Spatial and temporal background modelling of non-stationary visual scenes

Russell, David Mark

Spatial and Temporal Background Modelling of Non-stationary Visual Scenes

David Mark Russell

Submitted to the University of London in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Queen Mary, University of London

2009

Abstract

The prevalence of electronic imaging systems in everyday life has become increasingly apparent in recent years. Applications are to be found in medical scanning, automated manufacture, and perhaps most significantly, surveillance. Metropolitan areas, shopping malls, and road traffic management all employ and benefit from an unprecedented quantity of video cameras for monitoring purposes. But the high cost and limited effectiveness of employing humans as the final link in the monitoring chain has driven scientists to seek solutions based on *machine vision* techniques. Whilst the field of machine vision has enjoyed consistent rapid development in the last 20 years, some of the most fundamental issues still remain to be solved in a satisfactory manner.

Central to a great many vision applications is the concept of segmentation, and in particular, most practical systems perform *background subtraction* as one of the first stages of video processing. This involves separation of 'interesting foreground' from the less informative but persistent background. But the definition of what is 'interesting' is somewhat subjective, and liable to be application specific. Furthermore, the background may be interpreted as including the visual appearance of *normal activity* of any agents present in the scene, human or otherwise. Thus a *background model* might be called upon to absorb lighting changes, moving trees and foliage, or normal traffic flow and pedestrian activity, in order to effect what might be termed in 'biologically-inspired' vision as *pre-attentive selection*. This challenge is one of the Holy Grails of the computer vision field, and consequently the subject has received considerable attention.

This thesis sets out to address some of the limitations of contemporary methods of background segmentation by investigating methods of inducing *local mutual support* amongst pixels in three starkly contrasting paradigms: (1) locality in the spatial domain, (2) locality in the short-term time domain, and (3) locality in the domain of cyclic repetition frequency.

Conventional per pixel models, such as those based on Gaussian Mixture Models, offer no spatial support between adjacent pixels at all. At the other extreme, eigenspace models impose a structure in which every image pixel bears the same relation to every other pixel. But Markov Random Fields permit definition of arbitrary local *cliques* by construction of a suitable graph, and

are used here to facilitate a novel structure capable of exploiting probabilistic local cooccurrence of adjacent Local Binary Patterns. The result is a method exhibiting strong sensitivity to multiple learned local pattern hypotheses, whilst relying solely on monochrome image data.

Many background models enforce temporal consistency constraints on a pixel in attempt to confirm background membership before being accepted as part of the model, and typically some control over this process is exercised by a learning rate parameter. But in busy scenes, a true background pixel may be visible for a relatively small fraction of the time and in a temporally fragmented fashion, thus hindering such background acquisition. However, support in terms of temporal locality may still be achieved by using Combinatorial Optimization to derive short-term background estimates which induce a similar consistency, but are considerably more robust to disturbance. A novel technique is presented here in which the short-term estimates act as 'pre-filtered' data from which a far more compact eigen-background may be constructed.

Many scenes entail elements exhibiting repetitive periodic behaviour. Some road junctions employing traffic signals are among these, yet little is to be found amongst the literature regarding the explicit modelling of such periodic processes in a scene. Previous work focussing on gait recognition has demonstrated approaches based on recurrence of self-similarity by which local periodicity may be identified. The present work harnesses and extends this method in order to characterize scenes displaying multiple distinct periodicities by building a spatio-temporal model. The model may then be used to highlight abnormality in scene activity. Furthermore, a Phase Locked Loop technique with a novel phase detector is detailed, enabling such a model to maintain correct synchronization with scene activity in spite of noise and drift of periodicity.

This thesis contends that these three approaches are all manifestations of the same broad underlying concept: *local support* in each of the space, time and frequency domains, and furthermore, that the support can be harnessed practically, as will be demonstrated experimentally.

Declaration

I hereby declare that this thesis has been composed by myself and that it describes my own work. It has not been submitted, either in the same or different form, to this or any other university for a degree. All verbatim extracts are distinguished by quotation marks, and all sources of information have been acknowledged.

Some parts of the work have previously been published as:

1. D. Russell and S. Gong. Multi-layered Decomposition of Recurrent Scenes. In the proceedings of the *10th European Conference on Computer Vision (ECCV)*, Marseille 2008.

2. D. Russell and S. Gong. Exploiting Periodicity in Recurrent Scenes. In the proceedings of the *19th British Machine Vision Conference (BMVC)*, Leeds 2008.

3. D. Russell and S. Gong. Segmenting Highly Textured Non-stationary Background. In the proceedings of the *18th British Machine Vision Conference (BMVC)*, Warwick 2007.

4. D. Russell and S. Gong. Minimum Cuts of A Time-Varying Background. In the proceedings of the *17th British Machine Vision Conference (BMVC)*, Edinburgh 2006.

5. D. Russell and S. Gong. A Highly Efficient Block-based Dynamic Background Model. In the proceedings of the *IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, Como 2005.

David Russell.

London, March 2009.

Acknowledgements

First and foremost I would like to thank my supervisor Professor Shaogang Gong for his seemingly endless stimulating input, rewarding discussion, encouragement, criticism, and most especially his abiding patience, in the completion of this research and thesis.

I would like to thank and acknowledge members of the academic staff and PhD students past and present for their enlightening discussions and general help in all manner of ways. In no particular order: Tao Xiang, Fabrizio Smeraldi, Lourdes Agapito, Peter McOwan, Christof Monz, Tassos Tombros, Andrew Graves, Hayley Hung, Alex Leung, Alessio del Bue, Caifeng Shan, Jun Li, Samuel Pachoud, Milan Verma, Bryan Prosser, and Yogesh Raja.

Many thanks also to members of the Systems Support department: Tim Kay, Lukasz Zalewski, Matt Bernstein, Tom King, Keith Clarke, and David Hawes, for solving countless software problems, more often than not of my own making.

I would like to express my gratitude to members of the departmental administrative staff, without whose help I would frequently have been lost: Joan Hunter, Carly Wheeler, Julie Macdonald, Sue White, Karen Finesilver, Rupal Vaja, and Roger Law.

I am very grateful to the Engineering and Physical Sciences Research Council (EPSRC) for sponsorship of the work described herein.

I would also like to thank Vladimir Kolmogorov for use of his C++ implementation of the MinCut/MaxFlow algorithm, available at: http://www.adastral.ucl.ac.uk/ vladkolm/software.html

Finally, I would like to thank my parents and friends for their continued support and encouragement throughout my recent return to academic pursuits.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Interpretation of natural scenes is a process which happens largely automatically in the human brain. Yet scientists from a whole range of fields such as Biology, Psychology, Neurology, and indeed Computer Vision, have long wanted to know precisely how it happens. Nakayama et al. [87] observe that:

> "Retinal images are formed on the back of our eyeballs, upside down; they are very unstable, abruptly shifting two to four times a second according to the movements of the eyes... ...yet the visual scene appears to us as upright, stable and homogeneous."

This is a truly impressive example of image processing, but it immediately raises the question of where the scene is actually perceived - not on the retina evidently. The same authors maintain that understanding the mechanism of this process will go a long way to explaining thought and perception generally. But in any case, as appears usual with biological systems, nature has parsimoniously evolved a human vision system fit for one particular purpose: survival.

But why should humans possess such curiosity regarding the detailed functioning of the human visual system? Perhaps from their purely inquisitive approach to science and nature generally. Or perhaps to be able to supplant such a system with a synthetic alternative, maybe even with several potential improvements.

The ever-growing reliance of man on machine has meant that a diverse range of 'intelligent' devices based on computational elements has pervaded many aspects of human existence. Increasingly such devices possess some type of camera interface, and algorithms in Computer

Vision have helped to propel the interaction and usability of the devices to new heights. But as a vital part of the vision process, the machine must perceive *individual* objects forming the world around it in order to interact in a useful way. The capability to logically separate objects, from each other and more especially from the background, is a basic attribute of vision systems. However, despite continued research effort, truly reliable solutions have yet to be attained, especially for use under challenging variable lighting conditions, and where the background may not be stationary, in a physical or statistical sense.

But traditional algorithms for distinguishing foreground from background rarely make best use of *local* image information to achieve accurate separation. This thesis is concerned with finding strategies for improving on current techniques by exploiting *local connectivity* between image elements which are adjacent in space, time, and repetition rate, in order to encourage co-operative decisions about the location of boundaries between foreground and background.

1.1 Surveillance

Amongst potential exponents of vision algorithms, few applications have become more prolific than visual surveillance in recent years. According to McCahill and Norris [81] it is estimated that there were at least 400,000 video surveillance cameras in London alone as of 2002, and more than 4 million spread throughout the UK. Considering that these devices operate continuously, the quantity of video data generated is enormous. Both metropolitan authorities and private security firms employ people to watch and monitor ongoing events in the hope of identifying criminal activity, untoward behaviour, and serious but non-malicious situations. Evidently, due to the sheer volume of data, the task of surveillance becomes increasingly difficult to manage even in a well staffed camera control room, such as that shown in Figure 1.1. Most humans can maintain the required level of concentration for as little as 20 minutes at a time before fatigue and boredom erode their powers of attention, this being partly due to the fact that for almost all of that time, nothing of interest actually happens. According to a report by Gill and Spriggs [41] many quieter, relatively uninteresting, city areas receive little or no attention for long periods of time for these very reasons. Figure 1.2 shows police officers attentively monitoring popular tourist areas in London, but it is not practical to deploy such intensive surveillance everywhere.

However, thanks to techniques in the field of computer vision, it may now be possible to relegate the bulk of the thankless surveillance task to machines which never tire or lose interest

or job satisfaction. Enter the synthetic vision system. And perhaps just as in nature, one of the crucial goals is *survival*. Not of the machines themselves, but this time of their creators, against the threat of bombs, theft, and other less serious crimes.

Figure 1.1: Control Room at Newham Borough Council in London, catering for approximately 500 video feeds from surrounding public and restricted areas. Although all cameras are recorded, not all can be *viewed* at once in real time, especially by such a limited number of operators.

Figure 1.2: Police at New Scotland Yard monitor key security locations. But such focussed surveillance leaves little resource for coverage of quieter areas. Photograph: Kirsty Wigglesworth/AFP/Getty Images.

1.2 Machine Vision

By employing *machine learning* algorithms, it should be possible to accumulate details regarding activity in a scene covered by a fixed view camera in a *statistical* sense, such that any unusual event could be highlighted and brought rapidly to the attention of a human observer. Thus the

overall monitoring task might then be accomplished with *fewer* people watching *more* stimulating activity for a greater proportion of the time, and in a smaller control room consisting of fewer TV screens. Instead of relying on easily fatigued human concentration, the monotonous 'watching' process would go on in the CPU of a computer, continuously fed with digital video data from one or more cameras.

Such a system may be required to generate immediate alarms, or trigger certain events in critical situations. But another paradigm is that of *retrospective investigation*, whereby people or vehicles might be tracked through archived data across disparate Closed Circuit Television (CCTV) networks with a view to solution of some particular crime.

But the implementation of software and algorithms suitable for this purpose is a non-trivial task, and full realization of the above situation still lies in the future. Although machine learning has been around for many decades, and is a diverse topic that has facilitated numerous applications that impinge daily on human lives, it is nevertheless still a rapidly developing topic. Many aspects of computer vision depend critically on principles rooted in machine learning, since scene appearance and activity patterns for example, cannot practically be programmed directly.

In general, activity implies movement of objects in a scene such as people, vehicles, and trees, which cause localized changes in the scene's appearance due to occlusion of the background. The presence or absence of stationary objects also affects local appearance in a similar fashion. At the same time, variations in prevailing weather and lighting conditions cause changes in scene appearance on a *global* scale, as illustrated in Figure 1.3. In both global and local cases, the activity manifests itself as changes in colour and intensity at some subset of pixels. The objective is thus to identify unusual events by modelling the spatial and temporal intensity characteristics of the scene on a frame by frame basis. The ideal model is constructed or 'learned' incrementally from such data, and matches all aspects of normal activity. Rare behaviour in the scene is highlighted where it occurs because it *does not* fit the learned model.

1.3 Image Acquisition

In order for a machine to start interpreting a scene, something physical about that scene must first be measured. By far the most common type of sensor available is the standard video camera based on a CCD (Charge Coupled Device) target, measuring light intensity levels in the three spectral areas Red, Green, and Blue (RGB) as a regular 2-D array of pixels. Because most objects are not

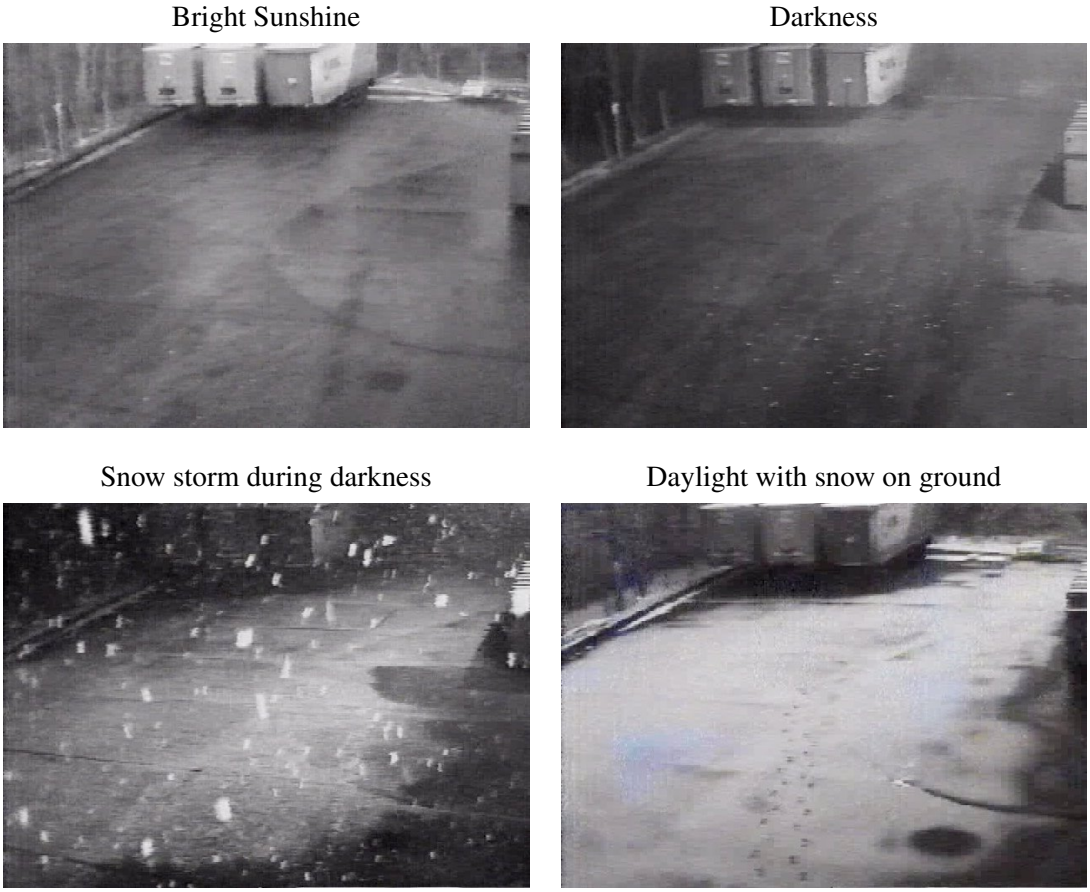


Figure 1.3: Typical surveillance images from a factory goods yard, in which the same scene is depicted under widely varying lighting and weather conditions. However, despite the marked change in appearance, from a surveillance novelty standpoint these views should all be classed as background. Such diversity is challenging, but not uncommon in practical situations.

themselves emitters of any light, use of such cameras is usually restricted to situations involving daylight or some artificial light source.

However, whilst the visible part of the electro-magnetic spectrum forms the basis of perception in human vision, other parts of the spectrum are also found to contain useful information. Most notably, infra-red imagery [76] presents a pattern of thermal emission from the scene. Again, many passive objects exhibit no emission of their own here either. However, objects which are themselves sources of heat, for example, people, animals and engine-driven vehicles, betray themselves by a significant heat signature. For this reason, infra-red imaging performs a vital role with regard to covert surveillance. At still longer wavelength, Terahertz radiation [119] remains strongly directional and thus is also a contender for such application. A method using a fusion of visible *and* thermal channels (RGB+T) has also been proposed [124].

But measurement of electro-magnetic radiation intensity, whether emitted or reflected, only gives the observer an impression of *appearance*. In the far field this amounts to a two-dimensional representation of scene content. Range Images [2], on the other hand, provide a measure of *distance* from the sensor, and can therefore lead directly to a three-dimensional representation of the scene from a single sensor's point of reference. Such 3-D images may be deemed considerably more informative than their 2-D counterparts because of the potential to resolve ambiguities regarding 3-D shape, and reason about occlusion. But the much higher system cost renders these range imaging devices less popular, and consequently perhaps hard to justify from a commercial point of view.

Stereoscopic vision systems with two cameras also permit 3-D *scene reconstruction* [48] but are not commonly used in practice for surveillance, at least partly due to cost considerations, but also perhaps because of the more complex setup which generally hinders ease of deployment. A more likely scenario is a network of cameras at a site in which there is some overlap of camera coverage [65], although this does not guarantee a comprehensive 3-D scene model. In reality, many practical cost effective solutions to surveillance problems are ultimately provided by relatively inexpensive video camera technology used within a multi-camera environment, regardless of whether spatial proximity is specifically exploited.

1.4 Unit Formation

Before attempting to define any type of artificial vision system, it seems worth taking a moment to gain insight into how the Human Visual System (HVS) manages to interpret visual scenes so successfully. Crucial to this process is the concept of *scene decomposition*, or *segmentation*. In HVS parlance, segmentation is often referred to as *unit formation* [64]. As an involuntary cognitive process, the image falling on the retina is 'parsed' into separate surfaces in order to establish the extent of the entities in the image, a process known as *individuation*. Surfaces are defined by their boundaries, but it is the association of a particular boundary with a given surface which controls the perceived juxtaposition of the surfaces in terms of depth. By a process known as *amodal completion*, introduced in work by Michotte [84] and analyzed by Kanizsa [63], surfaces separated by occlusion may be reunited to enable perception of the entire scene in a 3D space, in spite of the occlusion, and *without* the need for stereoscopic vision. Perception of whole entities is advantageous since at a higher cognitive level, interaction with the environment

is with discrete complete objects, as described by Spelke et al. in [123].

Such a form of discretization is a form of pre-attentive selection, by which concentration at higher cognitive levels can be directed towards distinct concepts in the scene, e.g. tiger attacking from the left. This prioritization becomes necessary when the brain has only a finite capacity for dealing with separate objects. A crucial point is that parsing of surfaces into objects happens at a relatively low level, and according to very specific but highly adaptive rules, as described by Nakayama et al. in [87].

Much of the above explanation is supported by inference from experiments involving optical illusions and demonstrations, and a much more detailed account of the unit formation process is presented by Shipley and Kellman in [64].

In an *artificial segmentation* application it would seem useful to replicate some type of unit formation scheme, although with a rather simpler rule set. Applications are to be found in the computer vision literature regarding use of Markov Random Fields to group together pixels with similar appearance, according to the prior knowledge that most uniformly coloured regions tend to form continuous surfaces, as proposed by Schindler and Wang in [111]. But however it is achieved, identifying individual entities amongst an image of pixels is the final objective. Temporal consistency of an object is undoubtedly an influence in the HVS, as reported by Shipley in [118], and should be exploited to advantage in any type of synthetic equivalent system.

1.5 Background Modelling

Although complete segmentation of all objects in a scene from each other is a laudable target, it may not be easily achieved, or actually even necessary. Often it suffices to distinguish what is interesting in a scene and denote it *foreground* with respect to objects which are normally in the scene: the *background*. By definition the background is persistent, and consequently in general more data is available to describe it. For this reason, the type of model capable of distinguishing *foreground* from *background* is usually termed a *background model*.

In contrast, some systems described in the literature achieve reliable foreground identification in context by specifically *tracking* individual objects and their boundaries through a scene [56]. Whilst this might achieve good results in ideal circumstances, such tracking algorithms are computationally intensive, especially when the number of objects is large, but in addition, may be hindered significantly by image clutter and noise. Temporary loss of tracking, and the subse-

quent re-establishment of correspondence is also a problem. So in general, explicit *background* modelling is more widely adopted.

Effective background modelling is a crucial first stage in most computer vision applications, especially in outdoor environments. The reliability with which potential foreground objects can be segmented and subsequently identified directly impacts on the efficiency and performance level achievable by subsequent processing stages such as tracking, recognition and threat evaluation. The nature of background is considered to be intrinsically statistical. Whilst the concept of statistical scene modelling suggests that there is no exact distinction between what constitutes foreground and background, a useful practical definition for surveillance in a busy urban scene is that people and the objects they cause to move are foreground. Buildings, fixtures, trees and permanent objects, together with any environmental change in lighting such as shadow caused by moving clouds, form the background. Critically, it is considered that background is in general necessarily *amongst* foreground, i.e. it can be literally *behind* and *in front* of foreground objects, especially in urban outdoor scenes, as shown by the examples in Figure 1.4. The task of a background model in such a setting is to discriminate between the two classes under a potentially wide variety of lighting conditions. Evidently, confusion might still arise, since trees sway in the wind, tending to become foreground, whilst people park their cars, which are eventually subsumed by the background.

The most commonly encountered models are based on per pixel techniques such as adaptive Gaussian Mixture Models [125, 131], or subspace analysis methods [94, 73], and both ap-

Figure 1.4: Examples demonstrating how background can be both behind and amongst foreground. As the person moves behind the tree leaves, the background partially occludes the foreground. The question is how to achieve useful segmentation from such fragmented evidence.

proaches have been used with success in many applications. However, as might be expected, neither class of approach turns out to provide completely reliable segmentation in real world situations. In particular, outdoor scenes suffer significantly from the effects of lighting variation due to prevailing weather conditions and the time of day. For example cloud cover alters the intensity and diffuseness of ambient lighting, and the contrast between areas illuminated by direct sunlight and those in the shade. The physical appearance of a scene can vary dramatically as a result of these effects, yet for most purposes, a background model would be required to classify the same scene as background regardless of absolute appearance under a range of lighting conditions, as Figure 1.3 clearly illustrates.

A further problem is that wind disturbs many types of leafy vegetation. Whilst the *textural* appearance of a bush might be largely constant, the absolute detailed pattern of pixels representing it is likely to change radically over time. Again, the ideal background model would always classify these regions as background in spite of the local chaotic motion. Random specular reflection from moving water presents a similar set of problems.

But the two classes of approach mentioned previously, the per pixel model and the global subspace model, represent two extremes in terms of pixel connectivity. By definition, the per pixel model operates on each pixel in spatial isolation, and hence is *unable* to exploit information about objects in the scene encoded in the similarity of neighbouring pixels. This situation is shown in Figure 1.5, where many isolated pixels are highlighted as foreground whilst evidence from surrounding pixels is in contradiction. On the other hand, the subspace model attempts to form a spatially holistic model of the scene. But given that the model is restricted to eigenvectors (representing image *modes*) linked to only a set of the *largest* eigenvalues, the descriptive power of each model component is still potentially distributed across the whole scene. If regions of the scene are largely visually independent of each other, then this descriptive power could be used to greater effect if it could be concentrated more locally. Figure 1.6 illustrates the distribution of model eigenvectors across a typical road junction scene. Whilst some regionalization of the eigenvectors is visible, many overlap extensively, and are spread over large and diverse image areas. It is possible that some connectivity between certain regions arises accidentally from inadequate training data, when in reality there is no such valid relationship. Figure 1.7 shows the result of a typical segmentation using a subspace model constrained to 14 eigenvectors. The model clearly has trouble expressing some scene regions effectively.

Figure 1.5: Left: Typical busy traffic junction. Right: Segmentation using a 5 component Gaussian Mixture Model. Many random noise pixels are highlighted even though they should be part of the background. Each pixel acts as a process in isolation in this type of model, so no local support or consensus can be drawn upon during segmentation. A more useful segmentation may be obtained if decisions for a pixel were formed co-operatively.

So overall, it would appear that permitting *local* support between pixels, dependent on local correlation observed during training, is likely to lead to a more informed foreground/background segmentation of an image. The gestalt theory from psychology [114] suggests that local groupings are an essential contributor to the success of the Human Visual System, so it seems not unreasonable to exploit such local support in a background model. But taking the idea still further, spatio-temporal *persistence* of objects is anticipated in human cognition, and this too is a form of grouping or *local temporal support*.

In the light of the previous discussions and examples, it appears logical to devise mechanisms which augment accuracy of segmentation by employing *local* rather than global or pixel-wise support, be this of a spatial, temporal, or other nature. The focus of this thesis is to explore models which harness the benefits of such support in three different ways.

The approaches to be examined are characterized by treating the image as a *signal*, applying algorithms homogeneously across the sequence in all dimensions. Such an approach intrinsically lends itself to easier implementation on cellular type processing arrays, or in hardware, with all the flexibility and practical advantages in deployment that this brings to real applications. As an example, in [147] a motion segmentation algorithm is implemented on a graphics card array processor having Single Instruction Multiple Data (SIMD) capabilities, achieving a 12 times increase in throughput compared with a CPU only implementation.

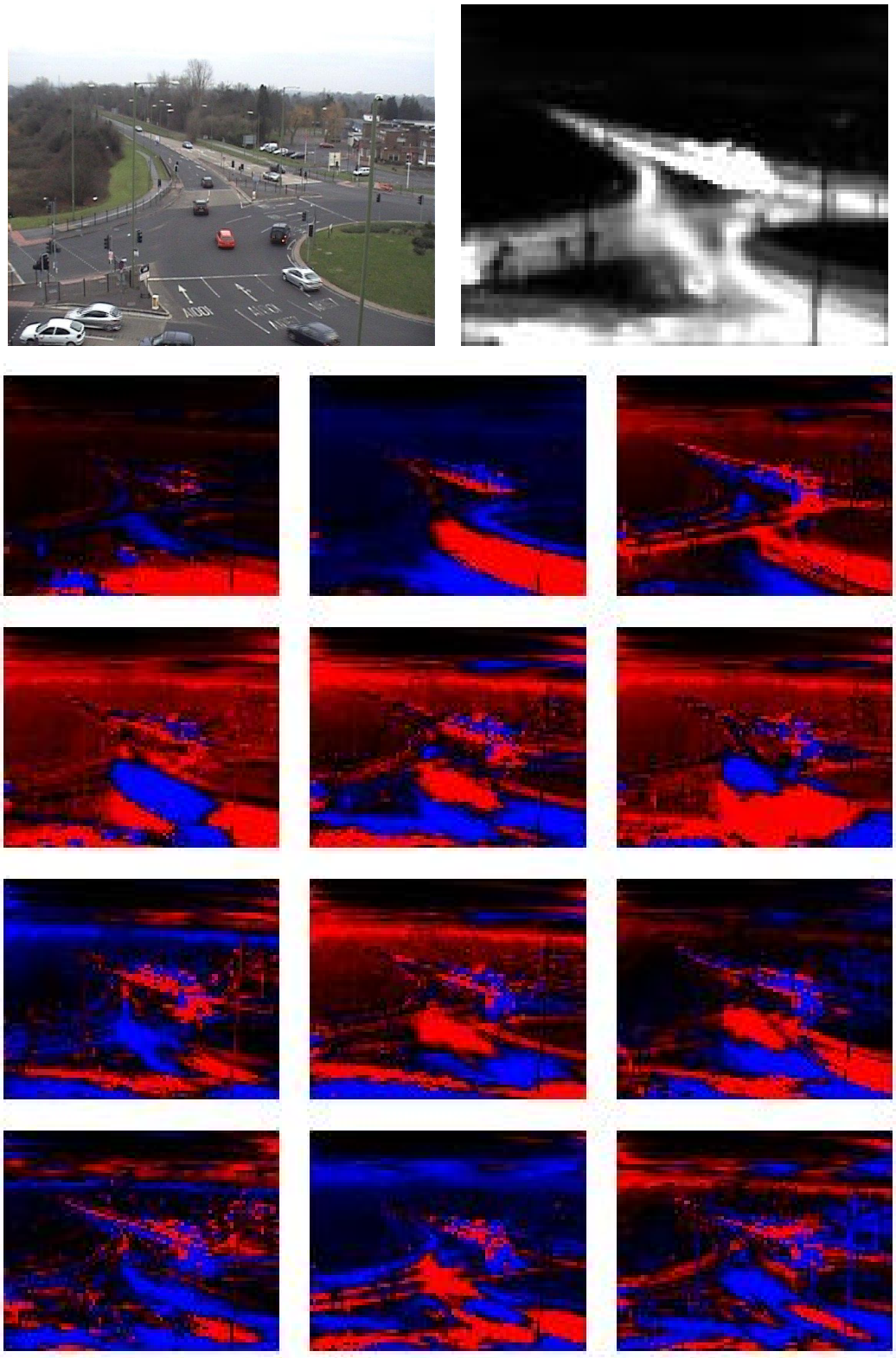


Figure 1.6: Top left: Typical traffic junction. Top right: Regions of activity from pixel variance. Bottom: First 12 eigenimages from decomposing covariance matrix derived from 7500 input frames taken over 1 hour. Red and blue represent +ve and -ve eigenvector elements respectively.

Figure 1.7: Left: Typical busy traffic junction. Right: Segmentation using a Subspace Model consisting of the first 14 eigenvectors corresponding to the largest eigenvalues of the training set covariance matrix. Many items, especially road markings are highlighted even though they should be part of the background. This may occur partly because there are too few eigenvectors trying to act globally across the image, as in the eigenvector distributions of Figure 1.6.

1.6 Approach

1.6.1 Pattern-based Background Identification

A logical starting point is to consider local *spatial* support in the Foreground/Background decision. By defining a simple *rotationally sensitive* Local Binary Pattern (LBP), local image gradient is approximated by a discrete symbol set possessing a limited range of values. Accumulation of local cooccurrence statistics exhaustively between all directly adjacent pixel sites then becomes tractable in terms of storage requirements. The accrued joint distribution over the symbol set at each pixel pair is then be used to evaluate the relative conditional probability of symbol occurrence between the two sites in previously unseen data.

Casting the problem as a Markov Random Field, a binary graph cut permits evidence at one pixel site to give weight to surrounding segmentation decisions, based on the concept of achieving a global optimum. The merit of such an approach is to preserve the mutual information between adjacent pixel sites that is encoded in the cooccurrence map, and to use that acquired data to best advantage in creating a more informed segmentation mask.

Previous methods have imposed an *intensity* smoothness constraint between adjacent pixels, but the novelty in this thesis is to base the constraint on *potentially different* LBP symbols which are consistently observed simultaneously at the two respective neighbouring sites, regardless of

the actual symbols or patterns they represent. The advantage of this approach is that it is automatically well suited to modelling highly textured regions such as leaves and vegetation. Furthermore, the cooccurrence map encodes *all* commonly cooccurring symbol pairs between two adjacent pixels, allowing the model to accommodate dynamic texture such as *moving* vegetation.

Although already useful, the technique described so far does not exploit any type of *temporal persistence*, an attribute which might equally be considered as *locality in time*.

1.6.2 Estimation of Time-varying Backgrounds

Taking advantage of locality in time is tackled by considering short blocks of images in the training data. Using a technique based on Combinatorial Optimization, a set of pseudo-optimal short-term background estimates is derived from these blocks of images in order to effect a degree of pre-filtering. Optimization of the new background estimates is arranged such that local temporal inconsistencies caused by moving objects are rejected in favour of spatio-temporally stable areas of the training block pixel volume. One pre-filtered background estimate is produced per block of training images, which then contributes to a conventional eigenspace model.

Although the short-term estimates still contain some foreground artifacts, the level of *contamination* by foreground objects is considerably lower than without the pre-filtering stage, and thus the resultant eigenspace model requires far fewer eigenvectors to model a similar percentage of overall image energy. The purer eigenspace model permits a tighter detection threshold, yielding a more sensitive and discriminative system overall. This approach is particularly well suited to busy surveillance scenes in which the anticipated level of background contamination is high throughout a substantial fraction of the training data.

But local *temporal* persistence is only one aspect of the time domain behaviour at a pixel. If there is regular repetition of some characteristic in the image, then the successive cycles of repetition can be seen as sharing locality in elapsed time from one cycle to the next, which amounts to locality in *periodicity*.

1.6.3 Exploiting Periodicity in Recurrent Scenes

Many road junctions are controlled by traffic lights which have a precisely controlled cycle time. The repetition frequency of activity on such a junction is likely to form a dominant part of any dynamic description of the scene. This aspect is exploited here in order to build a *predictive* model of scene activity, based on the premise that the long-term time average over the junction

cycle is *cyclostationary* in character.

By modelling the scene as a ring of 2D arrays of histograms over some appropriate feature, rare events can be detected when the content of previously unseen images contains objects or events which violate the acquired model. The method has the advantage that it may be generalized to any particular feature or combined feature set for which a Probability Density Function (PDF) may be obtained.

Model construction evidently relies on knowledge of the fundamental period over which road junction activity takes place. The period is determined by extending a method previously applied to period estimation in *gait recognition*. Here, a dissimilarity matrix based on Kullback-Leibler divergence is built for each spatial block or region in the training sequence, such that the divergence between all possible pairs of histograms in the temporal dimension is represented. Periodicity is finally estimated from structure in the autocorrelation of the dissimilarity matrix.

Possibilities exist to derive a separate periodicity for *each individual* spatial image block, or at the other extreme, a single periodicity for the whole scene, if one should exist. The latter approach is applicable in cases where the whole image is occupied by a single road junction, comprising vehicles and pedestrians all regulated by the same set of traffic signals.

Exploiting periodicity in this way is tantamount to imposing local support in the *frequency* domain. A model specifically based on periodicity is better placed to take advantage of recurrent scene behaviour in terms of detection sensitivity than existing techniques based solely on stochastic mechanisms. The approach represents one way to perform a *Dynamic Scene Decomposition*, in which normal activity is separated from both static background *and* unusual foreground, to form a 'midground', and hence a three layer model. The midground component could potentially lead to a form of spatio-temporal *activity segmentation* map as a descriptor of the scene.

1.7 Goals and Contributions of the Thesis

1.7.1 Goals

The overall objective of the work is to research and define methods to augment the performance of established background segmentation techniques by applying the general concept of local support amongst scene elements (pixels). This aspect is explored in the following 3 domains:

- **The Spatial Domain:** A method of inducing local *spatial* support based on cooccurrence of features between adjacent pixels will be formulated, developed and demonstrated. This

will show that rather than relying on spatial smoothness of some feature, a generalization to an arbitrary relationship of that feature between adjacent sites encourages a more informed foreground/background segmentation. This work is presented in Chapter 3.

- **The Temporal Domain:** A hybrid background model will be proposed, consisting of a conventional eigenspace model and a pre-filtering stage based on combinatorial optimization operating on short blocks of training frames. Temporal support will be demonstrated by compiling a single background estimate frame from each block of training frames, such that parts of the spatio-temporal volume representing movement and non-stable intensity characteristics are rejected in favour of exemplar pixels from more stable regions. Validity of the overall technique will be shown by illustrating that an eigenspace model built from the short-term estimates is more compact than one built directly from the unfiltered training data. The effects of several critical parameters relating to input sampling rate and the combinatorial optimization stage will also be determined in detail. This work is presented in Chapter 4.

- **The Frequency Domain:** A method of characterizing the spatio-temporal behaviour of periodically recurrent scenes will be developed. The model will permit detection of scene anomalies which do not fit the expected behaviour encountered in the training data. A generalized technique for estimation of the dominant fundamental repetition frequency of a scene or region from the PDF of some arbitrary feature will be detailed. A practical method for maintaining synchronization of the periodic model with scene activity even in the presence of noise will also be illustrated. This work is presented in Chapter 5.

The validity of each of the approaches described above is demonstrated experimentally with datasets drawn from real world situations.

1.7.2 Contributions

Apart from the methodological approach detailed above, along with its development and application, the following technical points are highlighted:

- Introduction of the RSLBP_4 operator to characterize local 2D intensity gradient, whilst retaining orientation information. An important feature of the operator is that it can take

on only 16 discrete symbol values, and thus a *cooccurrence* map between adjacent pixels needs only 256 bins, rendering the approach tractable from a storage point of view [108].

- Application of a cooccurrence histogram between two adjacent sites over some arbitrary feature with a limited symbol set. The map permits mutual inference between the sites so that their level of agreement can be determined, and the notion of *smoothness* can be applied in spite of the absolute symbol values being arbitrary and different. The cooccurrence prior may be used to derive arc weights between pixels in a graph cut to improve segmentation [108].

- A novel way of using a binary graph cut to implement conditional local support between image elements. Use of unequal *forward* and *reverse* arc weights between pixels permits expression of asymmetric conditional probabilities between two pixels [108].

- Use of Combinatorial Optimization to pre-filter cluttered scenes in order to remove *most* foreground objects, based on rejecting local spatio-temporal instability amongst a short block of sample frames. The short-term estimates, representing pre-filtered background data, may then be used to construct a conventional eigenspace model which is more compact than without pre-filtering [107].

- Study of important parameters involved in the Combinatorial Optimization algorithm: the number of input images per block, the input frame rate, the effect on final labelling of initial label values, the effect of constants β and λ which control the balance between the *data term* and the *pairwise interaction term* in the objective function. The study reveals that the model is not unduly sensitive to the choice of these parameters.

- Application of autocovariance of a similarity matrix to find *dominant periodicity* using symmetric Kullback-Leibler divergence as the metric instead of a directly measured parameter such as pixel intensity. This approach renders the technique independent of the chosen feature space, provided that a PDF over that space is available [110].

- A cyclic ring of histograms learned separately for each spatial sub-block of an image sequence over some chosen feature in order to model recurrent scene events and activity. The size of the ring is determined by establishing the dominant fundamental period of the data described by the histograms. The histogram set may be used to evaluate the rarity of

activity in subsequent unseen data frames, and thus highlight anomalous events. The fundamental period may be defined separately for each sub-block, or as a single global value for the whole scene [110].

- Use of a software-based Phase Locked Loop (PLL) to synchronize the local model phase accumulator to learned scene activity *after* training. The technique helps to combat the effects of noise, temporary loss of signal, drift in absolute event timing, and errors in initial periodicity estimation [109].

- A novel phase detector sub-block for the PLL based on histogram comparison. Again using the Kullback-Leibler divergence, comparing the current model with a complete cycle of recent data at all possible different phases, the optimal state counter phase may be determined [109].

1.8 Thesis Structure

The arguments of the thesis are presented in the following chapters, the breakdown of which is as follows:

- **Chapter 2** reviews literature relevant to the three proposed lines of research, and provides insight into the reasons underlying their choice.

- **Chapter 3** investigates spatial support based on local pattern cooccurrence and minimum cuts on a related graph. Experiments are performed on a monochrome dataset in which the desired foreground is fragmented and partially occluded by background. Comparison is made between this result, and what is possible without the benefit of local support, and also what is possible without the new RSLBP_4 operator. The conclusion is that the combination of techniques proposed *does* provide a measure of improvement in segmentation in such challenging circumstances.

- **Chapter 4** demonstrates the use of short-term temporal support to derive background approximations from which an eigenspace model is constructed. The experiments show how the approach can be used to advantage in a very busy road traffic scene, for which some parts of the true background are visible for less than half of the time. By comparison with an eigenspace model based directly on unfiltered data, it is shown that for a given number of eigenvectors, superior performance is achieved by the proposed combination technique.

- **Chapter 5** describes a model exhibiting support based on exploiting repetitive periodic activity. Formulation of the frequency estimation technique, and construction of the subsequent periodic model are detailed. Experimental results from three different traffic scenarios are presented, with the common goal of detecting anomalous behaviour which contradicts that generally depicted in training data. Two scenarios use a simple object aspect ratio feature, whilst the third employs object optical flow as the feature. The results of the latter experiment are particularly convincing. Further experiments show how the frequency estimation stage may be extended to yield an individual fundamental period for each spatial image block. The final experiment details use of a PLL to successfully maintain model synchronization in the presence of corrupted data.

- **Chapter 6** concludes the thesis, summarizing results and ground covered in the research, and suggests various promising directions for future studies based on the results achieved so far.

Chapter 2

Literature Review

Scene decomposition manifests itself most frequently in the literature as Background/Foreground segmentation, yet more commonly termed *Background Modelling*. A great many computer vision systems require such a scheme as a front-end or pre-processing stage in order to function correctly, especially in outdoor scenarios, and a wide variety of techniques have been proposed to achieve a segmentation suitable for the application at hand. In general the requirement is to identify and localize objects in the scene which are interesting or salient in some way compared with the normal appearance of the scene. The Background Model forms a static or dynamic characterization of the typical scene, with the intention of highlighting as Foreground those irregularities which cannot be accounted for.

General surveys of the most commonly used techniques are to be found in Piccardi [97], Radke et al. [104], and McIvor et al. [83], whilst Cheung and Kamath [17] summarize contemporary methods as applied specifically to traffic scenes. A review by Buxton [13] on the other hand outlines various methods of modelling scene activity from a behavioural point of view.

2.1 Local Representation

Whilst in general the range of raw features available is restricted to those offered by commercially viable sensors, by comparison the number of features which can be derived by subsequent numerical processing of the same data is almost without limit. Linear or non-linear transformation of the basic RGB colour space is often performed for reasons of coding and transmission, but can also be beneficial in scene modelling. For example, the colour space defined as Hue, Satura-

tion and Value (HSV) can rendered invariant to absolute intensity simply by ignoring the 'Value' component [37], whilst conversely retaining this component alone directly yields a monochrome representation of the signal. Useful results based on separating intensity from chromaticity are exemplified in work by both Matsuyama et al. [80], and Horprasert et al. [51].

Extraction of information from monochrome signals is of considerable importance, since according to [45], many surveillance video signals possess little or no significant chrominance content. Image gradient [24], optical flow [5, 77] are popular features, whilst for texture-based analysis Haar wavelets [95] and Local Binary Patterns (LBP) [93] have been proposed.

2.1.1 Heuristic Methods

Although strictly mathematical approaches to scene modelling with provably good solutions are to be highly commended, many less justifiable 'engineering approaches' have been applied to specific real-world problems, nevertheless achieving effective results. Heuristic methods substitute a simpler problem for the actual problem with a view to producing empirical results sufficiently good for some particular application.

Accordingly, the term *midground* is introduced in recent work by Valentine et al. [133], whereby objects persistent in the foreground make the transition to midground after a specified time limit. It is claimed that the human visual system is less effective in recognition the presence of objects over medium time-spans, and the technique is proposed with the *abandoned luggage* scenario in mind. Eventually midground is subsumed by the background after time elapses past a second threshold. Such thresholds are invariably a compromise and difficult to estimate in general unconstrained situations, but may work effectively enough to produce a valid algorithm in a given situation.

An adaptive block-based algorithm presented by Russell and Gong in [106] uses a priority stack by which to judge the novelty of a given intensity pattern in a block. The heuristic is to promote patterns to a higher priority as they re-occur in new images, but otherwise by default let them be displaced down towards low priority and eventually lost. The overall effect is to retain estimates of the most commonly occurring patterns, from which a per block foreground mask may be obtained via an L1 distance metric.

As is common in the case of heuristic methods, it is often necessary to set vital parameters of a system, such as offsets, factors and thresholds, to particular fixed values in order to make an algorithm work. But all too often these parameters need to be chosen manually for each

different operational scenario, unless a suitable estimation mechanism can be found. Overall, more rigorous mathematical formulations of problems, from which parameter values emerge in a natural way, are to be preferred.

2.1.2 Per Pixel Models

In the simplest pixel-based models, each pixel position in the image is processed in spatial isolation from all the other pixels. Each pixel has its own private set of model parameters, and its state is dependent solely on the history of that pixel position alone. Such models are very popular, not least because of their conceptual simplicity both in theory and implementation.

The very simplest approach is to calculate the individual pixel differences between a frame and its predecessor as described by Jain and Nagel in [58]. The resultant image, whether colour or monochrome, highlights all changes in appearance between the two frames. An equally obvious way forward is to accumulate a *mean image* either from training data, or on-line as a Moving Average (MA) filter with rectangular history window, or amnesic average as defined by Weng et al. in [137] with exponentially decreasing weight for earlier data. A mean image may be subtracted from any new frame to discover the *novel* information.

Whilst mean pixel values remain somewhat susceptible to perturbation by outliers, a more robust statistic is the median filter, which has been employed successfully by McFarlane and Schofield in [82].

Evidently some threshold must be breached if a hard binary decision on novelty is required at each pixel. A global threshold across the image will always be a compromise since abnormality for highly variant pixels will only be reached at higher levels of deviation from the mean. Hence a standard deviation, maintained separately for each pixel, is the logical solution. A pixel may then be termed foreground if it exceeds a globally selected number of standard deviations from the mean. Models based on the assumption of Gaussian distributions have been widely adopted in order to cast this problem in a more formal probabilistic manner.

The Pfinder system described by Wren et al. in [142] uses a model based on a single three dimensional Gaussian to approximate the Probability Density Function (PDF) of background colour and intensity. Thus each pixel has a three element vector representing the mean and a 3×3 covariance matrix describing the variability from that mean. The model is updated incrementally with each new frame using an exponentially weighted amnesic average [137].

Extending the technique slightly, it is possible to model the PDF as a mixture of simpler PDFs

[31]. Such a Gaussian Mixture Model (GMM) is described by Stauffer and Grimson in [125] in which multiple hypotheses are supported simultaneously by the super-position of weighted Gaussians, each with its own particular mean and covariance. In this work, a mixture of up to six Gaussians per pixel is supported, although only those with the largest weights which sum to exceed a certain threshold are actually used for evaluation. Thus the number of Gaussians in the pixel model adapts approximately to the number of hypotheses found in the data.

The GMM technique is used with three weighted Gaussians by Friedman and Russell in a traffic tracking application [38] to enable the background model to represent views of road, shadow and vehicle. Evaluation and maintenance of the model parameters, i.e. means, covariances and mixture weights, becomes quite a complex process because the class labels are completely unknown. Use of the Expectation Maximization (EM) algorithm due to Dempster et al. [27] is proposed, in which the lack of this so-called *missing information* is dealt with by an iterative process of alternately re-estimating the model parameters and the sufficient statistics until a convergence criterion is met. However, this algorithm is not guaranteed to find the globally optimum solution for any given mixture.

On-line incremental versions of EM are to be found by Nowlan in [92] and by Neal and Hinton in [89], although these only provide approximations to the original batch EM algorithm of Dempster et al. [27], since they have to track a potentially non-stationary distribution.

2.1.3 Subspace Methods and Incremental Learning

Instead of modelling the behaviour of pixels in an image sequence separately, it is possible to analyze how changes in image intensity are related globally between pixels, or further, between pixel colour channels. By performing Eigenvalue Decomposition (EVD) [127] on the covariance matrix from a set of vectorized images, the pixel interrelations may be discovered. The technique known as Principal Components Analysis (PCA), due to an original concept from Hotelling [52], and later work by Jolliffe [60] is a popular method of achieving this. The vectors comprising the most significant portion of the eigenvalue decomposition, the *principal components*, are retained to form an eigenspace model which represents the largest possible variance in the data. This forms the basis of the so-called *eigen-background* technique in which PCA is used to model the background of an image sequence.

In an application for tracking and modelling human interactions, Oliver et al. [94] describe how an eigen-background model has been implemented using simple batch mode evaluation.

The idea of adaptive thresholding when calculating the segmentation mask is also proposed, in an attempt to compensate for large shadows in the image. It is maintained in this work that the eigen-background technique is computationally more efficient than per pixel GMMs for a given level of performance.

Early work on incremental update of eigenspaces did not start with applications in computer vision, rather in statistics. One of the earliest records of incremental update of the Eigenvalue Decomposition was by Golub [44]. Further research by Bunch and Nielsen [12] added a level of robustness to the technique and developed a method for ensuring convergence. Work by DeGroat and Roberts [26] dealt with the excessive roundoff and truncation errors encountered by other methods to date in applications involving very large numbers of incremental steps.

