MOTION SEGMENTATION OF SEMANTIC OBJECTS IN VIDEO SEQUENCES

DAVID J.THIRDE

A thesis submitted in fulfilment of the requirements
of Kingston University for the degree of Doctor of Philosophy

March 2007

Abstract

The extraction of meaningful objects from video sequences is becoming increasingly important in many multimedia applications such as video compression or video post-production. The goal of this thesis is to review, evaluate and build upon the wealth of recent work on the problem of video object segmentation in the context of probabilistic techniques for generic video object segmentation.

Methods are suggested that solve this problem using formal probabilistic learning techniques, this allows principled justification of methods applied to the problem of segmenting video objects. By applying a simple, but effective, evaluation methodology the impact of all aspects of the video object segmentation process are quantitatively analysed.

This research focuses on the application of feature spaces and probabilistic models for video object segmentation are investigated. Subsequently, an efficient region-based approach to object segmentation is described along with an evaluation of mechanisms for updating such a representation. Finally, a hierarchical Bayesian framework is proposed to allow efficient implementation and comparison of combined region-level and object-level representational schemes.

Contents

List of Tables

List of Figures

Acknowledgements

The author wishes to acknowledge the following:

Eliza for her support and patience.

His family for their support and encouragement.

His principle supervisor Dr. Graeme A Jones for his guidance throughout the course of this work.

His supervisors Dr. Darrel Greenhill and Dr. Vincent Lau for their guidance and support. Dr. Julian Flack and his colleagues[1] for their guidance and support.

Dr. James Orwell for his encouragement.

Dr. Ahmed Shihab for his helpful discussion on clustering algorithms.

Dr. Paolo Remagnino for his guidance and support.

Dr. Paul Giaccone for his assistance with LaTeX and ground truth.

Etienne Corvee for helpful discussion on optical flow techniques.

Patrick Bas [2] for providing the *Bream* and *Children* sequences.

Çigdem Eroglu Erdem [3] for providing the *Parrot* sequence.

[1] DDD Group plc; `http://www.ddd.com`

[2] Laboratoire des Images & des Signaux, Grenoble , France; `http://www.lis.inpg.fr/pages_perso/bas/`

[3] `http://web.boun.edu.tr/erogluc/`

Chapter 1

Introduction

The extraction of meaningful objects from video sequences is becoming increasingly important in many multimedia applications such as video compression or video post-production. The type of objects to be extracted from a sequence is dependent on the application. The goal of this thesis is to review, evaluate and build upon the wealth of recent work on the problem of video object segmentation in the context of probabilistic techniques for motion segmentation of semantic video objects. In this chapter, Section 1.1 defines semantic video objects and Section 1.2 introduces the type of video sequence that the objects will be extracted from. Section 1.3 reviews the traditional approaches for extracting video objects previously used in the context of video post-production. Finally, Section 1.4 details the aims and objectives for this work.

1.1 Semantic Video Objects

A semantic video object is a visible entity within a video sequence that is meaningful in some way, the concept of such an object is dependent on the application. An object may have different significance for the human visual system than it would for video compression performance. Examples of extracted semantic video objects are shown in Figure 1.1. The objects of interest in the video sequence may or may not represent real world objects (i.e. those seen by the human visual system). In general, real world objects cannot be extracted using automated analysis of simple features such as colour or motion.

This work is developed in the context of video post-production where objects are defined in a video sequence as the collection of pixels that correspond to the projection of a

