

version $\underline{\mathbf{b}}'_g$ obtained from the matched cluster's low-dimensional subspace of $m_c$ dimensions, where $m_c \leq m$.

$$\underline{\mathbf{h}} = \underline{\mathbf{b}}_g - \underline{\mathbf{b}}'_g \tag{5.34}$$

$$\underline{\mathbf{b}}'_g = \underline{\mu}_c + \underline{\underline{\mathbf{P}}}_c \underline{\mathbf{b}}_c \tag{5.35}$$

The residue vector $\underline{\mathbf{h}}$ is of the same dimensionality $m$ as the global eigenspace. The component of $\underline{\mathbf{h}}$ that corresponds to $m_c$ significant dimensions of the matched cluster's subspace is negligible. Thus, the component of the residue vector in the dimensions other than cluster's is represented by the external modelling error $\Delta\underline{\mathbf{h}}$, as defined in Equation 5.15. The magnitude $\|\Delta\underline{\mathbf{h}}\|$ is tested against the preset threshold $\Phi$. Two outcomes of the test are possible:

i) $\|\Delta\underline{\mathbf{h}}\| \leq \Phi$

The external modelling error is below the threshold $\Phi$ and is considered to be small. Therefore, the observation is classified as background, $\beta$ = background. (This situation corresponds to the semantic case 1 described in Section 5.2.5.)

ii) $\|\Delta\underline{\mathbf{h}}\| > \Phi$

The external modelling error exceeds the threshold $\Phi$. At this stage, it is not obvious whether the observation is a new unknown background (semantic case 2) or contaminated with foreground (semantic case 3). Thus, an additional error test in the full dimensional image space is needed to classify the observation. The error vector in the image space

and the *rms* error are calculated using Equation 5.31 and 5.32. The classification decision is based on thresholding the *rms* error with a preset threshold $\varepsilon$. Thus, the classification label $\beta$ is

$$\beta = \begin{cases} \text{background} ; & e_{rms} \leq \varepsilon \\ \text{foreground} ; & \text{else} \end{cases} \tag{5.36}$$

The choice of the constant thresholds $\Phi$ and $\varepsilon$ used in the classification process is explained in Section 5.3.9. As a result of the classification, the observation is classified and labelled as a true background or contaminated with foreground objects. The information obtained about the new observation is used to update the model.

### 5.3.7 Model update

This section details the update methodology of the multi-modal eigen model, which was introduced in Section 5.2.4. First, the methodology for adding new dimensions to the updated subspaces is outlined. Then, the update of the mode-matched cluster's mean and the eigen-subspace is explained. Finally, the update of the weighting coefficients of the modes is described.

#### Adding a new dimension to the subspace

The novel approach to incremental update and adding new dimensions was described earlier in Section 5.2.4. Here, it is applied to update of the mode-matched cluster.

The residue vector $\underline{\mathbf{h}}$ represents the error of modelling within the low-

dimensional eigen-subspace of the mode-matched cluster. It is calculated as the difference between the observation projection in the global eigen-space $\underline{\mathbf{b}}_g$ and its version recreated from the eigen-subspace of the mode-matched cluster $\underline{\mathbf{b}}_g^{'}$, as defined in Equations 5.34 and 5.35. The residue vector $\underline{\mathbf{h}}$ is of the same dimensionality $m$ as the global eigen-space. The components of $\underline{\mathbf{h}}$ that correspond to the few significant dimensions of the mode-matched subspace are negligible. The component of the residue vector in the dimensions other than clusters is represented by $(m - m_c)$ dimensional external modelling error vector $\Delta\underline{\mathbf{h}}$.

The magnitude $\|\Delta\underline{\mathbf{h}}\|$ determines whether the eigen-subspace of the mode-matched cluster requires a new dimension added during the update step. If $\|\Delta\underline{\mathbf{h}}\|$ is below a preset threshold $\Phi$ a new dimension is not required. If $\|\Delta\underline{\mathbf{h}}\|$ exceeds the threshold, the dimensionality $m_c^{'}$ of the updated eigen-subspace of the matched cluster increases to $m_c + 1$.

$$m_c^{'} = \begin{cases} m_c + 1\,; & \|\Delta\underline{\mathbf{h}}\| > \Phi \\ m_c \quad\; ; & \text{else} \end{cases} \tag{5.37}$$

$$\widehat{\underline{\mathbf{h}}} = \begin{cases} \dfrac{\underline{\mathbf{h}}}{\|\underline{\mathbf{h}}\|}\,; & \|\Delta\underline{\mathbf{h}}\| > \Phi \\ \\ \underline{\mathbf{0}} \;\; ; & \text{else} \end{cases} \tag{5.38}$$

The unit residue vector $\widehat{\underline{\mathbf{h}}}$, defined in Equation 5.38, is used as a new orthogonal vector added to the set of orthogonal vectors spanning the subspace of the mode-matched cluster. It corresponds to the *new dimension* which is added to the space if required. When the external modelling error of the

$m_c$-dimensional mode-matched cluster is below the threshold $\Phi$, the new orthogonal vector $\widehat{\underline{\mathbf{h}}}$ is reduced to $\underline{\mathbf{0}}$ and therefore a new dimension is **not** added to the updated space.

### Updating the mean of the mode-matched cluster

Based on the new background observation only the mean of the mode-matched cluster $\mathcal{M}_c$ is updated. The means of unmatched clusters remain unchanged. The updated mean $\underline{\mu}_c^{'}$ is calculated as

$$\underline{\mu}_c^{'} = \alpha\, \underline{\mu}_c + (1-\alpha)\, \underline{\mathbf{b}}_g \tag{5.39}$$

where $\alpha$ is a weighting function, defined by Equation 5.58, which controls the way the current model accommodates the new observation.

### Updating the local eigen-subspace of the mode-matched cluster

The local eigen-subspace of the mode-matched cluster $\mathcal{M}_c$ is updated by incorporating the new observation which was mode-matched to it. This section details the update steps 8 and 9, which were outlined in Section 5.3.2 and shown in Figure 5.2.

The updated subspace rotation matrix $\underline{\underline{\mathbf{P}}}_c^{'}$ can be derived from the eigen decomposition of the updated subspace covariance matrix $\underline{\underline{\mathbf{S}}}_c^{'}$.

$$\underline{\underline{\mathbf{S}}}_c^{'}\, \underline{\underline{\mathbf{P}}}_c^{'} = \underline{\underline{\mathbf{P}}}_c^{'}\, \underline{\mathbf{\Lambda}}_c^{'} \tag{5.40}$$

The new covariance matrix is defined as

$$\underline{\underline{\mathbf{S}}}_c^{'} = \alpha \, \underline{\underline{\mathbf{S}}}_c + (1 - \alpha) \, \delta\underline{\mathbf{b}}_g \delta\underline{\mathbf{b}}_g^T \tag{5.41}$$

where $\underline{\underline{\mathbf{S}}}_c$ is the old covariance of the matched cluster and $\delta\underline{\mathbf{b}}_g$ is the new observation difference vector in the global eigen-space.

The new subspace rotation matrix $\underline{\underline{\mathbf{P}}}_c^{'}$ is updated by inclusion of the residue unit vector $\widehat{\underline{\mathbf{h}}}$, defined in Equation 5.38, as follows

$$\underline{\underline{\mathbf{P}}}_c^{'} = [\underline{\underline{\mathbf{P}}}_c, \widehat{\underline{\mathbf{h}}}] \, \underline{\underline{\mathbf{R}}}_c \tag{5.42}$$

Substituting Equations 5.41 and 5.42 in 5.40 yields another eigen decomposition

$$\underline{\underline{\mathbf{D}}}_c \, \underline{\underline{\mathbf{R}}}_c = \underline{\underline{\mathbf{R}}}_c \, \underline{\mathbf{\Lambda}}_c^{'} \tag{5.43}$$

where

$$\underline{\underline{\mathbf{D}}}_c = \alpha \begin{bmatrix} \underline{\underline{\mathbf{S}}}_c & \underline{\mathbf{0}} \\ \underline{\mathbf{0}}^T & 0 \end{bmatrix} + (1 - \alpha) \begin{bmatrix} \underline{\mathbf{b}}_c \underline{\mathbf{b}}_c^T & \underline{\gamma} \, \underline{\mathbf{b}}_c \\ \underline{\gamma} \, \underline{\mathbf{b}}_c^T & \underline{\gamma}^2 \end{bmatrix} \tag{5.44}$$

and $\underline{\gamma} = \widehat{\underline{\mathbf{h}}}^T \delta\underline{\mathbf{b}}_g$.

The solution to the update problem is therefore found. The set of updated subspace eigenvalues $\underline{\mathbf{\Lambda}}_c^{'}$ is directly calculated as eigenvalues of $\underline{\underline{\mathbf{D}}}_c$. Matrix $\underline{\underline{\mathbf{R}}}_c$ is then computed as a set of eigenvectors of $\underline{\underline{\mathbf{D}}}_c$ and by substitution in Equation 5.42 the updated subspace rotation matrix $\underline{\underline{\mathbf{P}}}_c^{'}$ is obtained.

It is assumed that the updated space contains one additional dimension.

If at a later stage the newly added eigenvalue is proved to be too small it can be discarded together with the corresponding eigenvector. All eigenvalues equal to zero are discarded straight away. Furthermore, when the error represented by vector $\Delta\underline{\mathbf{h}}$ is small, as shown in Equation 5.38, the updated space does not require new dimensions. In this case, the new orthogonal vector $\underline{\widehat{\mathbf{h}}}$, and therefore the vector $\underline{\gamma}$, both become zero. By substituting $\underline{\gamma} = \underline{0}$ in Equation 5.44 the initially added new dimension is eliminated.