Incremental update is important for practical implementations of subspace models because the number of initial training images is often limited. As a consequence, the resultant covariance matrix is unlikely to be full rank. An on-line application has the chance to augment the covariance matrix with new information by utilizing subsequent input frames as further exemplars as the algorithm runs. This is in addition to the obvious advantage that on-line update enables the model to track non-stationary image distributions.

One of the earliest uses of incremental learning in computer vision was by Murakami and Kumar [86] whose work involved efficient determination of eigenimages. Research into face characterization by Sirovich and Kirby [120] was one of the first applications to use the now widely-adopted *low dimensional* method in which eigenvalue decomposition of extremely large, but limited rank, matrices is avoided. Instead a smaller but equivalent matrix may be decomposed with far less computational effort to yield the eigenvalues directly, and the eigen-vectors via transformation. Use of this method proves crucial in the implementation of practical eigen-background models, and calculations performed by Chandrasekaran et al. [16] demonstrate this to be the case.

The need for robustness in incremental learning was tackled by Xu and Yuille [146]. The incremental technique in general lends itself well to robust extension because the model accumulated so far acts as an ideal *prototype* against which new data may be validated. Incorporation of the M-estimator technique, originally due to Huber [54] does permit robustness, but in Xu and Yuille's implementation a whole image is considered an outlier even if only one pixel is wayward, although other researchers subsequently solved this problem. In particular, De la Torre and

Black [72] propose an improvement to the M-estimator technique of Xu and Yuille [146] which is more tolerant of single pixel outliers. However, this is still an iterative method, thus incurring the expected computational penalties.

One of the major problems with all the methods to date is that whilst the eigenspace is allowed to evolve, the mean is prevented from doing so, thus limiting the models' usefulness as classifiers. Having recognized this, Hall et al. [46] extended the technique to permit variation in the mean.

The same work was also highly critical of the accuracy of the various incremental eigenspace methods available to date, in particular, ways of controlling eigenspace dimensionality are also called into question. Whilst Murakami and Kumar [86] always retained a fixed number of eigenvalues, the methods of Chandrasekaran et al. [16], and DeGroat and Roberts [26] allowed the eigenspace to expand or shrink by one dimension per iteration, in order to model the data with a specified degree of accuracy by retaining all eigenvalues larger than a certain threshold. In contrast, Hall et al. [46] support a strategy whereby the N largest eigenvalues are kept such that the total energy of the distribution exceeds a certain threshold, where energy is defined as the fraction of the retained eigenvalues to the total of all eigenvalues.

In separate work, Hall et al. [47] went on to detail methods for splitting and merging eigenspace models. The latter is potentially interesting in the context of incremental learning in which new data is appended to the model in small blocks rather than individually. In line with their previous work, the mean is properly dealt with in these operations.

In incremental learning experiments on face recognition, Skočaj and Leonardis [121] improve on the method of Hall et al. [46] by calculating exactly the same subspace but in a different way which includes weighting coefficients, although ultimately the robustness in their method is based on EM, an iterative technique.

By fixing the number of retained eigenvalues in a given application, Li [73] adopts a similar approach, but one which sports a much more efficient form of robust update. Using the current model as a prototype, an *influence function* is introduced, reducing the robustness problem to one of least squares once again. This directly provides a mechanism permitting robustness without the need for iteration, and thus represents an important development.

Incremental and batch mode methods are compared by Kwok and Zhao [71] in an image denoising application of hand-written digit recognition. They allude to the aforementioned mean shift problem and advocate use of a non-centred PCA method detailed by Jolliffe [60].

Work by Brand [11] concentrates on a method for incremental update of the Singular Value Decomposition (SVD) in the presence of missing or untrusted data. Favourable results are claimed in the method's use in a flow-based tracking application, along with low computational complexity compared with similar algorithms.

Although *Kernel PCA* methods have been proposed to deal with non-linearity [113], subspace models are inherently linear in nature, and the relationship between background pixels in typical image sequences may not always be captured effectively by a constrained number of eigenvectors. Contamination of the model by a highly cluttered foreground can only exacerbate the problem, but a method detailed later in Chapter 4 of the thesis addresses this particular issue.

2.1.4 Non-Parametric Models

The PDF of pixel-based models may also be represented by non-parametric techniques, whereby an approximation to the true PDF is constructed directly from training data. Interpolation to any arbitrary point in the distribution is achieved by summation over the data using some type of kernel function. This technique is also known as Parzen windows [31]. Although the kernel shape and size have to be chosen, unlike a Gaussian Mixture Model for example, there are no parameters to be estimated, and so the awkward Expectation-Maximization process is avoided.

The non-parametric approach is employed by Elgammal et al. in [33] with a multi-dimensional Gaussian kernel. To limit computational complexity, the model *training data* is based on the N most recent images. For each colour of each pixel, the kernel width is calculated at each new frame from the median absolute deviation in intensity, such that the distribution represents local image blur and not the step changes in intensity projected onto the pixel caused by occasional larger movements in the scene. For example when the pixel momentarily sees sky instead of a leaf. This work also demonstrates how normalization of colour can help to avoid the confusion caused by shadows, by isolating chromaticity from intensity for detection purposes.

The non-parametric approach is extended by Mittal and Paragios in [85] who employ a hybrid kernel density technique in a background model based on a feature vector containing two dimensional optical flow as well as normalized colour. In their PDF estimation, the kernel bandwidth is related to training data points as well as distribution estimation points.

But for non-parametric methods, density estimation can be computationally expensive if the model comprises many data points. In addition, *modes* of the distribution are implicit in a GMM, but for a non-parametric model need to be estimated, e.g. by the Mean Shift algorithm [20].

2.1.5 Region Based Features

In the per pixel models, pixels are treated completely in isolation, whilst in the eigenspace model, the combined connectivity of *all* image pixels is considered holistically. These are the two extremes of a more general paradigm in which a *number* of image pixels, perhaps a region, are treated simultaneously as a sub-image independent of all other sub-images. A possible approach to scene characterization along these lines is to split the image into square or rectangular blocks of pixels, such that the model treats the connectivity between changes in pixel values within a block individually. The reduction in holistic connectivity has two advantages. Firstly, the eigenspace model associated with each block does not have to waste eigenvectors expressing potentially irrelevant variation from other image areas. In essence, the image regions are not *bound* so tightly. Secondly, maintenance of many small eigenspace problems is more tractable according to the *curse of dimensionality* [3], since eigenvalue decomposition has a complexity $O(n^3)$ where n is the total number of pixels. Furthermore, the block-based approach is more suitable for a parallel processing implementation.

In [34] Eng et al. describe an application using a block-based background in which persons in an outdoor swimming pool are monitored for safety reasons. Here, the refraction of randomly disturbed water provides a considerable challenge for statistical modelling. Within each pixel block, a representative overall statistic is derived from training data. Firstly the vector median over time at each pixel is calculated, and then k-means clustering is applied to all the medians in the block. Foreground is obtained by evaluating the L1 norm between a new pixel and the 8-connected blocks around it using a threshold with hysteresis defined by Canny in [14].

In [80] Matsuyama et al. propose characterising $N \times N$ blocks of pixels by a Normalized Vector Distance (NVD). By rasterizing pixels in a block, a distance measure between a new block and a reference block may be measured as the distance between their respective vectors. Illumination invariance is achieved by the normalization process. Further, in the same work, a temporal cooccurrence matrix for a block is treated as a recurrence plot in order to identify periodic behaviour. The concept of the Recurrence Plot as a visualization tool is detailed by Casdagli in [15], and illustrates the evolution of a dynamical system in state-space in order to highlight cyclic behaviour by indicating repeating points at which the state-space is closely matched.

But breaking an image into spatial blocks entails an appropriate choice of block size, and a decision regarding whether to overlap the blocks, and by how much. If the blocks are not

overlapped in steps of one pixel, then treatment of the image is not homogeneous. On the other hand, single pixel overlap implies a large number of image blocks, and consequently poorer computational efficiency. Optimal choice of these parameters is likely to be highly dependent on scene content, and thus an estimation technique is necessary.

2.1.6 Saliency and Entropy Aspects

The relationship between *saliency* in an image and Background/Foreground segmentation is something of a conundrum. On the face of it, an object deserves to be classed as foreground if it is salient, and a foreground object should by definition be considered salient, so the two terms appear synonymous.

In work on spatial saliency, Kadir and Brady [61] determine an appropriate scale at which an image or image patch should be viewed in order to maximize a certain definition of saliency. Within their framework, the scale at which the *entropy* of the intensity distribution peaks is the desired correct scale, and the entropy becomes the measure of saliency. Interestingly, this scheme is not infallible, since a random pattern of black and white pixels represents a peak in entropy but would only appear *novel* at very small scales. Whilst this *Scale Saliency* algorithm is invariant under similarity transformations, later work by Kadir et al. [62] develops the idea using elliptic instead of circular sampling windows, to offer invariance under affine transformations as well. Extending the Scale Saliency algorithm to include the temporal dimension, Hung and Gong [55] define a measure of Spatio-Temporal (ST) saliency.

Motion salience in connection with surveillance applications is considered in work by Wildes [140]. Local motion in an image sequence is identified by a set of directional spatio-temporal energy filters as originally defined by Adelson and Bergen [1]. Pairs of energy components representing left/right and up/down movement are derived via a set of separable convolution kernels, before Gaussian filtering and normalization yield a measure of average opponent motion between each pair. Saliency is realized in image areas where the maximum of the horizontal and vertical imbalances between opponent pairs becomes significant over the area averaged at a particular variance of Gaussian. As such, the motion salience detected across the image is defined locally, but with respect to a scale imposed by choice of filter variance.

In the light of these works, it becomes clearer that saliency, as defined, is a measure of *local* novelty, either spatially or spatio-temporally, for which pixel history is not taken into account. So, whilst conceptually interesting, these methods may not prove directly useful with regard to

a Foreground/Background segmentation problem, in which the temporal scale needs to be much larger and non-localized. Within this context, saliency and foreground identification are not equivalent.

2.2 Spatial Correlation

Extensive work exists in the literature regarding the per pixel models [38, 125, 131] whereby a model of each pixel location is maintained independently of all others. At the opposite extreme of total connectivity between pixels, many variations on a theme have been proposed for the eigenspace type models [94, 132, 72]. Both approaches have been widely adopted by researchers and shown to be effective in many applications.

The binary mask resulting from thresholding in a given Foreground/Background segmentation is often treated subsequently by morphological operations in attempt to 'clean up' an imperfect segmentation [30]. Typically random pixels or small groups across the segmented image are highlighted as foreground, possibly due to noise, whilst evidence from most of the surrounding pixels does not agree. Similarly, within segmented foreground objects, odd pixels are still classed as background. The morphological operations of *erosion* and *dilation* can remove such inconsistencies, as shown in Figure 2.1, but two questions immediately spring to mind:

(1) Are the odd 'noise' pixels actually incorrect?

(2) Is their presence useful to the segmentation?

With regard to (1), evidently within the scope of the model, the rogue pixels are not in error, although they may have been caused by sensor noise or corruption of the signal during transmission. But if a discrepancy between the background model and a new frame of one pixel *is actually* in the observed scene for whatever reason, then it is question (2) which becomes pertinent.

For many applications, an isolated foreground pixel is not usually significant, since ideally by choice of a camera with suitable parameters, it would be arranged that even the smallest objects of interest occupy an area of many pixels in the image. At the point of quantization to a binary mask, information is inevitably lost: i.e. the degree to which a pixel's evaluated abnormality is greater or less than the threshold. A third question to be asked is therefore:

(3) Can a more informed segmentation at a pixel be achieved by taking into account the degree of abnormality of surrounding pixels?

A spatial or temporal Gaussian filter may be applied before quantization to reduce the effect

of the outliers [91], but the penalty would be loss of precision in object localization in space or time. Such a filter already actually imposes a primitive form of local support between pixels, but does not specifically by design attempt to eliminate isolated pixels or small groups from the segmentation.

On the other hand, the Linear Prediction used in Wallflower [131] automatically implies a level of *temporal* support at a pixel, and applying temporal hysteresis at a pixel adds some measure of resilience to noise as proposed by Canny in [14]. But a natural corollary is to consider the possibility of forming a *spatial clique*, whereby a given pixel's segmentation into foreground or background is influenced by *simultaneous* decisions dependent on the values of directly adjacent pixels.

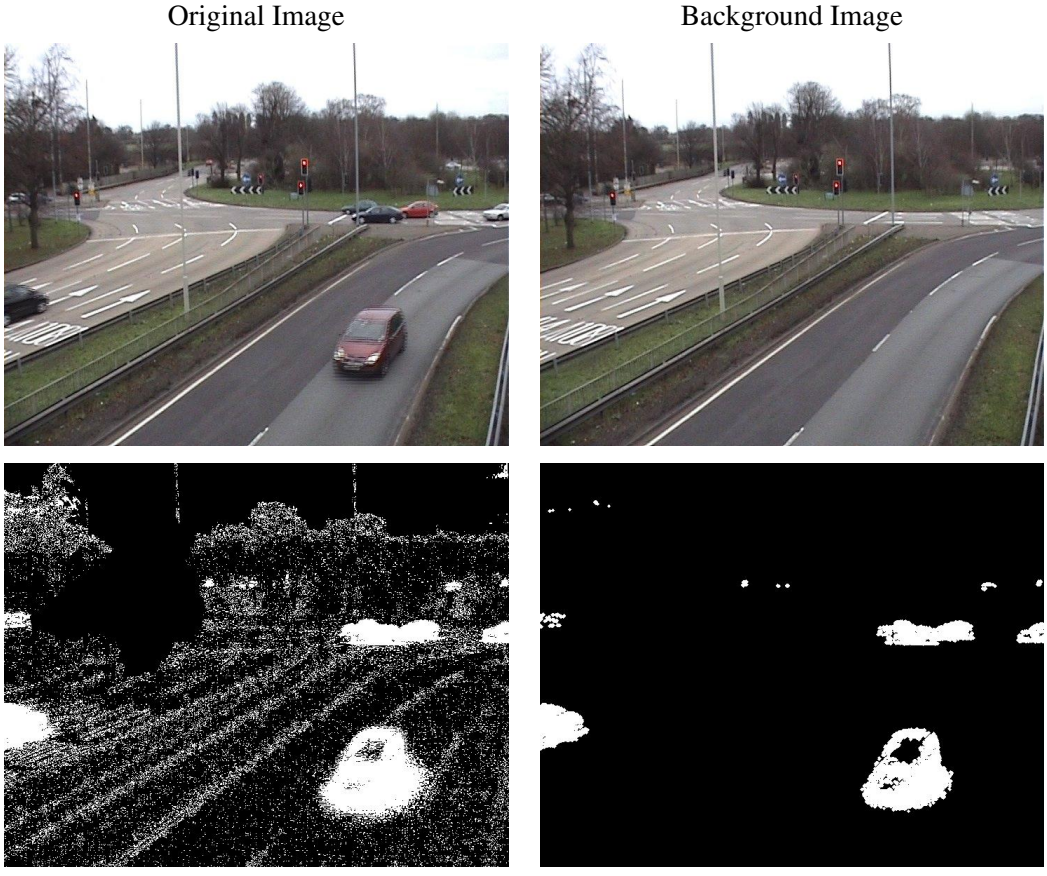


Figure 2.1: Background subtraction and morphological processing for a frame, showing how isolated pixel groups can be eliminated by *erosion* and *dilation*. But are isolated pixels actually in error or do they carry useful information? Elimination seems visually appealing, but may not always be the optimal strategy.

2.2.1 Local Binary Patterns

High entropy image content is commonly modelled as texture, as shown by Heikkilä and Pietikäinen in [50] and by Zhu, Wu and Mumford in [149]. This general approach does not encode *exact* pixel configurations, but rather *typical* patterns exemplary of the region. The LBP_8 operator described by Ojala et al. in [93] cleverly encodes a summary of local intensity gradient patterns in a 3×3 pixel block into one of ten different codewords in a way which renders it insensitive to both absolute illumination and pattern rotation. These are two crucial attributes in texture analysis. An application of LBP to background modelling is proposed by Heikkilä and Pietikäinen in [50]. Specifically, from training data they build a local histogram of the texture as represented by LBP in a circular area around each pixel. New images may subsequently be tested against the histogram to infer likelihood of background content.

2.2.2 Markov Random Fields

In their seminal paper [40], Geman and Geman introduce the effect of limiting the impact of local outliers as a form of *stochastic relaxation*. They describe a method of *image restoration* based on Markov Random Fields (MRF) with a maximum clique size of two. Small pixel groups which, according to local consensus, fail to match their surroundings, may be identified and hence corrected by an iterative *annealing* process. The outliers in this case might be intensity defects in film images, such as blotches, lines, and speckles. The clique size is the number of pixels considered to interact together to form one term of the field in optimization of the overall objective function. A pixel may form a separate clique with each of its nearest neighbours. The solution for the original image without defects is framed as a Bayesian estimation of the original image given the defective one, based on spatial correlation within cliques.

In a texture modelling application, Zhu, Wu and Mumford [149] estimate the optimal combination of features with which to approximate a training set using their Minimax Entropy principle. The Maximum Entropy principle, due to Jaynes [59], is used to determine a model representing a fusion of features resulting in a choice favouring simplicity in terms of maximum entropy. Then the Minimum Entropy principle is applied to produce a model with greater generality by incorporating sufficient features to minimize the Kullback-Leibler divergence of the model from the true distribution. Building this into their 'FRAME' model, they claim greater descriptive ability in resynthesizing various types of texture than previous MRF approaches.

More directly relevant to the removal of outliers in segmentation, in [111] Schindler and Wang also describe a framework based on Markov Random Fields in which each pixel forms a clique of size two with its 4-connected neighbours. In such a set up, the MRF is solved by determining the Foreground/Background state of each pixel, and for the special case of a maximal clique size of two [66] this may be achieved through a minimum cost cut on a graph having two terminal nodes, as well as a node for each pixel, as depicted in Figure 3.5. The cut is achieved by choosing the binary state of each pixel such that it is left joined to exactly one terminal, whilst no path is left between the two terminals. This is a particular problem in *Discrete Optimization*, which is described more generally by Cook et al. in [21], and carries with it the flavour of the desired spatial support.

Solving the above MRF for the minimum energy in terms of the cost of the partitioning cut is equivalent to finding the *most probable* Foreground/Background state for each pixel given the prior knowledge encoded in the graph's arc weights. Finding the optimal solution is closely linked to determining the *maximum flow* path through a network based on the graph.

Pioneering work in the solution of network flows was carried out by Ford and Fulkerson. Their 1956 paper [36] showed that determining the Maximum Flow from source to sink terminals of a network using the 'augmenting paths' strategy was a *polynomial time* algorithm relying on a *depth-first* type search. Furthermore, they also proved that the condition of Maximum Flow between terminals was equivalent to the problem of determining the Minimum Cut solution for the same network. Improvements and variations on the method have since been demonstrated by Edmonds and Karp [32] who proposed a *breadth-first* type graph search, and also by Dinic [29].

The augmenting paths strategy of Ford and Fulkerson is enhanced further in work by Boykov and Kolmogorov [9]. In contrast to the breadth first search, which they claim is costly in a vision application because it scans most pixels at each pass, they propose building two search trees, one rooted at each terminal. Extending, but subtly different from, the work of Dinic [29], the new algorithm re-uses the search trees instead of building them from scratch for each pass.

In terms of Foreground/Background segmentation, solution of an MRF by an appropriate graph cut algorithm is beneficial because the approach specifically *preserves* discontinuities, i.e. transitions between regions respectively labelled Foreground and Background. Given that boundaries have to appear somewhere, *energy minimization* has the effect of displacing the boundaries to the cheapest locations. If the strategy for assigning arc weights is well conceived, the result-

ing partitioning will be useful. Unavoidably, this strategy is inextricably linked to the particular chosen graph node interaction model, an example of which is the Potts model.

2.2.3 The Potts Model

In the aforementioned work by Schindler and Wang [111], a method is described whereby an MRF acts in conjunction with a conventional per pixel model using Gaussian Mixture Models. The data terms of the objective function are determined as the probability of a pixel being background if the pixel is deemed to belong to the background, or a constant if it is supposed to be foreground. Thus the notion of a *thresholding* constant does exist in a certain way.

However, the interaction terms between pixels is governed by a model originally due to Potts [102, 143], alternatively known as the Ising model [57] in the field of statistical mechanics. The model is very simple: if two adjacent pixels adopt the same (terminal) label in segmentation, zero penalty is incurred. On the other hand, if the pixels take on different labels, a *constant* penalty is charged. The effect of the interaction terms is to balance the cost of 'wrongly assigning' pixels in the segmentation against the total length of boundary perimeter between background and foreground.

Visually this appears to make objects seem more 'blob-like' with smooth boundaries. Additionally, across the image as a whole, isolated pixels and small groups are suppressed. Overall, the effect of applying MRFs to segmentation problems is to permit globally optimal solutions embracing a level of local spatial support between pixels. At first sight such an effect appears beneficial, and certainly higher level vision processes might be faster and more effective with a smaller number of more solid objects. But strictly mathematically it would seem that pixels have been ignored and object boundaries distorted for the sake of a more pleasing *visual* effect, with the prior being to minimize boundary length. With reference to questions (1) and (2) posed in Section 2.2, regardless of whether isolated pixels in the Foreground/Background mask *are* actually in error, they *have* contributed to the segmentation even if they get eliminated, but the assumed smoothness prior has not taken into account the *strength* of the correlation between adjacent pixels.

According to a method detailed in Seki et al. [116], a cooccurrence relationship is learned between adjacent image blocks based on the most significant Principal Components of their variation. Thus the possibility exists to support a change in one block by a completely different, but correlated change in its neighbour. Following a similar argument, it would seem possible to

arrange the neighbourhood interaction terms of an MRF to also take advantage of cooccurrence information, instead of just relying on the simplistic Potts model. The goal is still a 'clean' segmentation, but also a more precise one. With reference to question (3) in Section 2.2, a more informed segmentation will come from an enhanced pixel interaction penalty which can take on values *between* 0 and 1. Investigation into providing better local spatial support for Foreground/Background segmentation on this basis will be described further in Chapter 3.

This section has considered correlation due to locality in terms of the 2D image space. But temporal persistence is also a very natural phenomenon in the real world, so the next section explores the possibility of exploiting locality in time.

2.3 Short-term Spatio-temporal Correlation

The construction of a statistical scene model, either on-line or from training data, involves extraction of the most salient patterns and trends. A good model succeeds by distilling the vital aspects as well as more subtle characteristics into a compact efficient representation. Consequently the model will reflect contributions from *all* significant scene activity conveyed in the training data. But in some applications, notably surveillance, the requirement is to reliably detect, track and identify objects such as vehicles, people, and packages. Detection is achieved when an object differs significantly from the modelled scene content. The task of separating such items from more long-term image content by a statistical scene model is considerably frustrated if the model actually *includes* examples of the objects to be detected, as shown in Figure 2.2. Conversely, if a model contains no reference to an object, detection sensitivity for that object will be maximized in the sense of detection by matching new frames to that model.

Some proposed methods, for example that described by Paragios and Ramesh in [96], depend on the existence of a clean *reference frame* without foreground objects, obtained by somehow physically constraining the scene contents. But in most practical surveillance systems this is at best inconvenient, and may not be realistic at all. Furthermore, outdoor scenes are subject to considerable illumination variation, so a single reference frame would be of limited use.

2.3.1 Combinatorial Optimization

What is required is a continuous sequence of reference frames, devoid of contamination by the objects to be detected. A method is described by Cohen in [19] which attempts to eliminate

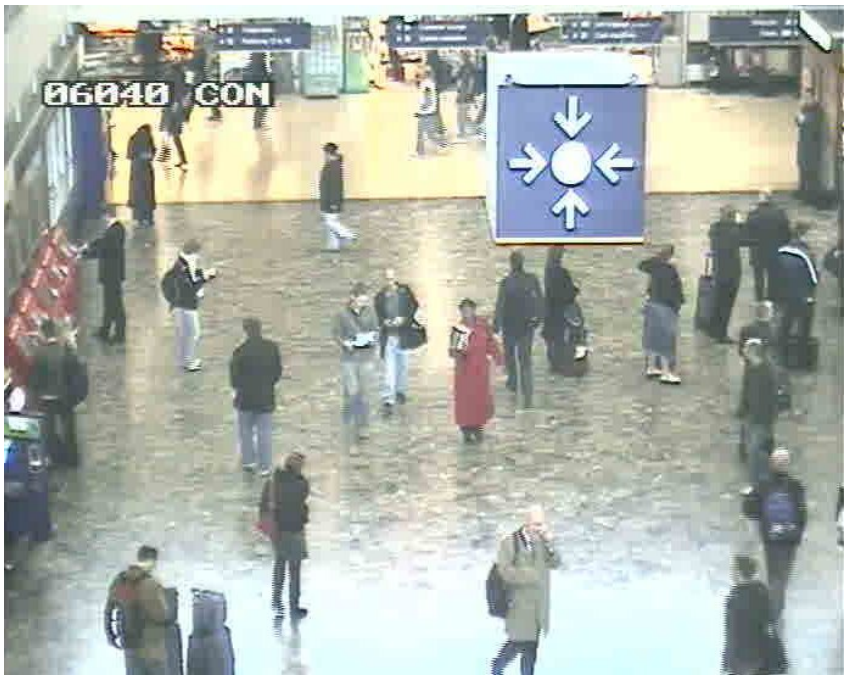


Figure 2.2: Busy station concourse scene in which it is extremely rare that all of the background is visible simultaneously. A background model which samples directly from this scene is likely to become contaminated by the very foreground objects it is trying to identify.

moving objects from a short sequence of frames to yield a single frame estimate of the stationary component.

The Cohen algorithm has been shown capable of compiling a *short-term background* image on a per pixel basis from a short block of input frames by casting the problem as an exercise in optimal labelling. Figure 2.3 shows an example of how 20 time-spaced frames from a continuously busy road junction can lead to a useful background approximation. The background is drawn from parts of any of the input frames which are found to be spatially and temporally consistent. Thus the solution comprises a set of labels or pointers, one for each pixel in the background image, specifying from which of the 20 input frames each pixel is to be taken. By employing an inventive set of cost penalty functions both temporally and spatially, boundaries between sets of pixels from different images are discouraged from occurring near moving objects, or image groups exhibiting temporal instability at a pixel.

Once again, based on solution of an MRF with maximal clique size of two, the method relies on minimum cost cuts on a graph. However, the optimization is not as simple as with the binary graph cut described in the previous section, and furthermore, the solution is only approximate.

Optimization by the Minimum Cut/Maximum Flow method of Ford and Fulkerson [36] is exact and runs in polynomial time, but pixels may only end up attached to one of *two* different terminals, e.g. Foreground and Background. But in the Cohen algorithm, each pixel may originate from one of k input frames, and so the resultant optimization problem is the minimum cut on a graph with k terminals. Much is to be found in the literature regarding the resultant multi-terminal k-cuts, but the one aspect carrying the consensus of view is that an exact solution to the problem is NP-hard, having time complexity that is exponential in k. Thus the maximum number of frames k which the Cohen algorithm can deal with is heavily constrained by available processing power, and practical systems involving large numbers of input frames cannot directly rely on the scalability of this approach in order to find longer-term background estimates based on exact k-cut solutions.

2.3.2 Multi-terminal Cuts

The concept of multi-terminal cuts is discussed at length by Dahlhaus et al. in [23] where the emphasis is on cuts on a graph with fixed non-negative edge weights. The principal application here is the minimization of communication costs in distributed parallel computing systems. From a practical point of view, Goldschmidt and Hochbaum [43] draw the assertion that for a fixed value of k, the complexity of the algorithm is reduced to being polynomial time, and some vision applications fall into the class of problems which do not demand significant scalability in k in order to be useful.

However, the algorithm described by Cohen [19] relies on a set of graph edge weights determined by the actual pixel-to-label assignments made between every clique of two neighbouring pixels. This more complicated problem is explored extensively by Boykov et al. in [10] where they introduce and compare their $\alpha\beta$-swap algorithm and their more efficient α-expansion approach to find *approximate* solutions in polynomial time.

The underlying principle is to break down the multi-terminal cut into a series of binary cuts which are individually soluble by the Ford Fulkerson algorithm [36]. From an arbitrary starting pixel labelling, the $\alpha\beta$-swap method repeatedly considers a binary cut between every pair of labels α and β from the set, allowing pixels to move between the two classes if the overall result represents a lower global labelling cost. After several iterations through all $k(k-1)/2$ pairs, an approximately optimal labelling evolves.

On the other hand, under the α-expansion model, the binary graph cut is between a nominated

label α, and all other labels together. Edge weights from a pixel to the particular α terminal in the graph are made infinite such that pixels can only migrate *into* the α class and not out of it. This time a complete iteration with each label taking on the role of α requires only k binary cuts.

As described by Kleinberg and Tardos in [66], convergence of the above algorithms is crucially dependent on the cost function between adjacent pixels conforming to the requirements of a true metric in terms of symmetry, and obeying the triangle inequality. Their work takes an in-depth look at what they have termed this *metric labelling problem*.

Kolmogorov and Zabih [67] conduct a comprehensive review and characterize various types of energy functions which *can* be minimized by graph cuts. Although they restrict the range of functions to those involving up to three binary variables, this is more than sufficient to generalize previous results.

But although the α-expansion method yields good short-term results, it still suffers from limitations of scalability in terms of the number of labels, suggesting that it can't be used directly to form long-term estimates. And in any case, a recovered background comprising parts under mutually independent lighting conditions is not useful.

Rather than trying to scale-up a multi-terminal cut algorithm by increasing the number of labels (frames) in order to achieve more long-term estimates, a potential solution may exist in which the α-expansion approximation to the multi-terminal cut is used as a *pre-processing* stage before a more conventional eigen-background model. The α-expansion produces good short-term background approximations with little contamination, whilst the eigenspace model assimilates the longer-term variations of the scene.

2.3.3 Pixel Labelling Applications

Other researchers have also used graph theory to optimize choice of pixel sets from different images. In a compositing application [25], Davis addresses the problem of creating a mosaic of a scene from multiple similar images containing moving objects. Although the bulk of this work concerns treatment of image registration issues, another objective is to produce an overall composite image in which the joins are invisible. A graph search method based on Dijkstra's *shortest path* algorithm [28] is used to find optimal paths along which to stitch adjacent images together on the basis of a cost minimization. Low cost between neighbouring pixels in the two candidate images is achieved where a good colour and intensity match is located, along with the absence of moving objects.

A similar approach is detailed by Wexler and Simakov in [138], whereby a panoramic view with minimal distortion is found by selecting vertical strips of pixels through a spatio-temporal volume according to a minimum cost criterion, again found by Dijkstra's algorithm.

A significant pixel labelling problem is that associated with 3-D stereo correspondence as described by Kolmogorov and Zabih in [68]. Here the labelling represents the offset of a pixel viewed in one image compared with its position in another. The graph cut enables a global optimization of offset on the basis of intensity matching between the pixels in the two images.

The above examples support the idea that graph theory has much to offer in terms of matching and stitching together potentially disparate scene regions.

Specifically regarding the Cohen α-expansion approach [19], the prerequisites for its success in recovery of a short-term background may be enumerated as follows:

1. That all of the required background is visible for some of the time
2. That the background is more consistently stable than any foreground pixel intensity
3. That each background pixel is time-independent

These conditions are rarely all satisfied, and the short-term estimates produced are unsuitable for direct use as background models. But crucially, the ability to assemble composite images whilst avoiding moving objects *does* provide a potential way to eliminate foreground clutter in general, and such an approach will be revisited as a pre-processing stage in Chapter 4 where, in conjunction with a standard eigenspace model, a more effective long-term background estimator is described. It will be demonstrated that the combination technique performs better than either part used on its own.

2.4 Dynamic Scene Decomposition

The techniques considered so far deal purely with the *appearance* of a scene at any given moment in time. They treat an entire scene as a set of stochastic processes which are highly correlated in the case of eigenspace models, unrelated in the case of per pixel models, or locally spatially correlated.

But although instantaneous appearance is the only observable quality available from the scene, additional information is encoded in *changes* in appearance over time. If the statistics

Figure 2.3: Example showing how Combinatorial Optimization may be used to compile a short-term background estimate from a block of frames. Left: 4 of the 20 input frames. Right: Recovered background.

of some image feature vary regularly and repeatably over time, the distribution is said to be *cyclostationary*. Analysis and characterization of cyclostationary processes is covered in detail by Gardner et al. in [39].

In the case of most models however, no assumptions are made about the temporal characteristics of behaviour of a pixel or a localized image feature, spectrally or otherwise structured. In both cases, this constitutes a waste of potentially valuable discriminative information. Behaviour of spatially distributed local features linked by adjacency in time has been addressed however, in the form of motion trajectory patterns.

2.4.1 Learning Motion Patterns

Considerable research interest has been directed towards learning motion patterns from video, since this is of immediate practical significance with regard to surveillance scenarios. The general idea is to analyze a large volume of exemplary data from a scene, such that anomalous activity may be detected, either within the existing data or occurring at a later time.

Often at the heart of such applications is a viable form of feature extraction and motion tracking to obtain object trajectories from the raw video data. To achieve this reliably is non-trivial, especially where multiple objects must be tracked without confusion through scenes exhibiting compromised lighting conditions and mutual occlusions. In any case, the body of work on acquisition and tracking is extensive, but beyond the scope of the current treatment. However, the two processes of *tracking* and *motion analysis* may be considered largely separate, and the

relevant detail for *this* thesis lies within the motion analysis and learning parts of contemporary approaches. Needless to say, algorithms which handle tracking failures gracefully are more likely to succeed, and this particular requirement is intrinsic to many recent works.

Amongst the approaches detailed in the following, some aim just to cluster together elements of spatio-temporal behaviour enabling inferences regarding cooccurrence of elements of activity. Other methods take into account *sequential* aspects of behaviour, whilst one technique attempts to attach semantic descriptors to groups of events. But none of the methods directly addresses *periodic* scene activity or attempts to exploit it, as the algorithm described in Chapter 5 does.

An approach involving spatio-temporal derivative filters is described by Pless in [99]. In this work, a scene is characterized by building a model of the intensity derivative in both spatial and temporal directions. These features are evaluated over a surrounding region for every pixel, and along with colour coordinates form a 4-D space for which single Gaussian and mixture of Gaussian models are produced. The gradient terms also permit estimation of localized optical flow using the Lucas-Kanade method [77]. These three methods are compared with an 'intensity only' model, and the Linear Prediction model from [131]. Results show that the performance of all the models is highly dependent on scale and diversity of local motion in the scene. This approach ignores all temporal aspects of scene activity.

Alternatively, Stauffer and Grimson [126] build a codebook of typical features from a large body of surveillance data. Initially they use x and y coordinates of an object's centroid along with the differentials, dx and dy, and overall object size as features. Applying vector quantization as the clustering technique, they develop a large but representative codebook of symbols from the accumulated data. Disregarding the order of symbols in a trajectory sequence, co-occurrence of the symbols is analyzed, leading to their hierarchical consolidation on the basis of probability mass functions, thus permitting discovery of a limited number of behavioural classes within the scene.

Within this paradigm, the authors claim that unusual events may be detected on two different levels. Firstly, by considering how well a new feature set matches *any* of the codebook entries, thus using the latter as a density estimator. Then secondly, by analysing the cooccurrence of codebook symbol collection from the novel event sequence after feature quantization. Thus the behaviour may be rated according to how well it matches previously learned activity classes.

In the same work, the technique is extended by augmenting the feature vector with binary

masks of motion silhouettes. Immediately the encoding of shape opens up the possibility of discriminating between object classes, regardless of motion, although at the expense of a very large feature vector. The authors describe experiments using a codebook containing as many as 400 different quantized prototypes.

Useful results on relatively small datasets have been achieved by Swears, Hoogs, and Perera in [129] using Hidden Markov Model (HMM) techniques to model object position and velocity in an aerial traffic monitoring application. Crucially, their approach explicitly deals with the *sequence* of object trajectory points by constraining certain entries in the HMM transition matrix in order to preclude return to a previous state. Observation parameters are learned such that a track is modelled piecewise by a series of elongated 2D Gaussian distributions. The authors have developed a set of HMM manipulation rules, including cluster creation, extension, observation testing, and track stealing, which act in an on-line fashion to accumulate a potentially cluttered scene of tracks into a collection reflecting normal behaviour. Automatic model order selection is achieved dynamically for each HMM according to the Akaike Information Criterion (AIC) [145].

Chains of Gaussian distributions are also used by Hu et al. in [53], and crucially in their algorithm too, the temporal aspect of trajectories is specifically modelled. As with other works, object position, velocity, and size form the feature vector, but during training, all trajectories are resampled to a constant vector length before a two level clustering scheme using a fuzzy K-means algorithm. Thus the *sequence* of points is represented in a high dimensional space, and the trajectories may be clustered firstly on the basis of position, and secondly on the basis of temporal order, before construction of the Gaussians depicting image tracks. The result is a model which is more sensitive to event order than is possible in approaches which do not model temporal information directly, but which rely solely on an object's directional velocity to support sequential order. There is certain similarity to the approach of Swears et al. [129] in that the final chain of Gaussians mimics the behaviour at the output of a uni-directional HMM.

The method adopted by Zhong, Shi and Visontai in [148] also relies on cooccurrence of features, but this time *without* specifically tracking individual objects. In their work, based on techniques from *information retrieval*, they build coarse spatio-temporal histograms from extensive video data, based on simple image features - colour intensity and motion information computed directly with a temporal Gaussian derivative filter. Following vector quantization into a dictio-

nary (codebook) of prototypes, just as in [126], the body of video data is split into short blocks. The equivalent of a *document-keyword* matrix is then constructed treating temporal blocks of video as documents, and prototypes within them as the keywords. The motion pattern learning task is then cast as a problem in information retrieval, relying on techniques from graph theory. The objective is to discover *concepts* by forming links between *informative* keywords, whilst ignoring the effects of common and random keywords. Computing such a correspondence relationship is non-trivial, and the authors introduce *transitive closure* and a *co-embedding* technique in their solution to the problem. An important aspect of the approach is *inferred similarity*, which permits expression of relationships between video blocks via similar prototypes. Unusual events in this context are defined as clusters which are spatially isolated in the embedding space. This method is attractive because the simplicity of chosen features makes it less dependent on specific scene characteristics. However, part of its ethos is explicitly avoiding temporal constraint of the prototypes.

Construction of a *semantic* scene model is the central theme of work by Makris and Ellis [78] in which the scene is modelled separately in both *topographical* and *topological* structures. The former represents image locations in terms of real-world ground plane coordinates, facilitating easier handling of object motion in perspective views, and opening up the possibility of data fusion from multiple cameras. The feature vector consists of object coordinates and velocities derived by Kalman filtering. Following accumulation of object trajectories, Expectation Maximisation (EM) is employed to express the distribution as a Gaussian Mixture Model (GMM). A rule-based scheme enables clustering into specific scene elements, such as entry/exit points, routes, junctions and stop zones. The topological map describes their interconnection in a graphical way, such that a Bayesian Belief Network may be constructed to make inferences about object behaviour. The compact nature of the graphical model permits *conceptual* queries about the data much more readily than the raw data would allow, representing a valuable feature in a surveillance context. Although this semantic approach is likely to be very powerful in many situations, still any intrinsic periodicity in the scene is ignored.

Object *saliency* is the key to irregularity detection in work by Boiman and Irani [7]. Within their philosophy, an object which can't be explained, or *exemplified*, by similar partial supporting object views from a database is considered to be salient, and thus an anomaly. A feature vector is formed from the temporal image gradient at all spatial pixel locations in a spatio-temporal

patch of video, and a database is built from many spatio-temporal patches at multiple scales throughout training data. A query block of video may be supported wholly, or in part, if it can be constructed from contiguous groups of patches from the database. The inference mechanism imposes geometrical restriction on the location and scale of candidate spatio-temporal patches from the database, with a view to inducing support over as large a spatio-temporal extent as possible. Such large areas then constitute a match, whereas evidence fragmented within the database affords less support for the normality of the query. The inference algorithm is based on a Bayesian network with belief propagation by message passing, but in addition, image and database topology is exploited to achieve tractable comparisons by progressive elimination of the search space. The emphasis of their algorithm is from the standpoint of *construction* of the query rather than its local dissimilarity from model data as with other approaches, and the *generalization* properties of the technique with regard to exact object pose and configuration are ensured by the wide variety of database patches and scales. Constraints in both sequentiality and absolute temporal distance are supported by this model, making it closest to the subject pursued in Chapter 5 of all techniques described so far.

2.4.2 Spatially Supported Linear Prediction

On the other hand, the per pixel Linear Prediction algorithms already described by Toyama et al. [131] rely solely on previous pixel values in order to anticipate and classify future ones. Such a method is likely to work best on pixel signals which exhibit *cyclostationarity*, whereby temporally cyclic behaviour content at a constant frequency is present. Here, a Wiener filter [139] forms a prediction of pixel intensity from a weighted sum of previous intensity values. The tap weights are calculated from the covariance of past values, where the goal is to achieve Minimum Mean Squared Error (MMSE) in the prediction. Experiments by Toyama et al. in [131] successfully demonstrated application of the approach to both indoor and outdoor scenes, using a 30 tap Wiener filter at a 4Hz frame rate. Evidently, the lowest frequency component that the predicted signal could exhibit is 7.5 seconds, which limits the range of effects that the model can express.

Integrating spatial support *and* temporal support in the form of Linear Prediction, Szummer and Picard [130] describe a method of modelling moving water, flames, and swaying trees as *temporal textures*. An Auto Regressive (AR) model is proposed in which a new frame may be synthesized such that each pixel is described by a weighted sum of previous versions of itself

and its neighbours, with an additive Gaussian noise process.

In work on Dynamic Textures, Soatto et al. [122] also describe a 2-D Linear Predictive algorithm based on the concept of System (Transfer Function) Identification from signal processing. An Auto Regressive Moving Average (ARMA) model is proposed whose coefficients are estimated by the EM technique of Dempster et al. [27] in the general case, although a closed-form solution is presented for simpler second order stationary processes.

Liu and Picard [74] describe scene dynamics in terms of the Wold decomposition of the 1-D temporal signals derived from each image pixel, giving rise to deterministic (periodic) and non-deterministic (stochastic) components. Background recovery is performed by a median filter of length 11, and a 1-D Fourier transform yields the temporal frequency spectrum at a pixel. A measure of temporal periodicity is defined here as the *ratio* between the harmonic energy and the total energy along a temporal line. Harmonic peaks are identified according to a method in their previous work [75] on the 2-D Wold decomposition for texture analysis. Experiments show the overall algorithm capable of distinction between various human and animal gaits, and other types of motion.

In general, there are various techniques described in the literature for determining sinusoids in arbitrary 1-D signals. Several of them exploit the orthogonality between the sinusoids and noise, by considering separate component and noise subspaces. Eigen-decomposition of the signal autocorrelation matrix then yields eigenvalues and vectors from which the sinusoidal components may be derived. The catch with these methods is that the number of sinusoids sought in a given signal must be known a priori. Pisarenko's harmonic decomposition [98] is one of the simplest approaches, but is rather sensitive, hence mainly of theoretical interest. The more robust MUSIC (MUltiple SIgnal Classification) first described by Schmidt [112] and its derivative root-MUSIC also test for a known number of sinusoids. Detailed description of these subspace algorithms is given by Hayes in [49]. Whether such methods may easily be generalized to multi-dimensional video signals is unclear. The general problem of frequency estimation and frequency tracking is dealt with at length by Quinn and Hannan [103].

The per pixel Waviz algorithm described by Porikli and Wren [141] exploits cyclostationarity of the signal. Here, the Short-Term Fourier Transform (STFT) is used to determine the spectral content of a pixel's behaviour within a recent time window. A set of coefficient magnitudes is accumulated from overlapping temporal blocks by an exponentially forgetting filter. Their simi-

lar, but later work [101] describing the Waveback algorithm, used the Discrete Cosine Transform (DCT) instead, claiming its superior low frequency performance. In both cases the coefficients of the most recent block may be tested against the accumulated coefficients using the L2 norm to determine novelty. Consideration of the STFT window size and filter time constant is important, but in spite of this, good results were achieved on a scene containing agitated vegetation. The authors claim a higher *specificity* compared with competing methods that do not take into account the temporal aspect.

Whilst the techniques described above certainly take into account periodicity in an image sequence, they have all used pixel intensity directly as the feature. Although descriptive enough for some applications, a more general approach operating on the *distribution* over some arbitrary feature would be considerably more flexible. It is also not clear how well the linear predictive approaches would scale to scene events occurring over a longer time-span. Both of these aspects are addressed in Chapter 5 of this thesis.

2.4.3 Perceptual Grouping

Considerable work has been published on the biological aspects of perceptual grouping [87, 123]. In terms of the human visual system this amounts to forming relationships between objects in an image. But such grouping also occurs in the temporal dimension, whereby human attention is drawn to objects whose appearances change together, and those whose appearance changes cyclically or periodically. At this point it is important to make the distinction between these two types of variation: Cyclic motion implies events repeating in a certain sequence, whereas Periodic motion involves events associated strictly with a constant time interval.

Within the field of biologically inspired computing, systems using networks of Spiking RBF (Radial Basis Function) Neurons have been used by Natschläger and Ruf in [88] to characterize and identify spatio-temporal behaviour patterns. Such a neuron generates a pulse of activity when the combination of its inputs reaches a critical threshold. The network of connections from input neurons to output neurons contains groups of parallel paths with varying synaptic delays whose relative weights are learned in a Hebbian fashion such that the delay pattern eventually complements (mirrors) the times between events in training data. By this mechanism, an output neuron can *learn* to fire when the appropriate events occur with correctly matched time delays, since only under this condition will all spikes reach the nucleus simultaneously, causing its threshold to be breached and hence firing it.

This idea is applied to a practical vision system by Ng and Gong in [90], whereby relations between pixels in the Motion History Image (MHI) over a sequence are learned for a simple shopkeeper/customer scenario. Abnormal behaviour is detected when a customer takes an item of stock but leaves the shop without paying the shopkeeper, thus violating the normal sequence of events. Similarly using MHI, in [6] Bobick and Davis discriminate between actions based on movement of the human body by matching against various learned templates. But these examples only identify sequences of learned events occurring at precise relative times, whereas overall the sequences themselves are asynchronous events - they might happen only once, or repeatedly but at arbitrary times.

A model described by Xiang and Gong in [144] forms relations between *asynchronous* but related scene events by dynamically adding links between parallel Hidden Markov Models, making it ideal for many situations where temporal invariance is paramount. A complicated airport scenario is analyzed, whereby the service vehicles and personnel attending a docked aircraft have to function within a constrained order.

Whilst the techniques just described have the flexibility to model sequences of events, they do not attempt to explicitly model periodic behaviour. In fact they deliberately avoid reliance on absolute periodicity, because in many scenarios it is inappropriate. But for other types of scene, periodicity *is* a dominant behavioural aspect, and therefore research towards a suitable model is motivated.

2.4.4 Relation to Gait Analysis

On an apparently unrelated problem, much is to be found in the literature concerning gait characterization, modelling and identification. Generally these methods work by analyzing the relative motion of linked body members, which are of course all related by the same fundamental frequency. However, the parallel between this and modelling traffic at a road junction is surprisingly close. Given extracted features, image areas may be likened to body limbs in that they will likely share fundamental frequency, but be of arbitrary phase and harmonic content.

Various forms of periodic human motion are characterized by Polana and Nelson in [100] by tracking candidate objects and forming their *reference curves*. After evaluating a dominant spectral component if it exists, an appropriate temporal scale is identified. This idea is further developed by Cutler and Davis in [22], who also consider periodic self-similarity, Fisher's Test for periodicity [35], and Time Frequency Analysis [18]. Meanwhile the previously mentioned

Recurrence Plot described by Casdagli in [15] is a useful tool for visualizing the evolution of gait-like processes in state-space, showing specifically when the state revisits a previous location.