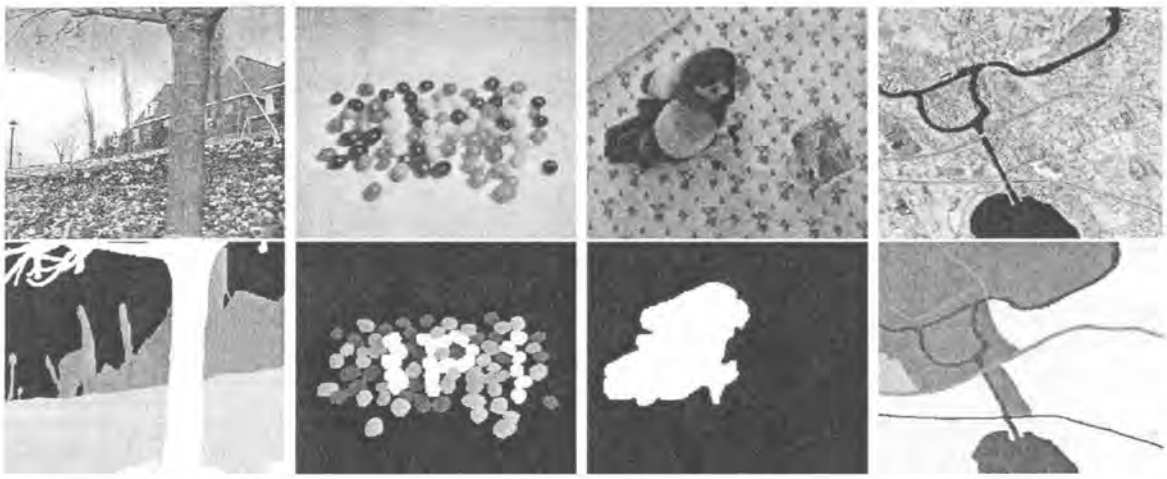


Figure 1.1: Examples of semantically extracted objects. (Far Left) shows objects extracted using motion as semantics for the 'Flower Garden' sequence. (Centre Left) shows objects extracted using colour for the 'Jelly Beans 2' scene. (Centre Right) shows the extracted foreground object using the human visual system for the 'Parrot' sequence. Finally, (Far Right) shows objects extracted using land use as semantics for the 'Aerial' scene.

real object into successive image planes of the video sequence [28]. This type of semantic definition describes objects that the human visual system *sees* within a scene and is demonstrated by the extracted foreground object for the 'Parrot' sequence in Figure 1.1. An object defined in this way may not exhibit any homogeneous properties and may only appear a separate object due to the prior information gained by experience of such objects. due to the complex nature of the human visual system, extracting this type of object is a non-trivial and challenging problem.

1.2 Generic Video Sequences

The properties exhibited by a video sequence can vary depending on the application. The term *generic* video sequence (or scene) is used to describe one where few parameters about the sequence can be assumed. The extraction of video objects from generic sequences therefore requires algorithms that operate with little prior knowledge about the sequence. A generic video sequence is defined as one with the properties shown in **bold** in Table 1.1, beyond this few assumptions are made about the content of the video sequence to be processed. This prior knowledge does not limit the algorithms proposed, indeed the same

algorithms can be applied, for example, to greyscale sequences with low signal-to-noise (SNR) ratio and compression artifacts. In this example a degradation in the quality of the extracted video objects would be expected.

Property	Examples
Arrangement of Sensors	**Monocular**, Binocular, Multiple with Overlapping FOV
Signal Type	Greyscale, **Colour**, IR
Frame Size	PAL, NTSC, CIF, QCIF, SIF
Length	Seconds, Minutes, Hours
Number of Channels	1, **3**
Sensor Motion	Stationary, Smooth, 'Jerky', Fast, Slow
Digital Capture Quality	High/Low Signal-to-Noise Ratio (SNR), Quantisation, **Cinematic**
Compression Quality	DV, MPEG, DivX, H.261/263, Cinepak, **Raw**
Subject	Natural, Man-Made, Augmented Reality, Computer Graphics, Near, Far, Large, Small

Table 1.1: Properties exhibited by video sequences. The properties defined for generic video sequences are listed in **bold**.

1.3 Video Object Extraction

The video objects to be extracted are an application dependent choice, information about a video object is often used in further processing or operator interaction. For example, an automatic vehicle registration plate recognition system can either extract an image of the number plate, a mask delimiting the registration plate or the vehicle registration number itself, depending on the next stage in the process. In the context of cinematic post production the object information required is a pixel-wise segmentation of the current scene into its constituent semantic video objects.

1.3.1 Traditional

In film and televisual industries there has long been the requirement to extract actors and props from one scene and place these into new composite sequences. In this section a brief

overview of traditional techniques for achieving this is given.