### Weighting coefficients update

The weighting coefficients capture the perceived relevance of clusters. They are used to identify the least relevant cluster, which will be replaced by a newly created cluster during the update process if needed.

At each time instance the weights of all existing clusters are updated. The update of the weight of a cluster depends on whether the cluster was matched to the new observation or not. The weight for a matched cluster increases ($q_i = 1$) while weights of all other currently existing clusters decrease ($q_i = 0$). The updated weight $\omega_i^{'}$ of the i$^{th}$ cluster is calculated as

$$\omega_i^{'} = \omega_i + \frac{1}{\min(t_i, T) + 1}(q_i - \omega_i) \tag{5.45}$$

where $i = 1, ..., K$.

## 5.3.8 Creating new cluster

Previously, Section 5.3.7 described the methodology for updating the multimodal model with a mode-matched new observation. This section explains

how an unmatched new observation is used for the multi-modal model update. This corresponds to the update step 10 outlined in Section 5.3.2 and shown in figure 5.2.

A new mode is created to model a new unknown background. The mode is represented by a new cluster - its mean, its eigen-subspace and its weight. The new cluster is initialised with the new unmatched observation point as its the mean, with a large initial variance and low initial weighting factor.

The new mode $\mathcal{M}^*$ is defined by its rotation matrix $\underline{\underline{\mathbf{P}}}^*$, set of eigenvalues $\underline{\mathbf{\Lambda}}^*$, mean $\underline{\mu}^*$, the weighting coefficient $\omega^*$, and the dimensionality $m^*$.

$$\mathcal{M}^* = \left\{\underline{\underline{\mathbf{P}}}^*, \underline{\mathbf{\Lambda}}^*, \underline{\mu}^*, \omega^*, m^*\right\} \tag{5.46}$$

where the elements of the new mode structure are initialised as follows.

### New mode dimensionality

Assuming that the initial model $\mathcal{M}^{(0)}$, defined by Equation 5.18, provides a realistic notion of the data variability, the dimensionality of the new mode is estimated from the dimensionality of the modes of the initial model. The dimensionality of the new mode, $m^*$, is set to the average dimensionality of the initial set of clusters $\mathcal{M}^{(0)}$.

$$m^* = \frac{1}{K}\sum_i m_i^{(0)} \tag{5.47}$$

where $m_i^{(0)}$ is the dimensionality of the i$^{th}$ initial cluster.

### New mode rotation matrix

The rotation matrix $\underline{\underline{\mathbf{P}}}^*$ is a $p$ x $m^*$ matrix, with orthogonal vectors spanning the new mode space as its column vectors. The top $m^*$ x $m^*$ submatrix is an identity matrix.

$$\underline{\underline{\mathbf{P}}}^* = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

### New mode variance

The variance of the new mode is set initially as high to ensure that following new observations with the same type of background are captured by the new cluster. This allows the new cluster to persist for long enough to grow beyond the initial stage. Assuming that the initial model $\mathcal{M}^{(0)}$ obtained during the training period provides a good representation of the variability of the data, the variance of the new mode is estimated from the variance of the clusters of the initial model.

The initial total variance of the new mode $V^*$ is estimated from the total variance of the initial set of clusters $\mathcal{M}^{(0)}$. It is calculated as a multiple, $a$, of the maximum total variance of the initial clusters, where this total variance

of the i$^{th}$ cluster is defined as

$$V_i^{(0)} = \sum_{k=1}^{m_i^{(0)}} \lambda_{i_k} \tag{5.48}$$

where $\lambda_{i_k}$ is the k$^{th}$ ordered eigenvalue of the i$^{th}$ initial cluster and $m_i^{(0)}$ is its dimensionality.

The initial variance of the new mode is then defined as

$$V^* = a\, V_j^{(0)} \tag{5.49}$$

where

$$j = \arg\max_i \{V_i^{(0)}\} \tag{5.50}$$

and $a$ is a multiplication factor (the choice of the value of $a$ is explained later in Section 5.3.10).

The new eigenvalues are derived using the *broken stick* function of $m^*$ segments, where the k$^{th}$ ordered eigenvalue $\lambda_k$ of the new mode is given as

$$\lambda_k = V^* \frac{1}{m^*} \sum_{j=k}^{m^*} \frac{1}{j} \tag{5.51}$$

where $k = 1, ..., m^*$.

The covariance matrix of the new mode $\underline{\underline{\mathbf{\Lambda}}}^*$ is a square diagonal $m^*$ x $m^*$ matrix with eigenvalues $\lambda_k$ on its diagonal.

$$\underline{\underline{\Lambda}}^* = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \lambda_{m^*} \end{bmatrix}$$

### New mode mean

The new mode mean in the global eigen-space is equal to the new observation point projection.

$$\underline{\mu}^* = \underline{\mathbf{b}}_g \tag{5.52}$$

### New mode weight and age

The new mode weight is set to a low value relative to the existing well established modes. The low value is chosen as the smallest weighting coefficient in the initial set of clusters.

$$\omega^* = \omega_j^{(0)} \tag{5.53}$$

where

$$j = \arg\min_i \{\omega_i^{(0)}\} \tag{5.54}$$

The age of the new cluster is initialised, $t^* = 1$.

### Removing outdated cluster

Whenever a new cluster $\mathcal{M}^*$ is created it replaces one of the existing clusters. The replaced cluster is the least relevant of all existing clusters at the time of creating the new cluster. The method for identification of the least

relevant cluster is inspired by the *fitness value* used to sort distributions in the Gaussian mixture model [Stauffer and Grimson, 1999]. (In this method the number of clusters is fixed. It could be argued that by removing the old clusters some information about the history of the data is lost. However, over a longer period of time, especially when large variations of lighting changes are present in the scene, the number of clusters may continue grow. In this large colection of clusters many of these will become redundant.) In the mixture model the fitness value is calculated for each distribution as the ratio between the current distribution weight and variance. A similar principle is applied to the multi-modal model, where the fitness value $\rho_i$ of the $i^{th}$ mode is defined as the ratio of the cluster's weight and its largest eigenvalue:

$$\rho_i = \omega_i / \lambda_i^{max} \tag{5.55}$$

Clusters are sorted by factor $\rho_i$ and the cluster with the lowest sorting factor $\mathcal{M}_j$ is replaced by the new cluster $\mathcal{M}^*$ as follows

$$\mathcal{M}_j = \mathcal{M}^* \tag{5.56}$$

where

$$j = \arg\min_i \{\rho_i\} \tag{5.57}$$

as a result, the outdated background condition modes are replaced by new more relevant modes enabling the model to evolve in time.

### 5.3.9 Definition of algorithm constants and variables

This section describes a number of parameters, constants and variables, used in the proposed algorithm. These are: the weighting function $\alpha$, the *age* of the cluster $t_i$, the update constant $T$, the probability of mode-matching $q_i$, the threshold constants $\Phi$ and $\varepsilon$, and the variance multiplication constant $a$.

- The update weighting function $\alpha$ controls the way the model adapts to the changes in the data. It determines how fast the model learns about the new lighting conditions in the scene and accommodates these changes. The parameter $\alpha$ is defined as

$$\alpha = \frac{1}{1 + \frac{1}{\min(t_i, T)}} \tag{5.58}$$

  Initially, the value of $\alpha$ is $\frac{1}{1 + \frac{1}{t_i}}$, which weights the new data relatively highly, and declines as new observations are added. To prevent this becoming infinitely small the weight is limited to $\frac{1}{1 + \frac{1}{T}}$ after $T$ observations. This follows the methodology used in [Stauffer and Grimson, 1999].

- The variable $t_i$ is the *age* of the i$^{th}$ cluster calculated as the time passed since the cluster was created to the present time. For the initial set of clusters each cluster is assigned the same age of $t_i = T + 1$.

- The constant $T$ is a time window of a fixed number of recent observations. During the initial period ($t \leq T$) after creation of a cluster

the algorithm gradually decreases the influence of incoming observations. The value of $T$ is estimated empirically for the chosen dataset in Section 5.3.10.

- The parameter $q_i$ is defined as the probability of mode-matching the new observations to the i$^{th}$ cluster. It can take values of

$$q_i = \begin{cases} 1, & \text{matched cluster} \\ 0, & \text{else} \end{cases} \tag{5.59}$$

- The threshold constant $\Phi$ is calculated as the inverse cumulative distribution function ($cdf$) of an $\chi^2$-distribution for a given probability of 0.95 with $(m - m_c)$ degrees of freedom, i.e.

$$\Phi = cdf^{-1}\left(0.95|\left(m - m_c\right)\right) \tag{5.60}$$

- The threshold constant $\varepsilon$ is estimated as the intrinsic noise of the dataset. It is calculated as the average difference between successive images in the training dataset.

- The multiplication constant $a$ determines the size of the initial variance of the newly created cluster, as shown in Equation 5.49. It is estimated empirically for the chosen dataset in Section 5.3.10.

### 5.3.10 Choosing the algorithm parameters

In this section, the values of the update constant $T$ and the new variance constant $a$ are determined for a dataset described in Section 5.4.1.