Seitz and Dyer [115] additionally consider the possibility that a given cyclic process, such as human gait, may not be entirely self-similar throughout its cycle due to temporal irregularity. A time-warping function is proposed which allows temporal contraction and dilation throughout the cycle provided that the *sequence* of the constituent changes is preserved. According to their formulation it is not clear if it is always possible to solve for or automatically learn the optimal warping function.

2.4.5 Phase Locked Loops

Instead of using Fourier analysis directly, in [8] Boyd employs Phase Locked Loops (PLLs) to discriminate between different gaits, on the basis that it is more efficient. The n-point Discrete Fourier Transform (DFT) is necessarily a block process requiring n recent samples and having $O(nlogn)$ complexity, whereas the PLL is a causal system for which simple update is performed at the arrival of each new sample. The finer *granularity* of the latter process is considerably more amenable to on-line applications.

According to Boyd's method [8], having identified some fundamental frequency for an object (person), application of a PLL per pixel in the relevant image area permits estimation of the magnitude and relative phase of this fundamental component for each pixel making up the object. The idea is that the *phase signature* for every object (person) will be different. The technique is rendered scale and translation invariant by matching these parameters as *shapes* in the complex plane using the Procrustes mean [79].

The PLL is a building block used extensively in electronics and communication in a wide variety of applications. Principally it is a servo loop which acts as a low pass filter for cyclic processes, constraining rate of change of frequency. Crucially, a local oscillator is synchronized in both frequency and phase to a single frequency in an incoming signal by the action of the feedback loop. When in the 'locked' condition, the local oscillator can *track* changes in the incoming frequency with a dynamic performance dictated by the loop filter transfer function, whilst exhibiting considerable robustness to noise. These particular qualities of the PLL will be harnessed to great advantage in Chapter 5 in order to permit a learned model of periodic scene behaviour to automatically maintain synchronization with the scene that it relates to. An extensive treatment of the design and analysis of PLLs is given by Best in [4].

2.5 Summary

The preceding discussion has covered essential techniques and works in the literature regarding scene decomposition, and in particular Foreground/Background segmentation. The breadth of techniques proposed in the literature is considerable, yet in general, approaches involving per pixel Gaussian mixture models, eigenspace models, non-parametric kernel-based models, and Linear Prediction, appear to dominate in the sphere of practical solutions.

In pursuit of an effective scene decomposition strategy, the following lines of research will be pursued in subsequent chapters:

1. In existing techniques Markov Random Fields have been used to induce local support in segmentation, but support based on mutual conditional probability between neighbours does *not* seem to have been explored extensively. As such, a pattern-based spatial support technique utilizing a simple but novel LBP operator and binary graph cuts will be proposed in Chapter 3.

2. Several algorithms have used graph cuts to optimally 'compile' one image from a number others with some particular goal in mind. To use such an approach to formulate a *pre-processor* for removing much scene clutter, upstream of a more conventional background model seems novel. Thus, a short-term spatio-temporal support technique using approximate minimum multi-terminal graph cuts followed by an eigenspace model will be investigated in Chapter 4.

3. Considerable work in the literature describes various approaches for exploiting temporal periodicity - mostly in the area of synthesis, prediction or gait characterization. But little is to be found about seeking the dominant global periodicity present in some particular types of scene, and how it might be used to advantage. Therefore, a spatio-temporal decomposition scheme for recurrent periodic scenes exhibiting a single dominant period, or multiple periodicities, is proposed in Chapter 5.

Chapter 3

Pattern-based Background Identification

Segmentation of an image into foreground (FG) and background (BG) can be considered as a pixel classification task which is generally performed on the basis of colour intensity measurement evidence from the pixel itself. However, when considering an image pixel in isolation, as a sample from an independent statistical process, there is only limited scope for validating the classification decision. Clearly, there is a strong possibility that neighbouring pixels, both in space and time, will exhibit similar or at least related behaviour. Evaluation of a pixel within the context of its surroundings to improve classification reliability is thus motivated. This chapter investigates a method of exploiting local *spatial* support which is novel in two particular ways. Firstly, a simplified Local Binary Pattern (LBP) operator characterizes a pixel's immediate neighbourhood, and secondly, a graph cut utilizes mutual conditional support between adjacent pixels based on this new operator to induce optimal segmentation. Although the focus here is on *spatial* interaction, the resulting model is nevertheless intended to represent *dynamic* backgrounds.

3.1 Scope of the Problem

The focus of this chapter is to tackle the challenging problem of modelling highly textured non-stationary backgrounds, and in particular, segmenting people moving amongst dense non-stationary trees and foliage excited by the wind as shown in Figure 3.1. Traditionally this has been a difficult problem to solve effectively due to the highly chaotic nature of such image areas containing branches and leaves. The high information content, or *entropy*, of patterns encountered and their temporal behaviour make them inherently incompressible and thus hard to model

compactly. In addition, it is not uncommon in such scenes to find that background is in general scattered amongst foreground, i.e. it can be literally *behind* and *in front* of foreground objects.

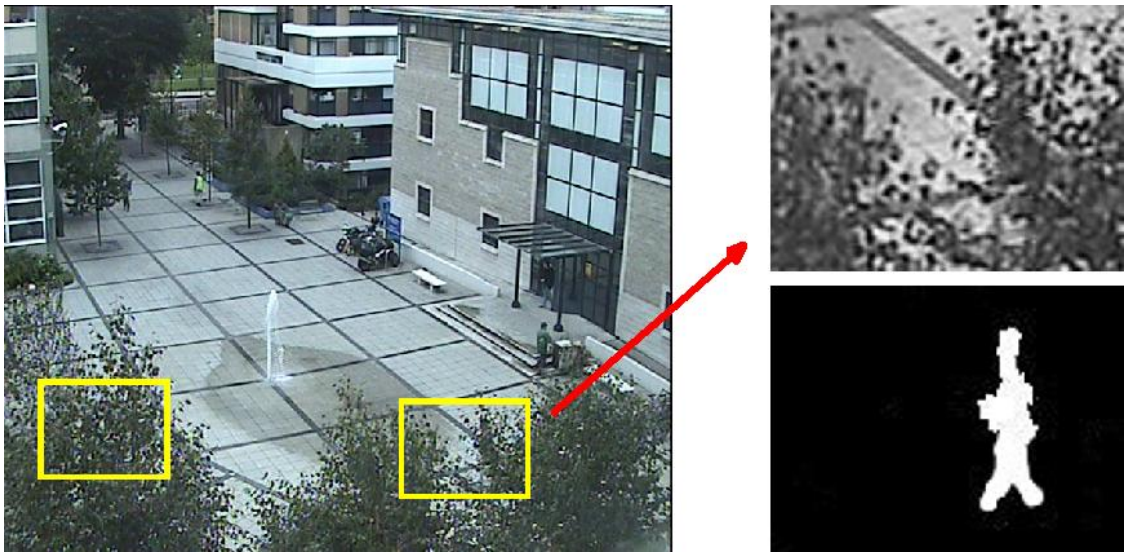


Figure 3.1: The challenge of pattern-based segmentation is to identify unusual objects amongst a highly cluttered background. An area of dense vegetation, shown enlarged in the top right image, largely obscures the view of a person shown as ground truth in the bottom right image.

3.1.1 High Entropy Scenes

Many typical scenes contain areas of high inherent complexity such as specular reflection from disturbed water and chaotic occlusion and appearance variation of vegetation moving under the influence of air flow. From the standpoint of information theory these represent *high entropy* sources [55], whilst signal processing tends to consider the effect spectrally, and refers to sources emitting *wideband noise*. In single frames, the chaos is a spatial property manifesting itself as texture, whilst in video such stochastic variation may occur temporally as well. Exact modelling of the precise characteristics of intensity over time in a high entropy image area is by definition almost impossible: the information is highly incompressible.

From a foreground/background detection point of view the goal is to highlight unusual state or behaviour of the objects in view, which might for example entail a person walking in front of a tree in leaf, or perhaps passing *behind* it, causing partial occlusion due to the *background* now being in front of the person of interest. In both cases, the requirement is to identify some less common pixel intensity configurations amongst a potentially broad range of common ones. Occlusion of the foreground, as in the second case, merely compounds the detection problem

by fragmenting the available useful evidence. A solution is sought which maximizes use of the available evidence in order to achieve useful segmentation.

3.1.2 Importance of Local Support

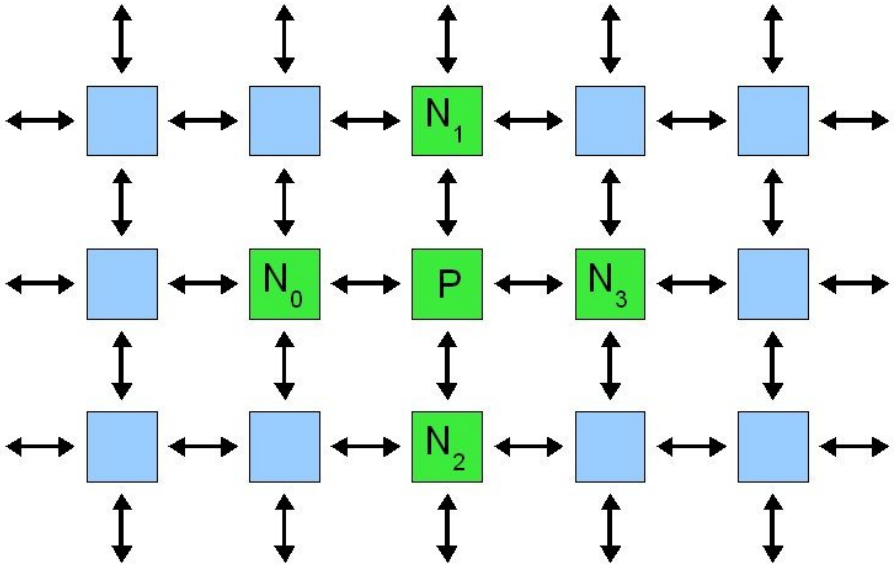


Figure 3.2: Diagram showing how local support may be applied to a pixel P within its 4-connected neighbourhood $N_0 \ldots N_3$. Interconnecting arrows couple mutually supported pixels.

Intuitively, allowing adjacent pixels to provide local support for each other seems highly reasonable. Equally, to *deny* the possibility of support could be seen as a *waste* of the available scene information encoded mutually between pixels. A possible scheme for local linkage with 4-connectivity is shown in Figure 3.2.

With subspace techniques as employed in [94, 73], the eigenvectors of image covariance represent linkage of pixel variations across the entire scene, and are thus inefficient at capturing independent *local* stochastic processes. Connectivity in the temporal dimension as exploited by Linear Prediction [131] is also likely to be ineffective due to the lack of cyclic components of intensity at a pixel.

On the other hand, the Gaussian Mixture Model (GMM) [125] has been shown highly effective when it comes to acquiring and adapting to the statistical characteristics of behaviour at a pixel. However, high variance (or covariance for a colour image) inevitably implies low selectivity for a Gaussian component, so unless the spread of common pixel values is confined to several narrow modes, there is the danger that a foreground object will fail to be detected reliably. In addition, the GMM is at the opposite extreme from the subspace model when it comes to connec-

tivity: in general it offers no mechanism for support regarding foreground/background decisions between pixels, either local or global.

3.1.3 Proposed Solution

A further additional requirement for a background model intended for outdoor use is that it must not be adversely affected by changes in scene illumination with regard to both intensity and chromaticity, although in some implementations [51] constancy of the latter is used to mitigate the effect of false positives caused by shadows.

From all the aforementioned observations it becomes apparent that a candidate solution should satisfy the following criteria:

1. Encode local pixel patterns
2. Provide local support among pixels
3. Have a probabilistic basis
4. Exhibit resilience to lighting variations
5. Be efficient in implementation

To this end, a solution which embraces three important aspects is proposed. Firstly, a *rotationally variant* simplification of the LBP_8 operator used as the image feature reduces susceptibility to illumination changes and provides an initial level of pattern sensitivity. Secondly, a cooccurrence map representing mutual conditional probabilities between adjacent pairs of pixel configurations lends local support to the foreground/background segmentation decisions, encoding a further degree of pattern dependence. Finally, the array of image pixels is treated similarly to a Markov Random Field (MRF), and an optimal realization of the segmentation in terms of pixel labelling in a combinatorial sense is arrived at by a minimum cut on a related graph. Experiments on a challenging dataset involving objects heavily obscured by tree branches demonstrate the advantage of this approach.

3.2 Rotationally Specific LBP_4

High entropy image content is commonly modelled as texture [149]. This general approach does not encode *exact* pixel configurations, rather it encodes *typical* patterns exemplary of the region.

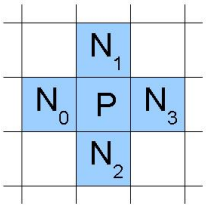
The LBP$_8$ operator described in [93] cleverly encodes a summary of patterns in a 3×3 pixel block into one of ten different codewords in a way which renders it insensitive to both absolute illumination and pattern rotation. These are both crucial attributes in texture analysis. Using such a scheme, segmentation on the basis of texture may be achieved by identifying regions with a similar probability distribution over the ten possible codewords as demonstrated by Heikkilä and Pietikäinen in [50]. Furthermore, the limited range of codewords is beneficial with regard to storage space, and facilitates adequate population of a histogram with only a modestly sized training set.

But the requirement for foreground/background segmentation is different. The interest here is not in regional *texture* statistics, but instead in *absolute pattern* statistics at a pixel, and furthermore, rotational invariance is not only unnecessary, but a hindrance with regard to the modelling requirement. Such a state of affairs leads naturally to the concept of a *Rotationally Specific* Local Binary Pattern (RSLBP) operator for grayscale images, obtained by simplifying LBP$_8$. As shown in Figure 3.3 the value of the RSLBP$_4$ operator at a pixel is given by subtracting the intensity value of the centre pixel from each of its 4-connected neighbours. The sign of the result of each subtraction contributes a single bit to form a 4 bit codeword. The spatial mapping from neighbour to bit position is immaterial as long as it is applied consistently throughout. This rotationally specific texture feature is quick and simple to compute, and yields a compact characterization of two-dimensional image gradient at a pixel fit for the current purpose.

Application of the RSLBP$_4$ operator, denoted here by $R(\cdot)$, to an image produces a symbol $S_r = \{0 \ldots 15\}$ at pixel location r. By considering the 16 bin histogram of these symbols at each pixel (x,y) over a set of K training frames I_k^T of size $M \times N$, where $k = \{1 \ldots K\}$, an estimate of Probability Density Function (PDF) representing pixel configuration over this feature is obtained:

$$p(r = S_r|x,y) = \frac{1}{K}\sum_{k=1}^{K} u \quad \text{where} \quad u = \begin{cases} 1 & \text{if } R(I^T_{x,y,k}) = S_r \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

A query image I^Q may be tested against this simply by evaluating the RSLBP$_4$ operator at every pixel and obtaining the appropriate probability from the histogram, which in turn is tested against a threshold to yield a rudimentary foreground/background segmentation.



$$RSLBP_4 = \sum_{i=0}^{3} (N_i > P).2^i$$

Figure 3.3: Kernel for the new $RSLBP_4$ operator: a 4 bit word is composed from the boolean results of thresholding the intensities of 4-connected neighbours against that of the centre pixel.

3.2.1 Cooccurrence Matrix

In order to provide local support between pixels, the training data is also used to build a cooccurrence matrix between every adjacent pair of 4-connected pixels both horizontally and vertically in the image. This two-dimensional histogram represents the joint probability of two separate $RSLBP_4$ symbols occurring simultaneously at the two adjacent locations. Although conceptually, cooccurrence between pixels horizontally and vertically is the same, from an implementation point of view it is preferable to consider it as two separate arrays, C_h of size $(M-1)\times N\times 16\times 16$ elements, and C_v of size $M\times(N-1)\times 16\times 16$ elements given by

$$C_h(x,y,i,j) = \frac{1}{K}\sum_{k=1}^{K} u \quad \text{where} \quad u = \begin{cases} 1 & \text{if } R(I^T_{x,y,k}) = i \;\&\; R(I^T_{x+1,y,k}) = j \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

$$C_v(x,y,i,j) = \frac{1}{K}\sum_{k=1}^{K} u \quad \text{where} \quad u = \begin{cases} 1 & \text{if } R(I^T_{x,y,k}) = i \;\&\; R(I^T_{x,y+1,k}) = j \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

at location (x,y) where $i,j = \{0,1\ldots 15\}$, $R(\cdot)$ is the $RSLBP_4$ operator, and I^T_k $k = \{1,2\ldots K\}$ is the training set. The cooccurrence matrices at each pixel are normalized to the number of training samples K such that they correctly reflect the joint PDF.

Now consider two horizontally adjacent pixels r and s in a query image I^Q at positions (x_r,y_r) and (x_r+1,y_r), having $RSLBP_4$ symbols S_r and S_s respectively. If on the basis of information from the training data solely about pixel r, it is decided that r is background, then a conditional probability of symbols over pixel s may be obtained from $C_h(x_r,y_r,S_r,S_s)$ due to the cooccurrence relationship. But in order for this to be a valid probability, normalization of C_h over its last dimension is required such that the conditional probability of s given r is:

$$p(s = S_s | r = S_r, x_r, y_r) = \frac{C_h(x_r, y_r, S_r, S_s)}{\sum_j C_h(x_r, y_r, S_r, j)} \tag{3.4}$$

However, the relationship between r and s is symmetrical, so if s were known to be background then the conditional probability over r comes from a similar expression. It is important to note that the normalization constant in the denominator must be obtained by summing along the *third* dimension of C_h this time:

$$p(r = S_r | s = S_s, x_r, y_r) = \frac{C_h(x_r, y_r, S_r, S_s)}{\sum_i C_h(x_r, y_r, i, S_s)} \tag{3.5}$$

The denominator in Equations (3.4) and (3.5) represents the *marginal probability* of symbol occurrence at a location (x, y), as illustrated in Figure 3.4. For pixels r and s which are *vertically* adjacent in I^Q at positions (x_r, y_r) and $(x_r, y_r + 1)$, again taking on RSLPB_4 symbols S_r and S_s respectively, the relative conditional probabilities are obtained as in Equations (3.4) and (3.5), but this time with reference to cooccurrence array C_v.

It becomes apparent that these mutually dependent results cannot be acted on sequentially, especially when it is remembered that a pixel is potentially supported by four neighbours. For any given query image there will be a *global* combination of foreground/background decisions amongst the pixels, i.e. a segmentation by pixel labelling, such that the labelling process is made optimal according to the localized support measure introduced above. Finding the optimal labelling of all pixels in a scene is then reduced to an exercise in Combinatorial Optimization, for which a solution may be sought through graph cut techniques.

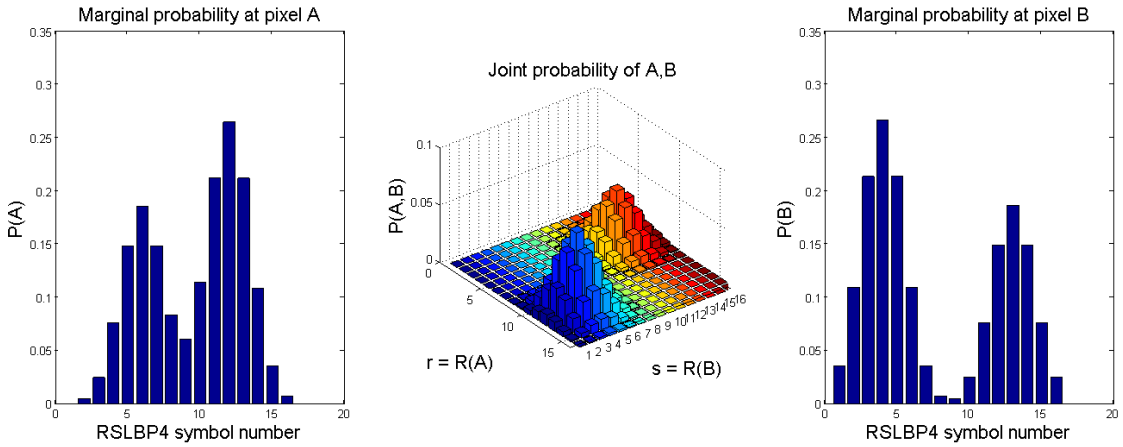


Figure 3.4: Example marginal and joint distributions for arbitrary adjacent pixels A,B using the new RSLBP_4 operator $R(\cdot)$ depicted in Figure 3.3.

3.3 Combinatorial Optimization

The problem of choosing a label for each pixel in an image from a finite set of labels according to a set of penalty expressions is the essence of discrete optimization. The objective is to separate the pixels according to their labels in the configuration which incurs the least penalty. If the penalty criteria are suitably designed, the optimal separation is useful in some way.

The labelling of pixels from a discrete set is directly equivalent to making a cut on a graph consisting of vertices and edges as shown in Figure 3.5. In such a graph there is a vertex for each pixel, and a special *terminal* vertex representing each element of the label set. Every pixel node is coupled by an edge to every terminal, but edges also exist between the pixels to represent their interdependencies. According to a scheme of penalties, every edge is assigned a weight determined by the cost of cutting that edge. The optimal solution is obtained by cutting enough edges to leave every pixel connected to exactly one terminal, thereby taking on that terminal's label, and yielding the combination of pixel to terminal assignments which gives the minimum cost cut of the graph, and hence the overall problem solution.

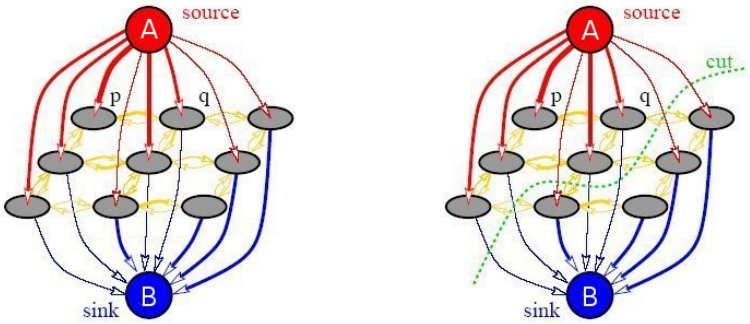


Figure 3.5: Graph for an array of only 9 pixels: *source* and *sink* nodes represent the two classes *A* and *B*. A cut must separate *A* and *B*: the MinCut/MaxFlow algorithm finds the cheapest. A practical graph contains a node for *every* image pixel. Figure taken from [9].

It was shown in [36] that for the special case of two labels, an optimal solution can be obtained in polynomial time using the Minimum Cut/Maximum Flow (MinCut/MaxFlow) algorithm. Fortunately the current foreground/background segmentation is just such a binary problem. Segmentation into more regions than this is potentially interesting, but the multi-way cut has been shown to be NP-hard [23], although [10] describes a way of achieving a *local* energy

minimum within a constant factor of the *global* minimum by their alpha expansion algorithm. At any rate, it is not yet clear exactly how to exploit the multi-way cut in the context of the current problem.

3.4 Inducing Local Support by Graph Cut

The graph cut problem has much in common with the solution of Bayesian networks and Markov Random Fields (MRF) [40], whereby a realization of the field encompasses the interdependencies of the nodes. A method described by Schindler and Wang in [111] demonstrates how local support can be achieved by considering the grid of pixels as an MRF utilising the Potts interaction model [102], in which the penalty for separating pixels is a constant. This leads to their goal of overall smoothness in the segmentation, which might look visually appealing, but may eventually not be accurate. The result of such a scheme is to favour reduction in the global total perimeter between foreground and background objects, yielding rounded-off corners, blob-like segmentation masks, and suppression of small pixel clusters. Ultimately the optimization goal may not induce convergence towards the ground truth, instead over-applying the heuristic that adjacent pixels belong to the same object - a potential distortion of the truth. This situation prompts the argument that the simplistic Potts interaction model is insufficient here.

In general, such combinatorial optimizations are considered to be *discontinuity preserving* in that they induce a trade-off between a spatial smoothness constraint and the individual likelihood of observed data. This is based on the premise that hard boundaries *do* occur in reality, but that they should only appear in segmentation where the data obviously supports them well. As an improvement to the approach of Schindler and Wang [111], a *variable* inter-pixel penalty is proposed here, with a view to supporting segmentation boundaries only where accumulated evidence is strong.

In solution of the binary label case by the MinCut/MaxFlow algorithm, one can imagine trying to transport as much water from the *source* node to the *sink* node by a system of pipes having capacity limits equal to the edge weights. When no more capacity can be added to the network, the path traced by the saturated pipes (edges) defines the minimum cut. In the current case of the mutual support problem, the capacity of a pipe depends on which way the water is flowing, i.e. which of its end nodes is joined to the source and which to the sink. This is crucial in determining which conditional probability, and hence penalty, is applied at the final

segmentation. In the proposed algorithm, illustrated for clarity here by only three pixels in Figure 3.6(a), the cost of a given labelling $\mathcal{L}$ is the energy function

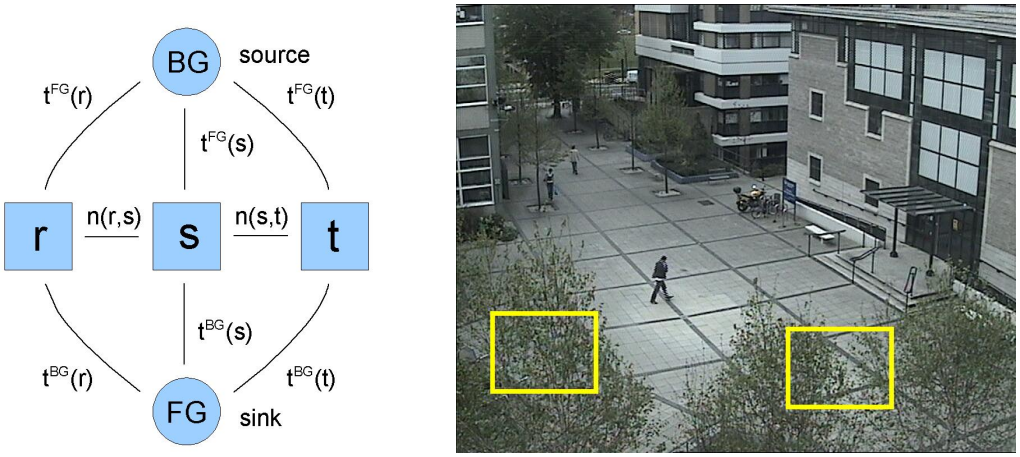
$$E(\mathcal{L}) = \sum_{r \in I^{\mathcal{Q}}} D_r(l) + \sum_{\{r,s\} \in \mathcal{N}} V_{rs}(p(r|s), p(s|r)) \tag{3.6}$$

consisting of penalty terms D_r derived from the probability of a pixel r being background as defined by Equation (3.1) and given by the $t^{BG}(r)$ and $t^{FG}(r)$ entries in Table 3.1, and an interaction term V_{rs} based on conditional probability derived from cooccurrence of adjacent pixels r and s, denoted by the $n(r,s)$ entry in Table 3.1. Determined by the way the graph edges are cut, D_r takes on one of the values $\{t^{BG}(r), t^{FG}(r)\}$, and V_{rs} takes on either the forward or reverse capacity value $n(r,s)$ if r and s have different labels, to yield the total cost for the cut $E(\mathcal{L})$ implied by choosing labelling $\mathcal{L}$. When adjacent pixels take on different labels $D_r = t^{BG}(r)$ and $D_s = t^{FG}(s)$ then V_{rs} assumes the Forward Capacity penalty, otherwise the Reverse Capacity penalty. In Equation (3.6) $\mathcal{N}$ represents the 4-connected neighbourhood of connections as shown in Figure 3.5 (not to be confused with the 4-connectivity earlier in RSLBP$_4$, even though it involves the same pixels).

Edge	Forward Capacity	Reverse Capacity
$t^{BG}(r)$	1	1
$t^{FG}(r)$	$\frac{\beta}{(p(r=S_r)+0.01)}$	$\frac{\beta}{(p(r=S_r)+0.01)}$
$n(r,s)$	$\lambda\, p(s=S_s \mid r=S_r, x, y)$	$\lambda\, p(r=S_r \mid s=S_s, x, y)$

Table 3.1: Table showing arc weight assignments for the graph representing image pixels.

The t^{FG} terminal link is weighted by an inverse function of probability, heuristically chosen to bias the penalty against symbols having a small chance of being background. Other formulations of this arc weight may also be found to work well. The V terms (n-links) can be seen as a penalty for separating pixels which, according to cooccurrence, should belong together and to the background. To cause them to end up separated, one would have to have a very low individual probability of occurring. The constants β and λ control the magnitude of the effect of the D and V penalties relative to each other, and also to the unity penalty assigned to the cost of being background. The process of calculating RSLBP$_4$ symbols is summarized by examples in Figure 3.7, whilst use of these RSLBP$_4$ values to index the correct conditional probabilities in C_h and C_v is shown in Figure 3.8.



(a) (b)

Figure 3.6: (a) More detailed graph for an array of only 3 pixels, showing Background as the *source* label and Foreground as the *sink*. Terminal and neighbourhood link edge weights are shown as t and n respectively. Cutting a lower t-link joins a pixel to the BG label incurring cost t^{BG} (b) Scenes chosen for the experiment lie within highlighted windows.

3.5 Experiment

To demonstrate the effectiveness of the new algorithm using RSLBP_4 and MinCut/MaxFlow, the challenging scene shown in Figure 3.6(b), containing a leafy tree in a courtyard, was chosen. The leaves move significantly in the wind whilst people pass *behind* the tree, but remain visible through the foliage. From a dataset of 2500 monochrome frames of size 128×96 pixels, 2000 are used as the training data to build the probability distributions and the cooccurrence matrices C_h and C_v. From the remaining frames an interesting subset is selected, in which people enter the scene and walk behind the trees, becoming partially occluded by leaves and branches. The new RSLBP_4 operator is compared not only with the standard LBP_8 operator, but also with a rather more primitive feature: a 16 level grayscale derived by merely truncating the pixel intensity to 4 bits. The MinCut algorithm and the previously tabulated weighting scheme was applied in all cases, and results are shown in Figures 3.9 and 3.10. The significant contribution of the MinCut stage is further demonstrated with a comparative result in which it is *not* used: Figure 3.11 shows what happens when individual pixel probabilities alone are used for segmentation using the RSLBP_4 operator. Even when the foreground detection threshold is optimized manually to 0.045, there is only a hint of the presence of a person, and most of the foreground pixels are noise. Figure 3.9 provides further evidence in support of the RSLBP_4 and MinCut combination, with

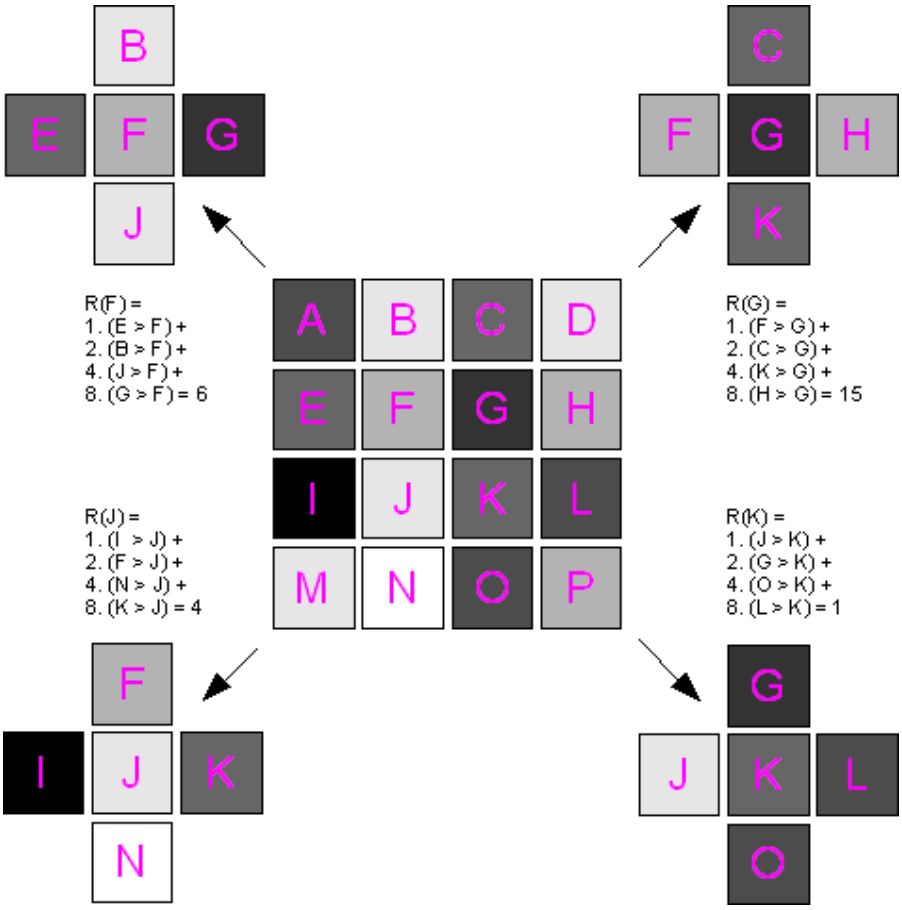


Figure 3.7: Evaluation of the new RSLBP$_4$ operator $R(\cdot)$ depicted in Figure 3.3 at the four adjacent pixel locations F, G, J and K.

images from the right hand window in the scene of Figure 3.6(b). For reference purposes, the figures also show results using a conventional Gaussian Mixture Model, consisting of 5 isotropic Gaussian components, and a learning rate time-constant set to 200 frames.

Although the Combinatorial Optimization algorithm chooses discrete labels as its solution, the notion of a detection threshold still exists in the form of the relative scaling of the various edge weights. In the present implementation, β controls the effect of the pixels' individual probabilities, whilst λ regulates the influence of the inter-pixel support. In each case, since the probabilities vary between 0 and 1, the two constants act as maximum values for their own particular type of edge. Empirically choosing $\beta = 7$ and $\lambda = 10$ scales the optimization favourably when the t^{BG} edges are set to unity. The inter-pixel support is limited to 4-connectivity here, as the important issue is demonstration of the basic principle. Using 8-connectivity may appear beneficial, but histogram sizes increase as the square of connectivity, leading to potential problems with storage requirement and data sparsity.

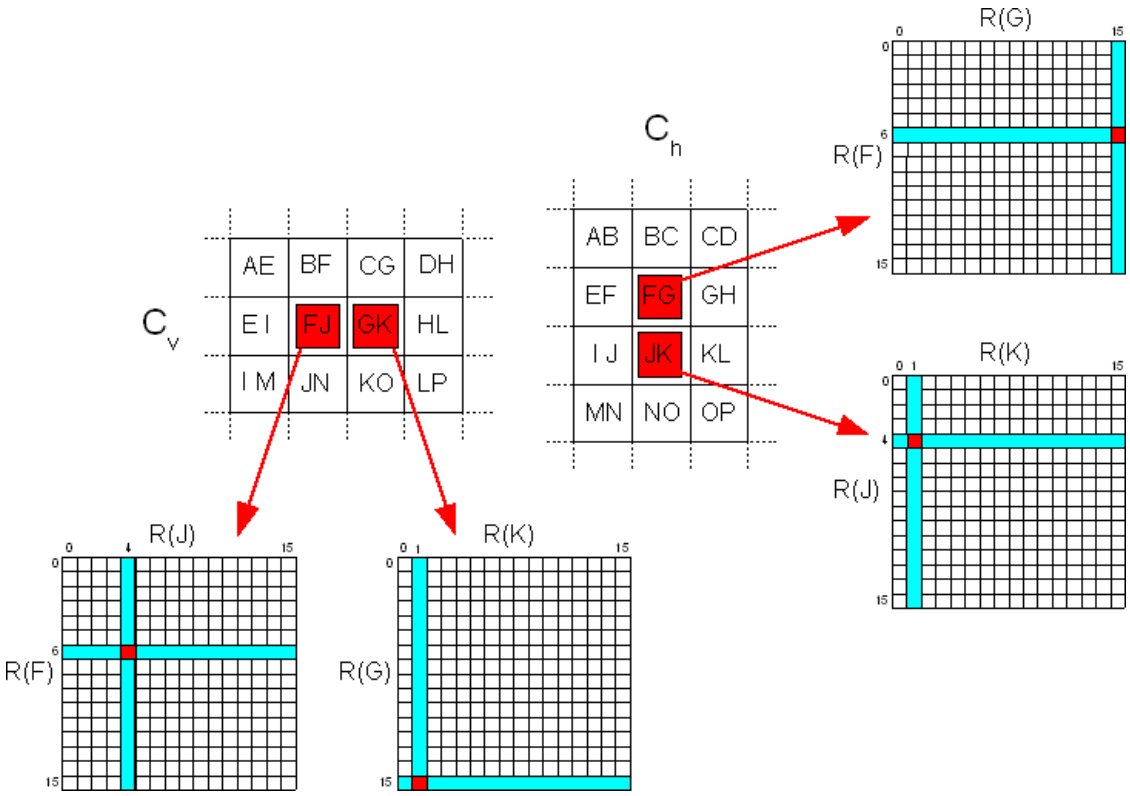


Figure 3.8: The $RSLBP_4$ values calculated in Figure 3.7 are used in pairs to index into the horizontal and vertical cooccurrence arrays C_h and C_v to find the appropriate conditional probabilities from which penalty weights may be evaluated for the graph in Figure 3.6(a) using Equations 3.4 and 3.5 with the expressions in Table 3.1.

3.6 Discussion

The $RSLBP_4$ operator can generate 16 different values as currently defined, leading to a cooccurrence matrix with only 16×16 entries. This compactness is convenient for two practical reasons. Firstly the memory required to store the inter pixel data is manageable, and secondly the quantity of training data to adequately estimate it remains modest. LBP_8 generates only 10 possible values, but as the experiments show, its rotational invariance renders it inferior in solution of the current problem. A rotationally *variant* version of LBP_8 generates as many as 59 combinations and is thus, according to the previous arguments, not so attractive.

Overall the favourable segmentation afforded by $RSLBP_4$ in the results in Figures 3.9 and 3.10 strongly supports the idea that it is a better choice than the other two commonly encountered features for the current application. Although LBP_8 yields acceptable performance in moderately challenging situations, as in Figure 3.9, it is no match for $RSLBP_4$ in the most difficult cases

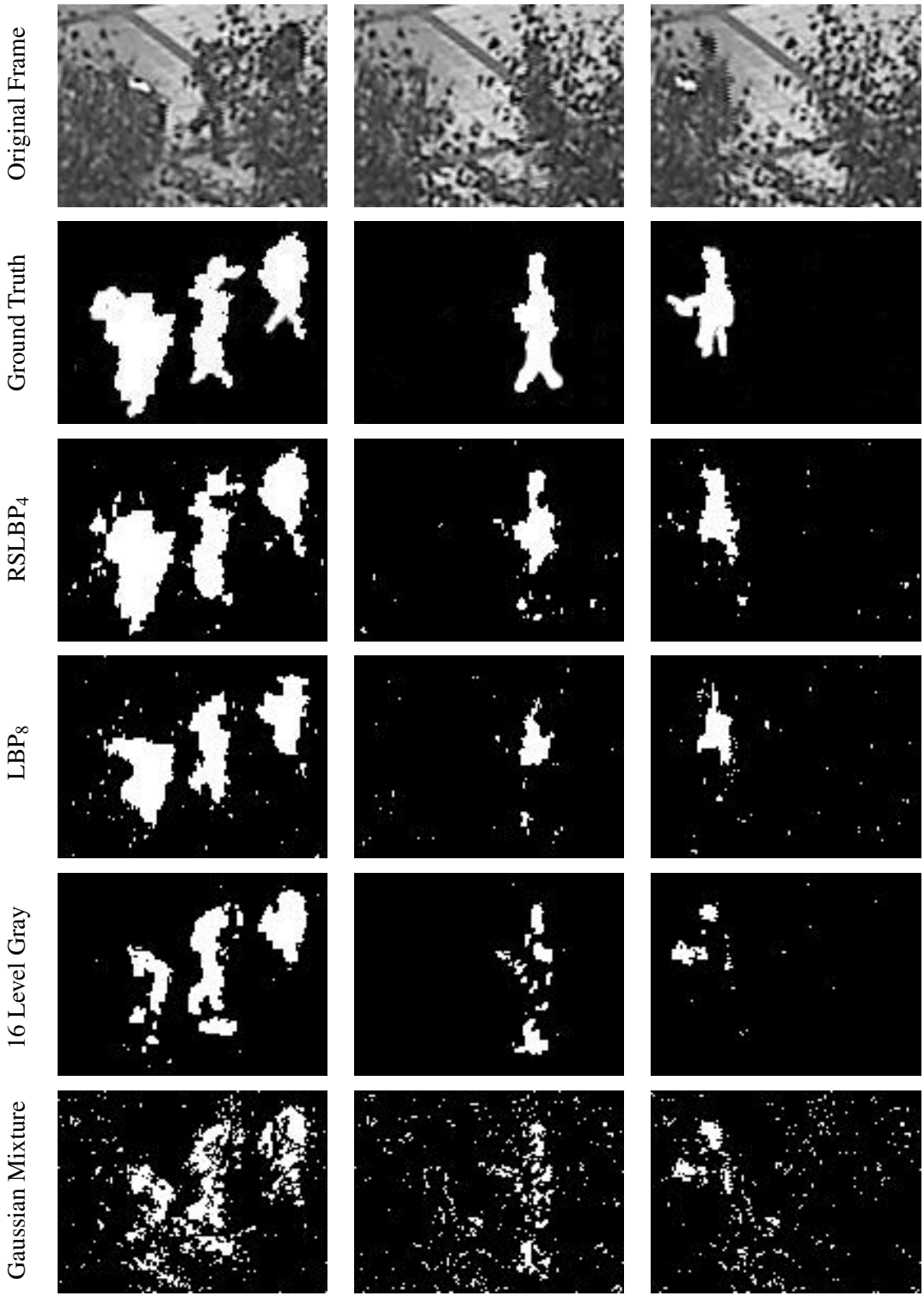


Figure 3.9: Challenging frames from the right window of Figure 3.6(b) in which people pass behind trees. Top to bottom: Original, Ground Truth, using $RSLBP_4$ operator, using LBP_8 operator, and using 16 level grayscale, all *with* MinCut, plus GMM. $RSLBP_4$ produces the best segmentation with LBP_8 a close second, whilst the grayscale and GMM methods perform poorly.

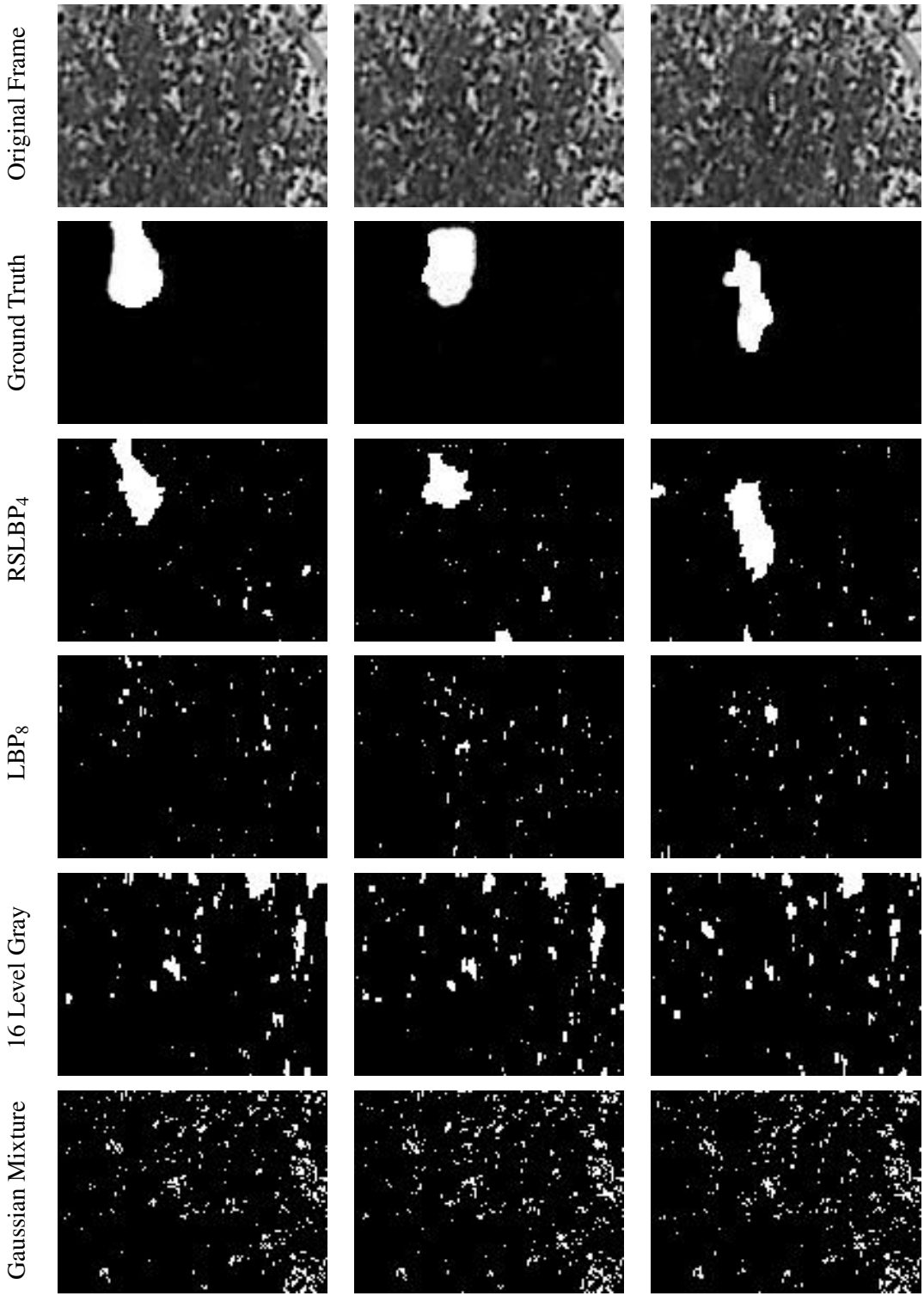


Figure 3.10: Very challenging frames from the left hand window of Figure 3.6(b) in which a person walks behind foliage. Top to bottom: Original, Ground Truth, using $RSLBP_4$ operator, using LBP_8 operator, and using 16 level grayscale, all *with* MinCut, plus Gaussian Mixture Model. Note that the $RSLBP_4$ operator is the *only* method which produces a useful segmentation here.

shown in Figure 3.10. Furthermore, the comparison between Figures 3.10 and 3.11 clearly shows that the graph cut technique contributes enormously to the overall quality of the segmentation, since without MinCut, all methods completely fail. It is believed that the 'double level' of local spatial support afforded by the partnership of the two techniques, $RSLBP_4$ and the graph cut, is the reason for the distinctive result.

On the other hand, the Gaussian Mixture Model does not perform well in any of the examples shown. The highly chaotic nature of moving foliage results in large values of component variance in the model, and consequently the sensitivity to foreground outliers is compromised. More importantly, the GMM doesn't benefit from the local spatial support which is inherent in the proposed new technique.

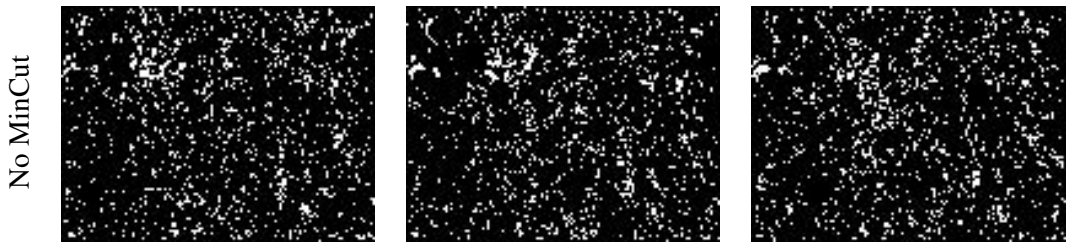


Figure 3.11: The same three frames as in Figure 3.10 using $RSLBP_4$ but *without* MinCut, and hence no local support. The previously visible person is now barely discernible amongst the noise. LBP_8 and Grayscale are similarly ineffectual without the vital MinCut stage. This result clearly demonstrates the importance of adding local support.