Chroma-keying

Chroma-keying is a well-established technique in video production. A large screen of uniform colour is placed behind the actors and props, a 'keying' unit built into the camera then removes any signal matching this colour from the scene, allowing the extracted objects to be superimposed on another video sequence to form a composite sequence. This approach has many drawbacks when used in film post production:

- It does not allow objects to be extracted without the chroma-key screen, hence cannot be used to extract objects from, for example, outdoor scenes.
- The screen must be well lit to ensure it has a uniform colour, this is difficult on large sets.
- It is difficult to ensure the screen is well lit whilst allowing artistic lighting on the objects within the scene. Scenes with low lighting levels pose particular problems.
- Objects may cast shadows onto the screen, changing the observed colour of the screen.
- The light from the brightly-lit brightly-coloured background can scatter, changing the hue of foreground objects giving an unnatural appearance

For real-time object extraction chroma-keying remains one of the fundamental techniques used, especially in televisual services. If real-time extraction is not required the limitations of chroma-keying can be overcome by performing manual extraction (or *rotoscoping*) of the video objects. This approach requires an operator to manually extract the video object by essentially drawing around the objects of interest. This is a time consuming and expensive approach, and has its own limitations due to the human interaction required. Rotoscoping is described in the following section.

Rotoscoping

Rotoscoping is the act of extracting objects from video sequences using human operators and computer based tools. Such tools (e.g. deformable geometric shape templates, "onion-skinning") are often implemented to improve the efficiency of human operators at the expense of final segmentation quality, even with such tools this process is operator intensive

THE FOLLOWING HAVE BEEN REDACTED AT THE REQUEST OF THE UNIVERSITY:

p16, figure 2.1

p32, figure 2.7

and can require a team of operators to convert a sequence. The problem with team-based manual video object segmentation approaches is a noticeable temporal incoherence (known as *bubbling*) as operators cannot define complex object boundaries efficiently and consistantly in successive frames. An example of the problem of balancing efficiency and accuracy is shown in Figure 1.2, where accurate manual segmentation of the tree object is compromised in favour of more efficient operator input. The figure on the right was generated using a tool that required operator supervision in the form of training points (i.e. image locations assigned object labels). This label was subsequently propagated to the surrounding pixels using a nearest-neighbour based classifier using spatial and colour features.

Figure 1.2: Manual extraction of video objects — efficiency vs. accuracy. (Left) A region of the original scene. (Centre) Manually extracted objects using Bezier curve based tools to approximate the boundary of the tree object. (Right) Extracted objects using supervised learning tool. (c)2004 DDD Group Plc

1.3.2 Computer Vision

With the predominant use of computers in the post-production industry, techniques for the extraction of semantic video objects can start to move away from time consuming, limited traditional approaches and instead utilise computer vision algorithms that can in many cases be used on a desktop computer. The extraction of video objects in the computer vision literature is often termed *Video Object Segmentation* and it is this terminology that will be adopted for the remainder of this thesis. The advantages of using computer vision methods to extract video objects are:

- Objects can be extracted from any video sequence.
- Reduces operator input by learning sparse supervised exemplars.
- In some sequences the objects can be extracted automatically (e.g. motion based).

- Accurate object boundaries can be located with repeated accuracy, leading to temporal coherence of extracted objects.

When inserting extracted objects into other sequences further processing is required to correct the appearance of the object to match the lighting conditions in the destination sequence. In this work the application of further processing stages is not considered and only the extraction of the objects from generic video sequences. While the work on computer vision based video object segmentation can be considered to be in its infancy; it is common sense to consider that in the longer term (with increased computational power and many years of research) there may well exist simple and easy to use programs that allow the accurate 'cutting and pasting' of objects between different video sequences.

1.4 Aims and Objectives

The principle aim of this thesis is to evaluate and propose methods to segment semantic video objects from generic video sequences. The work presented in this thesis aims to overcome problems associated with many existing approaches to video object extraction by applying formal methods in a well defined framework. The work is developed in the context of a supervised process, such that the objects are specified at key-frames by skilled operators. An important aspect of this work is to accurately and efficiently propagate these specified objects throughout the video sequence.

Figure 1.3 shows the input and output of the video object segmentation algorithm. The input is a video sequence and the output is a set of segmentation masks that delimit the semantic video objects to pixelwise precision. In this thesis methods are suggested that allow the solution of this problem using formal probabilistic representation techniques, this allows principled justification of methods applied to the problem of segmenting video objects. By applying a simple, but effective, evaluation methodology the impact of all aspects of the video object segmentation process can be analysed.