### Choosing the update constant T

Parameter $T$ is a time window of a fixed number of recent observations. It determines the relative influence of new observations on the updating of the model. During the initial period $t_i \leq T$ ($t_i$ being the age of the cluster) after creation of a cluster the model update gives priority to the most recent data to enable the new cluster to develop quickly. After this initial period the history of $T$ last observations has most influence over newly acquired observations.

The parameter $T$ controls the way new observations are incorporated into the model during the update stage. Weighting function $\alpha$, Equation 5.58, directly depends on the choice of $T$. Function $\alpha$ is associated with the current model, and $(1 - \alpha)$ with the new observation. Depending on the time when the update takes place, relative to $T$, the new observation will be given more or less weight compared to the accumulated knowledge about the data contained in the current model.

Figure 5.3 illustrates the effect of the $T$ parameter on the evolution of the model. A set of true background observations is analysed with a batch PCA and clustered into $K$ clusters in the eigen-space. Knowing the distribution of points in clusters, the adaptive algorithm is applied to the same data for different values of $T$. The graph shows the percentage of points that were not mode-matched to any of the clusters for different values of parameter $T$ when the adaptive cluster mode-matching was used. For small $T$ the model most strongly weights new observations over the history of the data. Newly created clusters are often quickly discarded as the least relevant having

not had enough time to grow. As $T$ increases the percentage of unassigned points drops. For $T = 100$ there is only 0.5% points left unmatched to any of the clusters. Further increase of $T$ does not contribute significantly to the reduction of the number of unmatched points.

At this point it may be assumed that the extrapolated graph, Figure 5.3, levels out or even continues to descend slowly. However, the same experiment showed that for large $T$ an increasing number of mode-matched tested images was actually assigned to wrong clusters. In other words, knowing in advance the distribution of points in clusters, the number of points not assigned to the expected cluster increased for values larger than $T = 100$. For large values of T the model will most strongly weight the history of the data causing a very slow evolution of the model. In this case, the changes in the data are not taken into account appropriately. The new point are incorporated into the older well established clusters instead of creating new clusters. This will eventually produce a very slow, inadaptable, inert model with outdated clusters, unable to accommodate observed changes in the background data. Therefore, the value $T = 100$ is adopted.

### Choosing the new cluster variance constant $a$

Assuming that the training dataset has provided a realistic notion of the data variability, the initial variance of the new cluster is calculated as a multiple, $a$, of the maximum cluster variance obtained during the training period, as shown in Equation 5.51.

Figure 5.4 illustrates how the accuracy of classification of new observations depends on the choice of the initial variance of the new cluster.

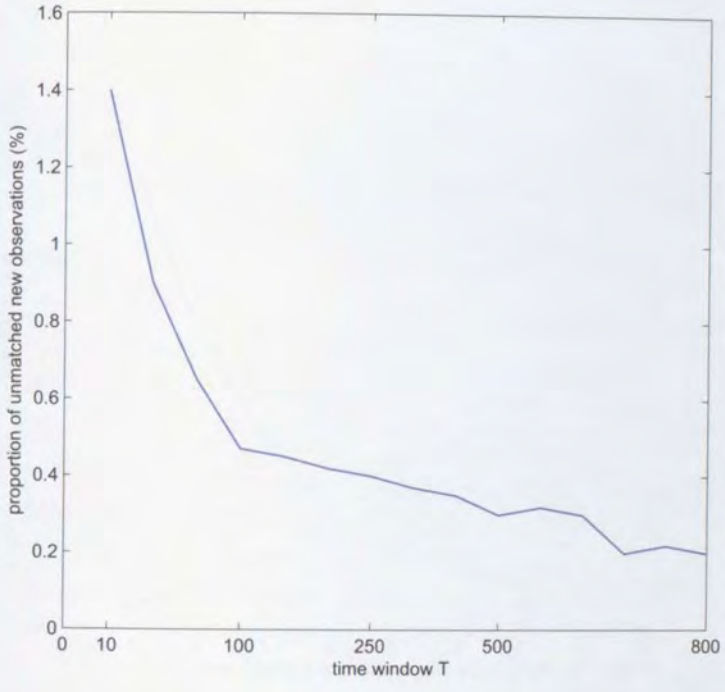



Figure 5.3: Choice of parameter $T$

The vertical axis represents the false alarm rate, where the false alarms are background-only images wrongly classified as contaminated with foreground. The horizontal axis represents the multiplication parameter $a$, which varies from 1 to 10.

It can be seen that relatively low initial variance results in high false alarm rate. When a new cluster is initially too small in volume (in multi-dimensional space), there is an increased risk that subsequent observations of similar background will not be captured by the new cluster. Instead, they may be wrongly associated with some of other existing clusters or a matching cluster will not be found. If the initial variance of the new cluster is set relatively high, it is more likely that the new cluster will capture more of the similar new observations. Further increase of the initial variance does

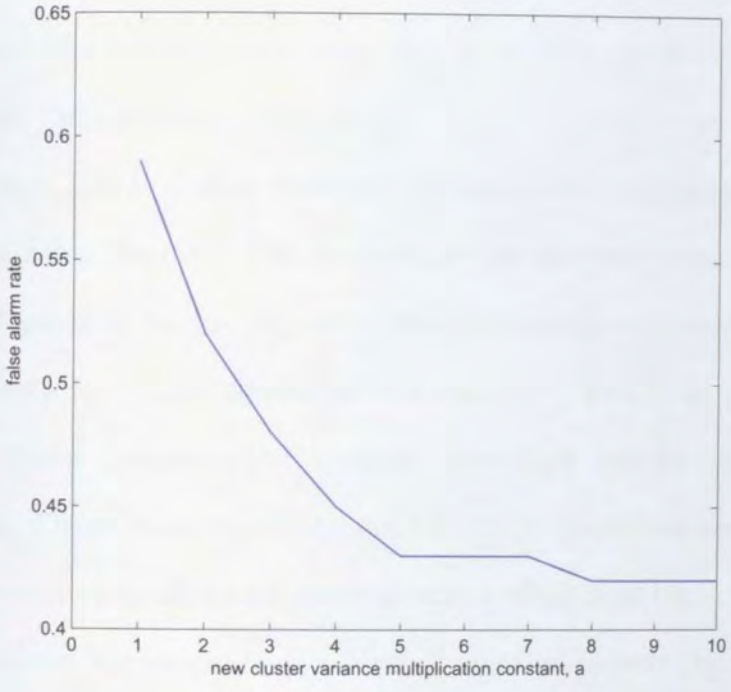



Figure 5.4: New cluster initial variance

not improve the performance of classification of background images.

The effect of reduced false alarms for large $a$ may be somewhat misleading when choosing the optimal value for $a$. The same experiment showed that, when the distribution of background points is known in advance, the final distribution of the points in the adaptive clusters was not as expected for large $a$. In a number of cases the tested points were assigned to wrong clusters. This occurred because very large clusters with growing initial variance captured more and more distant new observation. These large variances included images of very diverse types which normally cannot be modelled with the same cluster. It may be concluded that the increasing value of $a$ causes inaccurate clustering of background points. Therefore, the initial variance should be large enough to capture the images of the same type but not too

large to risk the accuracy of the model. The multiplication factor $a = 5$ was chosen to provide enough robustness for the newly created clusters without deteriorating the accuracy of the model.

The chosen initial cluster variance obtained for $a = 5$ does not, indeed, suppress all false alarms. This is because the mode-matching of the new observation point is performed in a low-dimensional eigen-subspace of the cluster of only few most significant dimensions. The false positives that persist are those images which contain significant sudden changes in the background. These background images are likely to contain significant components in many other dimensions of the space other than those of the cluster, which may cause the model to fail. Such images will always be wrongly classified as contaminated regardless the increase in the initial cluster variance; some examples are shown in the bottom row of Figure 5.5a.

## 5.4 Results

In this section, the proposed adaptive multi-modal algorithm is evaluated on a real video surveillance dataset. The remainder of the section is organised as follows. The dataset is presented in Section 5.4.1. The capability of the algorithm to detect the presence of the foreground of various sizes is tested and discussed in Section 5.4.2. Finally, the accuracy of the classification results are analysed and compared to those obtained by the uni-modal modelling approach in Section 5.4.3.

### 5.4.1 Dataset

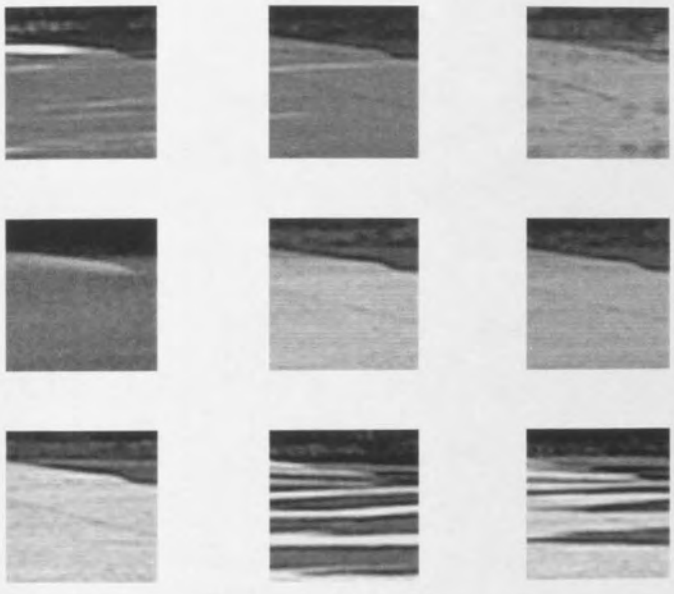
The adaptive multi-modal background modelling algorithm is evaluated on the dataset described in detail in Section 4.5.2. It is a video surveillance record of activities in a courtyard in front of an office building. It was recorded over a few weeks during the late spring and early summer, with long lasting daytime, at a frame rate of one frame per minute. Only daytime frames are used in the experiment.