3.7 Detailed Analysis

To provide a more comprehensive demonstration, the new technique was compared with competitors in four different lighting scenarios representing typically encountered practical situations:

1. High Occlusion - person moving behind dense foliage.
2. In Shadow - people moving through areas of high contrast shadow.
3. Against Clutter - person moving in front of highly textured background.
4. Open View - people moving across clear uncluttered background.

Ten example frames were chosen from each scenario and manually annotated with ground truth binary masks for the expected foreground. Representative frames are illustrated in Figure 3.12. All frames were processed with each of the algorithms previously described:

1. $RSLBP_4$
2. LBP_8
3. 16 Level Grayscale
4. Gaussian Mixture Model
5. $RSLBP_4$ without Minimum Cut

The resulting foreground/background masks were compared with the ground truth for each frame, and ROC (Receiver Operating Characteristic) curves produced comparing the various approaches for each scenario. For the graph cut based techniques (1-3 above) the parameter β was varied as the detection threshold in order to create the ROC curve. For the Gaussian Mixture Model (4 above), the threshold varied was the maximum permitted distance from the mean of a Gaussian component that a pixel value may assume whilst still retaining membership of that component. This parameter was set to 2.5σ in the original work by Stauffer and Grimson [125], but is here varied between 0.1σ and 10σ to form the ROC curve. For the $RSLBP_4$ without graph cut support (5 above), a simple threshold of probability of symbol occurrence varying between 0.01 and 0.3 was used.

3.7.1 Results

Typical segmentation results for all combinations of the four scenarios and five techniques are illustrated in Figure 3.13, whilst the ROC curves based on a range of 20 threshold values for each technique are shown in Figure 3.14. The results clearly show that under Heavy Occlusion conditions, the new $RSLBP_4$ operator with Minimum Cut support consistently produces the most accurate segmentation, whilst for the less challenging scenarios, the simple 16 Level Grayscale operator with Minimum Cut support yields the best results.

3.8 Validity of Asymmetric Flows

The graph cut technique detailed in Section 3.4 achieves a *Minimum Cut* on a graph by utilizing the *Maximum Flow* algorithm originally described by Ford and Fulkerson in [36]. But the method used here relies specifically on being able to impose *asymmetric capacity* limits on arcs between pixels. However, it is not clear from the literature whether the validity of the resultant partitioning

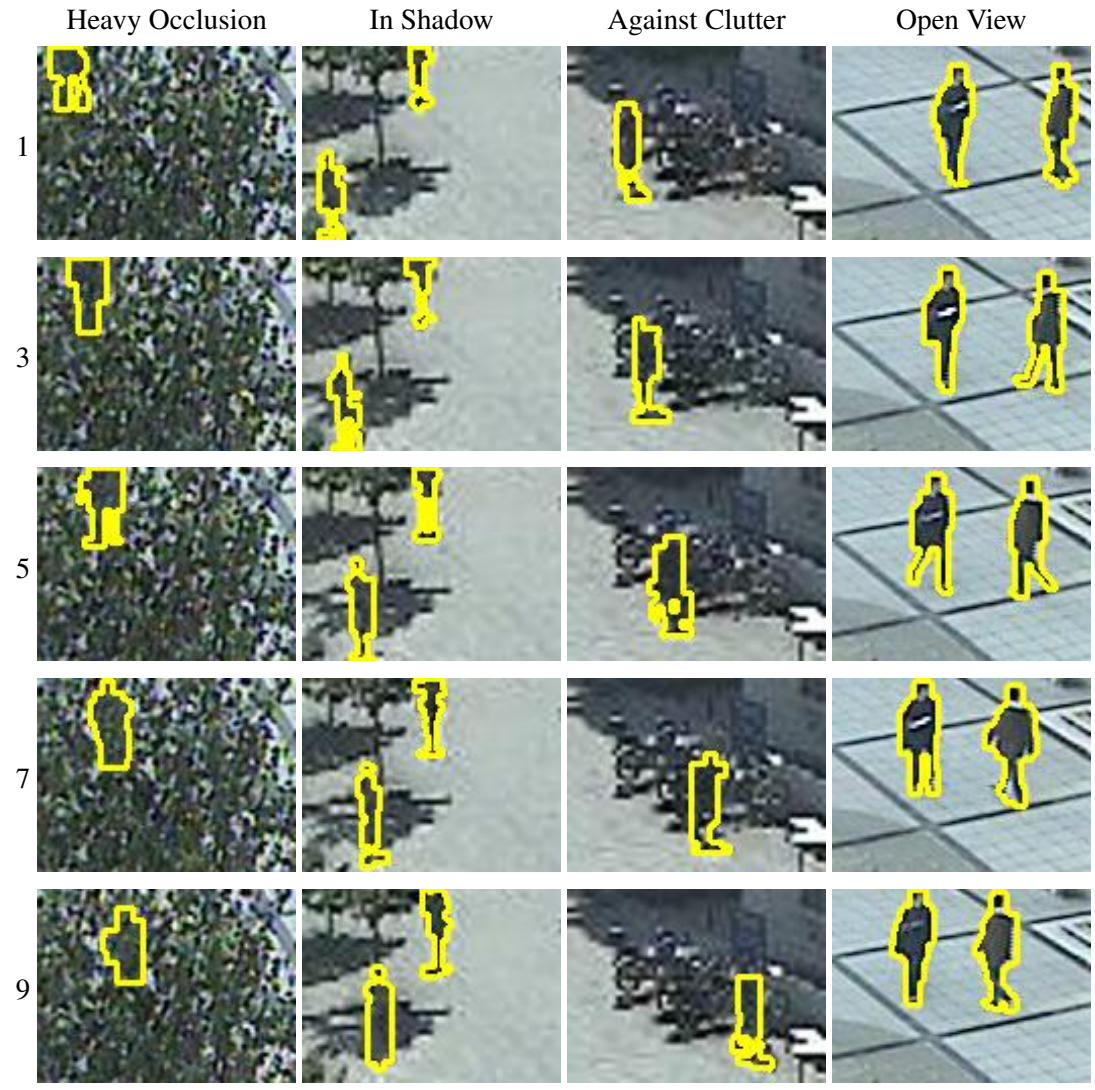


Figure 3.12: Five of the ten frames with ground truth annotation outlines for each of the four scenarios used to produce the ROC curves in Figure 3.14.

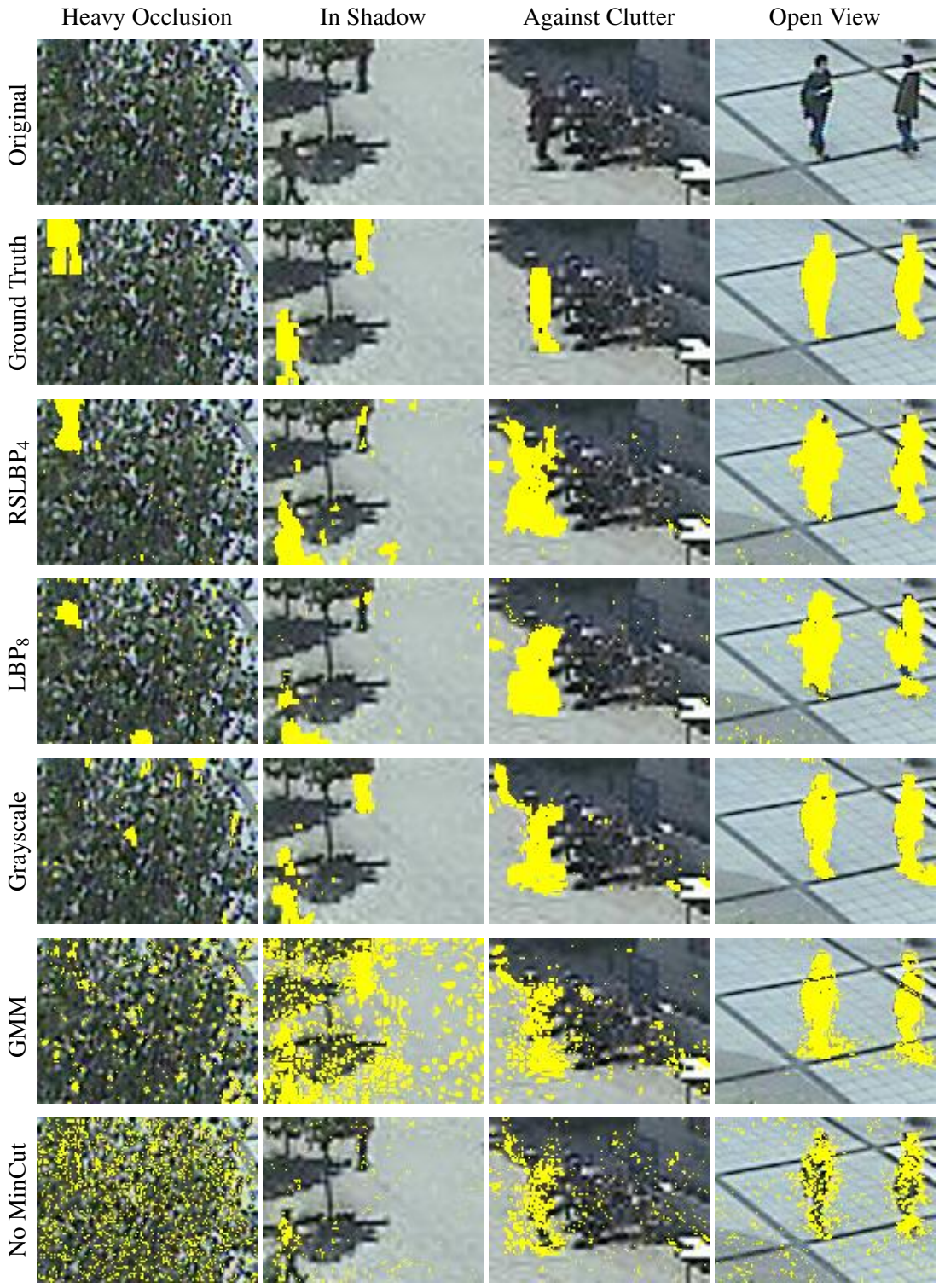


Figure 3.13: Segmentation results using each of the five different approaches detailed in Figures 3.10 and 3.11 for the first example of each of the four scenarios in Figure 3.12.

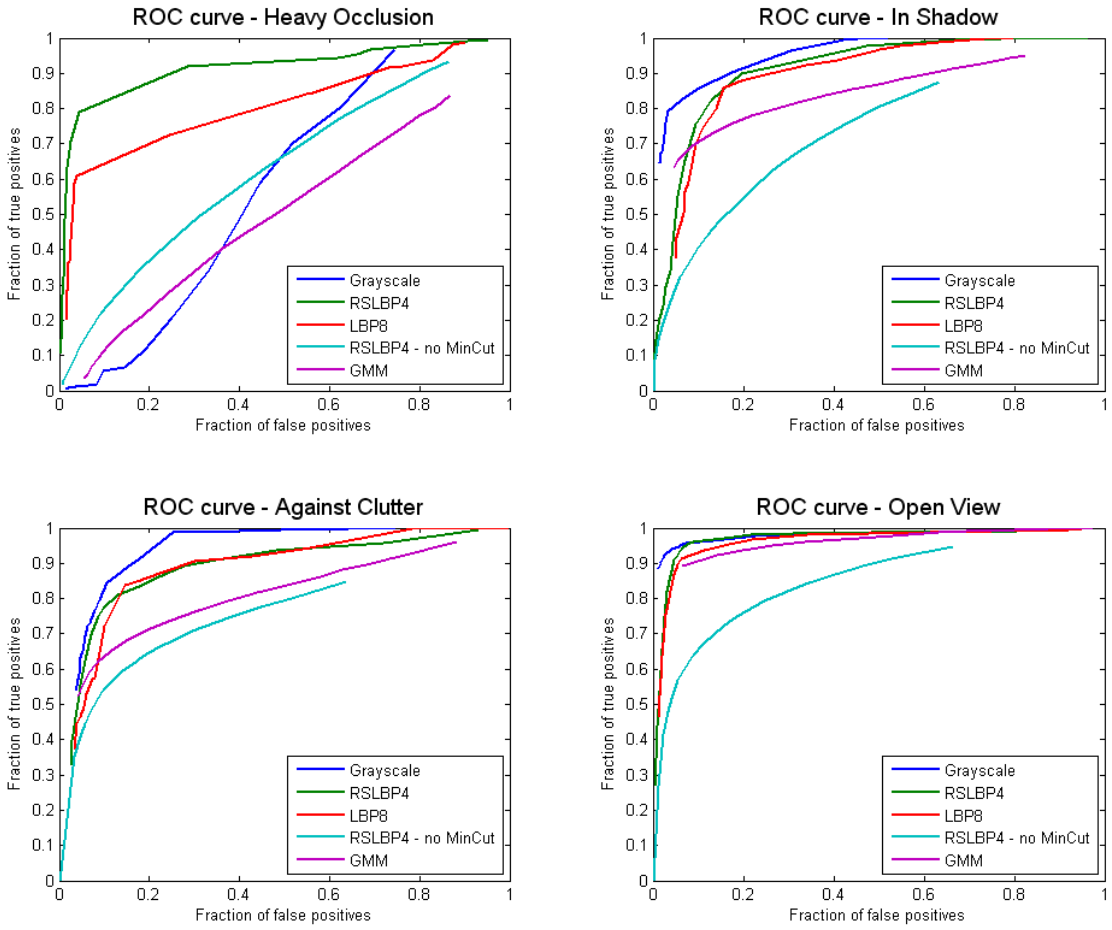


Figure 3.14: ROC curves comparing the performance of the five different approaches for the four scenarios depicted in Figure 3.12. The new $RSLBP_4$ operator excels under Heavy Occlusion conditions, whilst in less demanding situations the simple 16 Level Grayscale operator is more effective than other methods.

has been proved or disproved. To justify use of the approach in this chapter, arguments to support the validity are presented from both algorithmic and experimental points of view.

3.8.1 Depth First Graph Search

Under the Maximum Flow algorithm, the graph derived from a given frame is searched repeatedly from source to sink to find out if any further flow can be added to any of the paths. The algorithm terminates, yielding the maximum network flow condition, when no further capacity is available. As each arc of the graph is traversed, the current flow is compared with the capacity limit (weight) for that arc to establish the unused capacity. All that is required for the asymmetric weighting scheme to be valid is to establish available capacity relative to the *direction* of the proposed

current flow increment. This represents a minor complication of the algorithm, requiring the choice between the two directional capacities as each arc is encountered.

3.8.2 Exhaustive Search Experiment

To support the validity of asymmetric flow experimentally, a small graph was constructed as if from a 4×4 pixel image. Randomly generated weights were given to all t-links and asymmetric n-links. The graph was then analyzed by the Minimum Cut/Maximum Flow algorithm *and* analyzed by exhaustively calculating the total energy of all 2^{16} possible cut combinations. The experiment was repeated 1500 times with different sets of random arc weights. The Minimum Cut/Maximum Flow algorithm *never* failed to find the correct solution.

Whilst this demonstration in its own right is not conclusive proof of the validity of asymmetric weighting, and it is assumed that scaling up to 'real-sized' images presents no problem, it would seem that the algorithm is reliable enough to be used in practice.

3.9 Further Development

Although a model involving separate collection of training data is described here, it is anticipated that an adaptive on-line derivative would also be possible. In such a scenario, the cooccurrence database would be built and updated in the light of new incoming frames. Providing that a suitable learning rate can be found, the conditional distributions C_h and C_v between adjacent pixels will approximately converge, become more refined, and be tracked over time, exploiting the advantage of the ever-increasing body of training data.

The current model offers the possibility of inter-pixel support in the spatial domain only. It may be possible to extend support to additionally include a local temporal aspect. Using a 6-connected model in three dimensions would increase the model's discriminative capability if training data exhibits consistency in symbol *sequency*, but it is by no means clear whether typical scenes would benefit from this. Either way, the complexity of the graph cut would rise from four to six arcs per pixel - a modest price to pay if the result proves useful.

3.10 Summary

Detection of unusual objects amongst a highly textured background is a difficult problem, especially when the texture is manifest in the temporal dimension as well. Outdoor scenes involving

waving trees or moving water are examples of such scenarios, which are nevertheless frequently encountered in real world vision applications. This chapter has introduced a simpler new operator RSLBP_4 based on existing LBP methods, and shown how it can be applied to advantage in a probabilistic sense to tackling foreground/background segmentation of highly textured dynamic scenes. Its sensitivity to rotation, but resilience to overall illumination variations, both contribute vitally to its success in this application, and the restricted range of output symbols of RSLBP_4 permits tractable acquisition and storage of adjacent pixel cooccurrence data.

But as demonstrated, this alone is not sufficient for good segmentation in difficult circumstances. Cooccurrence of features in a pixel's local neighbourhood provides a powerful mechanism for boosting the reliability of the foreground/background decision task. By using the conditional probabilities yielded by pairwise cooccurrence of 4-connected pixels, and casting the problem as one of Combinatorial Optimization, results show that useful segmentation *is* possible from challenging dynamic backgrounds. It has been shown that cooccurrence data may be used to construct a graph, of which the minimum cost cut facilitates mutually supporting inferences between pixels, leading to a useful segmentation which would not have been easy to arrive at otherwise.

Whilst the method just described accumulates pattern distributions over time, it does not constrain or utilize information carried by the *temporal persistence*, or lack of, exhibited by the chosen feature. This particular aspect of background modelling is addressed in Chapter 4.

Chapter 4

Estimation of Time-varying Backgrounds

The most commonly encountered background models are based on per pixel techniques such as adaptive Gaussian Mixture Models [125, 131], or subspace analysis based methods [94, 73]. Both approaches have been used with success. However, in typical implementations it is difficult to avoid such background models being contaminated by foreground scene content, eventually resulting in a less discriminative model.

Whilst the previous chapter describes a technique exploiting purely *spatial* support in terms of cooccurrence of adjacent patterns, the focus of this chapter is investigation of a method seeking *local temporal* support for a recovered background by identifying consistency in appearance among nearby frames - an approach not generally directly exploited by conventional methods.

Motivated by the demand for a more effective background model, robust to non-stationary environmental changes in outdoor scenes, a technique using Combinatorial Optimization to extract near-optimal background estimates from *blocks* of temporally localized frames is presented here. Using an existing graph cut technique to derive these estimates as a pre-processing stage, in conjunction with subspace analysis, a novel hybrid background model is demonstrated. The combined approach exhibits results superior to those achievable with the latter technique alone, and especially suitable for background modelling in outdoor situations where variable lighting conditions prevail.

4.1 Scope of the Problem

An effective background model is a crucial first stage in most computer vision applications, especially in outdoor environments, where the simplistic *mean image* approach possible under heavily constrained indoor lighting conditions is inappropriate. The reliability with which the model identifies potential foreground objects directly impacts on the efficiency and performance level achievable by subsequent processing stages such as tracking, recognition and threat evaluation. The nature of such an unconstrained background is intrinsically statistical. Whilst the concept of statistical scene modelling suggests that there is no exact distinction between what constitutes foreground and background, a useful practical definition for surveillance in a busy urban scene is that people and the objects they cause to move are foreground, whereas buildings, fixtures, trees and permanent objects form the background. The task of the background model in such a setting is to discriminate between the two classes under a potentially wide variety of lighting conditions. Evidently, confusion might still arise, since trees sway in the wind, tending to become foreground, whilst people park their cars, which are eventually subsumed by the background. Without specifically distinguishing vehicles and people from other objects, the latter problem is unlikely to be completely soluble, but in spite of this, the temporal persistence of an object in a scene, or lack of persistence, constitutes strong evidence as to its novelty value.

Of the per pixel techniques, the adaptive Gaussian Mixture Model (GMM) [125, 131] is one of the most widely used background modelling techniques, and the PCA or eigen-background method [94, 73] is the most commonly encountered holistic modelling approach. Whilst both have been used successfully, both also suffer from the general problem of having to model foreground clutter *as well as* the background, since the model can't in general discriminate between the two. The resultant model is thus less compact, since it has to represent both foreground and background, but more seriously, the model's sensitivity to the foreground is compromised by the contamination. Such a system relies on the foreground being statistically quite rare in order that these two problems remain under control. However, in typical busy urban traffic scenes for example, such rarity cannot be relied upon.

In the case of the GMM, a higher model order with more modes may be required to encompass the extra hypotheses presented by foreground objects, which might be seen as outliers with regard to the background process. In order to render a GMM adaptive, Friedman and Russell [38] suggest classifying new pixel data in order to determine which Gaussian, if any, it matches before

using it to update the sufficient statistics of that Gaussian. However, this forms a feedback loop such that the classification outcome is a function of previous data. By the very nature of their dependence on pixel history, feedback systems are prone to various failure modes including:

1. Becoming trapped in a local minimum
2. Oscillation or instability
3. Limit cycles (see [128])

As an example of type 1, consider a red car parked at the side of the road during model initialization, which subsequently drives away leaving dark road-coloured pixels in its place. If the model always classifies these pixels as foreground, it will never use them to update the model, and the desired background distribution, the road, may never be encompassed.

In the case of PCA type methods, various attempts have been made to introduce robustness and mitigate the effect of foreground outliers, such as by Xu and Yuille [146] and De La Torre and Black [72]. An *influence function* is used by Li in [73] through which candidate background data is compared with the current model, which in turn acts as a prototype. In this case it could be argued that the model *can* discriminate between potentially useful background and irrelevant foreground. But applying the prototype comparison is necessarily a *feedback* process, possibly exhibiting non-linear characteristics because of the influence function, but in any case susceptible to the previously mentioned drawbacks.

Even the pattern-based cooccurrence method described in Chapter 3 suffers from the foreground contamination problem, since crucially the unwanted patterns generated serve merely to flatten the cooccurrence distributions and hence desensitize the model.

4.2 Short-term Background Estimates

On the other hand, a method detailed by Cohen in [19] has been shown capable of compiling a *short-term background* image on a per pixel basis from a short block of input frames by casting the problem as an exercise in optimal labelling. Figure 4.1 shows an example of how 20 frames from a continuously busy metro ticket hall can lead to a useful background approximation. The background is drawn from parts of any of the input frames which are found to be spatially and temporally consistent. Thus the solution comprises a set of labels or pointers, one for each pixel in the background image, specifying from which of the 20 input frames each pixel is to be taken.

The method described in [19] is an application of Combinatorial Optimization [21] achieving an approximately minimum cost solution using the Minimum Cut/Maximum Flow [36] and Alpha Expansion [10] algorithms. However, prerequisites for this approach to work are:

1. All of the required background is visible for some of the time.
2. The required background is more consistently stable than any foreground pixel intensity.
3. Each background pixel is time-independent.

But these conditions are not always satisfied. To address the problem, a method is proposed in this chapter whereby objects which are obviously foreground, under a given definition, are eliminated from input frames before allowing those frames to contribute to the construction of a background model. Such an approach yields a 'purer' representation of the true background, and hence one with heightened sensitivity. Obviously, if this pre-processing stage were totally effective, the task of background segmentation would already have been achieved. In reality, it only offers a useful measure of pre-processing. The new solution proposed here thus consists of the pixel-labelling method described above as a stage of pre-processing, operating on short blocks of input frames to produce a temporally localized background estimation per block. These estimates are then used to build an eigenspace model. Such a hybrid approach permits the latter to 'concentrate' on dealing with lighting and shadow variation rather than being contaminated with objects like cars and people which could more usefully be considered foreground, at least with regard to surveillance applications.

Furthermore, the proposed hybrid approach is entirely a *feed-forward* system. It thus avoids the previously mentioned instability and local minimum problems, and convergence towards the ground truth is more or less guaranteed.

From a model perspective, the new algorithm may be seen as offering a level of *temporal support* at a given pixel amongst all the frames comprising a block, by encouraging consistency in the choice of the estimated background. As a statistical model of the scene, a bias has been given towards retention of *persistent* elements of the scene structure, with a view to discriminating these from the relatively transient foreground.

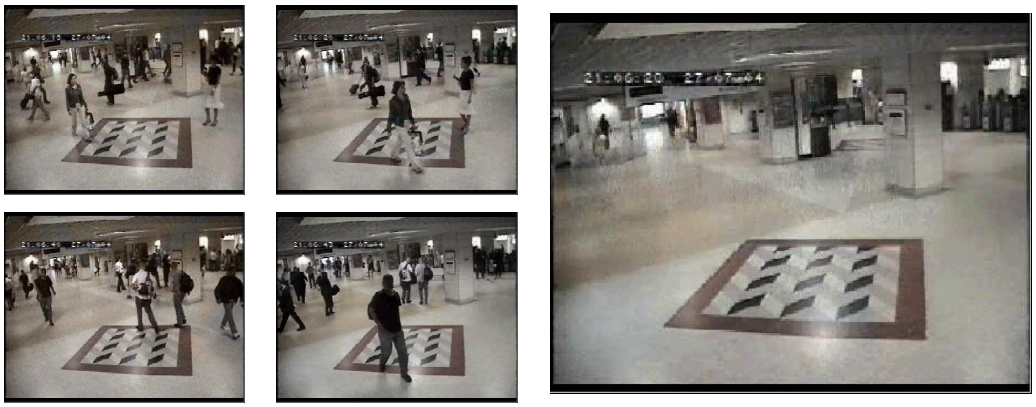


Figure 4.1: Example of short-term background recovery from video of a busy metro station ticket hall. Left: 4 of the 20 input frames. Right: Recovered background. Note how most of the moving objects have been eliminated. In this indoor environment with predictable lighting conditions, such a background may be used directly with little further processing, as shown in Figure 4.2.

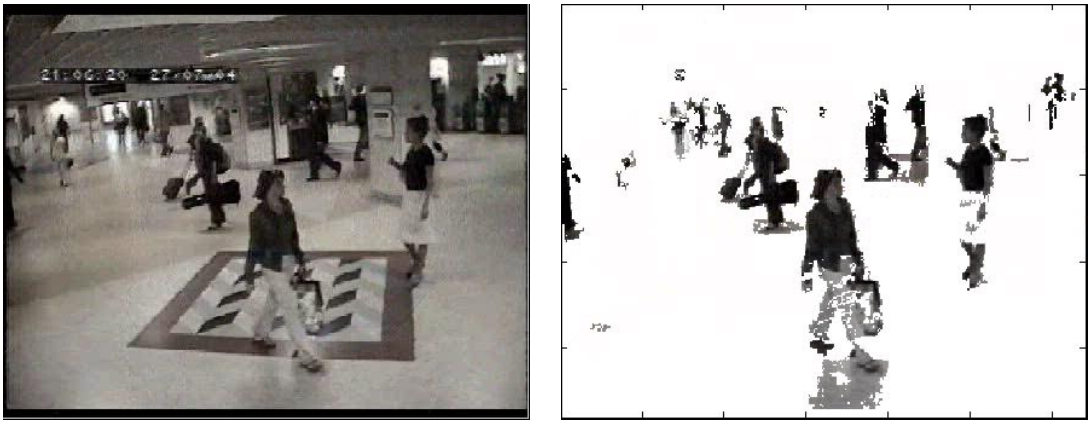


Figure 4.2: Foreground segmentation using the recovered background from Figure 4.1.

4.3 Combinatorial Optimization

Given a temporally localized set of F input frames of a scene each of $\mathcal{P}$ pixels, the requirement is to form an output image $I_{\mathcal{B}}$ to best represent the scene's background at that time. Thus a set of labels $\mathcal{F}$ is desired, consisting of one label per pixel, specifying from which input frame that pixel is to be taken. Evidently, the number of possible combinations is large, but finite. In essence, the idea is to assign a cost to each choice of label (1 of F) at each pixel, and then solve for the minimum cost over the image as a whole in order to yield the best set of background composition labels. For the algorithm to work, the cost assignment scheme for the pixels has to reflect lower costs for more appropriate combinations of labels. This process is encouraged by penalizing poor temporal or spatial correlation between adjacent pixels.

4.3.1 Binary Graph Cuts

The Ford-Fulkerson algorithm [36] permits exact solution of a combinatorial optimization problem in polynomial time by a minimum graph cut (Min-Cut) in a situation where there are only *two* class labels. Having defined a suitable costing model, an undirected graph may be constructed for the background image, consisting of a node for each pixel, plus two extra nodes known as the *source* and the *sink*, representing the two class labels. The pixel costs become the arc weights on the graph. However, there are F class labels representing the block of input frames, where F might typically be larger than ten or more. Although the exact solution of such a problem is possible, it has been shown to be NP-hard [10]. Instead, an approximate solution can be obtained rather more efficiently by applying the Min-Cut algorithm iteratively, with each class label taking its turn to be the source (α), whilst the other $F-1$ class labels become the sink (α'), as shown in Figure 4.3.

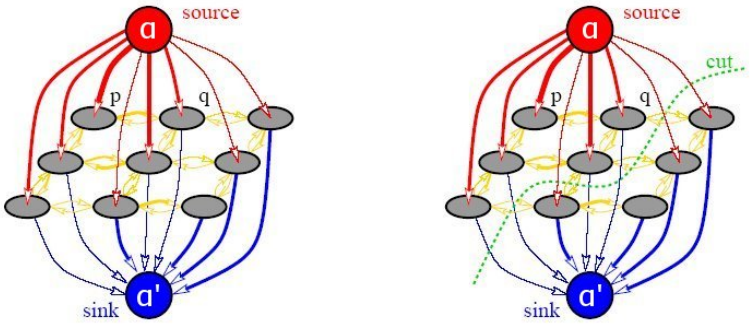


Figure 4.3: Graph for an array of only 9 pixels. Previously in Figure 3.5, the *source* and *sink* nodes represented two separate classes: Foreground and Background. Here the *source* represents the single label α chosen in the current iteration of the α-expansion process, whilst the *sink* node α' represents the union of all other labels. Weights *between* pixels stem from spatial continuity, whilst those connecting to the *source* and *sink* relate to temporal and motion continuity. The actual graph contains a node for *every* pixel in the image. Figure taken from [9].

4.3.2 Alpha Expansion

Under this scheme, at any given iteration, a pixel might already belong to the class label which is currently taking its turn at being α. In this case, the weight (cost) linking it to α is made infinite, so that the pixel cannot leave the class label at this iteration. The overall result is that as α takes

on each class label F, pixels from all the other class labels may leave in order to join α, but none may leave α. This is known as α-*expansion* which has been shown by Boykov et al. [10] to lead to an *approximately* minimum cost labelling solution after a number of cycles of α through the F class labels. Typical total image energy reduction over the first iteration is shown in Figure 4.4. According to a proof in [10], the objective function lies within a constant factor of the global optimum if the interaction penalty is a true metric obeying the triangle inequality. The optimal graph cut at any iteration is then obtained by a process drawing an analogy with network flow, in which arc weights are considered flow capacities, the objective being to achieve maximum flow (Max-Flow) from source to sink. Under this condition, the arcs which are saturated (i.e. have reached their flow capacity) are those which should be cut to achieve the optimal partitioning in the equivalent Min-Cut problem. To arrive at this situation, flow is added to the network incrementally in an iterative fashion until no further addition is possible because there are no remaining unsaturated paths from source to sink.

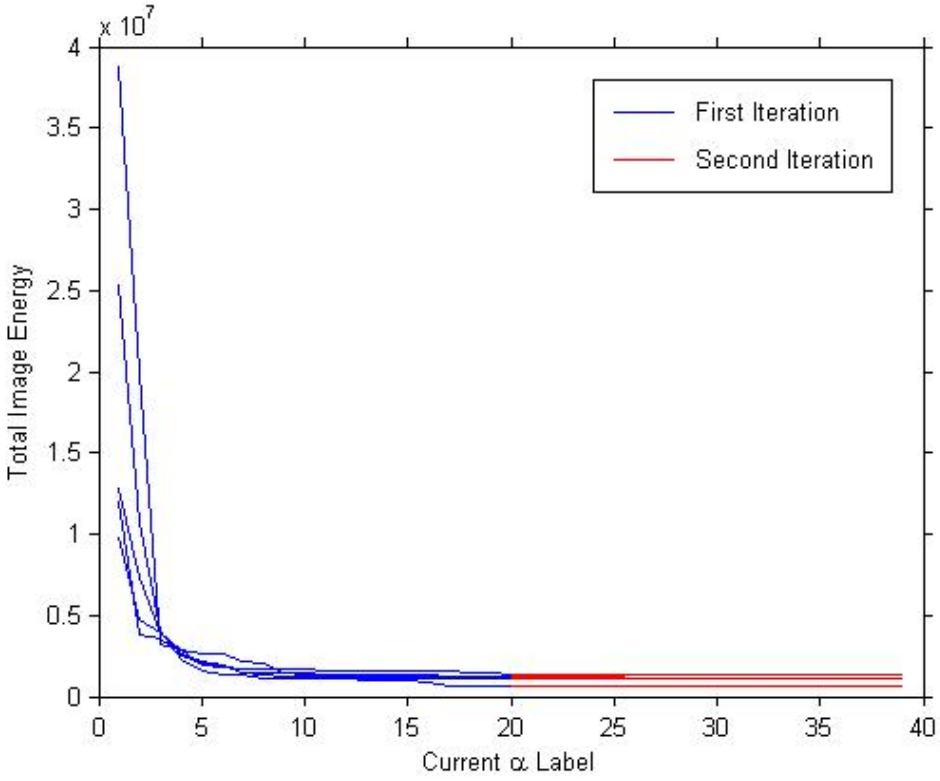


Figure 4.4: Energy reduction through the first two iterations of alpha expansion for the 5 test frame sets used in Section 4.6. Note how the energy is reduced dramatically during the first few graph cuts. The implication here is that quite good background estimates may be obtained even after only one alpha expansion iteration.

4.4 A Hybrid Pixel-Labelling and Subspace Model

4.4.1 Labelling Cost Functions

Following the notation of Cohen in [19], a set of input F frames are denoted as $I_1, I_2, \ldots, I_F$, and $I_f(p)$ is a colour intensity vector at pixel p where $p \in \mathcal{P}$ is the set of pixels in an image, and $f \in \{1 \ldots F\}$. A given labelling is defined as $\mathcal{F} = \{f_p\}_{p \in \mathcal{P}}$. The background estimation is formed by taking a pixel intensity vector at p from input frame f_p^* for all $p \in \mathcal{P}$ such that $\{f_p^*\}_{p \in \mathcal{P}}$ is the set of labels corresponding to the minimum cost background. The cost of a given labelling $\mathcal{F}$ is the energy function

$$E(\mathcal{F}) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{\{p,q\} \in \mathcal{N}} V_{pq}(f_p, f_q) \tag{4.1}$$

consisting of terms relating respectively to *temporal smoothness* at pixel p, and *spatial smoothness* between pixels p and q in a neighbourhood $\mathcal{N}$ around p. The temporal smoothness term $D_p(f_p)$ consists of two parts which are each evaluated independently at every pixel site according to the relationship

$$D_p(f_p) = D_p^S(f_p) + \beta D_p^C(f_p) \tag{4.2}$$

where β controls the balance between D^S and D^C. The first $D_p^S(f_p)$, termed the Stationarity Cost, penalizes choice of frames where the local temporal variance is high, evaluated over $2r$ adjacent frames, with pixels averaged over the three colour components as described in [19], so that

$$D_p^S(f_p) = \min\left(Var_{f_p - r \ldots f_p}(p), Var_{f_p \ldots f_p + r}(p)\right) \tag{4.3}$$

permitting the most stable r frames either before *or* after f_p to represent the stationarity at p. The overall variance over a range of frames is calculated as an equally weighted sum of the biased variances of each colour component in the range at that pixel

$$Var_{f_a \ldots f_b}(p) = \frac{1}{3}\left[Var(R_{p,f_a \ldots f_b}) + Var(G_{p,f_a \ldots f_b}) + Var(B_{p,f_a \ldots f_b})\right] \tag{4.4}$$

for a colour vector $[R\,G\,B]^T$ at pixel p and frame f.

The second part $D_p^C(f_p)$, known as the consistency cost, penalizes choice of frames in which there is a motion boundary for a pixel. Choice of a frame f_p is penalized if, at the pixel in question, there is significant temporal difference $M_{f_p f} = \left\|I_{f_p} - I_f\right\|_2$ from another frame f, but at the same time, the latter contains little spatial difference. A large ratio in the gradients of M and I implies a moving object in frame f_p, which should be excluded from the background. Using

the square of the L_2 norm, this ratio is defined as

$$\Omega_{f_p f}(p) = \frac{\left\|\nabla M_{f_p f}(p)\right\|^2}{\|\nabla I_f(p)\|^2 + \varepsilon^2} \tag{4.5}$$

where ∇ here represents the *spatial* derivative of a vector. So the numerator of Equation (4.5) represents the gradient magnitude of $M_{f_p f}$, and is large wherever image I_f changes from agreeing with I_{f_p} to disagreeing with it. The denominator of Equation (4.5) reflects the spatial gradient magnitude of the luminous intensity of image I_f, and thus overall, $\Omega_{f_p f}$ is large where there is a temporal boundary between I_{f_p} and I_f, but no intensity boundary in I_f to mitigate it.

Given that the colour intensity vector I of a pixel p located at position (x,y) in frame f is given by $I(p_{x,y}) = [R_{x,y,f}\ G_{x,y,f}\ B_{x,y,f}]^T$, then

$$\begin{aligned}\left\|\nabla M_{f_p f}(p_{x,y})\right\|^2 = &\left(\left\|\begin{bmatrix} R_{x-1,y,f_p} \\ G_{x-1,y,f_p} \\ B_{x-1,y,f_p} \end{bmatrix} - \begin{bmatrix} R_{x-1,y,f} \\ G_{x-1,y,f} \\ B_{x-1,y,f} \end{bmatrix}\right\| - \left\|\begin{bmatrix} R_{x+1,y,f_p} \\ G_{x+1,y,f_p} \\ B_{x+1,y,f_p} \end{bmatrix} - \begin{bmatrix} R_{x+1,y,f} \\ G_{x+1,y,f} \\ B_{x+1,y,f} \end{bmatrix}\right\|\right)^2 + \\ &\left(\left\|\begin{bmatrix} R_{x,y-1,f_p} \\ G_{x,y-1,f_p} \\ B_{x,y-1,f_p} \end{bmatrix} - \begin{bmatrix} R_{x,y-1,f} \\ G_{x,y-1,f} \\ B_{x,y-1,f} \end{bmatrix}\right\| - \left\|\begin{bmatrix} R_{x,y+1,f_p} \\ G_{x,y+1,f_p} \\ B_{x,y+1,f_p} \end{bmatrix} - \begin{bmatrix} R_{x,y+1,f} \\ G_{x,y+1,f} \\ B_{x,y+1,f} \end{bmatrix}\right\|\right)^2\end{aligned} \tag{4.6}$$

and

$$\|\nabla I_f(p_{x,y})\|^2 = \left\|\begin{bmatrix} R_{x-1,y} \\ G_{x-1,y} \\ B_{x-1,y} \end{bmatrix} - \begin{bmatrix} R_{x+1,y} \\ G_{x+1,y} \\ B_{x+1,y} \end{bmatrix}\right\|^2 + \left\|\begin{bmatrix} R_{x,y-1} \\ G_{x,y-1} \\ B_{x,y-1} \end{bmatrix} - \begin{bmatrix} R_{x,y+1} \\ G_{x,y+1} \\ B_{x,y+1} \end{bmatrix}\right\|^2 \tag{4.7}$$

The small constant ε in Equation (4.5) prevents the denominator from being zero, and ensures a low cost when there is little gradient in either M or I. Confidence about the identification of motion in f_p is gained by averaging $\Omega_{f_p f}$ over all frames

$$D_p^C(f_p) = \frac{1}{F}\sum_{f=1}^{F} \Omega_{f_p f}(p) \tag{4.8}$$

The quantity Ω bears a close relationship to the magnitude component of an optical flow field calculation. The images in Figure 4.5 show how D^C is developed from its constituent parts, whilst examples of typical relative values for D^S and D^C are shown in Figure 4.6.

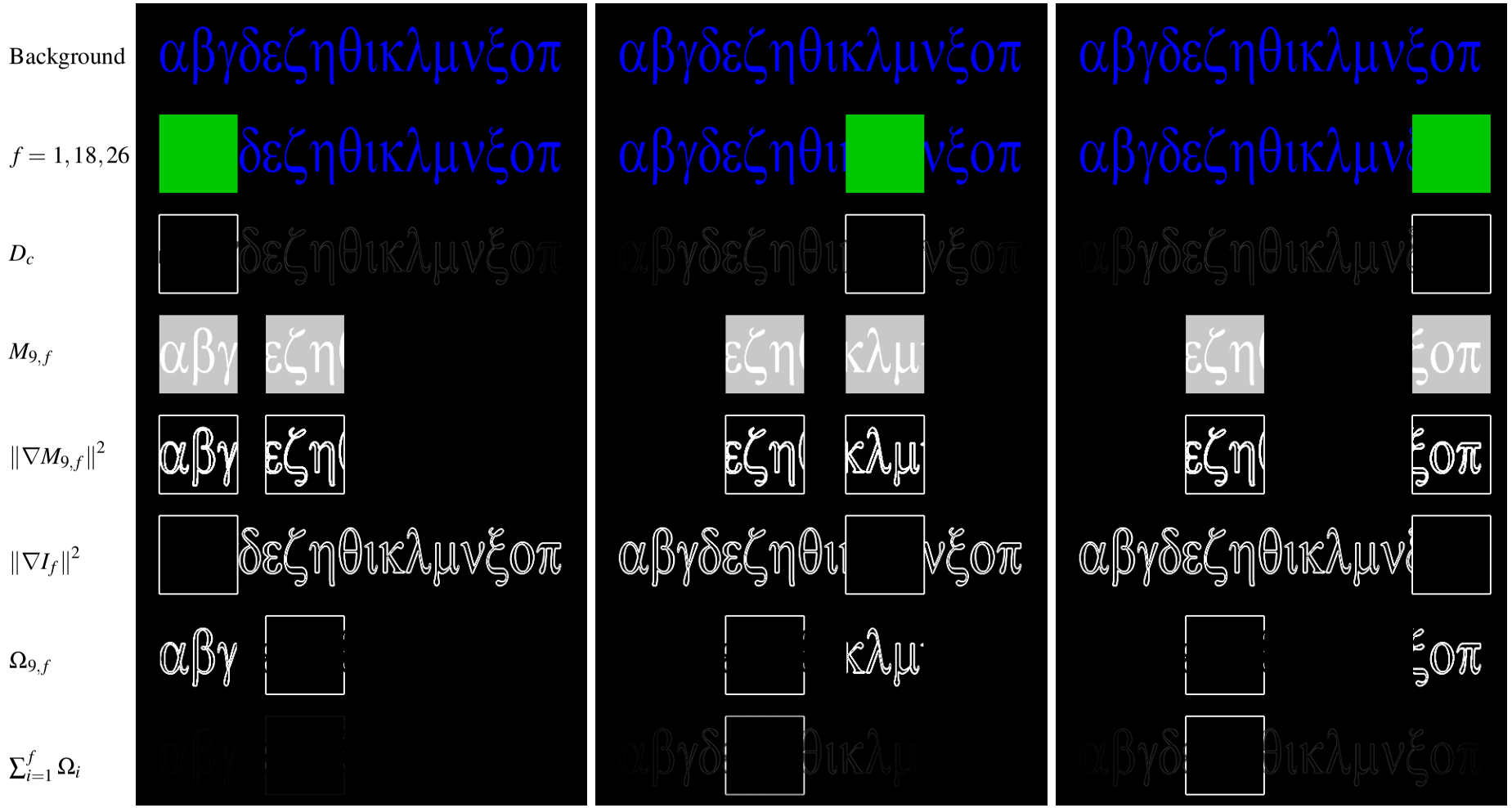


Figure 4.5: A simple example showing how D^C is derived from spatial and temporal properties of an image set. The Greek letters represent a background, across which the green square moves. The objective is to recover the background from the input frames depicted in the second row. Successive rows represent the quantities used in Equations (4.5) and (4.8).

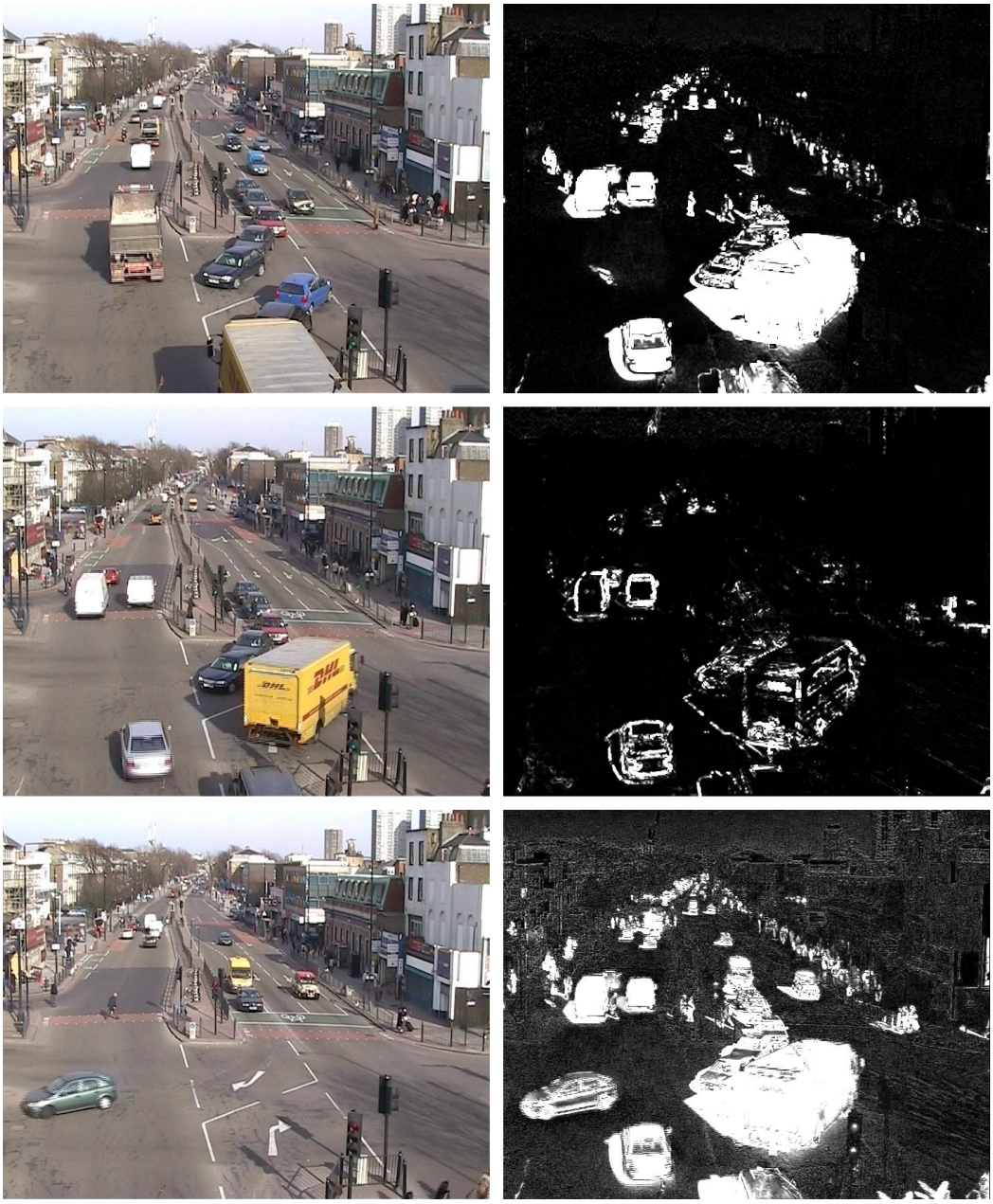


Figure 4.6: Relative values of the three cost mechanisms used in the algorithm for a typical candidate background frame. Higher intensity signifies higher cost of choosing a pixel from a frame. Left: The first 3 frames from a block. Top Right: Stationarity Cost D^S for the middle left frame relative to the other two frames. This cost is characterized by high local temporal variance between candidate frames. Middle Right: Consistency Cost D^C for the middle left frame relative to *all* other frames in the block. This cost is high at motion boundaries. Bottom Right: Spatial Continuity Cost V_{pq} between the left middle and lower frames, considering p and q as *horizontal* neighbours. It is high for a poor intensity match between p and q in the two chosen frames.