As part of this process the feature space is defined within which the video object distribution is modelled using probabilistic methods. An efficient region based approach to video object segmentation is subsequently suggested along with an evaluation of mechanisms for updating such a representation. Finally, a hierarchical framework is proposed to allow the region based approach to be defined within the local co-ordinate system of

the parent object; this framework raises many interesting areas of future research for video object segmentation work.

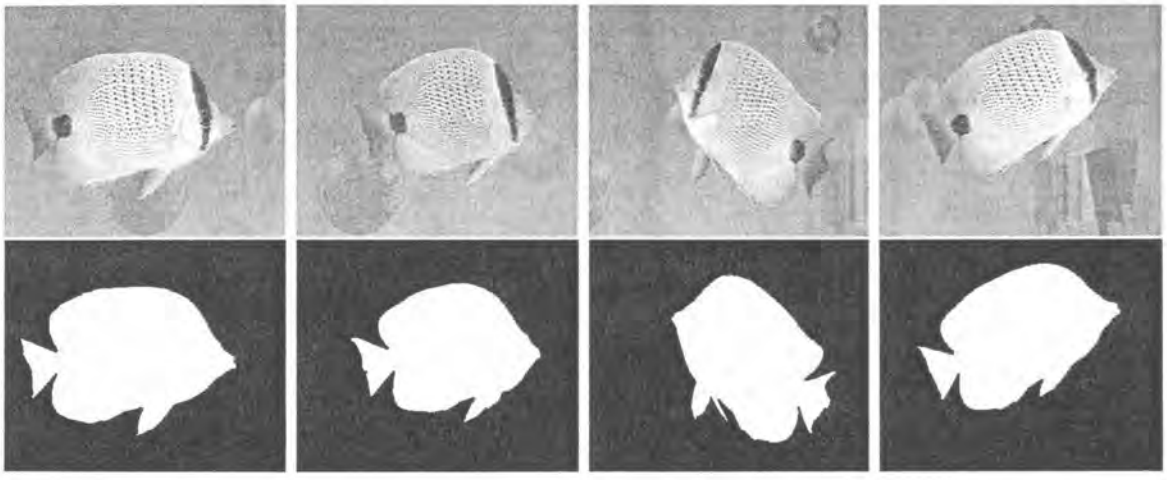


Figure 1.3: The input and output of the video object segmentation algorithm. (Top Row) The input to the system is a video sequence that is processed by the video object segmentation algorithm to output (Bottom Row) a set of segmentation masks for the semantic video objects in the scene.

Chapter 2 reviews the background material for video object segmentation using probabilistic methods. Chapter 3 evaluates a selection of popular features for video object segmentation; these are evaluated using a principled methodology and supervised image based segmentation. Chapter 4 introduces probabilistic representative models for video object segmentation and evaluates the performance in terms of the quality of the extracted segmentation mask. Chapter 5 improves the computational efficiency of object based segmentation schemes by modelling at the localised region based level. A selection of mechanisms for updating the region based representation are presented and evaluated over a range of test sequences. Chapter 6 introduces the hierachical framework, methods are suggested to fulfil the requirements of this framework. Finally, Chapter 7 reviews the thesis and draws conclusions and future directions for this work.

Chapter 2

Video Object Segmentation

This chapter gives background information that is useful for the remainder of the thesis. Detail is given about the application of computer vision methods to the problem of extracting video objects. The application of classifiers to learn the interaction of a human is first discussed, followed by a review of methods by which the object-based representation can be updated during the video sequence.

An overview of a generalised video object segmentation algorithm is presented in Section 2.1. Section 2.2 reviews the form of user input to the segmentation process. Section 2.3 introduces existing techniques for video object segmentation organised into three broad taxanomic categories. Section 2.4 describes the feature spaces that can be used to describe objects within a video sequence and Section 2.5 introduces the representational schemes that can be used to model the video object in the feature space. Section 2.6 follows this by giving an overview of the evolution strategies for the representational model and Section 2.7 reviews existing methods for evaluating the performance of a video object segmentation system. Finally, Section 2.8 reviews the state of the art methods for video object segmentation.