The dataset is rich with lighting changes, alternating sunny, cloudy and rainy weather conditions, shadows from surrounding trees creating distinctive patterns and moving across the scene, and background motion from swaying trees. The analysis is performed on the grey-scale images which are divided into smaller regions in order to provide a better understanding of the variations in the dataset. The region in the position {6,5} in Figure 4.29 is taken as an example of a stationary background where the variability of the data is caused mainly by global lighting changes.

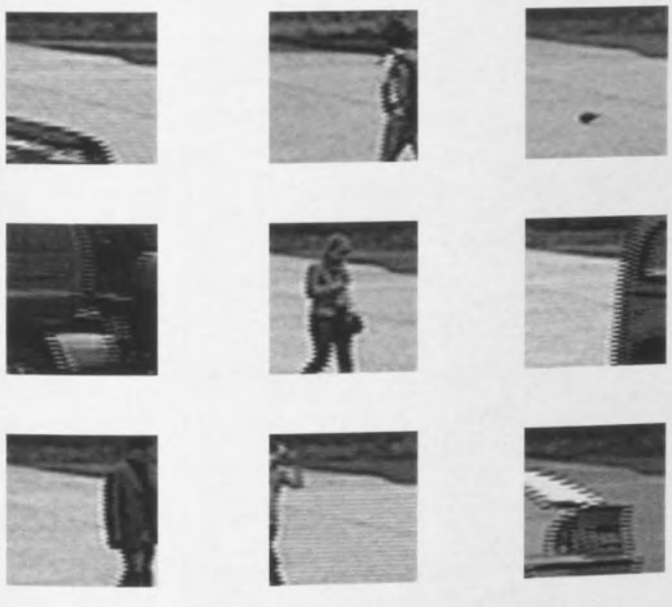
Foreground objects, moving people and vehicles, are sparse and typically appear in a single frame due to the low frame rate of recording. Some examples of background and contaminated observations are shown in Figure 5.5.

### 5.4.2 Detection of contaminated images

This section explores the ability of the model to detect and classify contaminated observations. An artificial contaminant in the form of a disc is introduced in the background scene. The contaminant increases in size obscuring between 1% and 80% of the observed image region as shown in Figure 5.6.



(a) Backgrounds



(b) Foregrounds

Figure 5.5: Dataset

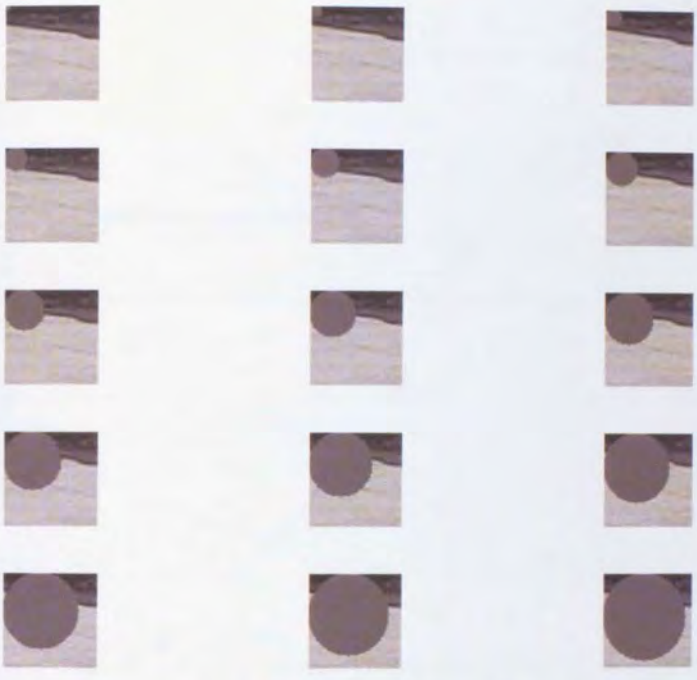


Figure 5.6: Contaminated observations

After the initialisation stage the algorithm processes 300 background observations incrementally adapting to the changing lighting conditions before a disc is inserted in the next new observation. The disc is of a uniform grey-scale equal to the average grey-level of the test image region. The same experiment is repeated for increasing disc size. A total of sixteen images with various contaminant size were processed and classified.

The final score of classification is thirteen contaminated images correctly classified as foreground - true positives (TP), and three wrongly classified as background-only - false negatives (FN). The ability of the algorithm to detect and correctly classify contaminated observations with different contamination sizes is now observed and discussed.

Figure 5.7 illustrates the final result of the classification. Each point on the graph corresponds to one of the classified contaminated observations;

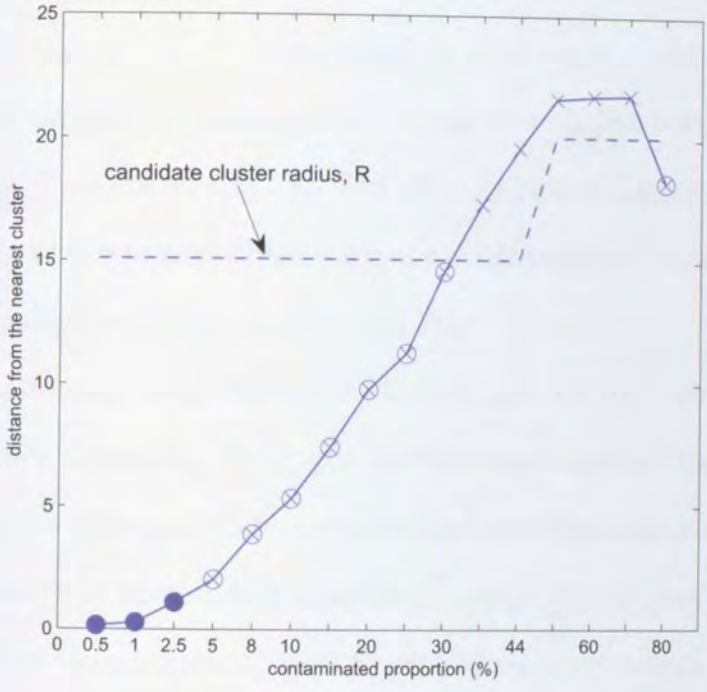



Figure 5.7: Distance from cluster

three false negatives (full circle) and thirteen true positives (circle-cross and cross). The increasing contaminant disc size, expressed as a percentage of the observed image region, is on the horizontal axis. The vertical axis is the distance between an observation and the mean of the candidate cluster $\mathcal{M}_c$, which was identified in the step 5 of the algorithm (see Figure 5.2). The solid line represents the distance between the observation projection and the candidate cluster $\mathcal{M}_c$. The dashed line represents the limiting radius (Equation 5.29) of the boundary of the candidate cluster's hyper-sphere; the radius depends on the number of dimensions of the cluster's eigen-subspace. Points marked by solid circles correspond to images with contaminants smaller than 5% of the image region; these contaminants remain undetected. These observations were mode-matched to a candidate cluster in step 5 and then wrongly

classified as background-only in step 6 of the algorithm. Points marked by a circle with a cross inside correspond to observations which were mode-matched to a candidate cluster in step 5, but nonetheless correctly classified as foreground in steps 6 and 7a. Finally, the points marked with a cross correspond to observations which were not mode-matched in step 5 and were correctly classified as foreground in step 7b.

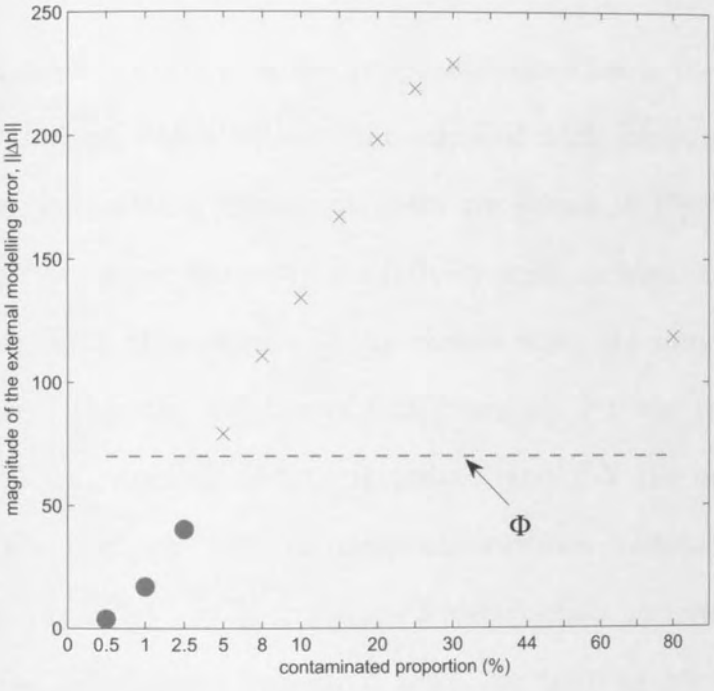
Figures 5.8a and 5.8b illustrate the outcome of the classification performed in steps 6 and 7a. Here, the contaminants smaller than 30% of the image region are expected to be captured (larger contaminants are normally detected already in step 5 and classified in step 7b). It can be seen from graphs that there are ten such observations. These observations were initially, in step 5, mode-matched to a candidate cluster in its low $m_c$-dimensional eigen-subspace. Seven of them (points marked with crosses) were then disqualified in the step 6 due to significant components being detected in other dimensions of the global eigen-space. Three observations (points marked with solid circles) with contaminants smaller than 5% of the image region, failed to be detected.