Given that the goal is to 'stitch together' a composite image from areas in candidate source frames, it is evident that the boundary between the prospective areas must occur somewhere. In order to cause minimal visual disturbance in the resultant output, the ideal location for the 'switch' is one where the candidate images possess high immediate similarity. Thus the spatial continuity cost between two neighbouring pixels p and q for two input frames f_p and f_q is

$$V_{pq}(f_p, f_q) = \lambda \left(\frac{\left\| I_{f_p}(p) - I_{f_q}(p) \right\|^2 + \left\| I_{f_p}(q) - I_{f_q}(q) \right\|^2}{2 \times (\text{number of colour planes})} \right) \tag{4.9}$$

The penalty of choosing f_p and f_q as different source frames for two neighbouring pixels p and q will be small if the frames differ by little in the vicinity of p and q, thus encouraging the switch from copying from one frame to another. Such a region is quite likely to represent background in this case. The constant λ controls the balance between V and the temporal cost D. The procedure for constructing the graph for one iteration of α-expansion is shown in Figure 4.7 for the simple case where one pixel is already in class α. For the more complicated situation when neither p nor q is in α, the reader is referred to [10].

4.4.2 Subspace Modelling of Min-Cut Labelled Background Pixels

From a sequence of M input frames of size $h \times v$ pixels, overlapping blocks of F frames are drawn to which the above background recovery algorithm is applied, yielding $N = M - F + 1$ candidate background frames $I_{\mathcal{B}_1}, I_{\mathcal{B}_2} \ldots, I_{\mathcal{B}_N}$. Thus $I_{\mathcal{B}_1}$ is derived from input frames $1 \ldots F$, $I_{\mathcal{B}_2}$ from frames $2 \ldots F+1$ and so on. The background images are then rasterized to form column vectors $\mathbf{x}_1 \ldots \mathbf{x}_N$ each of length hv elements. The mean vector $\mathbf{m}$ of $\{\mathbf{x}_1 \ldots \mathbf{x}_N\}$ is determined as

$$\mathbf{m} = \frac{1}{N} \left(\sum_{i=1}^{N} \mathbf{x}_i \right) \tag{4.10}$$

After mean subtraction, the vectors $\{\mathbf{x}_1 \ldots \mathbf{x}_N\}$ are concatenated horizontally to form a matrix $\mathbf{X} = [\mathbf{x}_1 - \mathbf{m}, \mathbf{x}_2 - \mathbf{m} \ldots, \mathbf{x}_N - \mathbf{m}]$. The covariance matrix for the background vectors $\mathbf{x}_n$ where $1 \leq n \leq N$ is then given by the outer product of $\mathbf{X}$ with itself $\mathbf{C} = \mathbf{X}\mathbf{X}^T$ with eigenvectors $\mathbf{v}_i$ and eigenvalues λ_i where $1 \leq i \leq N$

$$\mathbf{X}\mathbf{X}^T \mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{4.11}$$

However, such a matrix would contain $(hv)^2$ elements but only have a rank of at maximum N. In this case advantage of the low dimensional method in [73] may be taken, whereby Equation (4.11) is pre-multiplied by $\mathbf{X}^T$ in order to find the much smaller matrix $\mathbf{X}^T\mathbf{X}$ of size $N \times N$

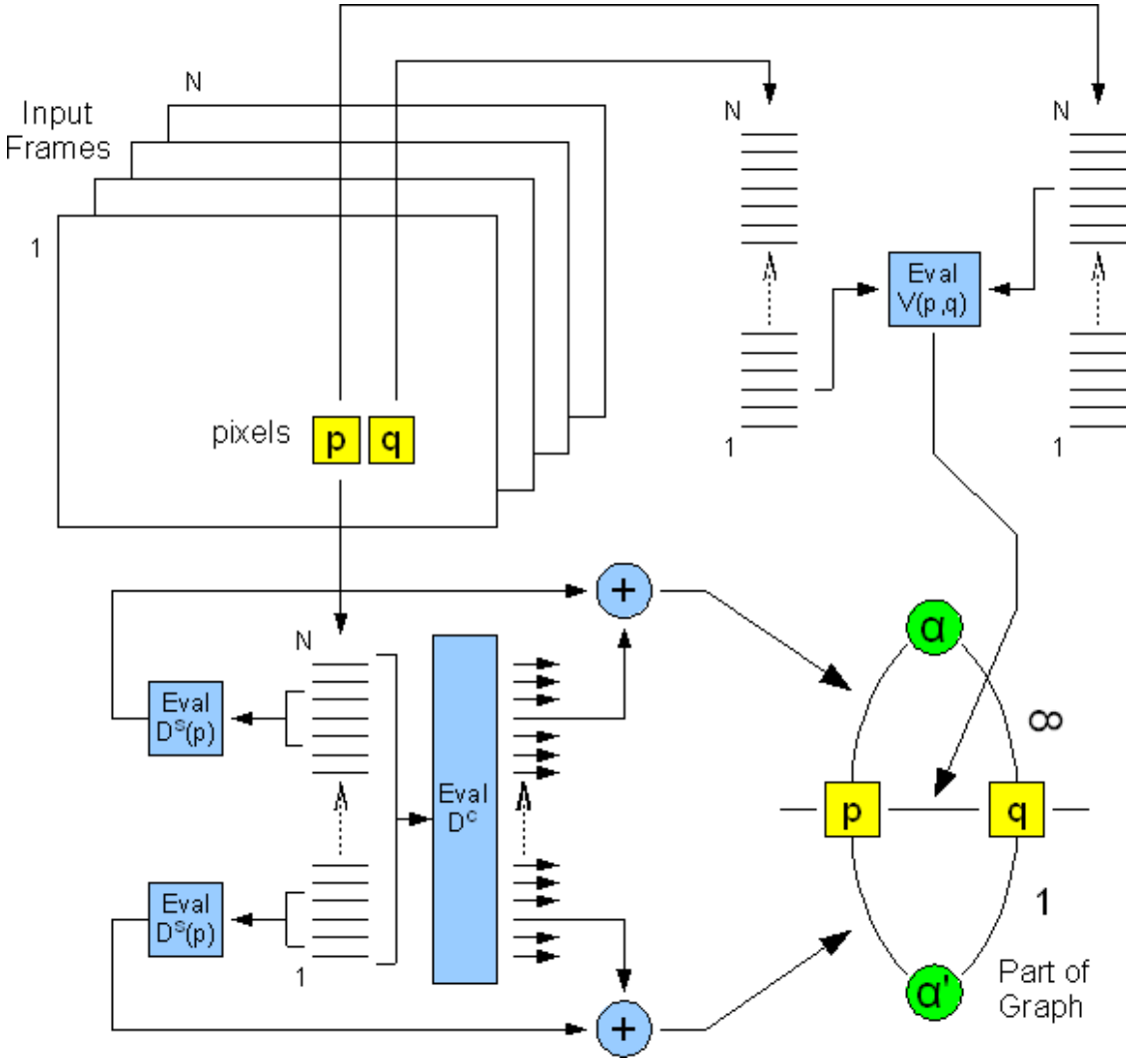


Figure 4.7: Diagram showing how the graph is constructed from a block of input frames using the three types of penalty weight: D^S, D^C, and V. In this example pixel q is already in α. Terminal α' indicates all labels *not* in class α. Although only shown here as a graph for a 1D image, in reality arcs are introduced between all adjacent pixels in a 4-connected manner for 2D images. For the more complicated situation when neither p nor q is in α, the reader is referred to [10].

which possesses the same eigenvalues as $\mathbf{X}\mathbf{X}^T$ and eigenvectors $\mathbf{u}_i = \mathbf{X}^T\mathbf{v}_i$

$$\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{v}_i) = \lambda_i(\mathbf{X}^T\mathbf{v}_i) \tag{4.12}$$

Thus eigen-decomposition is performed on $\mathbf{C}' = \mathbf{X}^T\mathbf{X}$, and the K eigenvectors corresponding to the largest eigenvalues of $\mathbf{C}'$ are retained such that

$$\frac{\sum_{i=1}^{K}\lambda_i}{\sum_{i=1}^{N}\lambda_i} \geq \gamma \tag{4.13}$$

to form a normalized approximate model

$$\mathbf{V} = \left[\frac{\mathbf{X}^T \mathbf{v}_1}{|\mathbf{X}^T \mathbf{v}_1|} \frac{\mathbf{X}^T \mathbf{v}_2}{|\mathbf{X}^T \mathbf{v}_2|} \cdots \frac{\mathbf{X}^T \mathbf{v}_K}{|\mathbf{X}^T \mathbf{v}_K|} \right] \tag{4.14}$$

where γ represents a given fraction of the original energy. A new image vector $\mathbf{y}$ may then be segmented into foreground and background by projecting into the subspace spanned by $\mathbf{V}$ to determine what parts of it are supported by the model. Re-projecting back into the image space and subtracting from the original image $\mathbf{y}$ leaves the residual image vector $\mathbf{r}$ as

$$\mathbf{r} = (\mathbf{y} - \mathbf{m}) - \mathbf{V}\left(\mathbf{V}^T(\mathbf{y} - \mathbf{m})\right) \tag{4.15}$$

Thresholding each element p of $\mathbf{r}$ against a constant τ yields a binary vector $\mathbf{B}$, that may be de-rasterized to the original image aspect ratio to form a binary segmentation mask, which is

$$B_p = \begin{cases} 1 & \text{if } r_p > \tau \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

4.5 Experiment

In order to demonstrate the effectiveness of the proposed scheme, a comparison is performed between the performance of a subspace model derived from pre-filtered backgrounds obtained by the Min-Cut optimization (the 'Min-Cut + Subspace' method) and that of a subspace model built directly from the N input frames (the 'Direct Subspace' method). Both systems were constrained to use only 14 eigenvectors, a number which permitted the former to represent at least 80% of its original covariance energy. In addition, the result of using the Min-Cut *alone* on frames taken from the input sequence is shown (the 'Min-Cut Only' method).

4.5.1 Dataset

For the experiment, a very busy urban scene at a road junction by a metro station containing continuous activity involving both people and vehicles was chosen (see Figure 4.8). Video data was collected over a one hour period in colour at a frame rate of 25Hz, producing 90,000 RGB image frames at a spatial resolution of 720×576 and 8 bit intensity resolution per colour. For the purpose of model building, every 300th frame was extracted from this to provide a set of $N = 300$ images taken at 12 second intervals.

Using $F = 20$ input frames to evaluate each pre-filtered background, the Min-Cut + Subspace model was constructed using 280 images, whilst the Direct Subspace model used the 300

Figure 4.8: Examples illustrating typical level of activity in the chosen challenging urban road scene, in which parts of the true background are persistently occluded.

unprocessed input frames. To accelerate the Min-Cut labelling process, the input frames were sub-sampled to 360×288 resolution. Although the resultant label set consisted of only this number of elements, the backgrounds were reconstructed using 1 label per 4 pixels in order to preserve the original image resolution. The cost balancing constants for the Min-Cut process were set at $\beta = 1$ and $\lambda = 4$, whilst in the consistency cost calculation $\varepsilon = 1$. The threshold for segmentation in both Min-Cut + Subspace and Direct Subspace methods was 20, given that the intensity range for the RGB data was $[0, 255]$. The Min-Cut Only method used 20 frames from the input sequence taken at 3 minute intervals, the binary mask being given by thresholding the difference from the single recovered background. Finally, for all methods, the binary masks were subjected to morphological filtering to remove single and small groups of pixels before display.

4.5.2 Results

The graph in Figure 4.9 illustrates the cumulative distribution of energy (eigenvalues) among the eigenvectors of the covariance matrix for the Min-Cut + Subspace and Direct Subspace models. It is clear that the former requires considerably fewer eigenvectors to reach a certain energy fraction, thus supporting the idea that the proposed hybrid technique attains a more compact model. The sharp rise of the Min-Cut + Subspace curve for energy fractions above 0.95 here strongly indicates the dominance of a small number of eigenvectors in the model, as intended.

Figure 4.10 shows typical output from the Min-Cut pre-processing stage. As illustrated by the left image, foreground object removal is not always complete. If the 20 input frames used to produce this particular background estimation contain the stationary car in most frames, it will

be indistinguishable from the background. Although such foreground objects still contaminate the subsequent subspace model, the pre-processing removes so much of the foreground clutter that overall, considerable advantage is gained.

The segmentation masks B for two typical input frames, which were *not* used to build the models, are shown in Figure 4.11 for all three cases of the experiment. The Min-Cut + Subspace model clearly demonstrates the cleanest segmentation of objects which, for a typical surveillance application, are required to be foreground.

In particular, for the Direct Subspace model, some of the white road markings and the shadows from the traffic signal posts in the bottom right of the images are breaking through into the foreground compared with the output from the Min-Cut + Subspace model. Both of these effects constitute *modes of variation* which a dynamic background model will attempt to subsume when exposed to training data. Many of the road markings are intermittently occluded by vehicles passing over them, whilst the shadows move slowly over time, but both cases count as variations as far as a background model is concerned.

As previously described, a subspace model can only absorb as many modes of variation as allowed by the number of eigenvectors from which it is constructed. The more scene elements changing over time that there are to be described, the less effectively a constrained eigenspace model can wholly satisfy the requirement. For the Direct Subspace model, most of the eigenvectors end up committed to explaining the vehicles and people moving in the scene, which are the dominant sources of variation. The ultimate result is inability of the model to adequately capture the more subtle variations present, hence the appearance of various nominally background elements in the foreground mask - in this case, particularly the aforementioned shadows and road markings. The knock on effect is thus desensitization of the overall model and therefore poor foreground/background discrimination.

On the other hand, the Min-Cut + Subspace model sees the white road markings as being largely constant due to the relatively clean short-term background estimates derived by the Min-Cut stage. Furthermore, the dominant modes of activity - vehicles and people - are also by and large removed. Thus the eigenvectors of the subspace part of this model remain free to express subtle variations, such as the gradual shift of shadows as the sun moves.

For the Min-Cut Only experiment, the images at the bottom of Figure 4.11 show considerable problems with shadows, not only on the road surface, but also at building corners and edges. Be-

cause no variability at all is catered for in the single recovered image, the changing shadows at the edges and walls of buildings have not been accommodated. Due to the result of the combinatorial optimization, the output of which is a single image compiled from samples taken throughout the whole hour of the input video, the chances of a lighting match with a single arbitrary input frame is small. Different parts of the background model will match different lighting conditions, but almost all sources of image variation will be wrongly represented for most of the training data. So the Min-Cut Only approach is of limited practical use.

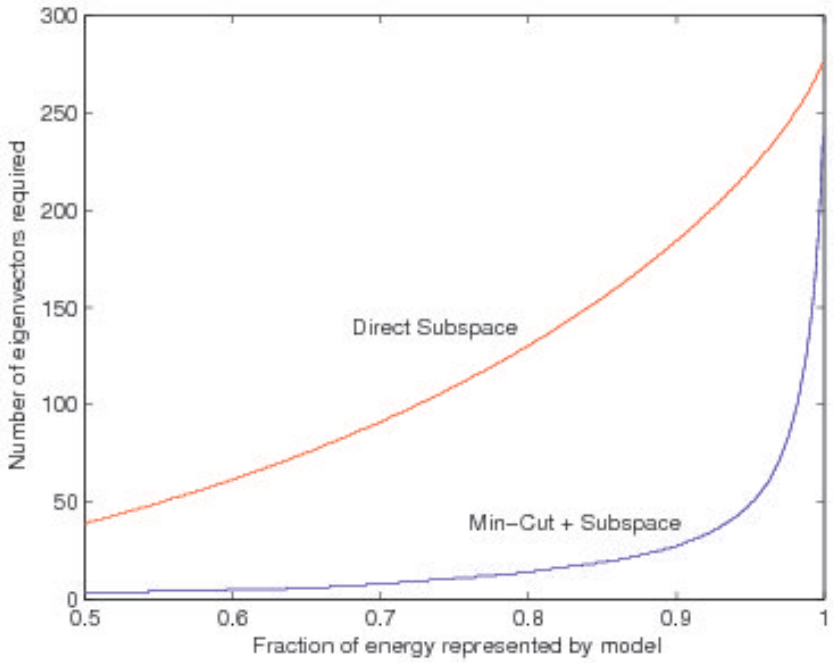


Figure 4.9: Graph showing that Min-Cut + Subspace consistently requires considerably fewer eigenvectors to retain a certain fraction of energy than the Direct Subspace method.

4.6 Accuracy of Short-term Estimates

The results of the previous section demonstrate the utility of the short-term background images produced by minimizing the objective function in Equation (4.1) over a block of candidate input frames. However, ultimately such background images are still only *static estimates* of the true background. Even amongst recovered backgrounds in which all the people and vehicles have been removed, minor differences in local shading and hue are apparent. Figure 4.12 illustrates the subtlety of variation in backgrounds only minutes apart. Although each pixel represents a legitimate local estimate of the scene, and the combination in which they are chosen is determined

Figure 4.10: Two examples of typical output from the Min-Cut pre-processing stage. Left: Imperfect object removal. Right: Near optimal background recovery.

globally according to Equation (4.1), even this may not carry any direct guarantee of validity or absolute consistency in the composite result. For example, some image areas may be locally brighter or darker with respect to any single frame if the ambient lighting conditions happened to shift throughout the duration of the block of candidate frames, as there is no *global* constraint term in the optimization to suppress unevenness. So overall, the accuracy, realism and ultimate usefulness of the short-term estimates might be called into question. However, it is the express purpose of the subsequent processing stage, in this case the eigenspace model, to compactly represent such image variation. The global inconsistency and realism artifacts inherent in this type of pre-processing are attenuated by the averaging effect of the following eigenspace model, which is in turn constrained to representing the most significant modes of variation by limiting number of eigenvectors. But at the pre-processing stage neither consistency nor realism need necessarily be seen as goals, it is the effectiveness of the overall system which is of primary interest, and removal of foreground clutter should be seen as the overriding requirement of pre-processing.

The large number of parameters to be determined, and the intrinsic stochastic nature of the input blocks can lead to somewhat chaotic results in terms of reliability of object removal, but general trends may be identified. The remainder of this section focuses on the effects on the outcome of the optimization process of the following controllable parameters:

1. Dependency on constants β and λ

2. Number of candidate frames in the input block

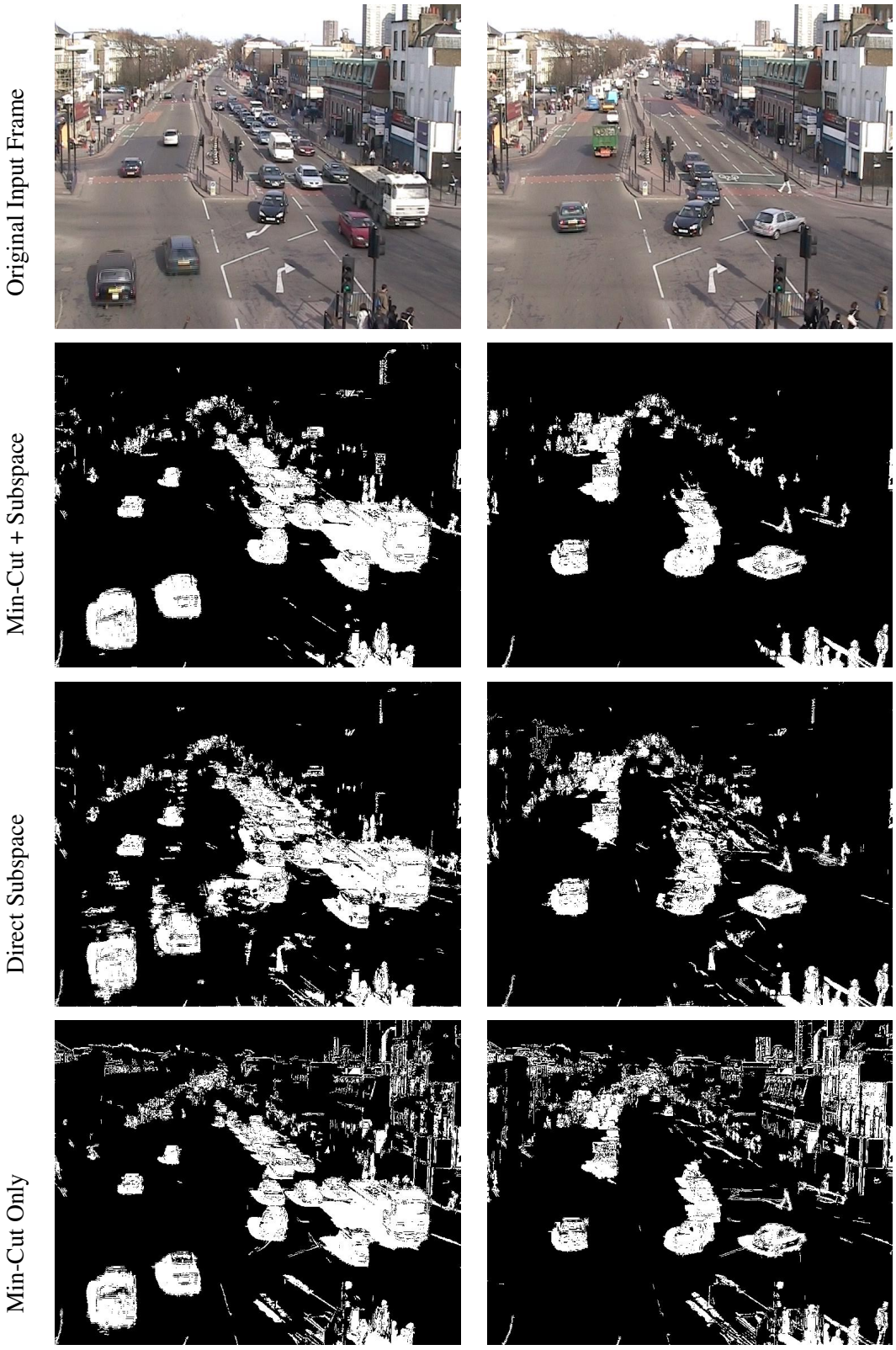


Figure 4.11: Segmentation of two frames using Min-Cut + Subspace, Direct Subspace, and Min-Cut Only methods. Min-Cut + Subspace shows the best segmentation here.

3. Sampling rate of candidate frames

4. Initial frame labelling of the pixels

The last three effects may be evaluated directly by considering the overall energy of a given labelling according to the minimized objective function. If a given set of parameters result in a higher final energy value than another set, then the resultant labelling is less successful - the true minimum has not been achieved, bearing in mind that α-expansion yields only an approximate optimum by design. The effect of varying β and λ cannot be evaluated this way, since the two constants directly affect energy values. However, the relative merit of a given combination of β and λ may be determined by considering the number of eigenvectors needed in an eigenspace model to represent a fixed fraction of the energy of variation. This is directly linked to the distribution of energy among the eigenvalues - better combinations of β and λ compress most of the energy into just a few.

4.6.1 Influence of Parameters β and λ

Combining Equations (4.1), (4.2) and (4.9) yields an expression for the objective function in which it is clear that the balance between the three optimization mechanisms, Stationarity cost (D^S), Consistency cost (D^C), and the Symmetric Pairwise cost (V) is controlled by the two constants β from Equation (4.2) and λ from Equation (4.9) according to the relationship

$$E(\mathcal{F}) = \sum_{p \in \mathcal{P}} D_p^S(f_p) + \beta D_p^C(f_p) \; + \sum_{\{p,q\} \in \mathcal{N}} V_{pq}(f_p, f_q) \tag{4.17}$$

The absolute value of E is immaterial here since it is only the minimum of E over the labelling space which is sought, hence two constants are sufficient. For all 100 combinations of 10 different β values and 10 different λ values, an eigenspace model was built from 25 recovered backgrounds taken from the scene in Figure 4.8 over a period of 33 minutes. Input blocks were made up from 20 candidate frames of resolution 360×288 taken at 12 second intervals, thus spanning a period of 240 seconds overall, with adjacent blocks overlapping by 160 seconds. For each combination, the cumulative distribution of energy was calculated starting with the largest eigenvalue. The number of eigenvectors required to represent at least 80% of the total energy is documented in Table 4.1.

These results show that there *is* a combination of constants leading to a minimum in required eigenvectors, and that this minimum remains fairly insensitive to the exact values chosen. This

Figure 4.12: Left: Three recovered backgrounds. Right: Recovered backgrounds enhanced for illustration purposes by subtracting 75% of the mean of the three, and multiplying by 4. Such backgrounds derived from a localized time period can exhibit very subtle differences in appearance due mainly to the shift in lighting conditions throughout the input block. Note the colour and intensity mismatches. Artifacts are apparent when adjacent image regions are selected from differently lit input frames. The optimization process does not impose *global* constraints on input pixel combinations.

$\lambda \Rightarrow$	0.01	0.25	0.5	1	2	4	8	16	32	64
$\beta = 0$	10	9	8	9	9	10	10	10	9	10
$\beta = 0.0625$	8	8	9	9	9	9	10	10	9	10
$\beta = 0.125$	8	8	8	10	9	10	10	9	10	10
$\beta = 0.25$	8	7	8	8	8	8	9	10	9	10
$\beta = 0.5$	8	7	7	8	8	8	8	9	9	10
$\beta = 1$	8	7	7	7	7	7	8	9	9	9
$\beta = 2$	9	7	7	7	7	7	7	7	9	10
$\beta = 4$	9	7	7	7	7	7	7	7	8	9
$\beta = 8$	9	8	7	6	7	7	7	7	7	9
$\beta = 16$	10	8	8	7	7	7	7	7	7	7

Table 4.1: Number of eigenvectors needed to account for 80% of background image energy for different combinations of β and λ. An optimum is found at $\beta = 8$, $\lambda = 1$. Crucially, according to this metric, the result shows stability at 7 eigenvectors over a wide range of β and λ.

is a useful feature of the technique from a practical point of view since a level of robustness is offered - even values fixed in advance are likely to yield plausible results. These findings are broadly in agreement with the less detailed results published in work by Cohen [19].

Figure 4.13 shows the recovered backgrounds and the corresponding label sets which control their make up for a *single* input block over 4 selected values each of β and λ across the range. Each label is represented by a unique colour, but especially for the simpler labellings, 10 or less of the available labels dominate. The figure clearly shows that a given combination of constants yields an ideal recovered background amidst other combinations which all lead to some defect or other. In line with expectation, larger values of λ give rise to simpler labellings with less different patches. As the relative cost of label boundaries due to the Symmetric Pairwise cost increases, the algorithm favours solutions with less total perimeter between labels. The same effect is also noticed as β is reduced, since in the absence of significant stationarity penalty in a region, the Symmetric Pairwise cost again remains the dominant term in Equation (4.17). It is also interesting to note the stability of solutions in terms of which labels are chosen - the label map mutates quite gradually as the value combinations are traversed. This again supports the concept of algorithm robustness.

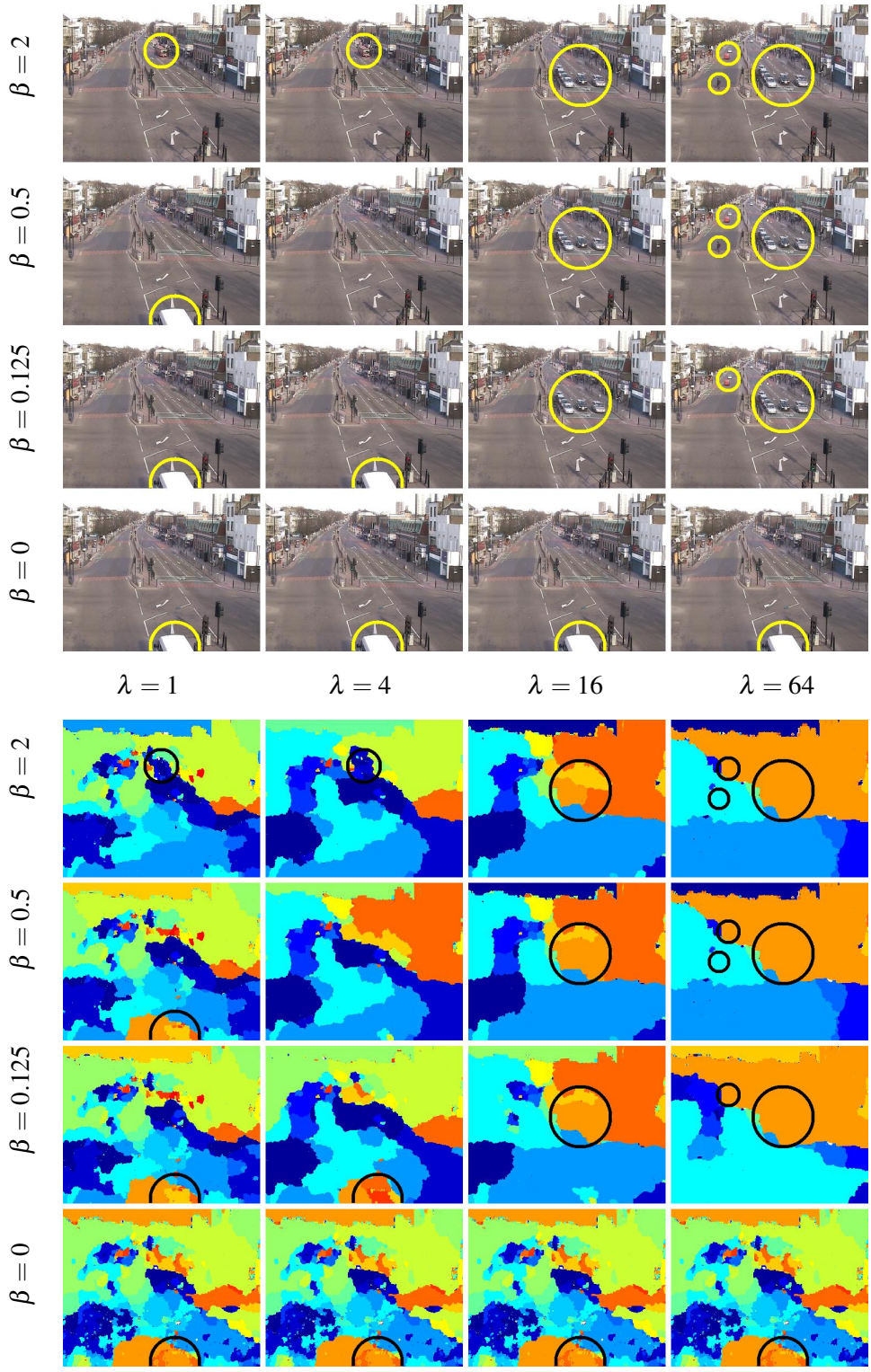


Figure 4.13: Effect of varying β and λ. Top: Recovered backgrounds. Bottom: Coloured labels of corresponding selected frames. The optimally 'clean' background is produced with $\beta = 0.5$ and $\lambda = 4$ in this case. Other combinations of β and λ lead to the highlighted defects. Note how smaller λ values on the left lead to a more fragmented label selection.

4.6.2 Input Block Size

Each recovered background is compiled from an input block of candidate frames, whereby pixels may be chosen from any of the frames according to the optimization of the objective function in Equation (4.17). However, the question arises as to how many candidate frames there should be to choose from, and how far apart in time they should be spaced.

The ideal solution would be to make *every* input frame from the source over a long period of time available for contribution to the background. However, the computational complexity of both α-expansion and eigenspace model maintenance prevents this from being possible given the processing power of the average PC, and say a 25Hz frame rate. For the α-expansion algorithm, the most costly process, complexity is essentially linear in N, where N is the number of frames to choose between, although more α iterations through the N labels may required to achieve stability if there are many more very similar input frames. In the current implementation, stability is usually achieved after 3 or 4 iterations with $N = 20$, meaning that *no* expansion moves occurred during the whole of the most recent iteration, at which point the algorithm terminates, being unable to reduce the objective function energy further.

Evidently, even if N is constrained by available resources, the timescale over which the N frames are distributed is still variable. In the case of the cyclic traffic junction scenario of Figure 4.8, the possibility of finding a realistic background estimate is dramatically increased if the input block spans *at least* a complete cycle of the traffic lights. Under this condition, the algorithm is exposed to the maximum number of different phases of behaviour exhibited by the junction traffic, and thus the best chance for enough regions of stable background to be visible. On the other hand, arranging N such that many cycles are spanned leaves open the strong possibility that shorter phases of behaviour in a cycle are missed. So overall, it is a question of achieving adequate *sampling rate* for a given constraint on N, and a particular timescale of events in the target scenario.

The experiments so far have shown that acceptable results for the traffic junction may be obtained with 20 frames taken at 12 second intervals over a period of 240 seconds, where the natural junction cycle time is around 112 seconds. The algorithm can only perform as well as the supplied data permits - realistically, piecing together a good background estimate relies on finding frames where all of the background is visible and stable over several frames at some point during the input block.

For the traffic junction scenario of Figure 4.8, N was varied between 2 and 20 frames distributed equidistantly in time over 9 separate 240 second sections of video. Figure 4.14 shows the final energy achieved by α-expansion averaged over the 9 clips. Clearly, for this particular scenario, the results show that little is to be gained by selecting $N > 12$ from an energy minimization point of view. Meanwhile, Figure 4.15 shows visible results of varying N over a single 240 second section of video. Various defects are to be seen in backgrounds where $N < 14$, which is broadly in agreement with the previous energy-based result. Thus it would seem that there is a significant connection between absolute α-expansion energy and visible purity.

A practical algorithm would benefit from automatic selection of N, even though it has an upper constraint imposed by processing time. A possible algorithm could be based on regulating N so that system operation occurs on the lower gradient part of the curve in Figure 4.14. This might be achieved by continually trying to reduce N in opposition to a criterion of maximum negative gradient.

4.6.3 Input Sampling Rate

For a given value of input block size N, the *sampling interval* may be varied to accommodate different activity time-spans into the input block. A parallel may be drawn here with the *Nyquist* rate from sampling theory [117] which stipulates a minimum sampling rate dependent on the bandwidth of the signal being sampled. The general idea is to enforce a sufficiently high sampling rate to prevent transient detail being missed between samples. The principle is equally applicable here in order not to miss areas of stable background if they remain unoccluded for short periods of time. This imposes a *minimum* on the sampling rate, whereas by similar argument, a block of candidate frames which does not span a wide enough time frame to encompass all modes of scene activity may also fail to produce a useful result. For constrained N, this imposes a *maximum* on sampling rate. The sample rate thus needs to be suitably *matched* to a scene's activity timescales.

To study the effect of varying the sampling rate, a set of eighty equally spaced frames representing exactly one single cycle of traffic junction activity was selected from video data. Using an input block size of N=16, blocks of frames were taken at 8 different sampling rates by picking every m_{th} frame from the set, where $m = \{1 \ldots 8\}$, taking modulo-80 frame indexes for large values of m. Furthermore, to avoid bias introduced by asymmetric scene activity, the final energy from α-expansion was determined at all 80 potential block starting points for each sampling rate, and the mean calculated. The graph of Figure 4.16 clearly shows a minimum in final energy at a

sampling interval of 175 frames, corresponding to $m = 5$. It turns out that at this sampling rate, the $N = 16$ input frames optimally span a complete cycle of junction activity. Given that lower energy is associated with more satisfactory background recovery, it seems that a feedback mechanism capable of minimizing energy with respect to sampling rate could provide an automatic technique for adapting the latter to scene activity time-span.

In general, the possibility of there being an optimal sampling rate depends on the behavioural content of the scene. The concept of periodic scene activity and how to exploit it is dealt with extensively in Chapter 5.

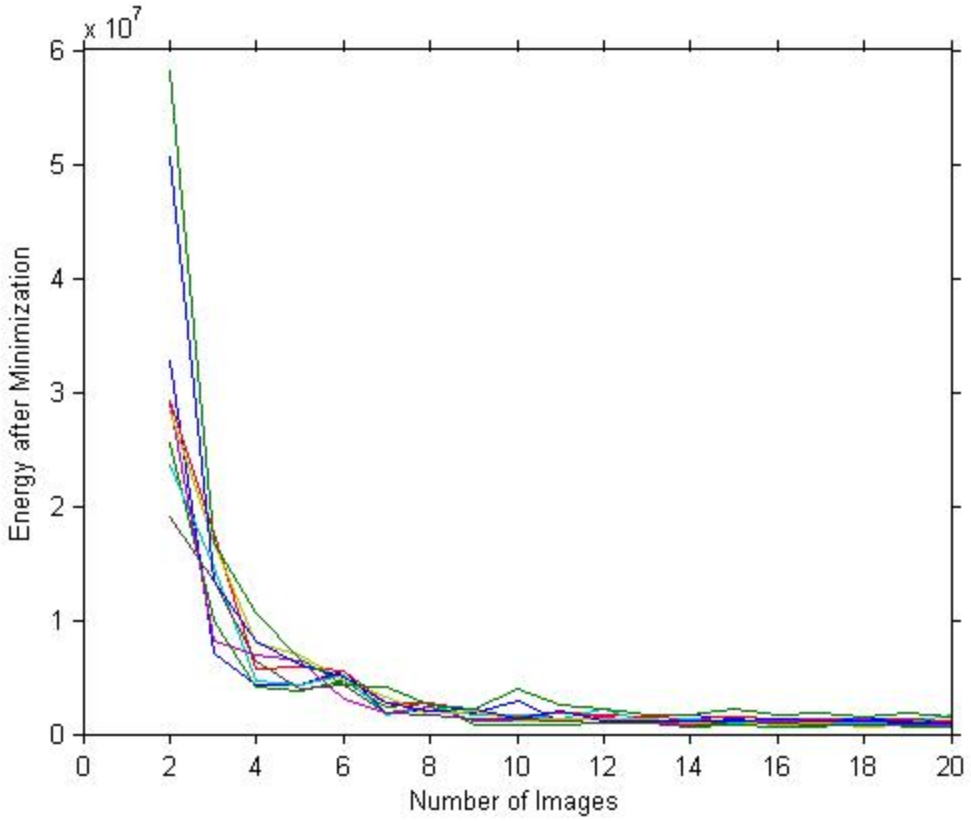


Figure 4.14: Effect of varying the input block size on the minimum energy achieved by α-expansion. Examples using between 2 and 20 input images for 9 different parts of a video sequence are shown. A law of diminishing returns is evident with more than about 8 input images per block for this particular scenario.

4.6.4 Initial Pixel Labelling

The α-expansion process detailed in Section 4.3.2 leads to approximate solutions of the N-cut problem by iteratively applying binary graph cuts to a set of pixel labels in order to form the

Figure 4.15: Examples of recovered background produced by α-expansion using various numbers of input frames. Good foreground rejection is achieved for $N = 14$ and above.

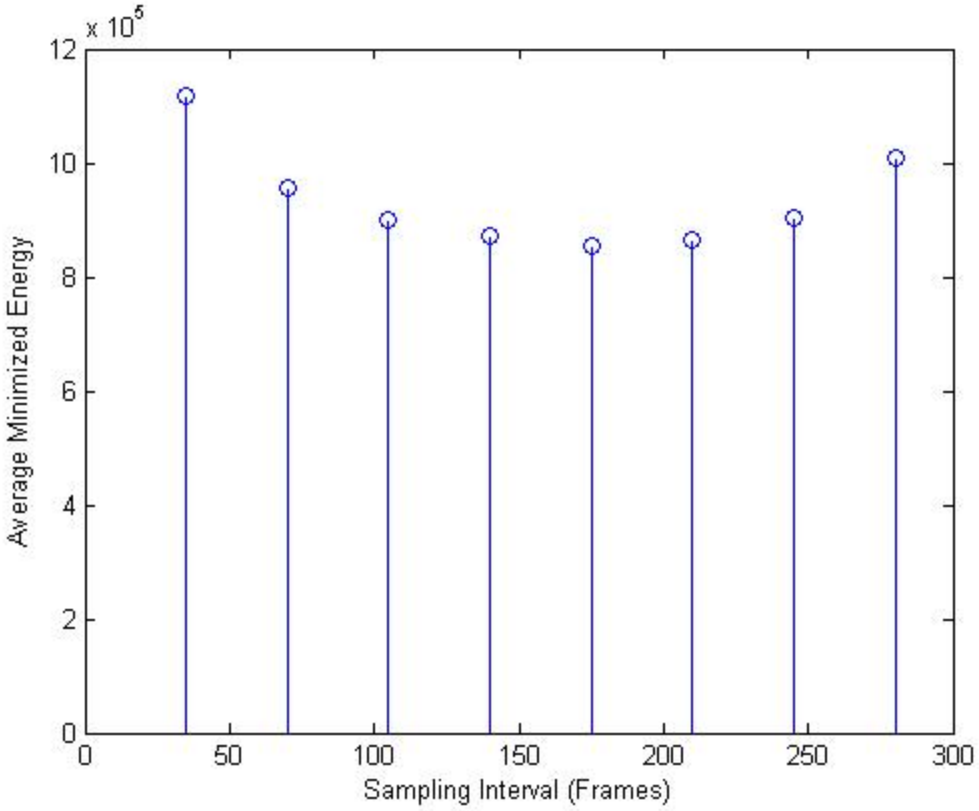


Figure 4.16: Minimized energy as a function of input frame sampling rate for the scenario in Figure 4.8 using N=16 frames per block. From the graph, the lowest energy value is achieved with samples approximately 175 frames apart, which at 25FPS corresponds to a sampling period of $\frac{175}{25} = 7$ seconds. Thus the $N = 16$ input frame block is equally distributed over a $7 \times 16 = 112$ second period - which closely matches the cycle time of the junction. Only under these conditions is the algorithm presented with the most comprehensive combination of scene representations from which to work. The shape of the plot above thus opens up the possibility of automatic sample rate adjustment based on a minimum energy criterion.

desired background image. Under this scheme, each frame represents one of the unique candidate frame labels, and a complete iteration involves a set of N such graph cuts, with each label taking its turn at being α. As a direct result of each binary graph cut, a pixel may migrate into the α class from any other class if such a move is energetically favourable according to the objective function. Before *and* after each cut, all pixel labels in the set forming a complete image must have a current label from the N candidate frames.

The question is *which* initial label should each pixel take on before the α-expansion optimization process begins?

Three potential answers spring to mind:

1. Random assignment
2. Fixed pre-determined pattern
3. Starting point determined from the data itself

At first sight, the third choice seems to rule itself out since the whole point of alpha expansion is to *find* the optimal label pattern. However, from Equation (4.17) it becomes obvious that the data terms D^S and D^C depend only upon an individual pixel's own labelling rather than that of its neighbourhood. Thus a plausible starting label for each pixel may be chosen as that which minimizes a weighted sum of these components $D^S + \beta D^C$. Moreover, from a practical point of view, this quantity has to be evaluated in any case either before or during α-expansion.

A fixed pre-determined pattern might reasonably consist of setting all initial labels to point to a single frame. The frame numbered N seems a good choice, since there is maximum chance of most α-expansion moves occurring during the first iteration of alpha, thus leading to the most stable solution as quickly as possible.

Random assignment would seem to be the worst way to initialize the labelling, since chaos is maximized and all potential prior information is disregarded.

The relative merit of the three initialization schemes was investigated experimentally using 5 sets of N=20 candidate input images taken from the scene in Figure 4.8. The final global energy achieved after minimization by α-expansion was used as the metric for comparison. As discussed previously, lower resultant energy is associated with more satisfactory labelling. For each of the 5 image sets, the initial labels were set globally to each of the N labels in turn, to 30 different random label sets, and to the label set dictated by $D^S + \beta D^C$. The resulting energies are shown in Figure 4.17 in blue, green, and red respectively. In general these results show that the final energy is broadly similar regardless of initialization scheme. But in more detail, it seems that the random labelling reliably produces amongst the lowest energies achieved, although only in image set 4 was it significantly better than the calculated labelling. The single label initializations produced the most variable (shown in cyan) set of results, and the worst of the methods. This may be because such labelling initially incurs very little energy contribution from the pairwise interaction terms (V), leaving the label set certainly in a minimum, but unfortunately a *local* minimum from which the algorithm fails to break out.

Analysis of the actual labellings returned by α-expansion also supports this conclusion. Figure 4.18 illustrates the difference between all pairs of resultant labellings for each of the 20 single label, 30 random, and 1 calculated starting points. When comparing two label sets, a different label at a pixel counts as one, and matching label as zero. The labelling differences are summed over the 5 input image sets, and displayed as a disparity matrix in which higher intensity signifies greater disparity. Most notably, the single label initializations (which already exhibited the worst energies), tend to cause final label combinations which disagree with each other considerably more than those produced by random and calculated initializations.

Overall it can be concluded that for this particular application, α-expansion produces quite consistent labelling results and final energies, and that *random labelling* is at least as effective an initialization as anything else.

4.6.5 Automated Parameter Exploration

Whilst it is reasonable to obtain qualitative measurements of the effects of the parameter adjustments described in this section, an altogether more extensive and rigorous analysis is also desirable. If the *ground truth* for a given background at a particular time were known, it would be possible to assess any other derived background with respect to it, possibly just by summing the absolute colour intensity difference of all pixels across the image. Then the search space of the parameters described could be covered systematically until a minimum overall 'background image error' were discovered. Furthermore, plots could be obtained of background accuracy in relation to the particular selection of any given parameter.

Even though the true background is not *actually* known, reasonably good backgrounds, such as that depicted with $N = 14$ in Figure 4.15, could be obtained with modest amounts of human intervention. Typically backgrounds without any obvious people or cars are to be sought. Although the effect of subtle image variations, such as those revealed in Figure 4.12, are not directly accounted for, this problem may be mitigated to some degree by obtaining a series of 'clean' estimates and extracting mean and variance for each pixel intensity. Hence, if the distribution of acceptable colours at a pixel is assumed to be normal, any query image may be assessed by how many of its pixels are within a specific number of standard deviations of the mean.