2.1 The Basics

Video object segmentation refers to a process that takes as input a raw video stream and outputs segmentation masks that delimit the (semantic) objects within the scene. The video can be processed using batch or sequential algorithms — batch algorithms process a video volume (consisting of multiple frames) whereas sequential methods process on a per-frame basis with either forward or backwards propagation of the current frame into the next. A

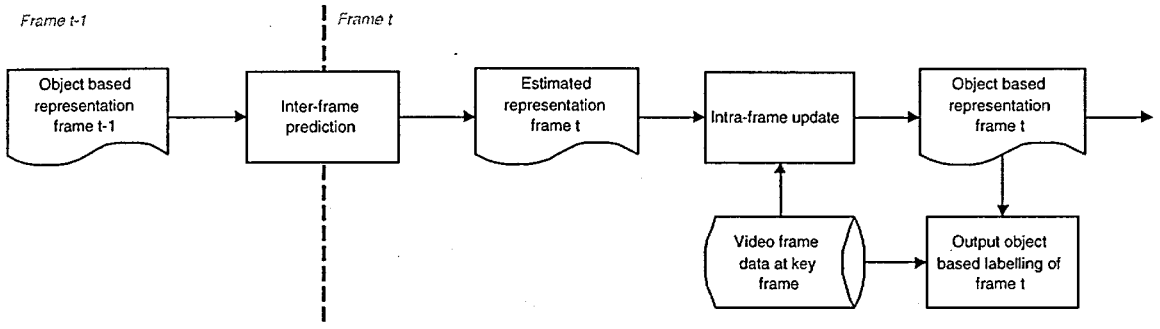


Figure 2.1: Generalised framework showing the per-frame update process of a video object segmentation algorithm.

review of video object/region segmentation methods is given by [151, 185].

In this section video object segmentation is introduced using a sequential framework with forward propagation of video objects. This is perhaps the most common type of framework and is easily modified to perform backwards propagation. Batch based processing schemes are not covered in this thesis due to the large data storage requirements of the video and the limited scalability of the approach.

A generalised sequential framework for feed forward video object segmentation at a frame t in a video sequence is presented in Figure 2.1. There are four main components that form this framework — feature space extraction from the video frame data, video object representation (i.e. the per-object model), inter-frame prediction and intra-frame update of the representational model. The feature space extraction is the step that takes the raw video frame data (for the current frame) and converts this into a feature space within which the video objects can be delimited. The video object representation is the model of the objects of interest that are tracks in the scene. The inter-frame prediction stage updates the representational model between frames. The intra-frame update stage corrects the prediction with the newly observed video frame data. The model update scheme requires mechanisms to give robustness to object interactions and the ability to innovate new objects or parts of objects to maintain the representativeness of the model for the duration of the video sequence. Objects or parts of objects that are no longer supported by the observed video data are terminated as part of this process.

The input to the system (on the left) is the output from the previous time step, that is, the object representation at frame $t-1$. This result is modified between the frames

(the inter-frame prediction step) to give the predicted object representation for the current frame. Using the current frame video data (which may be pre-processed) an intra-frame update stage is used to update the representative model for the video objects. This model subsequently becomes the output from frame t, along with the (final) segmentation mask of objects at that frame. Methods for representing the video objects are discussed in Section 2.3 and Section 2.5. Techniques for updating the object representation are presented in Section 2.6. Many of the existing methods for video object segmentation can be fitted into this framework. For some algorithms there are trivial steps in this process e.g. [98, 135, 61] do not have an inter-frame prediction scheme and simply use the previous frame result in the current frame.

The video can be pre-processed before application in the system to extract a multi-dimensional feature space within which the video objects can be distinguished. This extraction process can convert the colour space in a linear or non-linear way, measure texture-like features, extract gradients, edges, corners[1] , measure motion over adjacent frames etc. The conversion of the raw data into such features is often neccessary to allow enhanced analysis of the video sequence. The popular features derived from video sequence data are discussed in Section 2.4.