Figure 5.8a shows the outcome of the step 6. The horizontal axis is the size of the contaminant. The vertical axis shows the magnitude of the external modelling error, $\Delta\underline{\mathbf{h}}$, in $(m - m_c)$ dimensions outside the mode-matched cluster. The dashed line represents the threshold $\Phi$, which was defined in Equation 5.60. The points found below the threshold (solid circles) correspond to observations wrongly classified as background-only (FN). The points found above the threshold (crosses) correspond to points which are yet to be classified in step 7a.

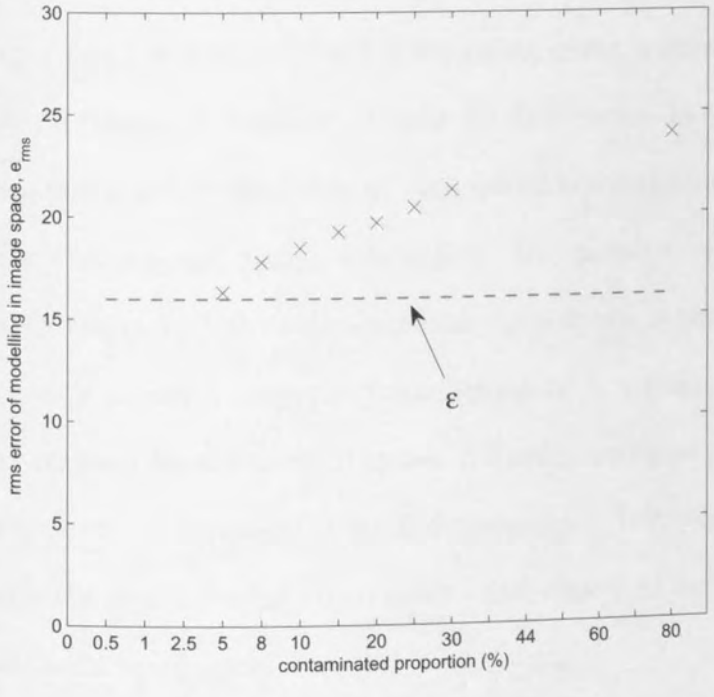
Figure 5.8b shows the outcome of the step 7a. The vertical axis represents the *rms* error of modelling in the image space, $e_{rms}$, which was defined in Equation 5.32. The dashed line corresponds to the noise threshold $\varepsilon$ defined in Section 5.3.9. Points above the threshold (crosses) are classified as foreground in step 7a. It can be seen that in this case all contaminated observations previously left unclassified in step 6, are now classified as foreground.

The experiment confirms that all observations with contaminants of size 5% or more of the image region are detected by the algorithm and correctly classified as contaminated. Hovever, it can be seen from the graphs that very large contaminants which occlude more than 80% of the image region may confuse the model. In this particular case, the artificial contaminant is of an uniform grey-level close to the average grey-level of the image region. The contaminated image was mode-matched to an existing cluster in step 5 and later classified as foreground in steps 6 and 7a. In real life situations, large objects are unlikely to be of a strictly uniform colour and are expected to be detected immediately in step 5 and classified as foreground in step 7b of the algorithm.

It was demonstrated earlier, in the case of the non-adaptive batch modelling in Chapter 4, that the off-line multi-modal approach was capable of detecting very small contaminants, less then 1% of the occluded image region, Figure 4.12. Compared to the contaminat size of 5%, the batch approach performs slightly better than the adaptive in detecting small contaminations. This is because at any time the batch model has more accumulated information about the dataset than the adaptive model. However, the batch approach is not suitable for online applications; the adaptive model is needed.




(a)




(b)

Figure 5.8: (a) External modelling error magnitude, (b) *rms* error in image space

### 5.4.3 Classification accuracy

The classification accuracy of the proposed algorithm is demonstrated on 2000 test frames of which 19 are contaminated with foreground objects of various sizes and colours (some examples are shown in Figure 5.5b). The dataset is rather sparse, therefore a relatively small number of contaminated frames is available. The results of the classification are summarised in Table 5.2 where $TP$ is the number of true positives, $FP$ the number of false positives, $TN$ the number of true negatives, and $FN$ the number of false negatives. The positives refer to image observations contaminated by foreground. The negatives are observations which contain background only.

The global eigen-space is derived from the training set using a batch PCA. The dimensionality of the eigen space is reduced to $m < p$ using the *broken stick* rule. The number of dimensions is then doubled to provide some robustness and ensures that retained dimensions contain enough variability of the original dataset. A number of $m = 50$ dimensions is obtained. This already represents a great reduction in dimensionality compared to $p = 4096$ dimensions in the original image space (i.e. the number of pixels in the observed image region). The $m$-dimensional eigen-space is clustered in $K = 10$ clusters which model background conditions of a similar nature. Each cluster forms its own local eigen-subspace of further reduced dimensionality $m_i \leq m$, where $m_i$ is typically 4 to 7 dimensions. Test observations are projected onto the multi-modal eigen-model and classified as background or contaminated with foreground objects.

To test the performance of the multi-modal model the results of the clas-

sification are compared with those obtained using the unimodal model. The results are summarised in Table 5.2. The unimodal model correctly detected all 19 contaminated images, while 28 background images were wrongly classified. The multi-modal model detected 13 contaminants, while failing to correctly classify only 9 true backgrounds. The 6 undetected contaminated observations contain very small objects of less than 5% of the image region.

| | unimodal | multi-modal |
|---|---|---|
| $TP$ | 19 | 13 |
| $FP$ | 28 | 9 |
| $TN$ | 1953 | 1972 |
| $FN$ | 0 | 6 |

Table 5.2: Classification accuracy

The results show that the unimodal approach performs better in the case of small contamination detection then the multi-modal model. This is caused by the greatly reduced dimensionality of cluster subspaces (only 4 to 7 dimensions in this example), where the loss of information about the data is significant. This caused reduced capacity of the multi-modal model to represent small variations in the data. At the same time, the multi-modal model achieved lower number of false alarms because of its improved ability to suppress the background noise.

## 5.5 Conclusion

Due to the high dimensionality, modelling of backgrounds in outdoor scenes is generally time consuming and computationally costly. Further difficulty is introduced by the nature of lighting changes in outdoor scenes due to weather conditions and background motion. By means of eigen analysis it is possible to define a smaller set of dimensions which will capture most of the variation of the original space. Furthermore, the background model needs to constantly adapt to accommodate those changes in time.

One way to achieve an accurate modelling of such scenes is to define a number of background modes which model background observations of similar lighting conditions. The modes are defined as multivariate Gaussian clusters in a reduced dimensionality eigen-space. The clustered eigen-space is updated with every new observation in order to adapt to new background conditions. At any time instance, the model is estimated from the new observation and the model at the previous time instance. The model is updated using only the knowledge of the current covariance matrix and the residue vector occurring as an error introduced by the outdated model.

The novelty of the proposed algorithm is threefold. First, the principle of the well known Gaussian mixture model is applied to image regions rather than pixels, where image observations are modelled with a mixture of multi-dimensional Gaussian clusters in the eigen-space. Second, a previous unimodal incremental method [Hall et al., 1998] is adapted to a multi-modal model consisting of clustered eigen-subspaces. Third, the method for adding new dimensions to eigen-subspaces is modified in order to avoid unnecessary

increases of space dimensionality. In this way, the proposed approach aims to reduce the dimensionality of the model as much as possible while the significant proportion of variability in the data is still preserved.

The proposed algorithm was used to classify a dataset of images with artificial foreground. It was demonstrated that the proposed algorithm is capable of detecting contaminants of size 5% or more of the observed image region. Smaller contaminants were not detected because the reduced model dimensionality was too low to model very small changes in the data. It is possible to sufficiently increase the dimensionality to capture these small variations too. Furthermore, the adaptive multi-modal model was tested on a real dataset with foreground objects of various types. The classification accuracy of the proposed multi-modal model was compared with the unimodal model. Although the capacity of the multi-modal model to detect very small objects was somewhat reduced, the false alarm rate was significantly improved.

It can be concluded that the proposed adaptive multi-modal approach provided a suitable structure for modelling of multi-modal multi-variate high-dimensional data. The low-dimensional model has an ability to successfully suppress the system noise and reveal underlying variability in the data. The proposed algorithm is suitable for modelling of the changing background of outdoor surveillance scenes, especially in foreground detection applications where a low false alarm rate is required.

### Discussion and future improvements

This section discusses performance of the proposed approach and suggests possible improvements. The proposed adaptive multi-modal algorithm aimed

to detect the presence of foreground objects in an outdoor video surveillance sequence and classify the image observations as background-only or contaminated with foreground.

The performance of the algorithm was presented in Table 5.2, which compared the unimodal and the multi-modal on-line methods. The number of foreground detections deteriorated in the case of the multi-modal model because the model failed to detect very small foreground objects and those of a similar grey-level as the background. Nevertheless, the multi-modal algorithm performed well. The number of false alarms significantly improved due to model's ability to successfully suppress background lighting changes.