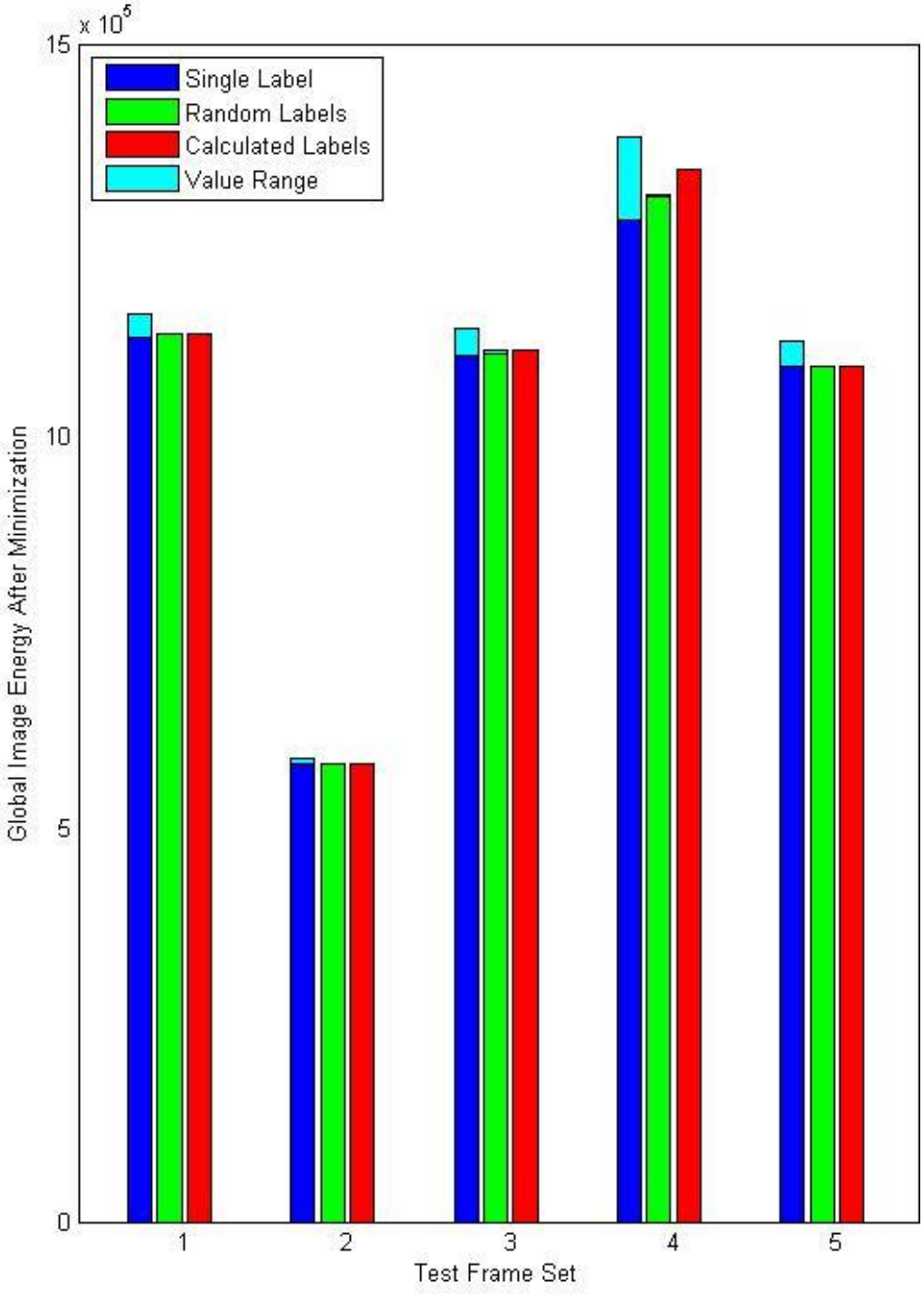


Figure 4.17: Effect of different initial labelling schemes on result of final α-expansion graph cut energy for 5 different input frame sets. Colour bars show results for: all pixels starting with each of the N possible labels (blue), pixels assigned random labels in 30 different configurations (green), and pixels assigned initial labels based on the minimum of invariant data term $D^S + \beta D^C$ (red). The consistently low energies and small variances associated with random initial labelling make this the favoured technique.

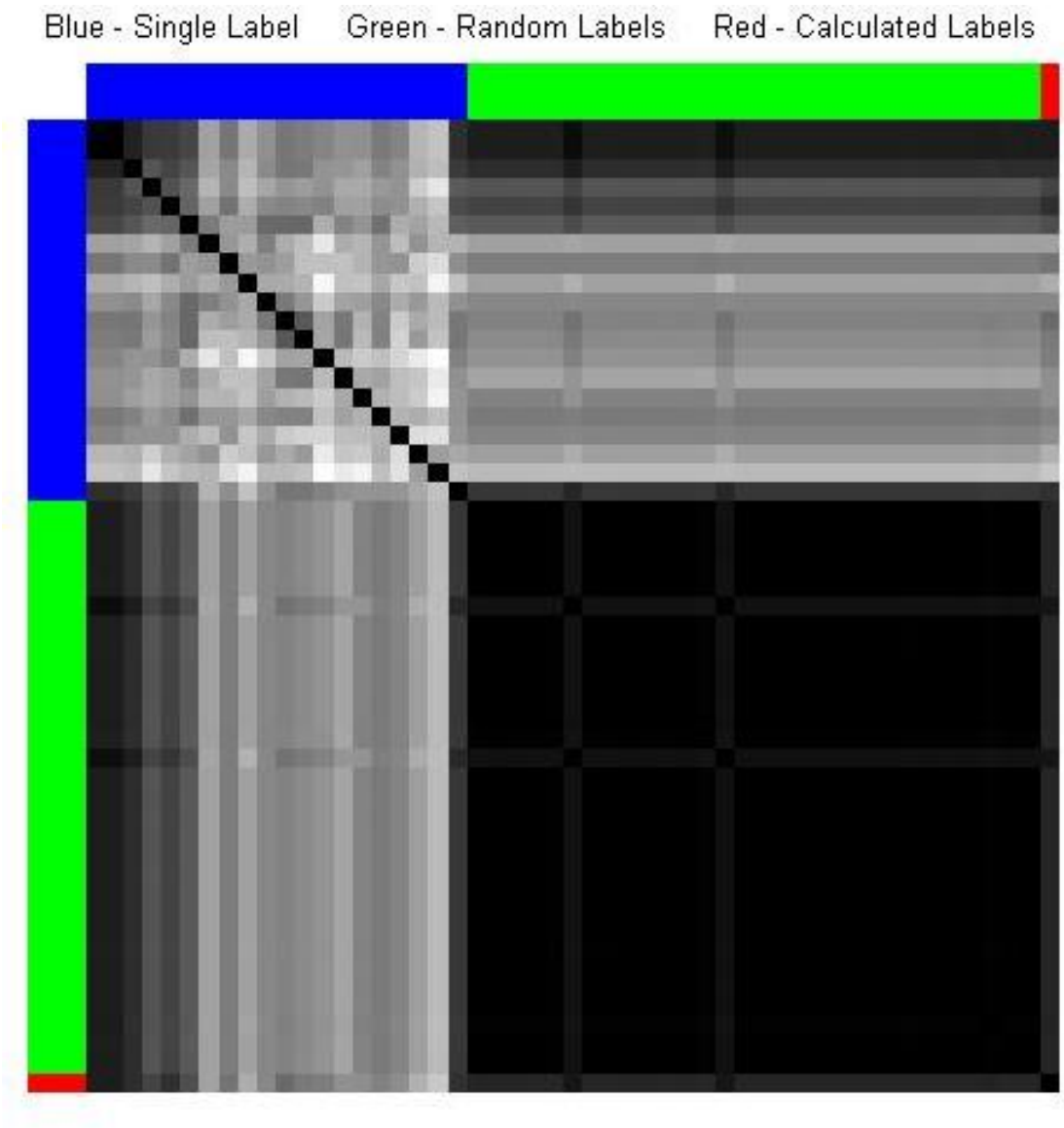


Figure 4.18: Disparity matrix showing the disagreement between computed label sets averaged over the 5 test frame groups. Higher intensity signifies a greater number of pixel locations which do not agree between the respective column and row label coordinates. Initialization of all image pixels to a single label (blue) tends to produce the greatest disparity amongst the final solutions. On the other hand, the random (green) and calculated (red) initializations almost always converge to the same result. No labelling disagrees with itself, hence the zeros on the leading diagonal.

4.7 Discussion

The success of the hybrid Min-Cut algorithm may be explained by consideration of its two constituent parts separately. The more effectively that one can eliminate foreground objects from the *short-term background* images at the Min-Cut stage, the more compact becomes the eigenspace

model for a given energy fraction. The Min-Cut process can only remove foreground objects if they are not consistently placed in the F source frames. Turning this around, the true background can only be found if it is found to be dominant in relation to the costing rules defined.

There is considerable scope for determining an optimal selection of source frames from real-time incoming video. The imperfect object removal illustrated in Figure 4.10 is typical of what happens when the choice of source frames is unsuitable. The present method of taking $F = 20$ frames at 12 second intervals is perhaps rather arbitrary and crude. Naturally, the combinatorial optimization will take longer if blocks of more than 20 frames are processed, but using fewer frames might cause some areas of true background never to be discovered.

The optimal sampling interval will depend on the temporal content of the scene. In the experimental example, the activity of people and cars is governed largely by the sequence of the traffic lights on the junction, the cycle time of which was measured to vary between 98s and 116s. Waiting cars accumulating at a red light could, for instance, constitute background if most of the F frames were taken while the cars waited.

An altogether more intelligent way of selecting frames for the optimization stage is required in order to maximize the capability of the pre-processing for elimination of unwanted foreground. One possibility would be to add a further term to the cost function in order to exclude choice of pixels or frames which are too distant, according to some metric, from a current version of the model. This hints at an on-line algorithm with the capability of automatic adaption. However, this should be pursued with care, since the resultant system would contain a feedback loop which may invite bootstrapping and instability problems if the prototype does not initialize properly.

Although a subspace model was selected for the second stage, possibilities certainly exist for incorporating other techniques. A per pixel model might need less Gaussian components, or perhaps even only one, if the pre-processing tends to reduce multi-modality in colour space. Dispensing with the Expectation-Maximization stage [27] that usually goes with Gaussian Mixture Models could lead to considerable saving in processing time.

However, it is believed that the subspace model as chosen here has the best possibility of success, since it excels in modelling the global linkage of changes between pixels rather than the spatially localized disturbances which the Min-Cut stage tends to attenuate. Such a property makes it ideal for a compact model of some aspects of daylight variability.

Recent work in [134] shows how accelerated Mincut/Maxflow graph cuts can be imple-

mented on the array processor of a common graphics card. Using a distributed version of the *Push Relabel* algorithm due to Goldberg and Tarjan [42], the processor achieves 150 graph cuts per second on an image of size 640×480 pixels, thus paving the way to a realizable background modelling system based on common hardware and the technique proposed in this chapter.

4.8 Summary

It has been demonstrated that a hybrid background modelling scheme consisting of a pre-processing stage based on the combination of a Min-Cut/Max-Flow algorithm *and* a conventional subspace model shows advantage over the conventional subspace model operating alone. Suitable for application in outdoor environments, this chapter has succeeded in developing a system tolerant of lighting changes, whilst showing robustness to a high level of activity in a complex scene.

Although rather computationally intensive, the new algorithm produces useful improvements when running at a sub-multiple of the true frame rate. With refinements in the software architecture, it is believed that the Min-Cut + Subspace method *does* have a useful role to play in practical applications, but in any case is valuable as a vehicle for future research in this direction.

However, in spite of the method's success in providing support for scene elements which are persistent in time, at the expense of those which are not, there are situations in which this behaviour is still not quite suitable. Objects which appear in a scene at regular intervals in time convey no new information in entropy terms, since they are entirely predictable, and as such may legitimately be considered part of the background. Yet the model just presented would eliminate these periodically appearing objects thus constraining them to be foreground. The next chapter will explore a model specifically capable of explaining *dynamic periodic* behaviour in a scene.

Chapter 5

Dynamic Scene Decomposition

Many traffic junctions are regulated by lights controlled by a timing device of considerable precision, and it is in these situations that a model which learns periodic spatio-temporal patterns is advocated, in order to highlight anomalous events such as broken-down vehicles, traffic accidents, or pedestrians jay-walking. More specifically, by estimating autocovariance of self-similarity, used previously in the context of gait recognition, a scene can be characterized by identifying its global fundamental period. As a model, a spatio-temporal grid of histograms built in accordance with some chosen feature is introduced. This model is then used to classify objects found in subsequent test data. In particular the effect of such characterization is demonstrated experimentally by monitoring firstly the bounding box aspect ratio, and secondly the optical flow field, of objects detected on a road traffic junction. The results enable the model to discriminate between activities of people and cars sufficiently well to provide useful warnings of adverse behaviour in real time. For example, it should be possible to identify a pedestrian trying to cross a road at a time when cars are normally moving through the junction. Namely, this calls for a model possessing a certain *temporal contextual awareness*, applicable directly to scenes exhibiting *recurrent* background activity. Other potential applications include heart image analysis and monitoring of industrial systems involving repetitive motion.

5.1 Periodic Scene Activity

There is considerable interest in techniques capable of identifying anomalies and unusual events in busy indoor and outdoor scenes, e.g. shopping malls and road junctions. Currently countless

people are deployed to watch and monitor CCTV screens in the hope of identifying criminal activity, untoward behaviour, and serious but non-malicious situations. A fundamental challenge in computer vision research is to augment such human effort by devising algorithms capable of isolating and displaying events of interest in a clear, uncluttered way and with a relatively low false alarm rate. Considerable research effort has produced systems which learn *statistical* scene content both at the pixel level [125] and from a global perspective [94] with a view to segmenting an image into the usual (background) and unusual (foreground). Many approaches achieve this by exploiting deviations in spatial appearance from some expected norm accumulated by a model over time. Furthermore, by relating foreground object size, and possibly shape, to areas within the scene, it becomes possible to identify people and vehicles in the 'wrong' place. However, generally such models are still oblivious to *relative* event timing.

In this chapter it will be shown that much can be gained from explicitly modelling temporal aspects in detail. The desire is to extend the definition of 'unusual' to the temporal domain such that the presence of an object is treated explicitly in a spatio-temporal context rather than modelled as a deviation from a single accumulated distribution. This approach is aimed specifically at modelling scenarios in which periodic behaviour is present, with the aim of identifying people and vehicles not just in the wrong *place*, but also those in the *right* place but at the wrong *time*.

Many examples of visible periodic behaviour are encountered in everyday life, displaying periodicity on a wide variety of timescales. Rotating structures such as fan blades, a swinging pendulum, and shadows caused by the sun moving across a cloudless sky day after day, represent three scenarios in which appearance repeats itself regularly, so that each cycle is all but indistinguishable from any other. Such scenes thus exhibit *self-similarity* with respect to previous versions of themselves, as the generating agent returns to the same point in state-space previously visited. This is in stark contrast with other types of behaviour consisting of stochastic events, or groups of events initiated at random time intervals, whereby self-similarity is not anticipated or guaranteed at any time.

From an *information theory* point of view, periodic events carry no new information at all - they are entirely predictable. Thus the presence of such behaviour in a scene may be considered as background activity, being non-salient and uninteresting with regard to novelty. Crucially, for these very reasons, it would be a waste of available information *not* to eliminate such recurrent events from the foreground, since periodic behaviour is by definition a form of stationarity, and

thus belongs as part of a *dynamic* background. Taking the idea further, if the *statistics* of a scene recur in a periodic manner rather than just an object or event, then the same argument may be applied.

Evidently not all behaviour in a given scene is necessarily periodic. Different parts of a scene might possess a range of fundamental frequencies, depending on the elements from which the scene is constructed, and how they are linked. Some parts of a scene might be characterized by multiple non-harmonically related periodicities, whilst others display only approximate periodicity - in communication theory terms, a fixed centre frequency which is Frequency Modulated (FM) by noise. In practice much of a typical scene will exhibit no detectable frequency components whatsoever, and thus may be considered *non-periodic* or *aperiodic*.

Road junctions regulated by traffic lights are ultimately controlled by a precise timing unit, such that the various vehicle paths are each given a certain amount of time to cross the junction. Traffic engineering strategies for achieving effective junction throughput whilst constrained by safe operating practices are discussed extensively in [69]. However, signal timing controller operation may be broadly divided into three modes. The simplest is *Fixed-rate Mode*, whereby the time given to each flow direction is preset during installation or maintenance and doesn't otherwise vary. In *Actuated Mode*, the cycle is essentially fixed, but a given phase may be lengthened or inserted into the sequence by activation of pedestrian or vehicle sensors. The third, less commonly found, is *Adaptive Mode*, in which timing plans are modified automatically in response to changing demands in traffic load as measured by a variety of different types of sensor. Fortunately, many junctions fall into the first category, and the constant cycle time opens up strong possibilities for exploiting the regular traffic behaviour as part of a dynamic background model.

On an apparently unrelated problem, much is to be found in the literature concerning gait characterization, modelling and identification, as detailed in [22, 100, 115]. Generally these methods work by analyzing the relative motion of linked body members, which are of course all related by the same fundamental frequency. However, the parallel between this and modelling traffic at a road junction is surprisingly close. Given certain extracted features, image areas may be likened to body limbs, sharing fundamental frequency, but being of arbitrary phase and harmonic content.

It is anticipated that a scene consisting largely of a single junction will possess a dominant fundamental frequency determined by the traffic signal cycle time, and that the queuing and

movement of all vehicles on the junction will be related to this. In addition, pedestrians using dedicated crossing points will also be somewhat regulated, whether or not their path across the road is controlled by a specific dedicated signal. Other activities are likely to be unrelated to the junction cycle time, perhaps completely non-periodic, or recurrent with a different periodicity, such as flashing signals, advertising signs, and animated billboard displays, both mechanical and electronic.

But a basic question arises as to exactly what feature an image or image patch possesses that is supposed to be periodic. The intensity at a pixel, or the presence or absence of a particular feature is a uni-dimensional quantity, for which the Fourier transform is well established. But the meaning of periodicity for a multi-dimensional quantity, such as an RGB triple or the pixels of a whole image patch, is much less clear. In the context of gait recognition, Cutler and Davis [22] explore the use of self-similarity of image patch intensity, in which the fundamental period is defined as the elapsed time between occurrence of matching patches. Of course, spurious matches might occur at any arbitrary time interval, so it becomes necessary to form an *average* period over a representative block of training data.

Here, the above concept is generalized to permit a much wider range of features to be used. Instead of comparing feature vector *values* between two different times, the idea is to compare the PDF over feature space between the two times. This has the following advantages:

1. Comparison is independent of the chosen feature, leading to a more decoupled, flexible algorithm overall.

2. All possible feature values are considered simultaneously in the comparison, yielding a broad consensus.

3. No metric on the feature itself is required, thus permitting use of features for which a precise metric is problematic to define, e.g. colour spaces.

But in order to obtain a *distribution* over feature space, a local spatial, temporal, or spatio-temporal extent containing examples from which to form the estimate must be defined. By necessity this imposes a degree of local low-pass filtering due to the *sampling aperture*, but resolution may be retained by overlapping the extents using as little as a one pixel or voxel step.

The important question arises as to whether to assume a single dominant period for a scene, leaving all other spectral content to be considered noise, or whether to split the scene into re-

gions and evaluate a dominant period for each. Ultimately, the particular application will dictate which is more appropriate. In any case, both avenues will be explored experimentally during this chapter.

Instead of using Fourier analysis of pixel intensity directly, [8] employs Phase Locked Loops (PLLs) to discriminate between different gaits. Having identified some fundamental frequency for an object (or person), application of a PLL per pixel in the relevant area permits estimation of the magnitude and relative phase of this fundamental component for each pixel in the object. The idea is that the *phase signature* for every object will be different. The technique is rendered scale and translation invariant by matching relative magnitude and phase parameters from a group of pixels to known exemplars as shapes in the complex plane using the Procrustes distance, as described in [79].

Choice of the PLL over Fourier analysis is significant for two reasons: efficiency and causality. An n-point Discrete Fourier Transform (DFT) has a time complexity of $O(nlogn)$ if n is chosen to be an integral binary power. But already implicit in the DFT approach is the need to accumulate the n most recent samples as a block before frequency estimation may proceed. On the other hand, the PLL is a causal system requiring a low complexity $O(1)$ update at every sample, which provides immediate frequency and phase estimates based on all previously encountered data in an amnesic fashion.

The PLL technique is similarly applied in this work in order to extract the fundamental frequency from temporal self-similarity in an image region. But here the purpose is to establish a local timing mechanism with some degree of *inertia*, which can maintain approximate synchronization with scene activity during periods of data corruption, and precise synchronization even if the scene's periodicity should drift from the original estimated value.

In this chapter, the goal is to construct an algorithm to characterize the periodicity of a scene based on its *temporal statistics* as a signal, rather than by performing explicit object tracking. This avoids the catch-22 problem of determining the appropriate scale at which to view a scene region, whereby the saliency of the region depends on the scale at which it is viewed, but the optimal viewing scale depends on the saliency [61]. Treating the recovered periodicity as a form of *temporal background*, the aim is to discover anomalies in both space and time simultaneously in previously unseen video. Experiments on datasets from three traffic junction scenes are shown, in which the effectiveness of such a model in performing anomaly detection is demonstrated. The

results show how scene activity may be decomposed into three layers based upon its dynamic content: static background, dynamic background, and foreground components.

5.2 Spatio-Temporal Model

The objective is to derive a model comprising a cyclic set of histograms as depicted in Figure 5.1 from a sequence of training frames. Given a video sequence $I_{x,y,t}$ consisting of t_{max} frames each of size $x_{max} \times y_{max}$ pixels in which (x,y) represents spatial pixel location, t the time index, and I the colour triple $\{R,G,B\}$, the data is split into two parts, the first for training and the second for evaluation. Obviously, the first image of the test sequence directly follows the final image from the training sequence - a fact which becomes crucial in ensuring the initialized model remains synchronized with the test data. This partitioning also enables a natural way of bootstrapping the model from limited initial exposure to the scene. Of course in a practical on-line situation, the second partition is temporally unbounded.

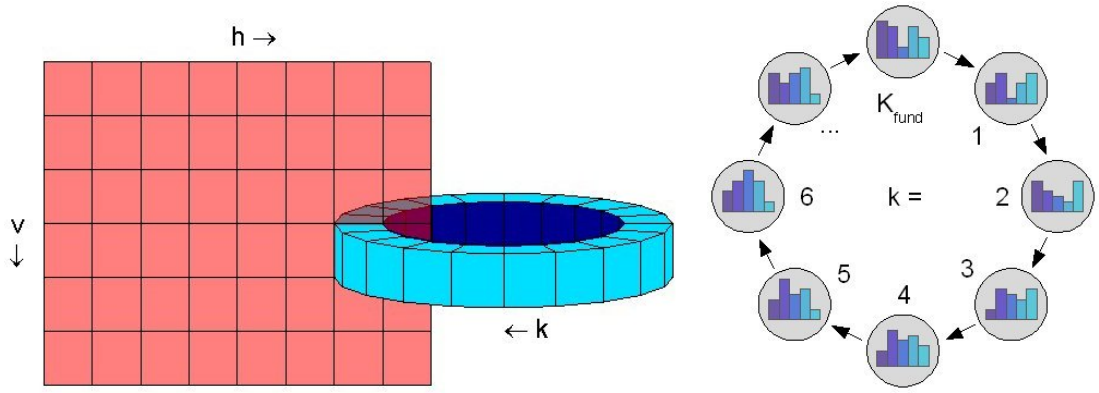


Figure 5.1: Diagram illustrating how the periodic statistics of a block of pixels at spatial location $\{v,h\}$ may be modelled over time by a set of histograms over some chosen feature space. The cyclic histogram set is indexed by a modulo-K_{fund} counter which references the appropriate behavioural representation for a given phase in the cycle, and increments regularly over time.

A *static* background model $I^B_{x,y,t}$ is evaluated from and maintained through both the training and test data according to the method detailed in Chapter 4. The overall algorithm for building the spatio-temporal model is shown in Table 5.1, and described in more detail in the following.

5.2.1 Feature Selection

A feature which summarizes some local characteristic of the image sequence must be selected. Later experiments employ optical flow, but for initial modelling of the traffic junction, the *aspect*

Step	Description
1	Derive a static background model from training sequence
2	Extract chosen feature from training sequence
3	Quantize samples to a coarse spatio-temporal grid forming linear state data
4	Find dominant fundamental period T_{fund} for the scene using the linear state data
5	'Roll up' Linear State Data using period T_{fund} starting from the end to form average State Cycle estimate (Figure 5.6)
6	Use State Cycle to classify previously unseen frames
7	Synthesize output from background and mismatched areas in new frames
8	Go to step 6

Table 5.1: Steps in the spatio-temporal modelling algorithm

ratio of an object's bounding box is chosen, anticipating that pedestrians will always be taller than they are wide, and vehicles will rarely be so under the majority of typical poses. In order to ensure symmetrical treatment of ratios greater and less than unity, a Log Aspect Ratio (LAR) feature $LAR_{x,y}$ is developed at position (x,y) by taking the natural logarithm and clipping to +/-1, resulting in ratio limits from $\frac{1}{e}$ to e

$$LAR_{x,y} = \max\left(-1, \min\left(1, \log_e\left(\frac{h_{x,y}}{w_{x,y}}\right)\right)\right) \tag{5.1}$$

where h and w are the object's bounding box height and width respectively. Bounding boxes are determined after applying morphological operations to a foreground binary mask $M^{FG}_{x,y,t}$ which removes shapes below a certain minimum pixel area. The binary mask $M^{FG}_{x,y,t}$ is in turn derived from the difference $D_{x,y,t}$ between the current image and the current static background I^{BG}, according to the L_1 (Manhattan) norm of the pixel vectors in colour space

$$M^{FG}_{x,y,t} = \begin{cases} 1 & \text{if } D_{x,y,t} > \tau \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

where τ is a constant and

$$D_{x,y,t} = \left\| I_{x,y,t} - I^{BG}_{x,y,t} \right\|_1 \tag{5.3}$$

Thus for each frame of video I_t, a (potentially empty) list L_t of valid bounding boxes $B_{t,m}$ is produced governed by the above rules

$$L_t = \{B_{t,1},\ B_{t,2},\ \ldots\ B_{t,m}\} \tag{5.4}$$

where the mth bounding box is characterized by the quad

$$B_{t,m} = \{x, y, w, h\} \tag{5.5}$$

in which (x,y) is the bounding box centre, and (w,h) is its size from which the LAR is calculated. The maximum value of m is determined by the number of objects detected in the current image. So the selected feature does not exist at every pixel, rather it will exist wherever valid objects are detected in the spatio-temporal volume. Figure 5.2 shows an example of accumulation of object LAR values over time in typical data from the experiment, showing how the feature discriminates between people and vehicles fairly effectively. Meanwhile in a plot of the image y-axis against time for a fixed x value, Figure 5.3 illustrates the inherently periodic nature of activity on the road junction.

5.2.2 Spatio-temporal Histogram

Thus far the training data is represented by a set of points in a 4-D space (x,y,t,LAR). In order to facilitate comparison of feature occurrence within the spatio-temporal volume, the idea is to build a spatio-temporal set of histograms over feature space, as shown in Figures 5.1 and 5.4. Therefore the volume is split into a grid of $h_{max} \times v_{max}$ equal sized square blocks of pixels spatially and n_{max} equal sized blocks of frames temporally. In later experiments, a sequence from the scene in Figure 5.2 consisting of $t_{max} = 90000$ frames each of $x_{max} = 360$ by $y_{max} = 288$ pixels was split into a grid of $h_{max} = 45$ by $v_{max} = 36$ spatial blocks by $n_{max} = 478$ temporal blocks. At each spatio-temporal grid position, consisting of a block of

$$\frac{x_{max}}{h_{max}} \times \frac{y_{max}}{v_{max}} \times \frac{t_{max}}{n_{max}} \quad = \quad \frac{360}{45} \times \frac{288}{36} \times \frac{90000}{478} \quad = \quad 8 \times 8 \times 188 \tag{5.6}$$

pixels, a histogram $H_{h,v,t}$ of b_{max} equal width bins is constructed over feature space. Evidently the spatial resolution is 8×8 pixels, whilst temporally the extent of a block is $\frac{188}{25} = 7.52$ seconds for a frame rate of 25Hz. The particular choice of h_{max}, v_{max} and n_{max} is ultimately a compromise

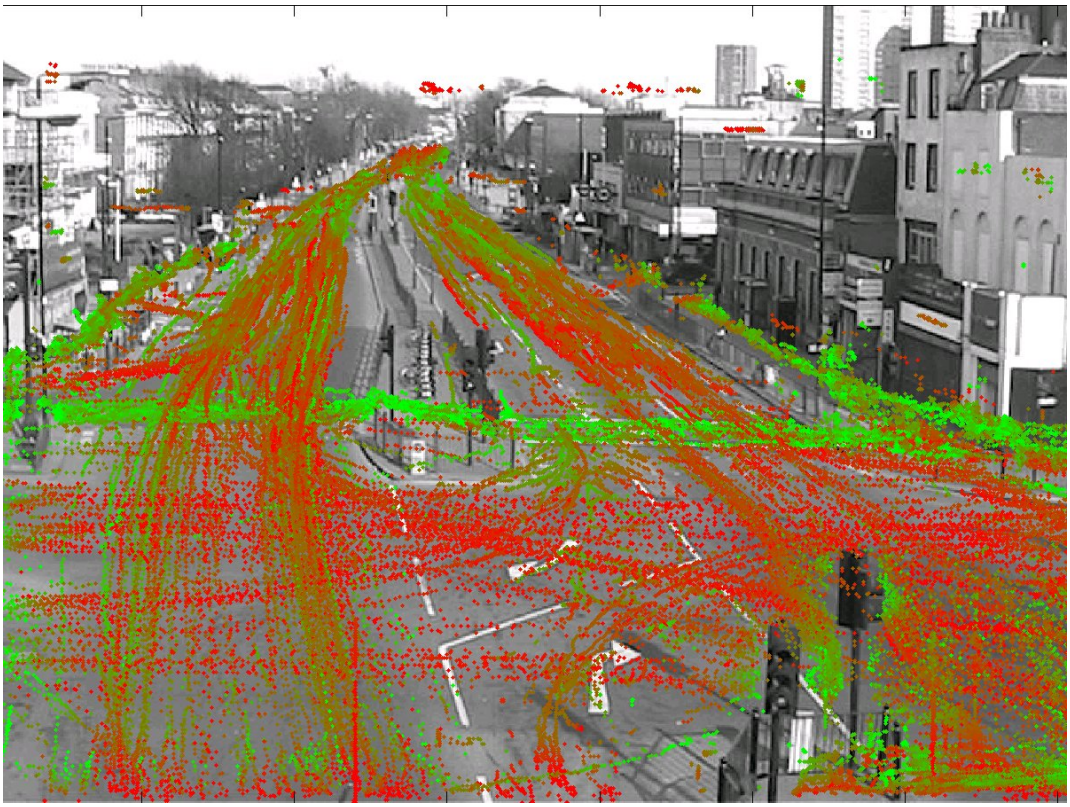


Figure 5.2: Bounding Box centres accumulated over time at a road junction scene. Colour represents aspect ratio: green samples have $h > w$ (pedestrians), red samples have $h < w$ (vehicles). Although not completely reliable, the feature is a fairly strong discriminator between vehicles and pedestrians, especially in the nearer part of the junction. The road areas are seen to contain mostly vehicles (red) whilst the pedestrian route across the junction, and the footpaths alongside the road are clearly dominated by pedestrians (green). The ratio for vehicles becomes unreliable in the far distance due to the unfavourable viewing angle.

between resolution and sparsity of histogram data. For LAR the bins are represented by the bounded 1-D set

$$H_{h,v,n}(b) = \{b_1,\, b_2,\, \ldots\, b_{max}\} \tag{5.7}$$

$$\text{where} \quad b = \lfloor \frac{b_{max}(LAR+1)}{2} + 0.99999 \rfloor \tag{5.8}$$

such that the range of the LAR feature ($-1 \leq LAR \leq +1$) is quantized and mapped uniformly onto bin number b, where $1 \leq b \leq b_{max}$. The inherent loss of resolution in all dimensions as a result of this down-sampling operation is countered by the advantage of being able to quantify

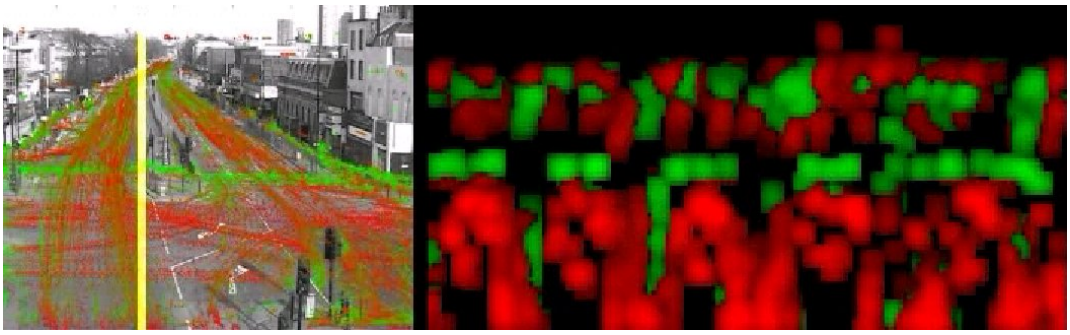


Figure 5.3: Y-T cut (right) through the spatio-temporal volume showing periodic behaviour of a road junction scene at the vertical yellow line (left) using bounding box aspect ratio as the feature. There is clearly a temporal structure to the data in various areas of the image.

the similarity between any two spatio-temporal regions on the basis of the selected feature purely by comparing histograms. In fact from this point onwards, the method becomes independent of the chosen feature and thus offers a degree of generality and considerable scope for matching any chosen feature(s) - a crucial strength of the approach.

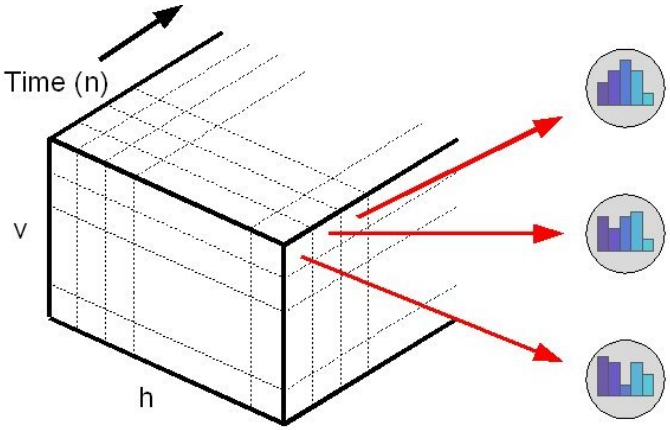


Figure 5.4: Diagram showing how an image sequence is divided into a uniform set of $h_{max} \times v_{max} \times n_{max}$ spatio-temporal blocks. For each block, a histogram over the chosen feature space is evaluated from training data.

5.2.3 The Sparsity Problem

It is quite possible that, given the relatively high dimensionality of the histogram containing the bounding box data points, the density of points is insufficient to yield meaningful distributions everywhere throughout the spatio-temporal volume. One potential solution is to decrease the number of blocks in the grid in the dimension(s) causing the deficiency. Alternatively a degree

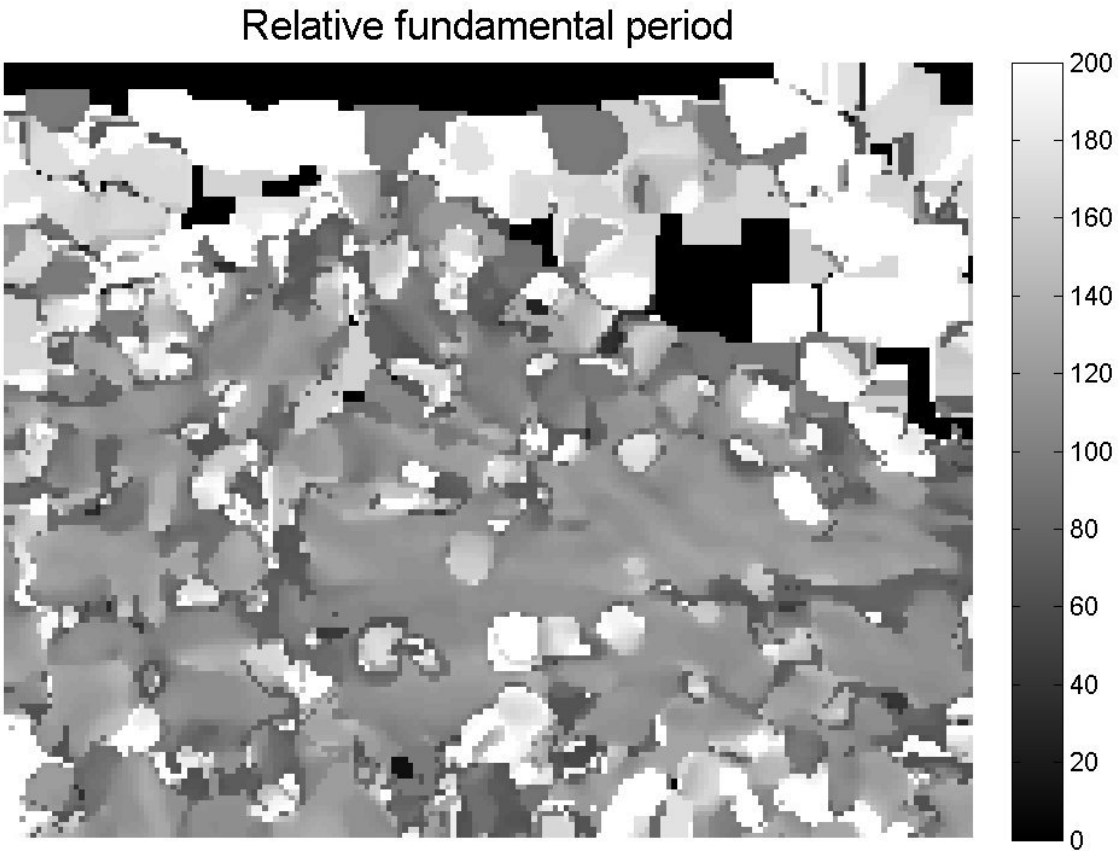


Figure 5.5: Relative fundamental period distribution of the scene in Figure 5.2 based on temporal autocorrelation of bounding box aspect ratio of 4×4 pixel blocks. Intensity, representing period in seconds according to the side bar, is given by the first significant peak of the autocorrelation function. Much of the central junction area is the same shade, indicating shared periodicity.

of data smoothing may be applied, both over the bins within each histogram and also between spatio-temporal histograms. It was found that experimental results benefited from convolution of the former with a normalized 1-D Gaussian filter, and of the latter with a 3-D Gaussian kernel having potentially different variance in the spatial and temporal directions. Inevitably there will be some regions which are poorly supported, and steps to mitigate the effects of this may become necessary in some situations. A certain advantage of the block-based approach from a spatial point of view is its tolerance to slight errors in registration due to camera movement induced either by wind or maintenance work.

5.2.4 Fundamental Period Estimation

To derive an estimate of the fundamental period over which scene changes occur is a non-trivial procedure, and as such it is dealt with separately in Section 5.3. Suffice to say at this point that

a scene may have a number of unrelated fundamental periods (including *none*) distributed over various regions, as shown in Figure 5.5, and optimally distinguishing them is a topic for future research. In this section where applications like the traffic junction are considered, it is assumed that there *is* a single dominant effect, for which the period is K_{fund} blocks each of t_{max}/n_{max} frames. Given a frame rate of F per second, the fundamental period is thus

$$T_{fund} = \frac{K_{fund}}{F} \frac{t_{max}}{n_{max}} \quad \text{seconds.} \tag{5.9}$$

Ideally the training data should be long enough to contain sufficient cycles of the fundamental period in order to permit the latter to be distinguished adequately from noise.

5.2.5 State Cycle and Model Initialization

The State Cycle $S^k_{h,v}$ for $k = \{1 \ldots K_{fund}\}$ of a grid location (h,v) is defined to be a temporal description of how the chosen feature varies throughout a single cycle of its fundamental period of K_{fund} phases. Given that the array $H_{h,v,n}$ contains a number of cycles of this temporal description in succession, the desire is to form an *average histogram* H_{fund} of size $h_{max} \times v_{max} \times K_{fund}$ representing a summary of the scene's typical behaviour over the c most recent cycles of the fundamental period, where $c = \lfloor \frac{n_{max}}{K_{fund}} \rfloor$ cycles. Thus taking the c most recent groups of K_{fund} blocks, the kth element of H_{fund} is the mean of the kth elements of the c groups

$$H_{fund,h,v,k}(b) = \frac{1}{c} \sum_{i=1}^{c} H_{h,v,n_{max}-iK_{fund}+k}(b) \tag{5.10}$$

where $k = \{1,\ 2,\ \ldots\ K_{fund}\}$. This 'rolling-up' of the training data is depicted diagrammatically in Figure 5.6. Normalization of H_{fund} over k,h,v, and b yields an estimate of feature probability P_{fund} which then forms the spatio-temporal model of the scene

$$P_{fund,h,v,k}(b) = \frac{H_{fund,h,v,k}(b)}{\sum_{k=1}^{K_{fund}} \sum_{h=1}^{h_{max}} \sum_{v=1}^{v_{max}} \sum_{b=1}^{b_{max}} H_{fund,h,v,k}(b)} \tag{5.11}$$

Assuming that continuous test sequence (e.g. real-time video streamed data) directly follows the initial training sequence, then the state counter k, initialized to 1, may be updated every $\frac{t_{max}}{n_{max}}$ frames according to the relation $k = \text{mod}(k, K_{fund}) + 1$ in order to keep track of the learned phases of periodic scene behaviour.

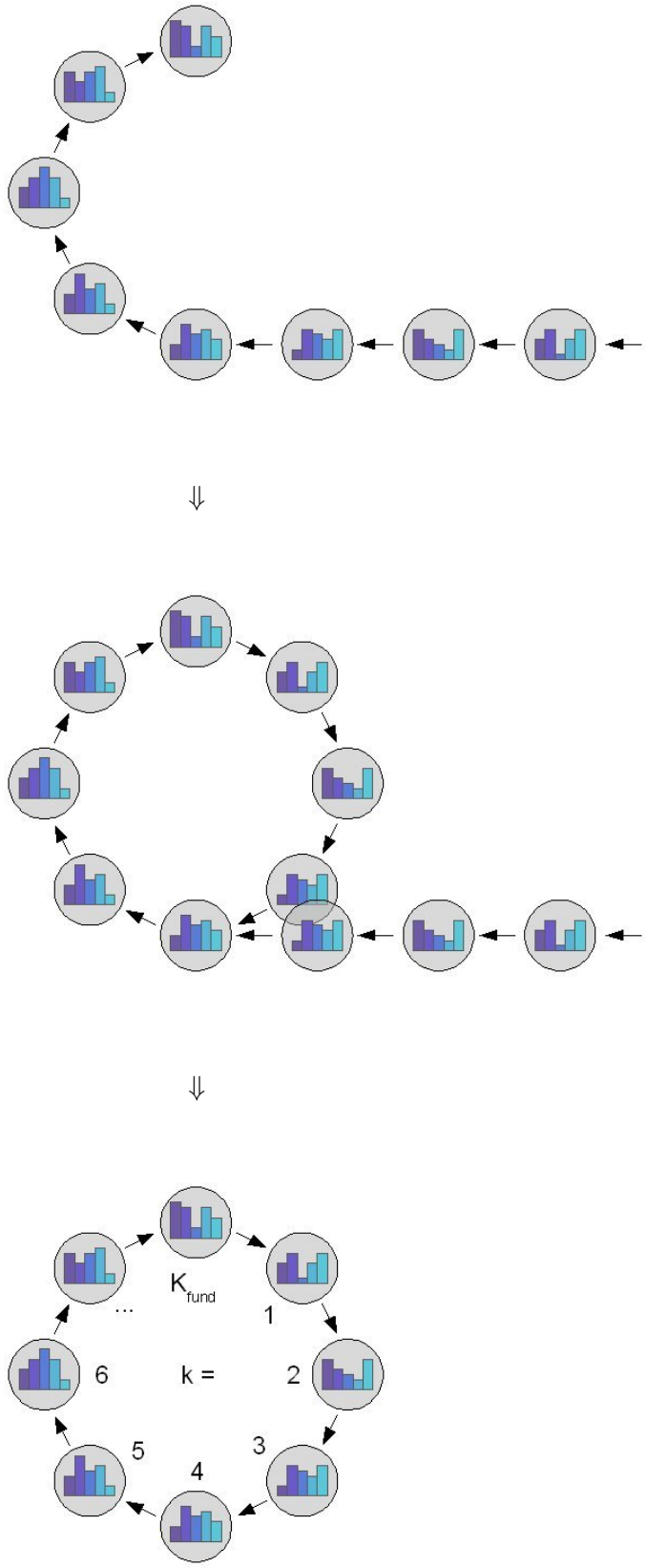


Figure 5.6: Diagram showing how the training data H is *rolled up* to form the single cycle average set of histograms H_{fund} depicted in Figure 5.1 which summarize scene activity.

5.2.6 Output Synthesis

The objective is to provide an output sequence from the algorithm showing only objects in the *wrong place* at the *wrong time*. For a query test frame I^Q appearing subsequent to model initialization, the foreground mask M^{FG} is obtained as in Equation (5.2), and valid object bounding boxes $B_{t,m}$ derived as in Equation (5.5). For each candidate bounding box, the LAR is evaluated from width and height using Equation (5.1) and b is given by Equation (5.8). Values for h and v are calculated using $h = \frac{x \times h_{max}}{x_{max}}$ and $v = \frac{y \times v_{max}}{y_{max}}$. Thus the estimated probability of that particular aspect ratio bounding box at position $\{h, v\}$ is given by the model for the current phase k, and may be compared with a threshold α in order to give a binary decision $M_{h,v}$ as to whether the object is sufficiently rare to be displayed

$$M_{h,v} = \begin{cases} 1 & \text{if } P_{fund,h,v,k}(b) < \alpha \\ 0 & \text{otherwise} \end{cases} \tag{5.12}$$

On the basis of $M_{h,v}$ being true, for each object in I^Q, M is used as a matting mask to re-insert pixels according to the bounding box dimensions from the new frame I^Q into the background I^{BG} for all objects determined to be anomalous with respect to the current model. The background with insertions forms the output image from the algorithm. Examples of this matting process are seen clearly in the experimental results under the 'Reconstruction' column of Figure 5.13, where the unusual object is highlighted *alone* in the static background.

5.3 Determining the Fundamental Period

The method described in the previous section relies totally on obtaining a robust estimate of the fundamental period of a region, or of the whole image area, using the 3-D spatio-temporal grid of histograms $H_{h,v,n}$ defined in Equation (5.7). The objective is to find the most common *lag* between instances of temporal self-similarity at times n_1 and n_2 over all possible combinations of n_1 and n_2. As a measure of the similarity between any two histograms, the general definition of the symmetric Kullback-Leibler Divergence (KLD) [70] is used, yielding a metric between distributions P_1 and P_2 of

$$D_{KL}(P_1, P_2) = \sum_i (P_{1,i} \log_2 (\frac{P_{1,i}}{P_{2,i}}) + P_{2,i} \log_2 (\frac{P_{2,i}}{P_{1,i}})) \text{ bits} \tag{5.13}$$

Thus over an arbitrary spatial region R in the grid, the *average dissimilarity* matrix S may be defined between two temporal planes at times n_1 and n_2 as

$$S_{n_1,n_2} = \frac{1}{\|R\|} \sum_{v,h \in R} D_{KL}(P_{n_1}(v,h), P_{n_2}(v,h)) \tag{5.14}$$

which after simplification yields

$$S_{n_1,n_2} = \frac{1}{\|R\|} \sum_{v,h \in R} \sum_{i=1}^{b_{max}} (P_{n_1,i} - P_{n_2,i}) \log_2 \left(\frac{P_{n_1,i}}{P_{n_2,i}} \right) \tag{5.15}$$

An example of the symmetric divergence relative to a single time is illustrated in Figure 5.7(a), and between all combinations of times as matrix S in Figure 5.7(b). Because it is the coincidences of *minima* in S that are of interest, representing the best distribution matches, S' is formed by subtracting the mean of S, leaving the minima now as negative *peaks*

$$S'(i,j) = S(i,j) - \frac{1}{i_{max} j_{max}} \sum_{i,j} S(i,j) \tag{5.16}$$

Then the normalized 2-D autocovariance matrix A is constructed from all possible lags (d_i, d_j) in both spatial directions

$$A(d_i,d_j) = \frac{\sum_{i,j} S'(i,j)\, S'(i+d_i, j+d_j)}{\sqrt{\sum_{i,j} S'(i,j)^2 \cdot \sum_{i,j} S'(i+d_i, j+d_j)^2}} \tag{5.17}$$

As shown in Figure 5.8(b), matrix A exhibits a regular structure of peaks spaced at the dominant period if one exists. The fundamental interval K_{fund} is identified by exploratory element-wise multiplication of A with a regular matrix of peaks generated by column vector $g(d)$ as shown in Figure 5.8(a), whereby varying the pitch d yields a peak in the overall temporal scene power observed. Thus the fundamental interval is given by

$$K_{fund} = \arg\max_{d} (g(d)^T A\, g(d)) \tag{5.18}$$

for $d_{min} \leq d \leq d_{max}$ and binary vector g such that

$$g_i(d) = \delta((i - n_{max}) \bmod d) \text{ where } 1 \leq i \leq 2n_{max} - 1 \tag{5.19}$$

Figure 5.9 shows how the scene's signal power peaks at a given value of d.