The proposed algorithm was tested on a dataset, which was selected because of its very challenging features such as constant and significant lighting changes due to weather conditions. An additional difficulty was that the sequence was recorded with a low frame rate which contributed to great differences between the successive image observations. A number of persistent false alarms resulted from extreme variations in the background, notably irregular moving patterns across the scene caused by long shadows of swaying trees during very sunny days (see the bottom row of the Figure 5.5a). The shadows were visible for long periods of time from the mid-afternoon to the evening on most days, sliding across the scene in constant motion. Yet, the model misclassified only 9 out of 1981 processed background images, or less than 0.5%. Out of 19 processed contaminated images, a total of 6 was left undetected. Three of them contained a small bird (the top right picture in Figure 5.5b). The other three contained small parts of objects of a grey-level similar to the background (the picture in the middle of the bottom row in

Figure 5.5b), primarily objects entering or leaving the scene. Considering the challenging nature of the chosen dataset it can be concluded that the algorithm performed successfully.

The algorithm may benefit from improvements which will address the problems of the inaccurate classification. The problem of the moving shadow patterns may be tackled by Markov modelling of transitions between the clusters providing an additional information about the nature of the data. Furthermore, the misclassification is due mainly to the reduced dimensionality of the model. Therefore, a more principled method is required for the dimensionality reduction of the local eigen-subspaces of the individual modes. Such method should ensure that the dimensionality is low enough to eliminate the noise and the effect of illumination changes in clusters of similar background conditions. At the same time, the number of retained dimensions must be sufficient to exclude foreground objects which are small or blend into the background. Furthermore, the detection of parts of objects entering and leaving the scene may be improved by providing some knowledge about the scene. Also, feedback from an object tracker may be used to refine the detection process. Finally, for an objective evaluation, an additional validation of the algorithm is required based on a comparative performance evaluation methodology, as proposed in Chapter 3, using well known datasets recognised by the research community.

# Chapter 6

# Conclusions

## 6.1 Summary

This work focused on the performance evaluation of motion detection algorithms and the problem of background modelling for visual surveillance applications. The aim of the thesis was twofold. First, it aimed to provide an objective and unambiguous tool for evaluation of foreground detection algorithms. Second, it aimed to provide a low-dimensional method for background modelling in challenging outdoor video surveillance scenes with an abundance of sudden and gradual lighting variations due to changing weather conditions.

### 6.1.1 Performance evaluation

The thesis presented a novel approach to performance evaluation of motion detection taking into consideration the desired performance of the algorithm in the context of the end-user application. Previous works, outlined in Sec-

tion 2.2, mostly focused on the evaluation at the pixel-level which failed to address the impact of foreground detection on the performance of object-based applications such as object tracking. The pixel-based approaches had the advantage of using ROC technique for the interpretation of metrics. Unfortunately, ROC tool is not applicable to the object-based approach. In an attempt to overcome this problem, a number of previously reported object-based techniques produced a maze of evaluation metrics exposed to subjective interpretations.

In the light of the preceding research, this thesis proposed a novel objective object-based framework, which enabled a straight-forward mechanism for both optimising and comparing motion detection algorithms.

### 6.1.2 Background modelling

In addition to the performance evaluation, this thesis proposed a novel adaptive multi-modal algorithm for background modelling. The goal was to provide a tool for efficient and accurate classification of image observations as background-only or contaminated with foreground.

The proposed methodology exhibits three main features. First, it exploits the advantages of modelling high-dimensional video data in low-dimensional eigen-space using PCA. Second, it considers the nature of lighting changes characteristic for outdoor scenes, such as weather conditions and moving shadow patterns. As a result of this consideration, types of resembling lighting conditions are identified as modes of background observations and represented as multi-variate Gaussian distributions in the low-dimensional eigen-

space. Finally, an incremental adaptive update approach is adopted, which enables the multi-modal model to evolve and accommodate newly observed changes in the background.

## 6.2 Discussion

### 6.2.1 Performance evaluation

It was demonstrated that a well-designed evaluation methodology for comparing motion detection algorithms reveals surprisingly complex issues. Two problems in particular were carefully addressed. First, the inevitable existence of evaluation parameters and the need to select appropriate values for these. Second, the absence of true negatives makes it impossible to use the well-known ROC methodology. In addition to these issues, an objective comparative methodology is required - a methodology that allows the definition of standardised application scenarios which provide context to the comparison.

From these considerations, a new object-based comparative methodology based on the *F-Measure* has been developed. The proposed methodology provides a single-valued ROC-like measure enabling both optimisation and comparison of motion detection algorithms. It includes a number of configuration steps. In summary, these are defining the application scenarios; determining the appropriate weighting parameters ($\alpha$) for each scenario; defining a method of associating detected objects with the ground truth objects; selecting the optimal values of the evaluation parameters; optimising

the parameters of each competing algorithm for each scenario; and finally computing and comparing the performance of each algorithm.

### 6.2.2 Background modelling in low-dimensional space

A novel adaptive multi-modal algorithm for background modelling and classification has been proposed. Several issues have been explored including dimensionality reduction, the possibility of subsampling of image observations, multi-modality of observed backgrounds, and incremental update of the model.

It has been demonstrated that it is possible, by means of eigen analysis, to define a smaller set of dimensions which will capture most of the variation of the original high-dimensional dataset. Generally, it is not obvious how many retained dimensions provide a sufficient dimensionality reduction while ensuring that the variability in the data is preserved. The rules which determine the cut-off dimension are intuitive and there is no universal answer as to which rule gives the most suitable number of retained dimensions. The *broken stick* rule is often said to be the most adequate for real data. To test this we explored the concept of the hyper-sphere of backgrounds which separates background-only observations from contaminated observation points in the eigen-space. For a particular number of dimensions all background-only observations remain inside the hyper-sphere, while the contaminated points are found beyond its boundaries. By comparison with the hyper-sphere test it was shown that the *broken stick* rule is an appropriate choice for dimensionality reduction. The number of dimensions obtained by the *broken stick*

correspond to the number of dimensions for which the hyper-sphere of backgrounds provides a good separation of the background-only and contaminated observations.

The possibility and limitations of subsampling a relatively small proportion of all the available data have also been explored. It has been suggested that depending on the contaminant size, it is possible to avoid subsampling from contaminated areas by subsampling from predefined image subregions. This controlled subsampling would provide accurate unimodal eigen-space classification of image observations from a very small amount of subsampled data. However, it was shown that in the multi-modal eigen-space, due to higher sensitivity to image changes, there is an increased risk of misclassification of subsampled data. Therefore, with an additional information i.e. some knowledge of entry points in the scene and/or expected contaminant sizes, this method may quickly detect contaminations of the background from a relatively little information about the scene.

### 6.2.3 Adaptive multi-modal background modelling

As one of the main contributions of this work, a novel adaptive multi-modal algorithm for background modelling and observation classification was developed. There are several novel contributions of the proposed algorithm. First, the principle of the well known Gaussian mixture model is applied to image regions rather than pixels, where image observations are modelled with a mixture of multi-dimensional Gaussian clusters in the eigen-space. Second, a known unimodal incremental method is adopted and applied to

a multi-modal model consisting of clustered eigen-subspaces. Finally, the method for adding new dimensions to eigen-subspaces is modified in order to avoid unnecessary continual increase of space dimensionality. In this way, the proposed approach aims to reduce the dimensionality of the model as much as possible while the significant proportion of variability in the data is still preserved.

By means of suitable experimentation a set of optimal model update parameters has been determined. The optimised algorithm has been used to classify a sequence containing contaminants of variable sizes. It has been shown that the proposed algorithm is capable of detecting small contaminants of size less than 5% of the observed image region. In addition, the performance of the multi-modal on-line algorithm was tested on a real dataset with foreground objects of various types, and the results have been compared with those of the unimodal model. It has been demonstrated that the multi-modal approach showed an improvement; even though the detection was slightly reduced, the false alarm rate was significantly improved. The slight deterioration of the foreground detection and the remaining false alarms are mainly due to the very challenging dataset containing constant and significant lighting changes due to weather conditions. The foreground objects appear in the scene typically for a single frame because of the very low frame rate. Encouragingly, only very small objects and those of a grey-level similar to the background failed to be detected. A number of remaining false alarms result from extreme variations in the background, notably irregular moving patterns across the scene caused by long shadows of swaying trees during very sunny days.

The proposed adaptive multi-modal approach has proved appropriate for modelling of multi-modal multi-variate high-dimensional data. The low-dimensional model has been able to efficiently suppress the system noise and reveal underlying variability in the data. It has successfully detected the presence of foreground objects and classified the image observations in a very challenging outdoor video surveillance sequence. The proposed adaptive multi-modal algorithm is suitable for modelling of the background of outdoor surveillance scenes, especially in foreground detection applications where a low false alarm rate is required.

## 6.3 Future work

### 6.3.1 Performance evaluation

The thesis discussed the significance of evaluating motion detection within the wider context of a surveillance system. However, some of the current weaknesses remain to be addressed: explicit methodology for choosing appropriate values for the weighting parameter $\alpha$; standardisation of application scenarios; comparison with more methods reported in the literature; and a bigger range of datasets. Further investigation is needed to explore whether this methodology might be practical further down the evaluation pipe-line i.e. measuring the impact of an early visual processing stage on the results of subsequent stages.

### 6.3.2 Background modelling

It was demonstrated that the proposed adaptive multi-modal algorithm successfully detected the presence of foreground objects and classified the image observations under the challenging conditions of an outdoor video sequence. However, in order to address the problems with the remaining inaccurate classifications, a number of possible modifications are suggested.

#### Markov modelling of mode transitions

Markov modelling of transitions between the clusters may help eliminate the problem of the moving shadow patterns and reduce the number of false alarms. It is expected that the image observations containing shadow patterns will repeatedly follow the same transitions between clusters. The likely transitions between clusters can be learned over time. If unlikely transition is observed it is possible that the image observation contains foreground objects.

#### Choosing the number of PCs

The number of undetected foregrounds is a direct result of the loss of information due to the reduced dimensionality of the model. Specifically, the very small objects or those of a grey-level similar to that of the background may remain undetected. Therefore, a method more principled than the *broken-stick* rule is required for the dimensionality reduction of the local eigen-subspaces of the individual modes. Such a method should ensure that the dimensionality is low enough to eliminate the noise and avoid false alarms while at the

same time provide a sufficient number of dimensions to detect foreground objects which are small or blend into the background.

### Alternative subsampling strategies

The detection of partial objects entering and leaving the scene may be improved by providing some knowledge about the scene topography. For example, the entry and the exit points, or the likely paths of the moving objects. This information may be used for a more selective subsampling from regions that are likely to contain foreground objects. If such a set of subsampled pixels does not indicate the presence of foreground in the observed image it may be concluded that it is very likely that the scene contains only background pixels. In a similar manner, feedback from an object tracker may be used to refine the search area and support the detection process.

### Comparative evaluation of the algorithm

The performance of the proposed adaptive multi-modal algorithm was tested by comparison with the results obtained by a non-adaptive algorithm. However, a more principled test is necessary. For an objective conclusion about the performance of the algorithm and recommendations for its application an additional validation is required. The validation of results should be based on a comparative evaluation methodology, as proposed in Chapter 3, using well known datasets recognised by the research community.

## 6.4 Publications

N. Lazarevic, J. Renno, D. Makris, G.A. Jones. An Object-based Comparative Methodology for Motion Detection based on the F-measure. *Journal of Computer Vision and Image Understanding, Special Issue on Intelligent Visual Surveillance*, 111:74–85, 2008

N. Lazarevic, J. Renno, and G.A. Jones. Performance evaluation in visual surveillance using the f-measure. In *Proceedings of the Fourth ACM International Workshop on Video Surveillance and Sensor Networks*, VSSN '06, pages 45-52, 2006.

N. Lazarevic, J. Renno, D. Makris, G.A. Jones. Designing evaluation methodologies: the case of motion detection. In *Proceedings of the Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, PETS'06, New York, U.S.A., pages 23–30, June 2006.

J. Renno, N. Lazarevic, D. Makris, G.A. Jones. Evaluating motion detection algorithms: Issues and results. In *Proceedings of IEEE International Workshop on Visual Surveillance*, VS'06, pages 97-104, Graz, Austria, May 2006.

# Bibliography

J. Aguilera, H. Wildernauer, M. Kampel, M. Borg, D. Thirde, and J. Ferryman. Evaluation of motion segmentation quality for aircraft activity surveillance. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 293–300, Beijing, October 2005.

M. Artac, M. Jogan, and A. Leonardis. Incremental pca or on-line visual learning and recognition. In *Proceedings of the 16 th International Conference on Pattern Recognition*, ICPR '02, pages 781–784, 2002.

F. Baf, T. Bouwmans, and B. Vachon. Type-2 fuzzy mixture of gaussians model: Application to background modeling. In *International Symposium on Visual Computing*, ISVC 2008, pages 772–781, Las Vegas, USA, December 2008.

Y. Benezeth, P.M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Review and evaluation of commonly-implemented background subtraction algorithms. In *19th International Conference on Pattern Recognition*, pages 1–4, Tampa, FL, Dec 2008.

J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. In *IEEE Workshop on Performance Analysis of Video Surveillance and Tracking*, PETS2003, pages 125–132, October 2003.

T. Bouwmans. Subspace learning for background modeling: A survey. *Recent Patents On Computer Science*, 2(3):223–234, 2009.

T. Bouwmans, F. El Baf, and B. Vachon. Background modeling using mixture of gaussians for foreground detection - a survey. *Recent Patents On Computer Science*, 1(3):219–237, 2008.

K.M. Branson and S. Agarwal. Structured principal component analysis. Technical report, Department of Computer Science and Engineering, University of California, San Diego, 2003.

P.J. Burt, J.R. Bergen, R. Hingorani, R. Kolczynski, W.A. Lee, A. Leung, J. Lubin, and H. Shvayster. Object tracking with a moving camera. In *Proceedings of the Workshop on Visual Motion*, pages 2–12, Irvine, CA, USA, March 1989.

Q. Cai and J.K. Aggarwal. Tracking human motion using multiple cameras. In *Proceedings of the 13th International Conference on Pattern recognition*, pages 68–72, 1996.

A. Cavallaro, E. Gelasce, and T. Ebrahimi. Objective evaluation of segmentation quality using spatio-temporal context. In *IEEE International Conference on Image Processing*, pages 301–304, September 2002.

S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and

H. Zhang. An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing*, 59:321–332, November 1997.

R. Chang, T. Gandhi, and M.M. Trivedi. Vision modules for a multi-sensory bridge monitoring approach. In *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, pages 971–976, October 2004.

S. Chaudhuri, S. Sharma, and S. Chatterjee. Recursive estimation of motion parameters. *Computer Vision and Image Understanding*, 64, 1996.

S.S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. *Visual Communications and Image Processing*, 5308 (1):881–892, 2004.

C. W. Cleverdon. On the inverse relationship of recall and precision. *Journal of Documentation*, 28(3):195–201, 1972.

T. F. Cootes, C. J. Taylor, D.H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proceedings of the British Machine Vision Conference*, BMVC, pages 9–18, 1992.

P. Correia and F. Pereira. Objective evaluation of relative segmentation quality. In *IEEE International Conference on Image Processing*, pages 308–311, Vancouver, Canada, September 2000.

R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, 2003.

R. Cutler and L. Davis. View-based detection and analysis of periodic motion. In *Proceedings of the 14th International Conference on Pattern Recognition*, volume 1 of *ICPR '98*, pages 495–500, Brisbane, Australia, 1998.

M. Daszykowski, K. Kaczmarek, Vander, and B. Walczak. Robust statistics in data analysis – a review: Basic concepts. *Chemometrics and Intelligent Laboratory Systems*, 85(2):203–219, 2007.

F. De la Torre and M.J. Black. Robust principal component analysis for computer vision. In *Proceedings of IEEE International Conference on Computer Vision*, volume 1 of *ICCV'01*, pages 362–369, Vacouver, Canada, 2001.

R.D. DeGroat and R.A. Roberts. Efficient, numerically stabilized rank-one eigenstructure updating. *IEEE Transaction on Acoustics Speech, and Signal Processing*, 38:301–316, 1990.

S. Denman, C. Fookes, S. Sridharan, and R. Lakemond. Dynamic performance measures for object tracking systems. In *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, AVSS '09, pages 541 –546, September 2009.

A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *6th European Conference on Computer Vision*, pages 751–767, Dublin, Ireland, June 2000.

C. Erdem and B. Sankur. Performance evaluation metrics for object-based video segmentation. In *10th European Signal Processing Conference (EUSIPCO)*, September 2000.

C. Erdem, A. Tekalp, and B. Sankur. Metrics for performance evaluation of video object segmentation and tracking without ground-truth. In *IEEE International Conference on Image Processing (ICIP)*, October 2004.

I. Everts, N. Sebe, and G.A. Jones. Cooperative object tracking with multiple ptz cameras. In *Proceedings of the International Conference on Image Analysis and Processing*, September 2007.

C. Faloutsos and K.-I. Lin. Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, SIGMOD '95, pages 163–174, 1995.

R. Fisher. Caviar - context aware vision using image-based active recognition. http://homepages.inf.ed.ac.uk/rbf/CAVIAR.

R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179 –188, 1936.

I. Fodor. A survey of dimension reduction techniques. Technical report, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 2002.

A. Franco, A. Lumini, and D. Maio. Eigenspace merging for model updating. In *International Conference on Pattern Recognition*, volume 2, pages 156–159, Quebec, Canada, 2002.

N. Friedman and R. Russell. Image segmentation in video sequences: A

probabilistic approach. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence*, UAI-97, pages 175–181, 1997.

X. Gao, T. Boult, F. Coetzee, and V. Ramesh. Error analysis of background adaption. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 503–510, 2000.

E. Gelasca, T. Ebrahimi, M. Farias, M. Carli, and S. Mitra. Towards perceptually driven segmentation evaluation metrics. In *CVPR 2004 Workshop on Perceptual booktitle in Computer Vision*, page 52, June 2004.

B. Georis, F. Bremond, M. Thonnat, and B. Macq. Use of an evaluation and diagnosis method to improve tracking performances. In *IASTED 3rd International Conference on Visualization, Imaging and Image Processing*, September 2003.

G. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14:403–420, 1970.

R. L. Gorsuch. *Factor Analysis.* Lawrence Erlbaum, 2nd edition, 1983.

D. Hall, J. Nascimento, P. Ribeiro, E. Andrade, and P. Moreno. Comparison of target detection algorithms using adaptive background models. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, October 2005.

P.M. Hall, D. Marshall, and R.R. Martin. Incremental eigenanalysis for classification. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 286–295, 1998.

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

J Heikkilä and O Silvén. A real-time system for monitoring of cyclists and pedestrians. In *Proceedings of the Second IEEE Workshop on Visual Surveillance*, VS '99, pages 74–81, 1999.