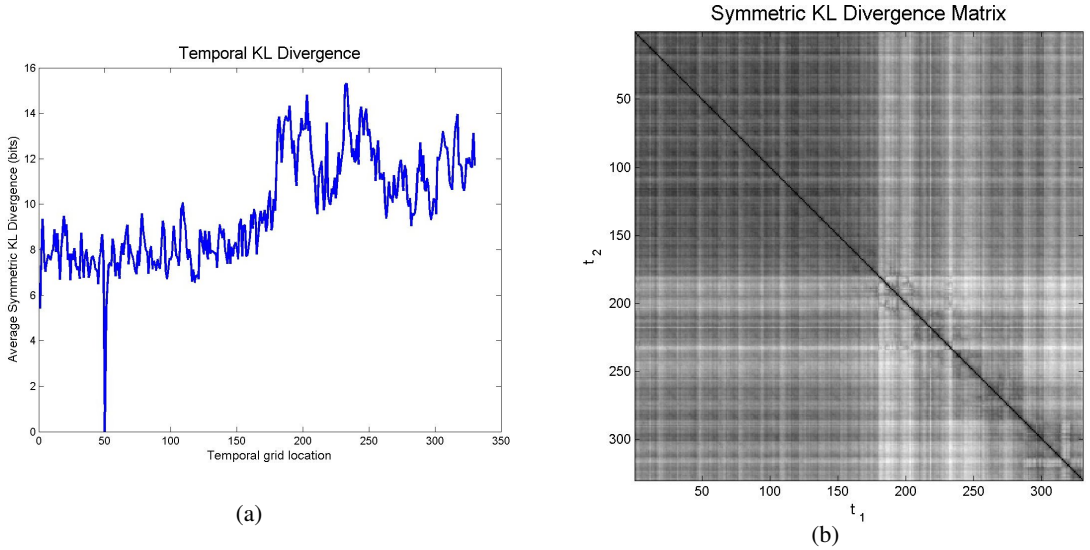
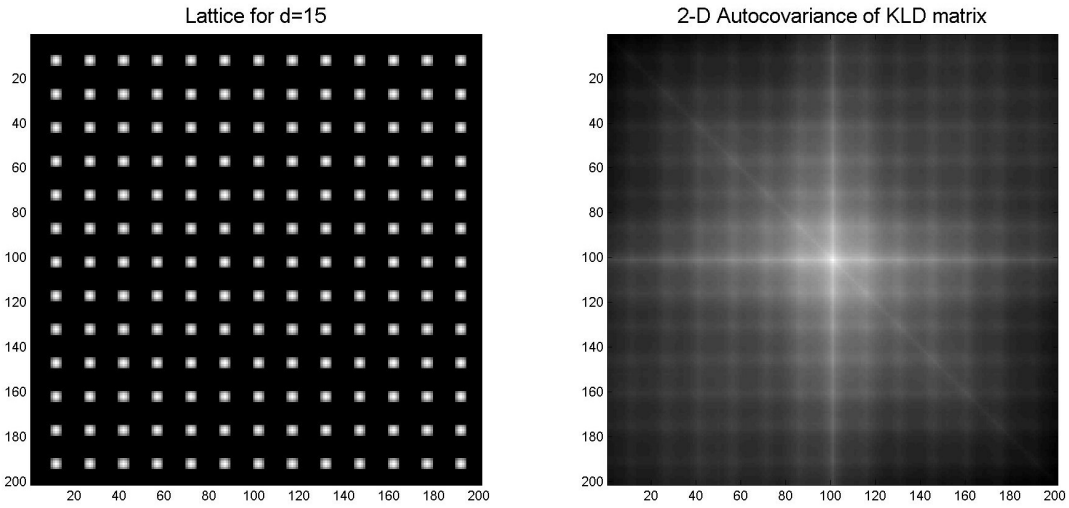


Figure 5.7: (a) Temporal KL Divergence at one grid position ($n_1 = 50$ on the x-axis) relative to all other temporal grid positions. Naturally the divergence is zero with respect to itself. (b) Average Divergence matrix between histograms at temporal grid positions n_1, n_2 for all combinations of n_1 and n_2. Using the Symmetric Kullback-Leibler formula, divergence is summed over all spatial grid positions of the scene, as well as over the histogram bins (Equation (5.15)).

In the current approach, the region R represents the entire scene, but this technique could equally well work with subsets of the scene, be they rectangular or square blocks, or even arbitrary shapes. A yet more elaborate scheme for analyzing the autocovariance matrix A is described in [22], in particular explaining that a diagonal equivalent of the matrix in Figure 5.8(a) is necessary to detect periodicity in certain scenes for which self-similarity of appearance peaks more than once per cycle, e.g. a swinging pendulum at its lowest point.

5.4 Experiment

For the experiments, three busy city-centre road junctions controlled by traffic lights were chosen. Each dataset consisted of 30000 frames of 720×576 pixel colour video at a frame rate of 25Hz, yielding sequences of 20 minutes duration. The data was spatially down-sampled to 360×288 pixels to ease computational load. The short-term background model was obtained as described using the method described in Chapter 4, based on blocks of 20 frames taken at 12 second intervals. The L_1 norm of the background-subtracted data was thresholded at a value of 30 given an intensity range of 0-255 per colour channel, and after morphological clean-up, identified object areas were thresholded to reject those below 70 pixels. The Log Aspect Ratio feature



(a) (b)

Figure 5.8: (a) Lattice for distance $d = 15$ generated by $g(d)g(d)^T$. Point-wise multiplication of such a lattice by the autocovariance matrix in (b) for a range of d identifies the fundamental period. (b) Autocovariance of the Divergence matrix in Figure 5.7(b), showing the strong lattice structure corresponding to a dominant fundamental temporal period in the video sequence.

range of +1 to -1 was split into $b_{max} = 5$ histogram bins, and the spatio-temporal histogram grid was 8×8 pixels wide spatially, and 180 frames deep temporally, giving $h_{max} = 45$, $v_{max} = 36$, and $n_{max} = 167$. For each sequence, the entire spatial extent of the spatio-temporal matrix was utilized to estimate the global fundamental period K_{fund} for the scene using the method described in Section 5.3. Allowing $c = 5$ cycles of this fundamental period of the temporal extent of the dataset to be used for training, the remainder was left for testing. Figure 5.10 illustrates how the state counter is correctly and consistently aligned with junction activity throughout the test sequence, as measured by the actual brightness of pixels representing the green traffic light at the bottom of the scene. Close synchronization is essential for the model to function properly.

The results for Scenarios 1 and 2, using bounding box aspect ratio as the feature, are shown in Figures 5.11 and 5.12, which have 3 rows of 5 images, with each row representing an example frame from the algorithm output. The left-most image is the original unprocessed frame, whilst the second image is the short-term static background which has been labelled as ‘Layer 0’. The objects detected to be anomalous according to the model are shown inserted into the static background and labelled as ‘Layer 2’ - the foreground. Similarly, the original image with background inserted where the object was detected, is shown as ‘Layer 1’ - the dynamic background.

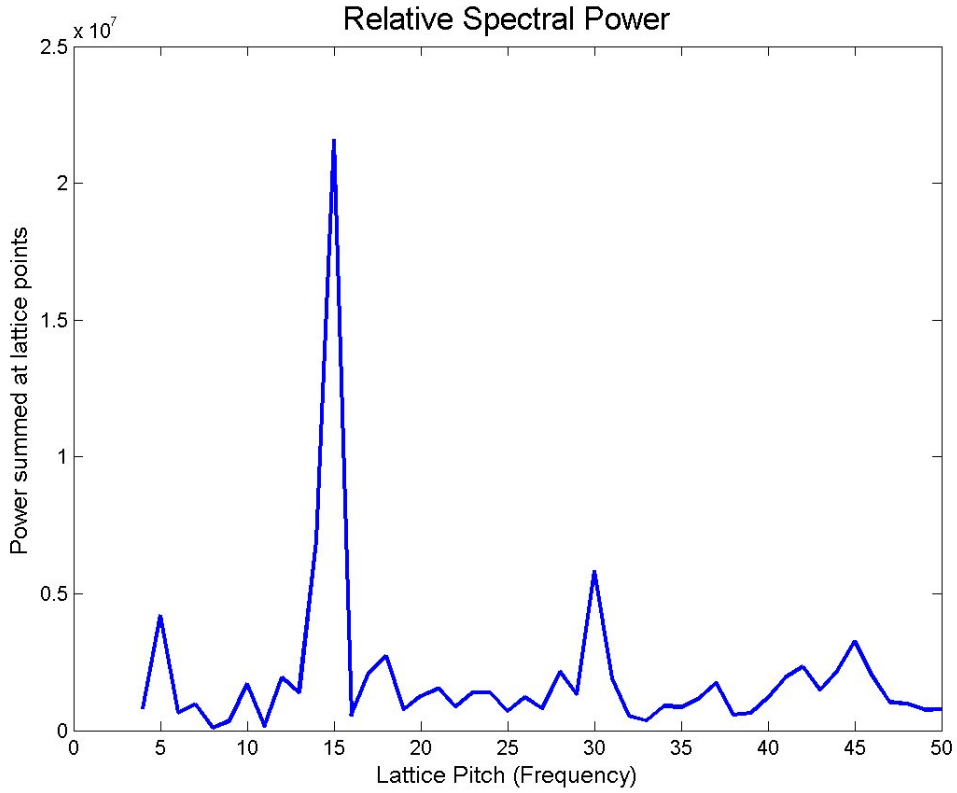


Figure 5.9: Relative spectral power of the scene in Figure 5.2 for values of d between 4 and 50, calculated as the sum of all values after point-wise multiplication of the lattice in Figure 5.8(a) by the autocovariance matrix in Figure 5.8(b). Note the fundamental at $d = 15$, giving a period of $15 \times 7.5s = 112.5s$ corresponding to the cycle time of the junction traffic signals.

Finally in the right-hand column, for comparison purposes, the result of classification using a non-temporal equivalent model derived from the same training data is shown. To achieve this, bin values of each histogram $P_{h,v,k}(b)$ are marginalized out over the time dimension to yield $P'_{h,v}(b)$.

Figures 5.13 and 5.14 show results from Scenario 3, this time using *optical flow* as the feature. Candidate objects in the scene were identified as previously, but here connected items in the foreground mask were used to estimate vehicle velocity according the Lucas-Kanade algorithm, whereby each pixel in the mask permits a contribution to an over-determined system of equations resulting in an estimate of the optical flow field at the object's location. Details of this algorithm are to be found in Appendix A and [77]. The two dimensional flow vector was coarsely quantized into a 3×3 bin histogram for incorporation into the model.

The results in Figures 5.13 and 5.14 encode the optical flow vectors as coloured areas for visualisation: hue represents direction, and intensity indicates vehicle speed.

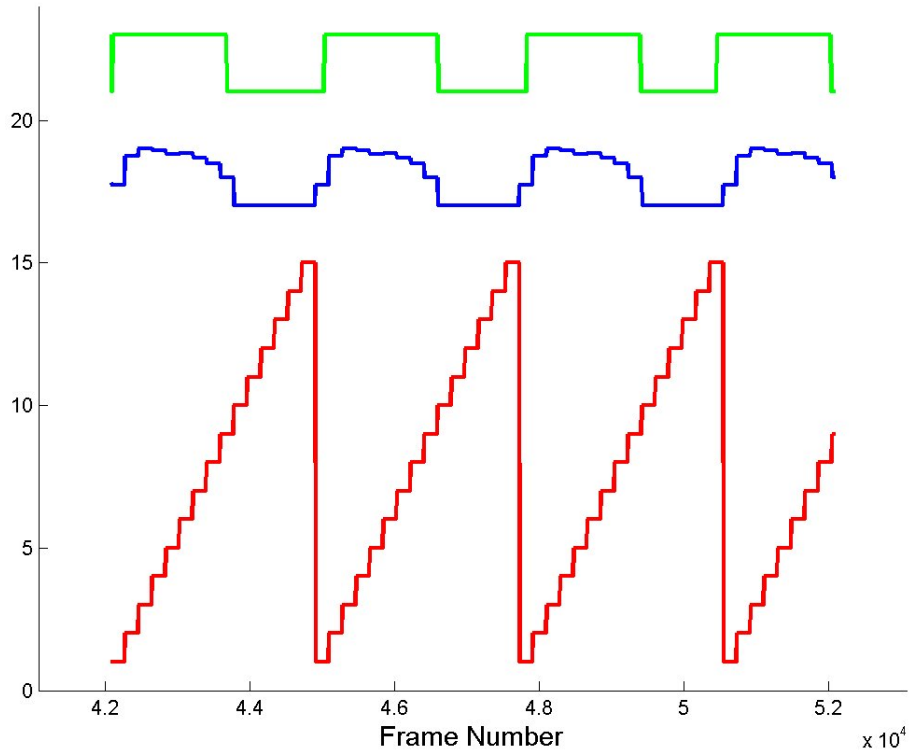


Figure 5.10: Timing diagram showing correct synchronization of model throughout test sequence. Top: Pixels from closest green traffic light in the scene, which is on for 9 of the 15 phases. Middle: Consensus of light over cycles in training data. Bottom: Internal state counter cycling through states 1 to 15. Note consistent and stable phase relationship between all three measurements, a vital condition for successful model operation.

Overall, when analyzing images, the algorithm achieves 3 frames per second throughput on a 2GHz Athlon-based PC, although initially building the model carries a considerably higher computational cost, dependent on the size of the training dataset.

5.5 Discussion

The results in Figures 5.11, 5.12, 5.13 and 5.14 demonstrate how, in spite of a background that is non-stationary, the new algorithm has managed to split scene activity into 3 distinct layers. This has been achieved partly by being able to make reliable estimates of true background amongst a busy scene by utilising the method described in Chapter 4, but mostly by classifying objects based on a spatio-temporal template learned from the scene during training.

What is termed Layer 0 takes on the non-stationary background, permitting detection of less persistently occurring objects such as people and vehicles. Having thus obtained reference to the latter in isolation from the background, the spatio-temporal model classifies them into Layer 1, objects of a suitable aspect ratio for the part of state-space they occupy, and Layer 2, objects which contradict the model. Within this framework, Layer 1 has taken on the role of a *dynamic background* in relation to what might usually be referred to as *foreground* objects. Such a dynamic background has three dimensions, two spatial and one temporal, and a match in all three of them is required as well as an acceptable probability value for the feature at those coordinates in order that the object is deemed acceptable as a dynamic background item. Thus the spatio-temporal model has gained more discriminative power than a spatial-only 2-D probabilistic model, which is oblivious to time.

By marginalizing out the time dimension so that the model degenerates into a more conventional temporally unaware type, one effectively increases the likelihood of an object at times in the cycle when it should be considered rare, and reduces its likelihood at times when it should be considered common. Thus the overall unwanted result is a desensitization of the model.

The upshot of this situation is that with no temporal processing (denoted as 'NTP' in the figures), too many relatively unimportant objects are detected, whilst use of the scene-synchronized spatio-temporal model reveals far more salient detection amongst the 'higher layers' of temporal change, associated with interesting and unexpected spatio-temporal events. Furthermore, all this may be achieved *without* prior knowledge of the size and location of potential triggering objects in the scene. Such a model clearly has benefits in a surveillance scenario.

In particular, among the results are examples of the new spatio-temporal model detecting objects of interest, whilst the model without temporal processing *fails* to highlight these, but identifies *less* truly interesting objects instead. That this remains so, however one decides to select the detection thresholds for the respective models, strongly supports the claim that the temporal aspect of the model is highly significant.

Figure 5.11: Examples from Scenario 1 show how the algorithm discovers objects not matching the learned spatio-temporal template, and thus splits the scene into 3 layers on the basis of its dynamic behaviour. Layer 0 is the continuously updated 'static' background, Layer 1 normal scene activity - the 'dynamic background', and Layer 2 carries 'novel' intrusions with respect to the training data. Some objects cannot be separated, regardless of threshold chosen. In (a) L2 correctly shows a car unusually pulling out onto the main road, whereas with No Temporal Processing (NTP), this cannot be distinguished from normal cars on the right. In (b) L2 spots the car over the waiting line, whereas NTP sees only a passing pedestrian. In (c) L2 finds pedestrians waiting at the crossing, whilst NTP wrongly highlights a car.

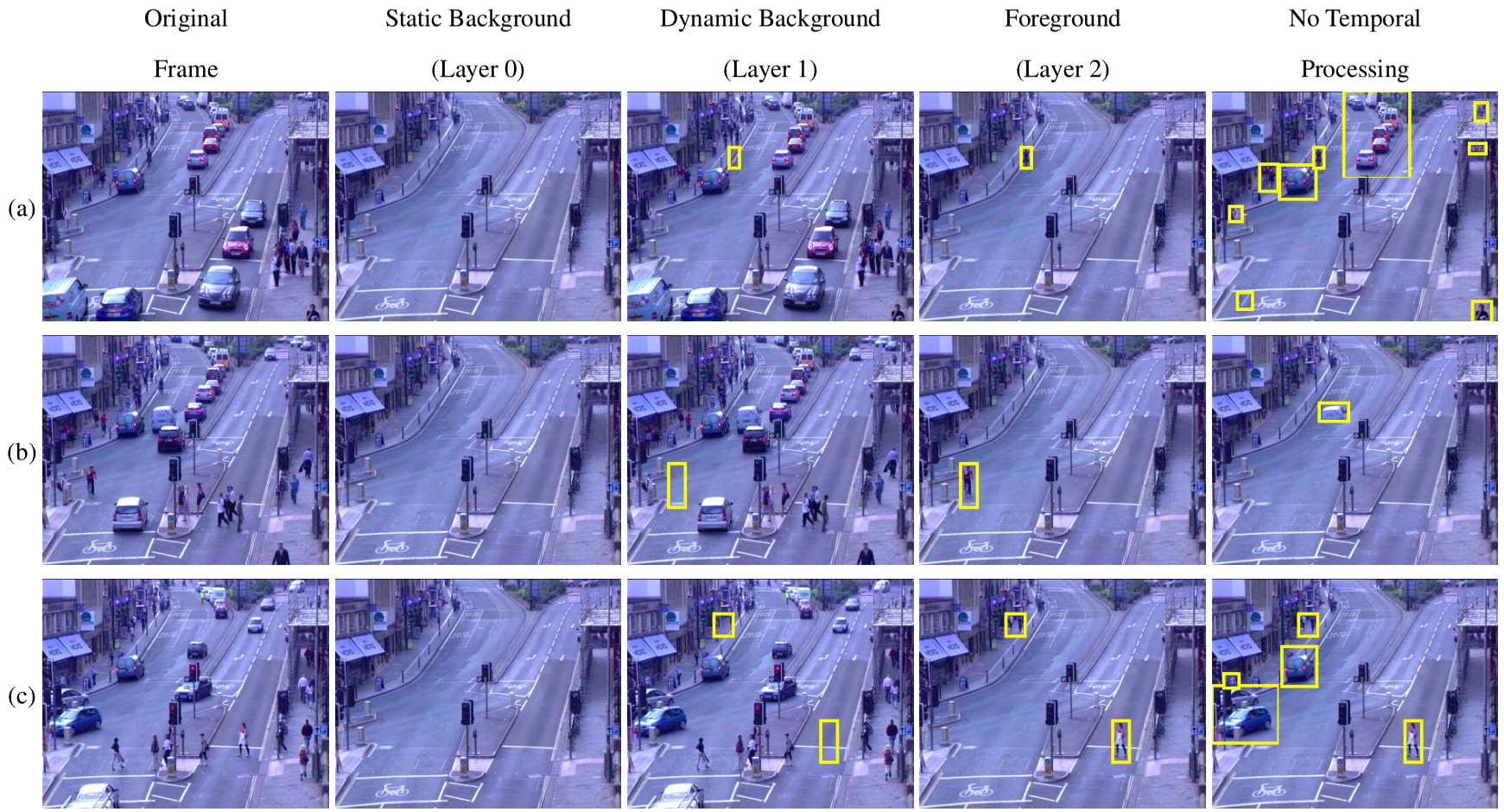


Figure 5.12: Examples from Scenario 2. From behind, cyclists tend to have an aspect ratio similar to people. Thus in (a) L2 singles out a cyclist close to the pathway, which with No Temporal Processing (NTP), cannot be separated. In (b) L2 has detected a different cyclist, again with the same profile as a person, where there should not be people, whilst NTP sees only part of a car in normal position. In (c) in the bottom right corner, L2 observes a person on the wrong part of the crossing, inseparable from vehicles on the junction with NTP.

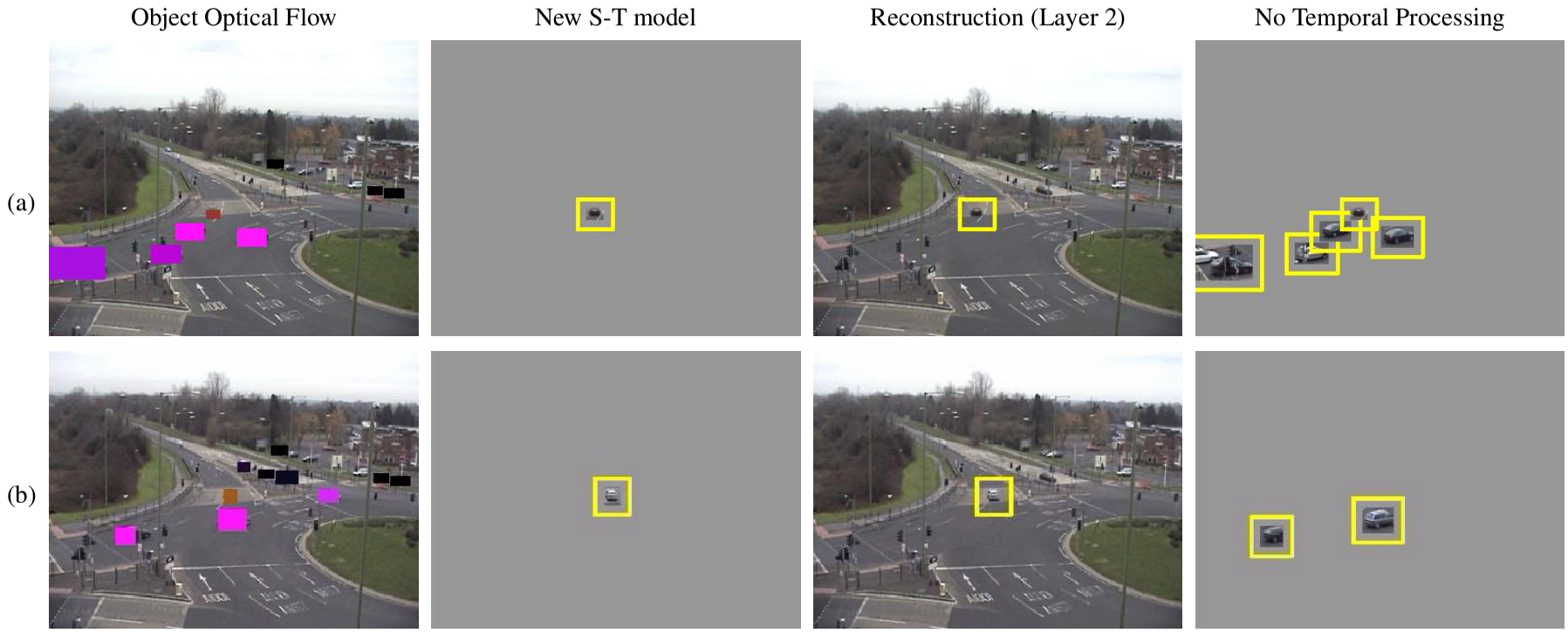


Figure 5.13: Examples from Scenario 3. Comparison between new S-T model and one with No Temporal Processing (NTP) based on optical flow. From left: Colour-coded optical flow, S-T model output, reconstruction from S-T model (i.e. Layer 2), NTP model output. In both (a) and (b), vehicles wrongly cross the lights going from bottom to top. The S-T model finds them, but the NTP model only highlights cars behaving normally.

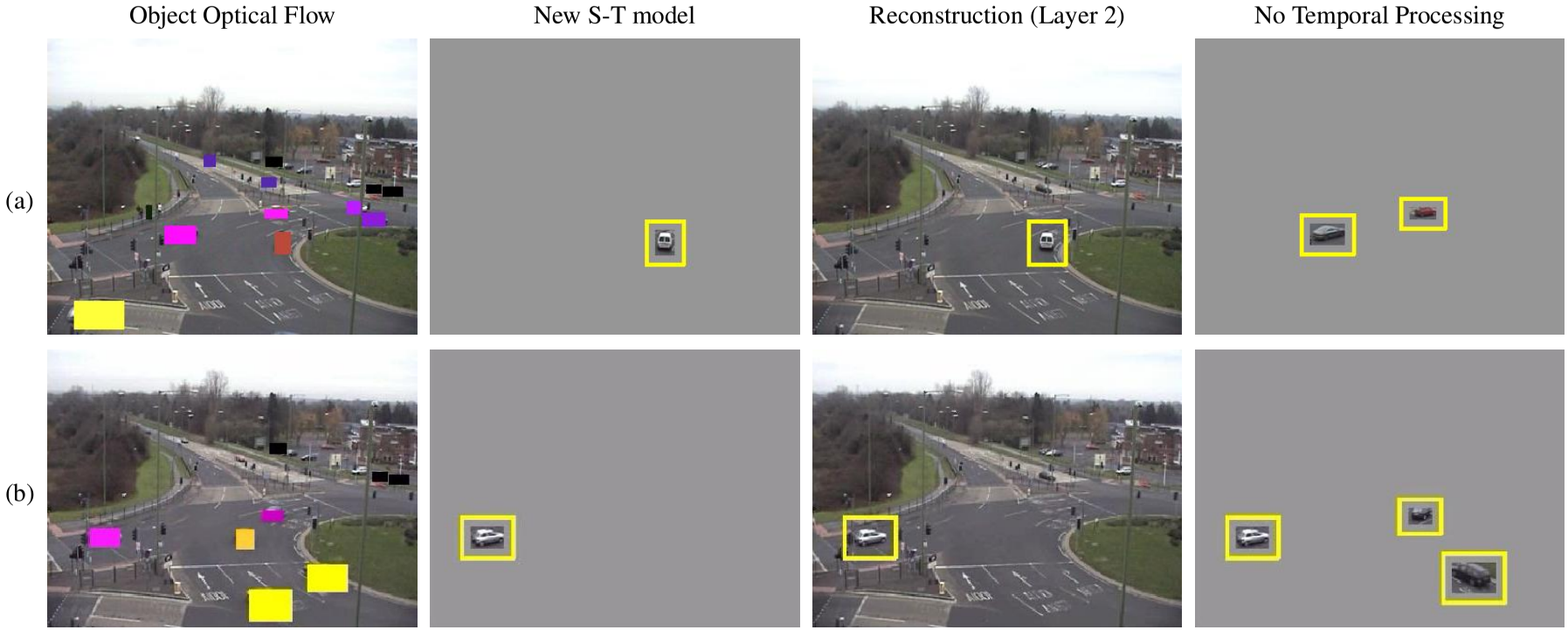


Figure 5.14: Examples from Scenario 3. Comparison between new S-T model and one with No Temporal Processing (NTP) based on optical flow. From left: Colour-coded optical flow, S-T model output, reconstruction from S-T model (i.e. Layer 2), NTP model output. (a): Car jumps red light by roundabout from bottom right. The S-T model sees it, but the NTP model only finds legal behaviour. (b): Car jumps red light from the left. The NTP model also highlights two other vehicles erroneously.

5.6 Scenes Exhibiting Multiple Periodicities

It is clear that many scenes will be composed of more than one harmonically unrelated periodic component. Instead of seeking a single global fundamental, the scene may be searched in a systematic fashion using the estimation technique previously described, but on multiple smaller regions. If somewhat optimal regions of common periodicity can be found, the 'rolling up' of periodic training data implemented above and depicted in Figures 5.1 and 5.6, is equally applicable to different image areas, each with its own K_{fund}.

As a step towards this, Figure 5.15 shows fundamental periods for Scenario 1 when split into a regular array of blocks. This time using RGB intensity values over 64 bin histograms ($4 \times 4 \times 4$) as the feature, with a spatial block size of 36×36 pixels and 4s temporal blocks, the locally evaluated periodicity is shown overlayed onto the image of Figure 5.2 for direct comparison.

From Figure 5.15, it is clear that different areas of the scene exhibit different periodicities, whilst some remain aperiodic with regard to the RGB feature space. But overall, many of the scene areas involving traffic still share the same periodicity of 112 seconds evaluated previously by the global method, as might be expected. To maintain an individual model per block, each with a potentially different periodicity, costs little extra resource in terms of computation time or storage over the method enforcing a single global periodicity. However, the reduced spatial area from which each local estimate of period derives an average may lead to less robust values of K_{fund} during training. A potential solution to this problem is presented in Section 5.8.

5.7 Verifying Periodicity Estimation

Demonstrating the effectiveness of the technique for identifying the fundamental period described in Section 5.3 requires detailed knowledge of the *ground truth* for a given scene. Whilst certain events, such as traffic light phase, may be readily analyzed manually by counting frames, other aspects of scene activity, such as trends in traffic or pedestrian density, are more subtle. Unlike a simple ground truth set consisting of a sequence of foreground masks for a single object moving once across a scene, characterization of spectral content depends on widely distributed temporal information, from which the fundamental period may not immediately be apparent using direct visualisation. A further problem with real scenes is that realistically they possess only a few periodic processes, some of which may be related anyway. For these reasons, the possibility for verification of the method in Section 5.3 using real data seems somewhat limited.

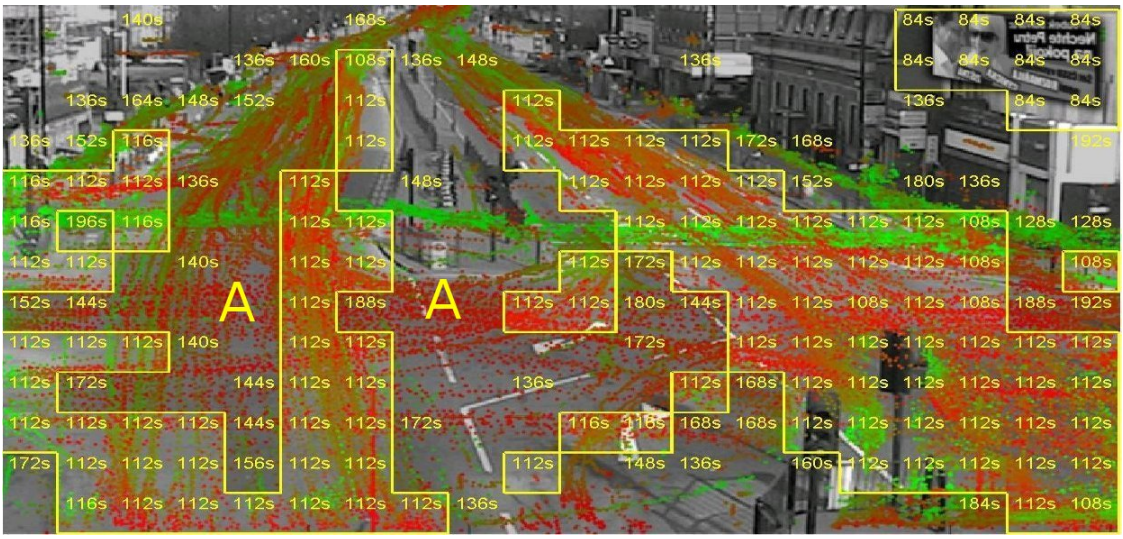


Figure 5.15: Periodicities from Scenario 1, but calculated over RGB feature space. Much of the junction shares the periodicity of the 112s traffic light system, whilst the rotating advertising board (top right) changes every 28s, cycling in sequence through three different advertisements in 84s. Areas marked 'A' are aperiodic. Pedestrians tracks shown in green, and vehicles in red are based on the previous bounding box ratio approach, and shown here for comparison purposes.

An alternative approach involves using a synthetic scene dataset for which the ground truth is completely known a priori. Although not directly representing real-world scenes, success with synthetic data gives at least some confidence in the method evolved for period estimation. Figure 5.16 shows two examples from a synthetic dataset of 16500 frames in which multiple randomly-sized coloured ellipses are embedded in additive white Gaussian noise. The shapes change colour independently according to a predetermined pattern consisting of a repeating set of 2 to 6 phases, each of length between 50 and 500 frames. The colour of each phase is also randomly pre-assigned from a set of 27 shades defined symmetrically in RGB colour space.

The results are shown in Figure 5.17 as a set of periods evaluated according to the method in Section 5.3 for each 18×18 pixel block, along with the ground truth from the synthesis process. It is evident that the technique is relatively successful, even when the elliptical shapes only partially fill a given square spatial block. The most obvious failure of the method is the cases in which the period estimation is in error by a factor of almost exactly 2 or 3. This is due to the general problem of the dataset being self-similar at all integer multiples of its own length. The spectral plot in Figure 5.18 illustrates the difficulty in establishing the true fundamental peak amongst several harmonically related components. Simply choosing the peak representing the

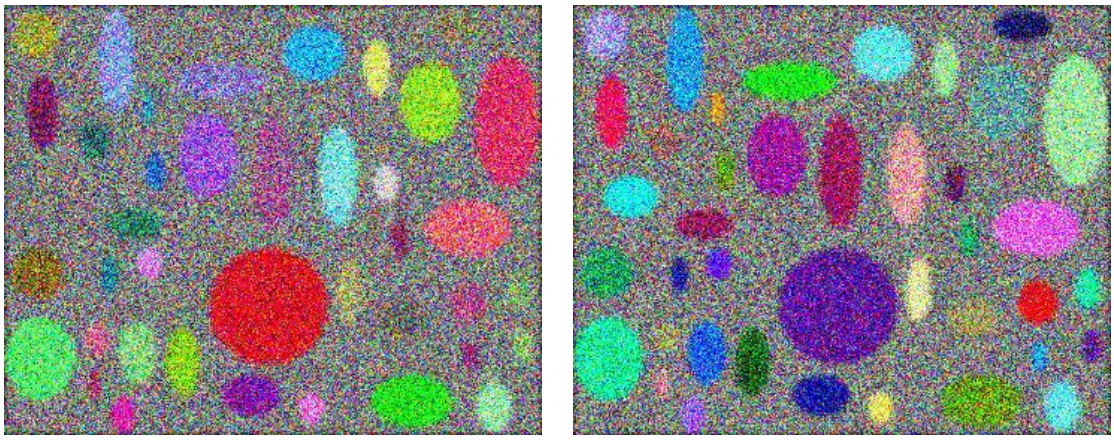


Figure 5.16: Frames from a synthetic sequence in which the elliptical shapes change colour according to a known repetitive predetermined random pattern. Additive white Gaussian noise of variance $\sigma^2 = 0.2$ represents severe sensor noise, compounding the estimation problem.

shortest period is the most obvious solution, but it is not clear how to determine the minimum acceptable amplitude threshold of the selected peak. A related problem is discovering areas with *no* intended periodicity - this is also a problem of choosing an appropriate minimum threshold. However, in many cases the true period is discovered to within 2 seconds without problem.

5.8 Phase-Locked Loop

The spatio-temporal model described so far relies completely on its synchronization with scene activity to provide meaningful results. Failure to maintain synchronization entails failure of the model as a whole. Two significant problems are apparent in relation to this aspect of model reliability. Firstly, the initial estimated periodicity of a block from training data may lack precision, and secondly, video data from the scene may be disrupted, corrupted, or some event in the scene may occur to radically shift the phase of the learned dynamic behaviour. In these and other possible cases, the model may become de-synchronized, and it is highly desirable that the state counter recover automatically from such situations. Essentially this is a question of ensuring *temporal registration* between model and scene. In [8], a Phase-Locked Loop (PLL) was used to recover the frequency and phase of oscillation in the characterization of human gait. The same technique is applied here in the context of the current problem, since the basic requirements of synchronization are common to gait analysis and traffic junction monitoring alike.

A PLL is a negative feedback servo mechanism encountered ubiquitously in electronic systems [4]. Implemented in digital or analogue hardware, or software it is usually constructed from

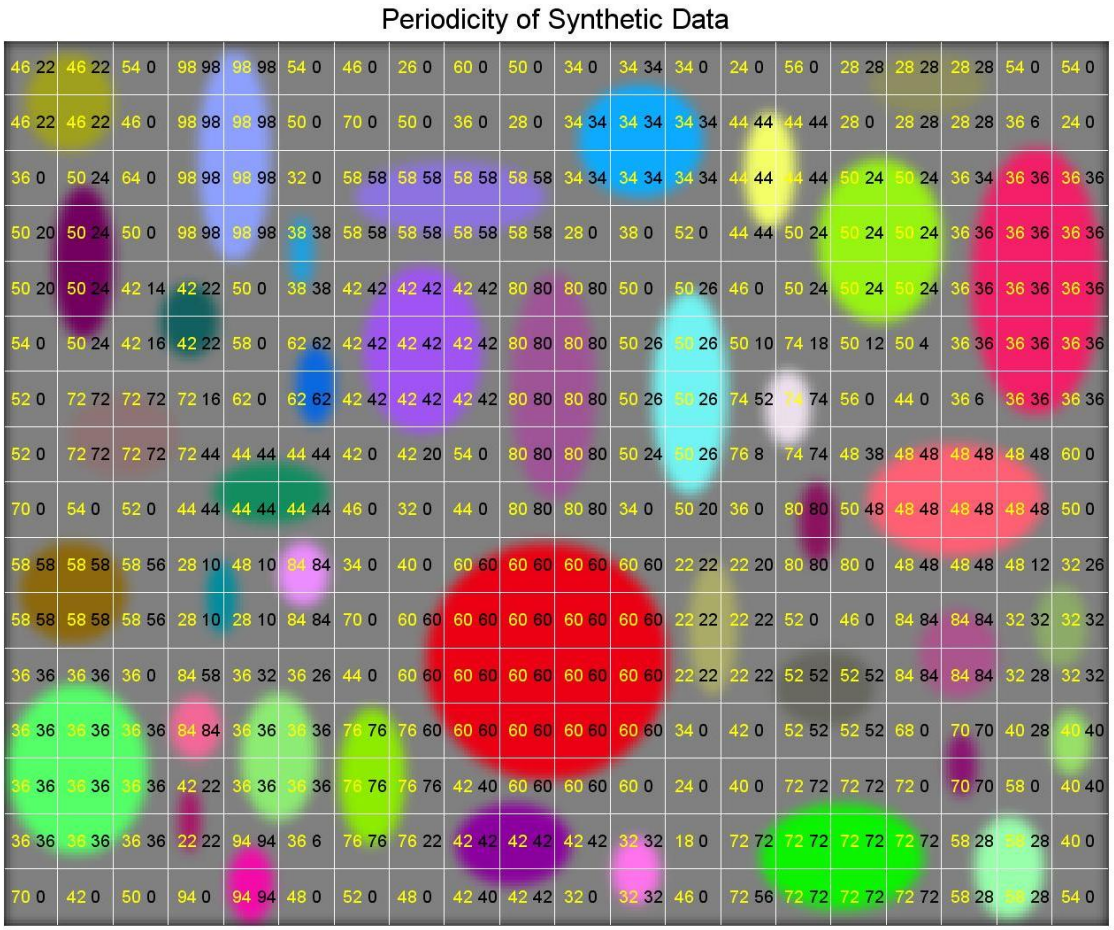


Figure 5.17: Results of periodicity analysis of the synthetic scene in Figure 5.16. Each white square represents an 18×18 pixel spatial block. The yellow figures are estimated period in seconds, whilst the black figures are the ground truth. Many periodicity values are correctly identified, whilst some are recognized as sub-harmonics.

the same functional building blocks as shown in Figure 5.19. It operates by synchronizing a local oscillator in both frequency and phase to a potentially noisy or variable frequency input signal, and is routinely used for demodulation, data recovery and frequency synthesis in communication and data systems. The behaviour of a PLL is largely controlled by the s- or z-plane transfer function of its loop filter, and is designed such that the PLL exhibits an overall system transfer function suitable for a particular task. Common configurations and design equations are analyzed extensively in [4].

Here use is made of the PLL's 'frequency filtering' property to solve the above mentioned shortcomings of the model. By this it is meant that short-term frequency variations, or *jitter*, are rejected, such that the output adopts the long-term average of the input frequency and phase.

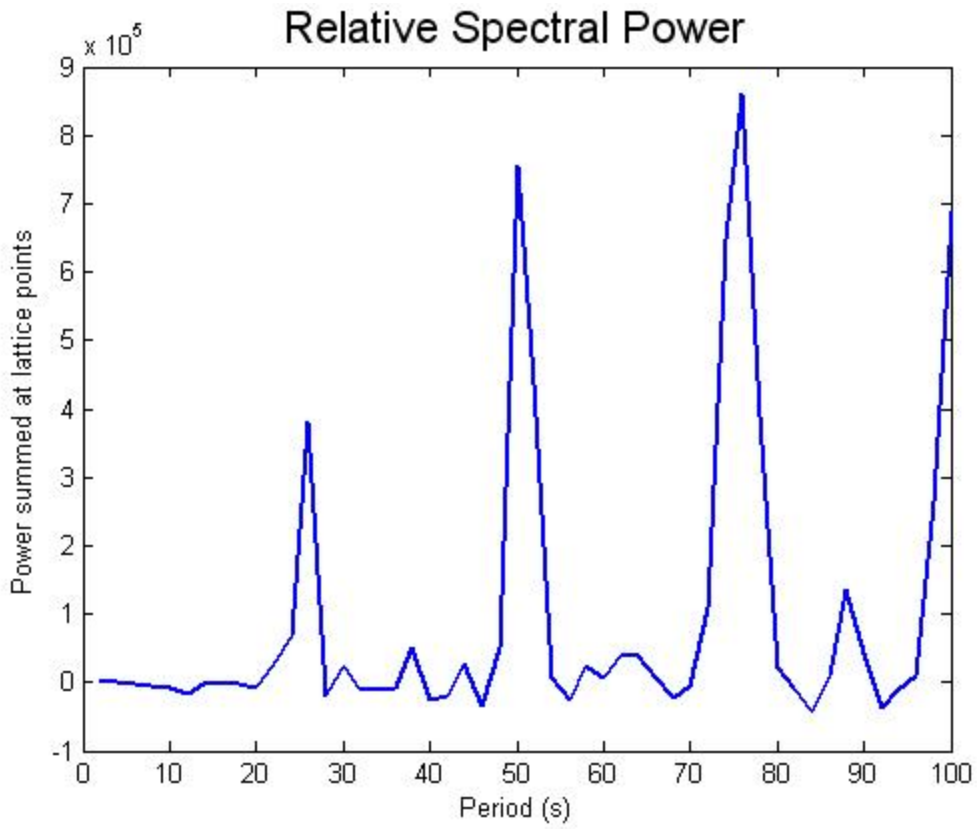


Figure 5.18: Spectral content of synthetic scene in Figure 5.17 at block location $(x,y) = (13,10)$ showing how it is sometimes unclear which peak represents the dominant period. The first peak at 26s is likely to be the fundamental, but components at twice and three times this period have higher energy content so are also contenders for contributing the dominant effect.

This relies on the fact that although the purpose of the spatio-temporal model is to detect *unusual* events in a scene, *on average* the behaviour will be largely consistent. Here, the $\div N$ counter in Figure 5.19 causes the oscillator to run at an integral multiple of the scene's fundamental frequency. The oscillator is implemented in software as a counter or *phase accumulator*, and the higher oscillator frequency yields a finer control of precision of output rate, and hence block periodicity. The *centre frequency* of the oscillator, is that frequency which it produces with zero at its control input. The constant K_{CO} in Figure 5.19 determines the amount of deviation from the centre frequency, up or down, with positive or negative bias at the control input.

5.8.1 Novel Phase Detector

One of the principle components of the PLL is the Phase Detector, which compares a reference input signal with the output of the oscillator, in this case the state counter. The difference between

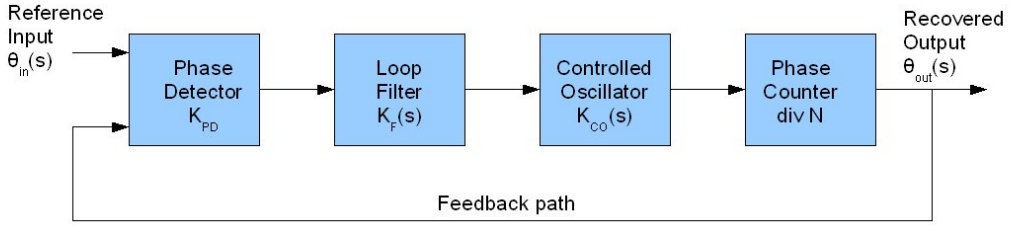


Figure 5.19: A typical Phase Locked Loop (PLL) System consisting of four basic building blocks arranged as a feedback network. Dynamic PLL behaviour is described by the s-plane transfer function $\frac{\theta_{out}(s)}{\theta_{in}(s)} = \frac{K_{PD}K_F(s)K_{CO}(s)/N}{1+K_{PD}K_F(s)K_{CO}(s)/N}$ if all components except the Phase Counter exhibit continuous-time functions, or an equivalent z-plane function if the components are implemented in the digital domain. Design methodologies tend to be somewhat involved, but generally the loop filter response $K_F(s)$ is tailored to yield the desired overall system response given the parameters of the other blocks. In the current application, the output $\theta_{out}(s)$ is designed to take on the long-term average frequency *and* phase of a potentially noisy input signal $\theta_{in}(s)$ derived from the traffic junction phase. Specific parameter selection is detailed in Section 5.8.2.

the two is then used to derive an error signal which corrects the frequency and phase of the oscillator such that it matches the reference input. In the present application, the reference input consists of data from previously unseen input frames. Using the KLD metric definition from Equation (5.13), a novel type of phase detector is introduced here which compares histograms at state l from T_{fund} most recent unseen frames in a circular fashion against the current model at all possible K_{fund} phases as shown in Figure 5.20, in order to determine the optimum

$$k^R_{opt}(l) = \arg\min_j \sum_{h,v\in R} \sum_k D_{KL}(P_{fund,h,v,k},\ Q_{h,v,\ \mathrm{mod}\ (j+l+k,\ K^R_{fund})}) \qquad j,k \in \{1 \ldots K^R_{fund}\} \quad (5.20)$$

for a given region R, where Q is the most recent set of K^R_{fund} normalized histograms from the scene. The phase k^R_{opt} at index offset j exhibiting minimum KLD is considered the optimal target towards which the state counter should be coerced to achieve convergence. Although potentially transiently in error, this phase measurement ensures that on average the model is synchronized to the scene, permitting accurate scene event classification. The loop filter is configured as a Proportional plus Integral type controller, the integral term allowing minimization of zero-order steady-state phase error between the state counter and scene.

A phase accumulator Φ^R_{acc} for region R maintains a high resolution phase representation, and

the state counter $k^R(l)$ for region R at time step l is then derived as a quantized version of this phase accumulator, mapped into the range $\{1 \ldots K^R_{fund}\}$ by the modulus operator

$$k^R(l) = \mathrm{mod}\,(\lfloor \frac{\Phi^R_{acc}(l)}{N} + 0.5 \rfloor, K^R_{fund}) + 1 \tag{5.21}$$

Calculation of the phase error signal k^R_{diff}, which drives the loop towards the locked condition, must be carried out using the following equation so that the *shortest* path around the ring to phase

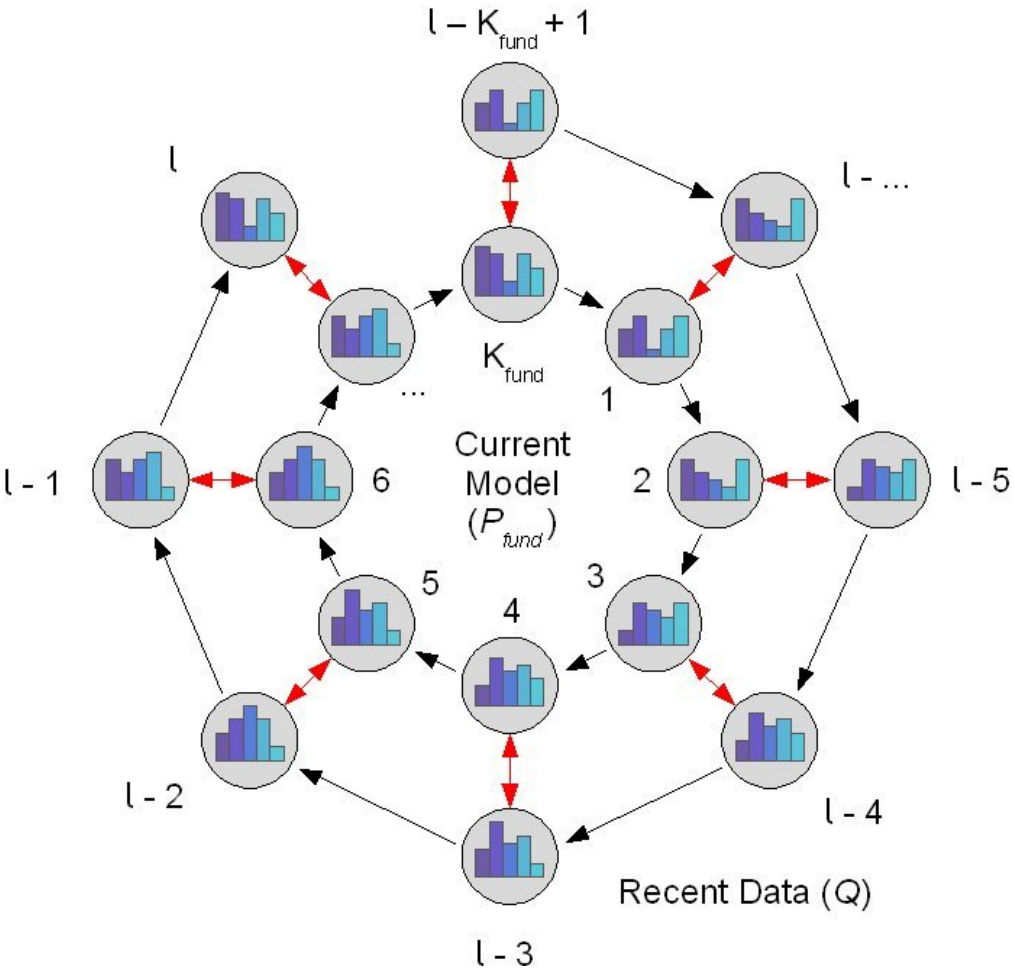


Figure 5.20: The novel phase detector compares the histogram from each model state (inner ring) with one from the K_{fund} most recent histograms from the scene (outer ring) using KLD (shown in red), and the results are summed together. The model (inner ring) is then imagined to 'rotate' in order to evaluate the match at all possible K_{fund} phases, the minimum of which is the optimum k_{opt} and the target for the model. The PLL advances or retards the phase of the model by fractionally increasing or decreasing the oscillator frequency, hence rotating the inner ring relative to the outer, the rate of convergence being determined by parameters G_P and G_I.

lock is always found

$$k^R_{diff}(l) = \begin{cases} k^R_{opt}(l) - k^R(l) - K^R_{fund} & \text{if } k^R_{opt}(l) - k^R(l) > K^R_{fund}/2 \\ k^R_{opt}(l) - k^R(l) + K^R_{fund} & \text{if } k^R_{opt}(l) - k^R(l) < K^R_{fund}/2 \\ k^R_{opt}(l) - k^R(l) & \text{otherwise} \end{cases} \tag{5.22}$$

The PLL oscillator, represented by the phase accumulator Φ^R_{acc} for a given scene region R is then updated by the following relationship

$$\Phi^R_{acc}(l+1) = \Phi^R_{acc}(l) + N + G_P\, k^R_{diff}(l) + G_I \sum_{i=1}^{l} k^R_{diff}(i) \tag{5.23}$$

where G_P is the proportional gain, G_I is the integral gain of the loop filter, and l is the current time index. Without the integral term G_I, a scene fundamental frequency different from the state counter's centre frequency would imply a non-zero steady state output from the Phase Detector, and hence an unwanted continuous offset error between model phase and scene phase.

5.8.2 PLL Experiment

Evaluation of the effectiveness of the PLL stage was performed using the same dataset as previously described for Scenario 1 in Section 5.4. Rather than showing results detailing traffic anomaly detection, the focus here is on maintaining synchronization between scene activity and the model state counter. Failure to maintain track of scene activity will result in failure of the spatio-temporal model as a whole, and this is bound to happen eventually in the absence of any mechanism to prevent it. Thus if maintenance of synchronization, and recovery from desynchronization can be demonstrated under various conditions, then addition of the PLL will be deemed successful.

A software-based PLL was constructed according to the previous description with a Phase Detector configured as in Equation (5.20). The region R was set to comprise the 5×5 group of blocks in the bottom right hand corner of Scenario 1 as depicted in Figure 5.15, having ascertained the local period to be 112 seconds, or 28 four second states. If a spatio-temporal model per block were required, then R would be equivalent to just the block in question.

The PLL loop filter was set up with a proportional gain of $G_P = 3$ and an integral gain of $G_I = 0.02$, these values being determined to be satisfactory by experiment, whilst the PLL multiplier $N = 100$, and the oscillator (state counter) was set to have a centre frequency of $N \times \frac{1}{112}\text{Hz} = 0.893\text{Hz}$, as governed by the local periodicity estimate of 112s.

The results are shown in Figure 5.21 as plots of the estimated scene phase, and the model phase as maintained by the PLL. Given that the vertical axis in each plot represents states in the range 1 to 28, it is clear that when the large vertical serrations of the graph between states 1 and 28 line up, then the state counter is synchronized with scene activity. The centre trace, which represents the state of the nearest green traffic light in the scene, supports the evidence that the PLL has correctly achieved lock.

The PLL starts from the unsynchronized state at $t = 0$, but by $t = 50$ is almost completely synchronized. A burst of noise at $t = 100$ makes it impossible to determine the instantaneous state of the traffic junction, but the *inertia* bestowed by the integrating loop filter ensures a plausible interpolation by the PLL until proper synchronization is re-established by $t = 170$.

Thus it can be concluded that inclusion of the PLL into the model framework has enabled a more robust approach to accurately tracking scene activity patterns. Furthermore, the Proportional plus Integral loop filter permits synchronization to a range of periods *close* to the fundamental period of the scene, so errors in initial estimation or quantization of K_{fund} are easily accommodated.

5.8.3 Assessing PLL Performance

The discussion so far has assumed that the PLL behaves in an 'ideal' way, performing perfect fundamental period recovery in the presence of unquantified noise mechanisms without losing lock. In reality, even PLLs modelled in software have performance limitations which must be mitigated in order for the overall system to operate satisfactorily. The principal requirements upon the PLL in the current application are that:

1. The PLL rapidly acquire the locked condition relative to scene activity, starting from the unlocked condition.

2. The PLL maintain lock during and in spite of significant levels of noise, which potentially may act as de-synchronizing information.

3. The short term frequency accuracy and stability of the PLL throughout the activity cycle be sufficiently good to guarantee acceptably low rates of false positive and false negative detections within the context of the application at hand.

5.8.4 PLL Parameters

Translation of a particular set of design requirements into a detailed specification of PLL building blocks is covered extensively in [4], but the main performance parameters are as follows:

1. Lock Time - This is the maximum time that the PLL takes to achieve the locked condition starting from any possible combination of oscillator phase and external signal phase.

2. False Lock - This is the situation where the PLL has locked to an integer multiple or fraction of the desired fundamental frequency. The harmonic content of the input signal and the design of the phase detector both affect the likelihood of false lock occurring.

3. Capture Range - The range of frequencies above and below the natural PLL centre frequency from which lock may be achieved. The absolute amplitude of the fundamental component, and the relative amplitude of other components may affect the extremes of the range.

4. Lock Range - The range of frequencies which may be continuously tracked without loss of lock given that the PLL is already in the locked condition. The presence of noise may affect achievable lock range.

5. Jitter Rejection - This represents the PLL's ability to attenuate short-term timing errors in the input signal, which might alternatively be viewed as Frequency Modulation of the fundamental frequency by a combination of noise and unwanted correlated components. This parameter, largely dictated by the loop filter, is directly related to the Lock Time and Lock Range. In particular, the requirements on the loop filter for a fast Lock Time (a high cut-off frequency) are directly opposed to those for achieving good Jitter Rejection (a low cut-off frequency), so a compromise must be struck. It is possible to get the best of both worlds by lowering the loop filter cut-off frequency once lock is achieved.

6. Phase Noise - The level of jitter produced by the oscillator itself. In a software model of a PLL this parameter barely exists, but more generally, as a negative feedback system the PLL should be able to attenuate its own jitter as well as that present in the input signal.

7. Transient Response - Closely related to the loop transfer function, this parameter defines the dynamic behaviour of the oscillator with respect to step changes in the input signal.

Design of a PLL to deliver a particular level of performance requires detailed knowledge of the input signal. In the case of scene analysis, signal characteristics are concealed within activity detected by the particular feature being derived. Although the fundamental period may be determined manually by counting frames between crucial scene events, such as by monitoring the pixels occupied by traffic lights, more subtle ground truth signal structure is much harder to perceive, being largely stochastic in nature, and distributed throughout the training set.

5.8.5 PLL Evaluation

In the case of the Phase Detector described in Section 5.8.1, the content of a given cycle of scene activity in isolation is of little value - it is the KLD of a cycle of recent histograms with respect to *other* cycles which is important. Analysis of a sizeable block of training data should permit a sufficiently accurate statistical description of the scene such as to enable sensible decisions to be made about the requirements of the PLL. The procedure would involve determining both short and long-term fundamental frequencies in the training data by the method described previously in Section 5.3.

In the absence of a large body of typical training data, a given PLL configuration may be analyzed based on synthetic data. Using artificial images generated as described in Section 5.7, cyclic scene elements may be fabricated with a wide variety of characteristics. Most significantly, the frequency may be modulated according to any arbitrary scheme, thus enabling the PLL's response in terms of Lock Time, Lock Range, Capture Range and Jitter Rejection all to be determined directly. However, whilst responses to the synthetic data may be proven in this manner, the subsequent suitability of a given PLL design for use in a *real-world* situation is still a matter for future research.

5.8.6 The PLL as a Frequency Estimator

Given that a PLL can acquire and lock to any frequency in a signal presented to its input, it might be thought that the complex process of determining K_{fund} described in Section 5.3 could be dispensed with. However, under certain conditions, it is possible for a PLL to lock to a frequency component which is unrelated to the dominant fundamental. Furthermore, locking to a multiple or sub-multiple of the dominant fundamental is also possible. For these reasons, the frequency *capture range* of the PLL needs to be restricted by design so that the centre frequency and useful operating range of the oscillator are in the vicinity of the true fundamental to be acquired. Thus,

separate approximate evaluation of K_{fund} *is* necessary, and subsequent PLL design parameters should be determined directly from it. In addition, the novel Phase Detector as defined in Section 5.8.1 also requires initial knowledge of approximate state cycle length.

5.9 Summary

An algorithm capable of automatically learning periodic activity within a scene has been described. The algorithm determines periodicity by analyzing self-similarity of PDF over some feature with respect to time, and then builds a data driven model across multiple cycles of this period from the training sequence. The model has been presented in two different ways, firstly imposing a single global periodicity across a scene, and then on a block-wise basis, supporting multiple periodicities within the scene. This *scalability* aspect is likely to be highly relevant in practical systems.

The spatio-temporal model has been shown to be useful in the context of road junction surveillance, where traffic regulated by consistently timed signals displays obvious periodicity. It has been demonstrated by experiment that the method is more discriminating with regard to the activity of a periodic scene than a model which is oblivious to recurrent temporal trends. The marked improvement in detection sensitivity comes exclusively from exploiting the learned model such that the expected instantaneous distribution over feature space is tightly coupled to the junction state cycle. Furthermore, the method is not tied to any particular feature, and may be deployed *wherever* a histogram over feature(s) is available.

By inclusion of a novel phase detector and control loop, it has been shown possible to maintain model synchronization in the presence of drift in the fundamental period, and to recover synchronization following disruption due to noise. The logical progression of the technique is to permit automatic on-line update of histogram data for a block when its PLL is known to be in the *locked* condition. Lock detection should be possible by monitoring the mean and variance of the Phase Detector output, which will both be small if the system is in lock.

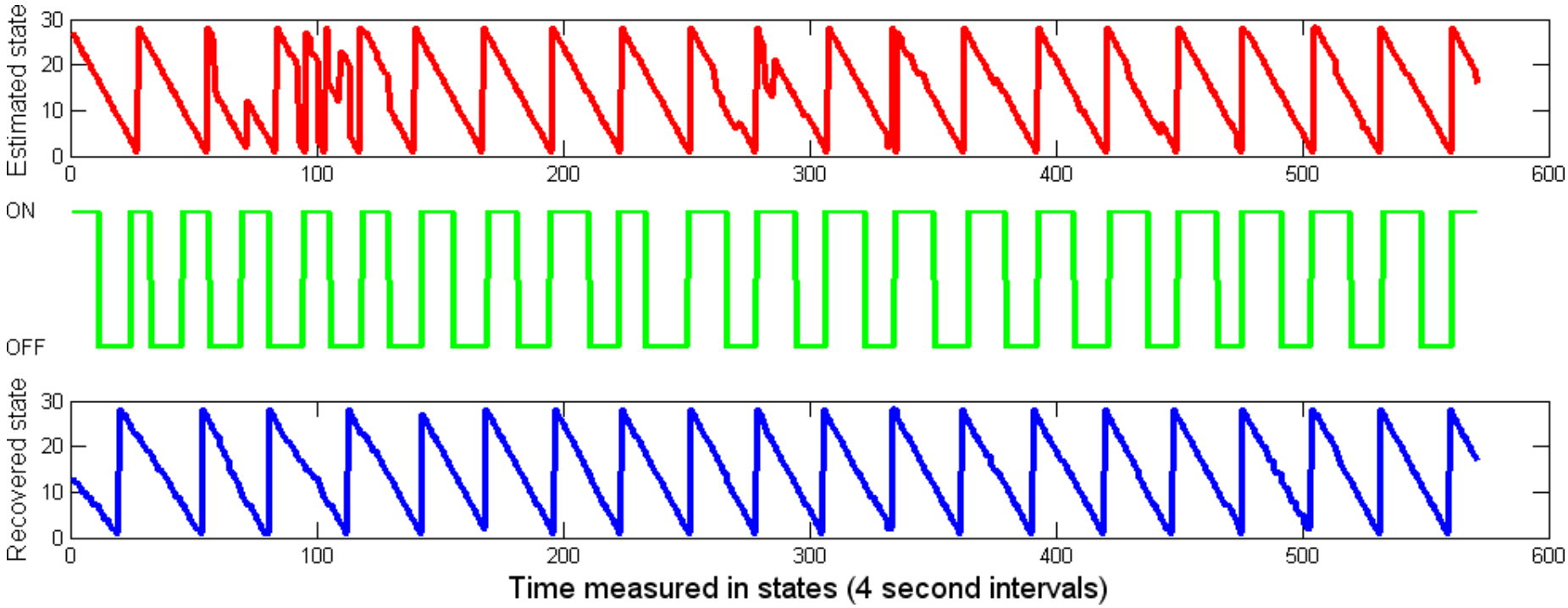


Figure 5.21: Benefit of PLL on model phase stability. The vertical axis depicts a modulo-28 decrementing state counter. The upper (red) trace shows the optimal *instantaneous* match between the model and the 28 most recently acquired 4 second blocks. However, in the presence of noise, this alone gives a poor estimate of the scene's real state, in this example especially at around t=100. The lower (blue) trace represents the state estimate from the PLL output. Note its effective acquisition of approximately correct phase soon after t=0, and how it maintains a useful output phase even during corrupted input data due to the *inertia* caused by the loop filter. Acting as a servo system, the phase detector and loop filter *advance* or *retard* the state counter's rate of change (frequency) in order to *chase* until optimal phase lock has been achieved. In the locked condition, the PLL output state (lower trace) matches the estimate (upper trace) *on average*, as is clearly indicated by the coincidence of vertical serrations representing transitions from state 1 to state 28. Crucially, this mechanism adapts to changes in fundamental period as well as phase. The middle (green) trace is the ON/OFF state of the nearest green traffic light in Figure 5.15.

Chapter 6

Conclusion and Future Work

This thesis has set out to explore the possibility of using various support strategies in segmentation of foreground from background in video sequences. Here the term *foreground* refers to any pixel or object which is deemed unusual in the context of the learned model, and *background* represents the normality, which may include dynamic objects and activity. The underlying motivation is enhancement in the reliability of the decision process by augmenting evidence at a given pixel with further consensus from its spatio-temporal locality, since there is a limit to what can be done when considering pixels in isolation. Applications are numerous, but abnormality detection in the field of surveillance and monitoring represents a strong candidate to benefit from advanced techniques, whether for on-line threat analysis, or retrospective search.

As concluded in Chapter 2, the available literature strongly suggests that there is considerable scope for providing or improving local support between pixels in a number of key ways, and the investigation of each is summarized below.

6.1 Pattern-based Spatial Support

In terms of spatial support, the two most popular current algorithms consist of the per pixel models [125] offering no pixel linkage, and at the opposite extreme, the subspace models of [94] imposing total global connectivity. Other contemporary methods use Markov Random Fields such as [111] in which a fixed penalty cost for a segmentation boundary helps to induce primitive local support. The algorithm described in Chapter 3 develops the MRF approach further by imposing penalties dependent on relative cooccurrence of a feature at adjacent image locations.

The RSLBP_4 operator is introduced as a feature representing local image pattern and gradient direction. This is a simple derivative of the commonly encountered LBP_8 operator [136, 93] which has deliberately been rendered sensitive to orientation. Because it generates only 16 different symbols, a cooccurrence matrix between two adjacent pixel locations has only 256 entries and is thus practical in terms of memory requirement, and may be adequately estimated with a modestly sized block of training data.

Cooccurrence between *every* pair of adjacent pixels in a 4-connected scheme is accumulated from training data, and used to calculate penalty terms related to the conditional probability of a symbol given its neighbour. The penalty terms form the inter-pixel weights in a binary graph cut, yielding a segmentation in which evidence from a pixel's locality lends support to the background/foreground decision. Asymmetric arc weights reflect the different mutual conditional probabilities associated with the direction of the support.

Experiments have shown a useful degree of segmentation improvement in difficult situations where objects are partially occluded by a fragmented chaotic background, in this case people walking behind moving foliage.

6.1.1 Future Work

With regard to spatial support, although a model involving separate collection of training data is described in Chapter 3, it is anticipated that an adaptive on-line derivative would also be possible. In such a scenario, the cooccurrence database would be built and updated in the light of new incoming frames. Providing that a suitable learning rate can be found, the conditional distributions C_h and C_v between adjacent pixels will approximately converge, become more refined, and be tracked over time, exploiting the advantage of the ever-increasing body of training data.

From the point of view of the RSLBP_4 operator, there are two obvious directions in which to extend the technique:

1. The scale at which the operator is applied may be too small to yield a useful cooccurrence distribution, in the sense that all pairs of neighbour symbols are equally likely, and no selective support can be gained. But downsampling the image before applying the operator may reveal a more stable structure and useful cooccurrence matrix.

2. Currently the operator deals only with the two spatial dimensions, but a similar operator in three dimensions based on a 6-connected set of pixels would also be possible. Of course

the set of generated symbols would increase to 64 and the cooccurrence matrix would then have 4096 entries, but the new operator would be capable of detecting directional gradients in the full spatio-temporal field, and thus have scope for capturing a yet wider variety of motion patterns, such as may be present in a particular application.

With regard to the asymmetric binary graph-cut, there are also attractive extensions:

1. The present algorithm only considers cooccurrence between adjacent pixels in the *spatial* plane, but there is no reason why *temporal* cooccurrence cannot be learned between the same pixel location in two successive frames, or across a wider time interval. Certainly the cooccurrence database would become much larger, commanding a greater volume of training data. But the required graph cut would still be binary, and hence soluble in polynomial time.

2. Applying a graph cut to induce local support is not limited to being used with RSLBP_4. Any other feature for which a cooccurrence map can be obtained could be substituted. In fact, any feature exhibiting asymmetric support between two locations may usefully be exploited using this strategy.

The principle problem with generalizing the cooccurrence concept is the constraint on storage of the cooccurrence map. For short discrete descriptors this approach is not an issue, but what is really required is an efficient method of expressing the relationship between adjacent pixels for *any* arbitrary continuous distribution. Direct storage of the map implies a dimensionality of twice that of the chosen feature itself, and the relationship has to be formed between *every* pair of adjacent pixels. But although some compression or approximation may well be possible, such as PCA or Local Linear Embedding (LLE) [105], the computational complexity is likely to be prohibitive.

6.2 Short-term Temporal Support in Busy Scenes

In many busy surveillance scenes, a significant part of the background is occluded by both vehicles and pedestrians for a large proportion of the time. Such a situation makes it difficult to reliably estimate the *true* background with much accuracy, whether using a Gaussian Mixture Model or an eigenspace model. Although it could be argued that the traffic *is* an integral part of

a dynamic background, if it is actually the traffic which is to be subsequently detected as foreground, then ideally for maximum sensitivity, the background model should be constrained to the truly static elements from which the scene is composed. Of course the question arises as to how to separate vehicles and pedestrians from everything else, such that they *can* be ignored.

A method is detailed in [19] for selecting or rejecting image regions principally on the basis of temporal stability and spatial consistency. Using binary graph cuts and the α-expansion technique [10] an approximately optimal short-term background may be obtained from a small temporally localized block of frames by combinatorial optimization. The estimate is achieved by pixel-wise assembly from the candidate input frames based on penalizing temporal stability and presence of motion boundaries as the *data* terms, and intensity matching as the interaction term between neighbouring pixels.

By applying the algorithm to successive blocks of frames, a series of estimates of the static background is obtained, from which a conventional eigenspace model may be built. Thus a degree of pre-filtering is applied to the data, resulting in the removal of much if not all of the unwanted clutter, and leaving longer-term effects such as illuminations changes to be incorporated into the subsequent model.

Chapter 4 describes experiments in which this process is put into practice, showing that the eigenspace model constructed from the pre-filtered 'backgrounds' models a much higher proportion of scene variance energy with fewer eigenvectors than a model with no pre-filtering. It is thus concluded that the approach has significant value in terms of background recovery from busy scenes. The overall result is short-term support amongst appropriately selected pixels, arrived at by rejecting combinations which fail some generic prerequisites.

Further experiments show that *initial labelling* of pixels from the candidate input frames has little practical effect on the exact final labelling, suggesting that the α-expansion process is reliably efficient, if not always exact. Experiments to vary the size and temporal spacing of the candidate input frame block, and the balance between the three penalty contributions show the algorithm to be fairly resilient to the exact choice of these parameters.

6.2.1 Future Work

The algorithm described in Chapter 4 currently relies on choosing the input sampling rate and the input block size to suit the scene being modelled. Although the method is not particularly sensitive to the exact values chosen, as the experiments show, it could be made more robust if the

two parameters were estimated automatically. Given that the effect of inappropriate parameter choices is to increase the amount of foreground clutter in the short-term estimates, automatic adjustment should be possible either by maximizing the stability globally across the scene over a number of estimates, or by monitoring the minimum values of the objective function being achieved.

The graph in Figure 4.14 shows that above a certain number of images per block, the advantage gained per extra image in terms of energy minimization becomes small. Thus the optimization should maintain the number of images needed to keep the gradient of this graph at a certain value, trading image estimate purity against processing requirement. Optimization of the sampling rate should be easier, since according to the curve depicted in Figure 4.16 a clear minimum in objective function energy is to be found.

6.3 Long-term Temporal Support in Recurrent Scenes

Much is to be found in the literature on the subject of learning motion and activity patterns in video data relating to a fixed scene, mainly from a surveillance point of view, and with the principal objective of detecting unusual behaviour with respect to training data [126, 78, 148].

But some types of scene depict activity of a repetitive nature, a prime example being road traffic junctions regulated by accurately timed traffic lights. Here the activity of vehicles and pedestrians at the junction may exhibit strong periodicity. In such cases, the task of identifying motion patterns takes on a new dimension - the possibility of casting the concept of 'unusual' behaviour as that which fails to be explained by a specifically periodic model of the road junction. In fact, in such situations, it should be considered a *waste* of the available information to ignore this attribute, yet little is to be found in the literature on solving the problem.

In general, the techniques cited above concentrate on the probabilistic motion of an object entering a scene, traversing some path across it, and leaving the scene, all as a single event unrelated in any particular way to other events, apart from their stochastic coincidence. However, Chapter 5 describes an approach which explicitly models periodic behaviour of objects in a scene, with a view to detecting vehicles and pedestrians which violate the normal trend of activity.

Crucial to the success of a periodic model is a method of determining the fundamental period of scene activity. This may be considered locally on a region or block basis in the case of scenes which manifest multiple periodicity in different areas, or as a single period for the whole image,

where the model amounts to a 'signature' of synchronized motion patterns within the scene. Both approaches are covered in the experiments described.

Either way, the scene is characterized by a cyclic ring of histograms over some appropriate feature, which is traversed by a timing index related directly to the phase of activity in the scene - in the case of the road junction, the traffic light sequence and timing. New frames may be compared with the model by extracting the same feature and comparing with the histogram selected by the time index, providing this index is correctly synchronized. Errant behaviour of several vehicles is depicted in experiments involving periodic road junctions by using optical flow or bounding box aspect ratio as the feature.

In acknowledgment of the need for reliable synchronization between internal model time index and scene activity in a practical system, Chapter 5 includes a description of an approach using a Phase Locked Loop (PLL) to explicitly maintain synchronization. By using a novel software *phase detector*, the PLL implementation ensures that the model regains synchronization following disruption of the incoming data stream, and has the capability to overcome small errors in estimation of the original period.

Overall, use of the periodic model described herein represents an exploitation of behaviour support due to locality in *periodic cycle* rather than linear time.

6.3.1 Future Work

So far the method described in Chapter 5 has only been tested on datasets exhibiting cyclostationary statistics. A practical system for monitoring traffic junctions would need to be able to adapt do scene statistics which vary over time, possibly daily, such as changes in relative traffic light timing which anticipate *tidal* traffic flow. Updating model histogram data from recent input frames should be possible, but it is essential that this learning process only be allowed to happen when the PLL is in the locked state. Only under this condition is it certain that scene statistics are contributing to the correct phase of the model.

The previously mentioned tidal traffic flow is also an example of a periodic process within another periodic process - the changes in traffic flow over several minutes due to the cycle of the traffic lights inside a daily shift in overall scene statistics due to the changing dominant flow of traffic between the beginning and end of the working day. The two periods will not necessarily be harmonically linked, so a more advanced form of fundamental frequency detection and model structure would be required. A similar situation might arise if the model were used in a medical

application to monitor heart activity, in which a separate and unrelated periodicity, perhaps due to breathing, might affect the visible area.

A basic limitation of the current model is that it deals only with *periodic* activity. On the other hand, previous work detailed in Section 2.4.1 describes methods which deal solely with *stochastic* behaviour. An important goal of future research would be to define a mathematical framework which naturally simultaneously encompasses both periodic and stochastic data. Although it is not yet clear how such a model may be formulated, the current model with a state counter based approach shares some similarities with the Hidden Markov Model technique detailed by Swears et al. in [129]. A hybrid model could seamlessly exploit periodic information where it happened to occur, yet automatically invoke probabilistic model aspects under other conditions.

6.4 Summary

Computer Vision research in general seems to be on a *quest* for the Holy Grail that is the 'perfect' background model, which learns all its own parameters, and avails ideal results in challenging environments. But in the end, it may well be the operational requirements and available computing power which dictate the 'optimal' approach for any given application. In any case, good techniques are likely to centre on *compounding* information between pixels to induce *spatio-temporal support* in a model, since considering pixels in isolation goes only so far.

This thesis has investigated three radically different support mechanisms in response to the quest for the ideal background model. None represents a panacea in its own right, but rationalization and unification with existing methods to maximize the depth of the modelled information should be a goal for future research.

It may take many years of research to develop an artificial vision system to rival its biological counterpart in terms of both high and low level visual processing. But some types of vision problem, notably surveillance, stand to benefit significantly from the machine vision approach. Computer systems hosting vision algorithms are deterministic, have vast storage capacity, and can communicate quickly over wide areas via high bandwidth data connections. All of these attributes facilitate an operational paradigm unavailable directly to biological systems, so it should not be surprising that agents based on computer vision may excel in quite different, perhaps complementary ways from humans.

Bibliography

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(2):284–299, 1985.

[2] E. H. Adelson and J. Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):99–106, 1992.

[3] R. E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.

[4] R. E. Best. *Phase-Locked Loops: Theory, Design and Applications*. McGraw-Hill, New York, 1993.

[5] M. J. Black and P. Anandan. A model for the detection of motion over time. In *International Conference on Computer Vision*, pages 33–37, Osaka, 1990.

[6] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[7] O. Boiman and M. Irani. Detecting irregularities in images and in video. *International Conference on Computer Vision*, 1:462–469, 2005.

[8] J. E. Boyd. Synchronization of oscillations for machine perception of gaits. *Computer Vision and Image Understanding*, 96(1):35–59, October 2004.

[9] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[10] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[11] M. Brand. Incremental singular value decomposition of uncertain data with missing values. Preprint TR-2002-24, Mitsubishi Electric Research Laboratory, May 2002.

[12] J. R. Bunch and C. P. Nielsen. Updating the singular value decomposition. *Numerische Mathematik*, 31:111–129, 1978.

[13] H. Buxton. Learning and understanding dynamic scene activity: a review. *Image and Vision Computing*, 21(1):125–136, 2003.

[14] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[15] M. Casdagli. Recurrence plots revisited. *Physica D*, 108:12–44, 1997.

[16] 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(5):321–332, September 1997.

[17] S. S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. In *Proceedings of SPIE Visual Communications and Image Processing 2004*, volume 5308, pages 881–892, January 2004.

[18] L. Cohen. *Time-Frequency Analysis: Theory and Applications*. Prentice Hall, New Jersey, 1994.

[19] S. Cohen. Background estimation as a labeling problem. In *International Conference on Computer Vision*, pages 1034–1041, Beijing, China, October 2005.

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

[21] W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley and Sons Inc, New York, 1998.

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

[23] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *Society for Industrial and Applied Mathematics Journal on Computing*, 23(4):864–894, 1994.

[24] N. Dalal and W. Triggs. Histograms of oriented gradients for human detection. In *International Conference on Computer Vision and Pattern Recognition*, pages 886–893, San Diego, 2005.

[25] J. Davis. Mosaics of scenes with moving objects. In *International Conference on Computer Vision and Pattern Recognition*, pages 354–360, 1998.

[26] R. D. DeGroat and R. A. Roberts. Efficient, numerically stabilized rank-one eigenstructure updating. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(2):301–316, February 1990.

[27] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[28] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[29] E. A. Dinic. Algorithm for solution of a problem of maximum flow in a network with power estimation. *Soviet Math. Doklady*, 11:1277–1280, 1970.

[30] E. R. Dougherty and R. A. Lotufo. *Hands-on Morphological Image Processing*. SPIE Press, Bellingham, WA, 2003.

[31] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons Inc, New York, 2001.

[32] J. Edmonds and R. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the Association for Computing Machinery*, 19(2):248–264, 1972.

[33] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *European Conference on Computer Vision*, pages 2:751–767, Dublin, Ireland, May 2000.

[34] H. L. Eng, K. A. Toh, A. H. Kam, J. Wang, and W. Y. Yau. An automatic drowning detection surveillance system for challenging outdoor pool environments. In *International Conference on Computer Vision*, pages 532–539, Nice, France, October 2003.

[35] R. A. Fisher. Tests of significance in harmonic analysis. In *Proceedings of the Royal Society A*, volume 125, pages 54–59, London, 1929.

[36] L. Ford and D. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[37] D. A. Forsyth and J. Ponce. *Computer vision modern approach*. Prentice Hall, 2002.

[38] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 175–181, Providence, Rhode Island, August 1997.

[39] W. A. Gardner, A. Napolitano, and L. Paura. Cyclostationarity: Half a century of research. *Signal Processing*, 86(4):639–697, 2006.

[40] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[41] M. Gill and A. Spriggs. Assessing the impact of CCTV. Technical Report No. 292, United Kingdom Home Office Research, Development and Statistics Directorate, February 2005.

[42] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. *Journal of the Association for Computing Machinery*, 35(4):921–940, 1988.

[43] O. Goldschmidt and D. S. Hochbaum. Polynomial algorithm for the k-cut problem for fixed k. *Mathematics of Operations Research*, 19(1):24–37, 1994.

[44] G. H. Golub. Some modified matrix eigenvalue problems. *Society for Industrial and Applied Mathematics Review*, 15(1):318–344, 1973.

[45] S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *International Conference on Computer Vision*, Nice, France, 2003.

[46] P. M. Hall, A. D. Marshall, and R. R. Martin. Incremental eigenanalysis for classification. In *British Machine Vision Conference*, pages 286–295, Southampton, UK, September 1998.

[47] P. M. Hall, A. D. Marshall, and R. R. Martin. Merging and splitting eigenspace models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1042–1049, September 2000.

[48] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2003.

[49] M. H. Hayes. *Statistical Digital Signal Processing and Modeling*. John Wiley and Sons Inc, New York, 1996.

[50] M. Heikkilä and M. Pietikäinen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):657–662, 2006.

[51] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *International Conference on Computer Vision Frame-Rate Workshop*, Kerkyra, Greece, 1999.

[52] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(0):417–441, 1933.

[53] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1450–1464, 2006.

[54] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, March 1964.

[55] H. Hung and S. Gong. Quantifying temporal saliency. In *British Machine Vision Conference*, pages 742–749, September 2004.

[56] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[57] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 31:253, 1925.

[58] R. Jain and H. Nagel. On the analysis of accumulative difference pictures from image sequence of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):206–214, 1979.

[59] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, May 1957.

[60] I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, New York, 2002.

[61] T. Kadir and M. Brady. Saliency, scale and image description. *International Journal of Computer Vision*, V45(2):83–105, November 2001.

[62] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *European Conference on Computer Vision*, Prague, 2004.

[63] G. Kanizsa. *Organization in Vision: Essays on Gestalt Perception*. Praeger, New York, 1979.

[64] P. J. Kellman and T. F. Shipley. A theory of visual interpolation in object perception. *Cognitive Psychology*, 23(2):141–221, April 1991.

[65] S. Khan, O. Javed, Z. Rasheed, and M. Shah. Human tracking in multiple cameras. In *International Conference on Computer Vision*, pages 331–336, Vancouver, 2001.

[66] J. Kleinberg and É. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *Journal of the Association for Computing Machinery*, 49(5):616–639, 2002.

[67] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[68] V. Kolmogorov and R. Zabih. Graph cut algorithms for binocular stereo with occlusions. In *Mathematical Models in Computer Vision: The Handbook*. Springer Verlag, 2005.

[69] P. Koonce. *Traffic Signal Timing Manual*. Federal Highway Administration, Turner-Fairbank Highway Research Center, McLean, VA, 2008.

[70] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[71] J. T. Kwok and H. Zhao. Incremental eigen decomposition. In *International Conference on Artificial Neural Networks*, pages 270–273, Istanbul, Turkey, June 2003.

[72] F. De la Torre and M. J. Black. Robust principal component analysis for computer vision. In *International Conference on Computer Vision*, pages 362–369, Vancouver, Canada, July 2001.

[73] Y. Li. On incremental and robust subspace learning. *Pattern Recognition*, 37(7):1509–1518, July 2004.

[74] F. Liu and R. W. Picard. Finding periodicity in space and time. In *International Conference on Computer Vision*, pages 376–383, 1998.

[75] F. Liu and R.W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:722–733, 1996.

[76] J. M. Lloyd. *Thermal Imaging Systems*. Kluwer Academic Publishers, Hingham, MA, 1975.

[77] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, Canada, 1981.

[78] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(3):397–408, 2005.

[79] K. V. Mardia and P. E. Jupp. *Directional Statistics*. John Wiley and Sons Inc, New York, 2000.

[80] T. Matsuyama, T. Ohya, and H. Habe. Background subtraction for nonstationary scenes. In *Asian Conference on Computer Vision*, pages 662–667, Taiwan, 2000.

[81] M. McCahill and C. Norris. CCTV in London. Technical Report No. 6, UrbanEye, Zentrum Technik und Gesellschaft, Technische Universität Berlin, June 2002.

[82] N. J. B. McFarlane and C. P. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8(3):187–193, 1995.

[83] A. M. McIvor, Q. Zang, and R. Klette. The background subtraction problem for video surveillance systems. In *Robot Vision, International Workshop RobVis*, pages 176–183, Auckland, February 2001.

[84] A. Michotte. *La perception de la causalité*. Leuven University Press, Leuven, 1954.

[85] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *International Conference on Computer Vision and Pattern Recognition*, pages 270–273, Washington, D.C., June 2004.

[86] H. Murakami and B. V. K. V. Kumar. Efficient calculation of primary images from a set of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(5):511–515, May 1982.

[87] K. Nakayama, Z. J. He, and S. Shimojo. Visual surface representation: A critical link between lower-level and higher-level vision. In D. N. Osherson and S. M. Kosslyn, editors, *Visual Cognition and Action: An Invitation to Cognitive Science (Volume 2) 2nd Edition*, pages 1–70. MIT Press, Cambridge, MA, 1995.

[88] T. Natschläger and B. Ruf. Spatial and temporal pattern analysis via spiking neurons. *Network: Computation in Neural Systems*, 9(3):319–332, 1998.

[89] R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

[90] J. Ng and S. Gong. On the binding mechanism of synchronised visual events. In *IEEE Workshop on Motion and Video Computing*, December 2002.

[91] M. S. Nixon and A. S. Aguado. *Feature Extraction and Image Processing*. Elsevier Science and Technology, Amsterdam, 2008.

[92] S. J. Nowlan. *Soft competitive adaptation: Neural Network learning algorithms based on fitting statistical mixtures*. PhD thesis, Carnegie Mellon University, 1991. CS–91–126.

[93] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, January 1996.

[94] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–841, August 2000.

[95] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, pages 555–562, Bombay, 1998.

[96] N. Paragios and V. Ramesh. An mrf-based approach for real-time subway monitoring. In *International Conference on Computer Vision*, 2001.

[97] M. Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man & Cybernetics*, pages 3099–3104, 2004.

[98] V. F. Pisarenko. The retrieval of harmonics from a covariance function. *Geophysics Journal of the Royal Astronomical Society*, 33:347–366, 1973.

[99] R. Pless. Spatio-temporal background models for outdoor surveillance. *EURASIP Journal on Applied Signal Processing*, 2005(14):2281–2291, 2005.

[100] R. Polana and R. C. Nelson. Detection and recognition of periodic, nonrigid motion. *International Journal of Computer Vision*, 23(3):261–282, 1997.

[101] F. Porikli and C. Wren. Change detection by frequency decomposition: Wave-back. Technical Report TR-2002-34, Mitsubishi Electric Research Laboratory, December 2005.

[102] R. B. Potts. Some generalized order-disorder transformation. In *Transformations, Proceedings of the Cambridge Philosophical Society*, volume 48, pages 106–109, 1952.

[103] B. G. Quinn and E. J. Hannan. *The Estimation and Tracking of Frequency*. Cambridge University Press, 2001.

[104] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, 2005.

[105] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[106] D. Russell and S. Gong. A highly efficient block-based dynamic background model. In *IEEE International Conference on Advanced Video and Signal based Surveillance*, pages 417–422, Como, Italy, September 2005.

[107] D. Russell and S. Gong. Minimum cuts of a time-varying background. In *British Machine Vision Conference*, pages 809–818, Edinburgh, September 2006.

[108] D. Russell and S. Gong. Segmenting highly textured nonstationary background. In *British Machine Vision Conference*, pages 550–559, Warwick, September 2007.

[109] D. Russell and S. Gong. Exploiting periodicity in recurrent scenes. In *British Machine Vision Conference*, pages 715–724, Leeds, September 2008.

[110] D. Russell and S. Gong. Multi-layered decomposition of recurrent scenes. In *European Conference on Computer Vision*, volume 3, pages 574–587, Marseille, October 2008.

[111] K. Schindler and H. Wang. Smooth foreground-background segmentation for video processing. In *Asian Conference on Computer Vision (2)*, pages 581–590, 2006.

[112] R. Schmidt. Multiple emitter location and signal parameter estimation. In *Proceedings RADC Spectrum Estimation Workshop*, pages 243–258, 1979.

[113] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Technical Report No. 14, Max Planck Institute for Biological Cybernetics, Tübingen, December 1996.

[114] B. J. Scholl. Objects and attention: The state of the art. *Cognition*, 80(1-2):1–46, 2001.

[115] S. M. Seitz and C. R. Dyer. View-invariant analysis of cyclic motion. *International Journal of Computer Vision*, 25(3):231–251, 1997.

[116] M. Seki, T. Wada, H. Fujiwara, and K. Sumi. Background subtraction based on cooccurrence of image variation. In *International Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, June 2003.

[117] C. E. Shannon. Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, 37(1):10–21, January 1949.

[118] T. F. Shipley. Spatiotemporal unit formation. *Behavioral and Brain Sciences*, 21(06):772–772, 2000.

[119] P. H. Siegel. Terahertz technology. *IEEE Transactions on Microwave and Millimeter Wave Techniques*, 50(3):910–928, March 2002.

[120] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of Optical Society of America, Series A*, 4(3):519–524, March 1987.

[121] D. Skočaj and A. Leonardis. Robust continuous subspace learning and recognition. In *International Electrotechnical and Computer Science Conference*, pages 275–278, Portorož, Slovenia, September 2002.

[122] S. Soatto, G. Doretto, and Y. N. Wu. Dynamic textures. In *International Conference on Computer Vision*, pages 439–446, Vancouver, Canada, July 2001.

[123] E. S. Spelke, G. Gutheil, and G. Van de Walle. The development of object perception. In D. N. Osherson and S. M. Kosslyn, editors, *Visual Cognition and Action: An Invitation to Cognitive Science (Volume 2) 2nd Edition*, pages 297–330. MIT Press, Cambridge, MA, 1995.

[124] L. St-Laurent, D. Prévost, and X. Maldague. Thermal imaging for enhanced foreground-background segmentation. In *Proceedings of the 8th conference on Quantitative InfraRed Thermography*, Padova, Italy, 2006.

[125] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *International Conference on Computer Vision and Pattern Recognition*, pages 2:246–252, Fort Collins, Colorado, June 1999.

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

[127] G. Strang. *Linear Algebra And Its Applications*. Thomson Learning Inc, Stamford, Connecticut, 1988.

[128] S. H. Strogatz. *Nonlinear Dynamics and Chaos*. Addison Wesley, Reading, MA, 1994.

[129] E. Swears, A. Hoogs, and A. G. A. Perera. Learning motion patterns in surveillance video using HMM clustering. In *IEEE Workshop on Motion and Video Computing*, Copper Mountain, CO, January 2008.

[130] M. Szummer and R. W. Picard. Temporal texture modeling. In *International Conference on Image Processing*, volume 3, pages 823–826, September 1996.

[131] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *International Conference on Computer Vision*, pages 255–261, Corfu, Greece, September 1999.

[132] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, January 1991.

[133] B. Valentine, S. Apewokin, L. M. Wills, S. Wills, and A. Gentile. Midground object detection in real world video scenes. In *Proceedings of the IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, London, September 2007.

[134] V. Vineet and P. J. Narayanan. CUDA Cuts: Fast graph cuts on the GPU. In *IEEE Workshop on Computer Vision on GPUs*, Anchorage, June 2008.

[135] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[136] L. Wang and D. C. He. Texture classification using texture spectrum. *Pattern Recognition*, 23(8):905–910, 1990.

[137] J. Weng, Y. Zhang, and W. Hwang. Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):1034–1040, August 2003.

[138] Y. Wexler and D. Simakov. Space-time scene manifolds. In *International Conference on Computer Vision*, pages 858–863, 2005.

[139] N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, Cambridge, MA, 1964.

[140] R. P. Wildes. A measure of motion salience for surveillance applications. In *International Conference on Image Processing (3)*, pages 183–187, 1998.

[141] C. Wren and F. Porikli. Waviz: Spectral similarity for object detection. Technical Report TR-2005-04, Mitsubishi Electric Research Laboratory, January 2005.

[142] C. R. 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(7):780–785, July 1997.

[143] F. Y. Wu. The Potts Model. *Reviews of Modern Physics*, 54(1):235–268, January 1982.

[144] T. Xiang and S. Gong. Beyond tracking: Modelling activity and understanding behaviour. *International Journal of Computer Vision*, 67(1):21–51, 2006.

[145] T. Xiang and S. Gong. Model selection for unsupervised learning of visual context. *International Journal of Computer Vision*, 69(2):181–201, 2006.

[146] 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, January 1995.

[147] Q. Yu and G. Medioni. A GPU-based implementation of motion detection from a moving platform. In *IEEE Workshop on Computer Vision on GPUs*, Anchorage, June 2008.

[148] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. *International Conference on Computer Vision and Pattern Recognition*, 2:819–826, 2004.

[149] C. S. Zhu, N. Y. Wu, and D. Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8), November 1997.

Appendix A

Calculation of Motion Vectors

Although the optical flow vector for an image does not generally directly represent the physical motion of objects in the scene, for the current purpose, use of the optical flow as a 'signature' for types of local motion is entirely reasonable. In a surveillance environment motion is one of the most important elements in detection of scene activity. However, the problem of deriving the optical flow field for an image sequence is in general ill-posed when images contain areas of constant intensity. The optical flow constraint equation

$$\nabla I.\mathbf{v}_p + \frac{\partial I}{\partial t} = 0 \tag{A.1}$$

where $\nabla I = [\frac{\partial I}{\partial x}\ \frac{\partial I}{\partial y}]^T$ is the spatial image gradient at point p, enables calculation of the motion vector $\mathbf{v}$ from the ratio of temporal gradient to spatial gradient in two dimensions. This computation is notoriously sensitive when the spatial gradient is small. The Lucas-Kanade algorithm solves a set of constraint equations for a local spatial and temporal volume, yielding a more stable solution. The equation

$$\mathbf{v} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{A.2}$$

$$\textit{where } \mathbf{A} = \begin{bmatrix} \frac{\partial I_1}{\partial x_1} & \frac{\partial I_1}{\partial y_1} \\ \frac{\partial I_2}{\partial x_2} & \frac{\partial I_2}{\partial y_2} \\ \vdots & \vdots \\ \frac{\partial I_{25}}{\partial x_{25}} & \frac{\partial I_{25}}{\partial y_{25}} \end{bmatrix} \textit{ and } \mathbf{b} = \begin{bmatrix} \frac{\partial I_1}{\partial t} \\ \frac{\partial I_2}{\partial t} \\ \vdots \\ \frac{\partial I_{25}}{\partial t} \end{bmatrix} \tag{A.3}$$

is the least squares solution to the over-constrained system of optical flow equations determined at, in this example, the 25 locations in a 5×5 pixel image block. A fast solution for Equation (A.2) may be formulated using integral images [135], such that the flow based on any arbitrary rectangle of pixels within an image may be arrived at in constant time. For the experiments in Chapter 5, the average flow vector for a whole vehicle was estimated directly from pixels within its bounding box using this method.

To further stabilize the calculation, a Gaussian filter of variance $\sigma^2 = 4$ is applied over a square 11×11 pixel block to each image before evaluating the motion vectors. Although sacrificing spatial resolution, this filter tends to locally turn sharp edges into gentle slopes, leading to more reliable motion estimates. Figure A.1 shows examples of the motion vectors from a typical surveillance sequence.

Figure A.1: Examples of the recovered motion vectors in which flow direction and magnitude for each pixel is derived from a 5×5 pixel block centred on it. Hue indicates direction whilst the intensity represents vector magnitude.