J. Herault and C. Jutten. Space or time adaptative signal processing by neural networks models. In *International Conference on Neural Networks for Computing*, CMCE '10, pages 206–211, 1986.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

H. Hotelling. Relations between two sets of variants. *Biometrika*, 28:321–377, 1936.

A. K. Jain, R.P.W. Duin, and J Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4–37, 2000.

C. Jaynes, S. Webb, M. Steele, and Q. Xiong. An open development environment for evaluation of video surveillance systems. In *IEEE Workshop on Performance Analysis of Video Surveillance and Tracking (PETS'2002)*, June 2002.

I.T. Jolliffe. *Principal component analysis.* Springer, 2002.

P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proceedings of the Workshop on Advanced Video Based Surveillance Systems*, Kingston, UK, 2001.

K.P. Karmann and A. Brandt. *Moving object recognition using and adaptive background memory*, pages 289–307. Elsevier Science Publishers, 2 edition, 1990.

D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proceeding of the 12th International Conference on Pattern Recognition*, pages 126–131, Jerusalem, October 1994.

P. Kumar, K. Sengupta, and A. Lee. A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system. In *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, pages 100–105, 2002.

D.-S. Lee, J. Hull, and B. Erol. A bayesian framework for gaussian mixture background modeling. In *Proceedings of IEEE International Conference on Image Processing*, Barcelona, Spain, Sep 2003.

W. Li, S. Yue H.H, Valle-Cervantes, and S.J. Qin. Recursive pca for adaptive process monitoring. *Journal of Process Control*, 10:471–486, 2000.

Y. Li. On incremental and robust subspace learning. *Pattern Recognition*, 37(7):1509 – 1518, 2004.

H. Lin, T. Liu, and J. Chuang. A probabilistic svm approach for background scene initialization. In *International Conference on Image Processing*, volume 3 of *ICIP 2002*, pages 893–896, Rochester, New York, Sep 2002.

T. List, J. Bins, J. Vazquez, and R.B. Fisher. Performance evaluating the evaluator. In *IEEE Joint Workshop on Visual Surveillance and Performance Analysis of Video Surveillance and Tracking (VS-PETS 2005)*, October 2005.

X. Liu and T. Chen. Shot boundary detection using temporal statistics modeling. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 781–784, Orlando, Florida, USA, 2002.

B.P.L. Lo and S.A. Velastin. View-based detection and analysis of periodic motion. In *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 158–161, Hong Kong, China, May 2001.

X-L. Lv, G-L. Zhao, and H. Meng. A new method for selecting gradient weight in incremental eigen-background modeling. In *International Conference on Information and Automation*, ICIA '09, pages 801–805, June 2009.

A. Manzanera and J.C. Richefeu. A new motion detection algorithm based on sigma-delta background estimation. *Pattern Recognition Letters*, 28(3): 320–328, 2007.

V.Y. Mariano, J. Min, J.H. Park, R. Kasturi, D. Mihalcik, H. Li, D.S. Doermann, and T. Drayer. Performance evaluation of object detection algorithms. In *16th International Conference on Pattern Recognition*, volume 2 of *ICPR*, pages 965–969, 2002.

D.R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 26(1), January 2004.

Z. Mayo and J.R. Tapamo. Background subtraction survey for highway surveillance. In *Proceedings of the Twentieth Annual Symposium of the Pattern Recognition Association of South Africa*, Stellenbosch, South Africa, Nov-Dec 2009.

N. J. B. McFarlane and C. P. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8:187–193, 1995.

A.M. McIvor. Background subtraction techniques. In *Proceedings of Image and Vision Computing*, Hamilton, New Zealand, 2000.

A. Mittal and L. Davis. Unified multi-camera detection and tracking using region-matching. In *IEEE Workshop on Multi Object Tracking*, 2001.

B. Moghaddam and A Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 696–710, 1997.

H. Murakami and V. Kumar. Efficient calculation of primary images from

a set of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 511 – 515, 1982.

J. Nascimento and J.S. Marques. New performance evaluation metrics for object detection algorithms. In *IEEE Workshop on Performance Analysis of Video Surveillance and Tracking (PETS'2004)*, May 2004.

J. Nascimento and J.S. Marques. Performance evaluation for object detection algorithms for video surveillance. *IEEE Transaction on Multimedia*, 2006.

H. Niemann. Linear and nonlinear mapping of patterns. *Pattern Recognition*, 12(2):83 – 87, 1980.

F. Oberti, A. Teschioni, and C.S. Regazzoni. Roc curves for performance evaluation of video sequences processing systems for surveillance applications. In *Proceedings of the International Conference on Image Processing, (ICIP)*, volume 2, pages 949–953, October 1999.

N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, Aug 2000.

J. Orwell, P. Remagnino, and G.A. Jones. Multi-camera color tracking. In *Proceedings of the Second IEEE Workshop on Visual Surveillance*, pages 14–21, Fort Collins, Colorado, USA, June 1999.

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

C.-J. Pai, H.-R. Tyan, Y.-M. Liang, H.-Y.M. Liao, and S.-W. Che. Pedestrian detection and tracking at crossroads. *Pattern Recognition*, 37:1025–1034, 2004.

M. Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3099–3104, Oct 2004.

W. Power and J.A. Schoonees. Understanding background mixture models for foreground segmentation. In *Proceedings of the Image and Vision Computing New Zealand*, pages 267–271, Auckland, New Zealand, 2002.

J. Provost and T. Fawcett. Analysis and visualisation of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, 1997.

P. Remagnino, A. Baumberg, T. Grove, D Hogg, T. N. Tan, A. D. Worrall, and K.D. Baker. An integrated traffic and pedestrian model-based vision system. In *Proceedings of the 8th British Machine Vision Conference*, BMVC'97, pages 380–389, 1997.

J. Renno, N. Lazarevic-McManus, D. Makris, and G. Jones. Evaluating motion detection algorithms: Issues and results. In *IEEE International Workshop on Visual Surveillance*, pages 97–104, Graz, Austria, May 2006.

L. K. Roweis, S. T.and Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

J. Rymel, J. Renno, D. Greenhill, J. Orwell, and G. Jones. Adaptive eigen-backgrounds for object detection. In *IEEE International Conference on Image Processing*, volume 3 of *ICIP 2004*, pages 24–27, Suntec City, Singapore, October 2004.

J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computing*, 18:401–409, 1969.

T. Schlogl, C. Beleznai, M. Winter, and H. Bischof. Performance evaluation metrics for motion detection and tracking. In *17th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 519–522, August 2004.

B. Schölkopf, K. R. Smola, and Müller K. R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

M. Seki, T. Wada, H. Fujiwara, and K. Sumi. Background subtraction based on cooccurrence of image variations. In *Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2 of *CVPR2003*, pages 65–72, June 2003.

D. Skočaj and A. Leonardis. Incremental and robust learning of subspace representations. *Image and Vision Computing*, 26:27–38, January 2008.

C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'2000)*, pages 246–252, Fort Collins, Colorado, June 1999.

L.D. Stefano, G. Neri, and E. Viarani. Analysis of pixel-level algorithms for video surveillance applications. In *11th International Conference on Image Analysis and Processing (ICIAP2001)*, pages 542–546, September 2001.

M. Surgue and E.R. Davies. Motion distillation for pedestrian surveillance. May 2006.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

D. Thirde, M. Borg, J. Aguilera, J. Ferryman, K. Baker, and M. Kampel. Evaluation of object tracking for aircraft activity surveillance. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 145–152, Beijing, October 2005.

K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 1 of *ICCV '99*, pages 255–261, 1999.

F.S. Tsai. Comparative study of dimensionality reduction techniques for data visualization. *Journal of Artificial Intelligence*, 3:119–134, 2010.

L.J.P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review. Technical report, 2008.

C.J. van Rijsbergen. *Information Retrieval.* Butterworths,London, 2nd edition, 1979.

R. Vidal, Yi Ma, and S. Sastry. Generalized principal component analysis (gpca). *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1945–1959, 2005.

P. Villegas and X. Marichal. Perceptually weighted evaluation criteria for segmentation masks in video sequences. *IEEE Transactions on Image Processing*, 13(8):1092–1103, August 2004.

X. Villegas, P. Marichal and A. Salcedo. Objective evaluation of segmentation masks in video sequences. In *Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'99)*, pages 85–88, May 1999.

S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2010, pages 1030 –1037, June 2010.

C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, 1997.

H. Wu and Q. Zheng. Self-evaluation for video tracking systems. In *Proceedings of the 24th Army Science Conference*, November 2004.

L. Xu and A.L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, 6(1):131–143, 1995.

J. Zeng, L. Xie, and Z.-Q. Liu. Type-2 fuzzy gaussian mixture models. *Pattern Recognition*, 41:3636–3643, 2008.

X.-Y. Zeng, Y.-W. Chen, and Z. Nakao. Image feature representation by the subspace of nonlinear pca. In *Proceedings 16th International Conference on Pattern Recognition*, pages 228 – 231, 2002.

J. Zhang and Y Zhuang. Adaptive weight selection for incremental eigen-background modeling. In *IEEE International Conference on Multimedia and Expo*, pages 851–854, July 2007.

S. Zhang, H. Yao, and S. Liu. Dynamic background subtraction based on local dependency histogram. In *IEEE International Workshop on Visual Surveillance*, VS 2008, pages 1–8, Marseille, France, Oct 2008.

Q. Zhou and J.K. Aggarwal. Tracking and classifying moving objects from video. In *Proceedings of the 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, PETS2001, 2001.