If the video objects of interest can be defined *a priori* using the features derived from the video data then an 'automated' process can be applied to extract them (although, as discussed in Section 2.2, these methods involve implicit supervision). For semantic objects in generic sequences there is no such automated extraction technique, therefore user interaction is required to delimit the object at frames of interest (termed key-frames) throughout the video sequence. The framework for the user interaction to such a process is shown in Figure 2.2. The operator interaction is used to drive (or constrain) the building of the object-based representation at the keyframe. This representation is the output from the supervision stage and subsequently forms the input to the next frame as shown in Figure 2.1. The operator supervision stage is discussed in Section 2.2.

[1]The term *corner* is traditionally used to represent feature points of interest extracted using the method of Harris [86].

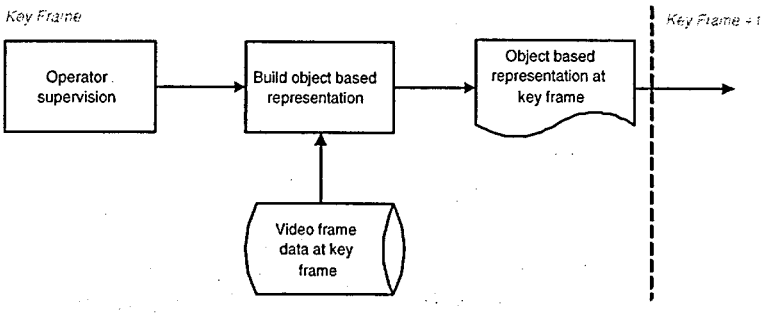


Figure 2.2: Generic framework showing the initialisation of a video object segmentation algorithm.

2.2 Operator Supervision

The type of objects to be extracted delimit real world objects captured in the video sequence. These types of objects to be extracted may not exhibit any homogeneous properties and may only appear a separate object due to the prior information gained by experience of such objects. This definition of a semantic object is ill-posed mathematically, hence there is no 'automatic' scheme for extracting the objects from sequences. Indeed, the concept of an 'automatic' scheme does not exist, there are always parameters that define how the algorithm performs and these can be thought of as a form of user supervision; the parameters are located deep in the program structure and therefore are generally not intuitive to the end user [26].

If the object segmentation scheme is *unsupervised* in this manner then arbitrarily labelled regions are found by clustering the data to find natural groupings in the feature space. Natural clustering of data is commonly applied in motion based segmentation schemes (e.g. [98, 66, 128]), where regions are grouped based on motion homogeneity (often appended by, for example, spatial and colour information). The types of 'objects' extracted by such approaches may correspond to homogeneous *regions* within the video sequence.

To allow real world semantic objects to be extracted many approaches to video object segmentation allow user supervision at *key-frames* throughout the video sequence [151]. This type of supervision allows the extraction of meaningful objects from video sequences and is common in video analysis techniques (e.g. [130, 186, 111, 61, 58, 83, 27, 88]); it also reduces the computational load by allowing the user to designate the semantically meaningful observations [145, 30]. The key-frame definition is analogous to the I-Frame

and P-Frames defined by ISO/MPEG-2 and applied to video analysis work (e.g. [83, 46]) where I-Frames are essentially key-frames and P-Frames are forward predictions from the I-Frame.

The key-frames can be chosen *a priori* by a human operator or using a temporal video segmentation algorithm that is capable of classifying shot types and hence extracting key-frames (e.g. Porter *et al* [140]). An example of key-frame extraction is shown in Figure 2.3, these were selected manually by a human operator. Manually extracted key-frames are generally frames selected to be representative of the interesting (i.e. semantic) content within a video sequence. The selected frame may be situated at a shot break (e.g. a camera change) or may contain a close up of one of the semantic objects within the scene.

Figure 2.3: Examples of key-frames extracted from a test sequence. (Bottom Row) The extracted key-frames (Left-Right, frames 00000,00060 and 00100) from the 'Table Tennis' sequence (Top Row, resolution 176×120, length 150 frames). The key-frames are chosen to be representative of the sequence, including the shot break that occurs $\sim$100 frames into the sequence.

The user supervision at the key-frames is generally used to perform one of two actions:

- Top-down constraint on the location and quantity of video objects, from which per object bottom-up models are generated.
- Labelling of bottom-up generated segments with object labels.

The top-down constraint method generally requires the user to create a dense (i.e. per pixel) segmentation map to delimit the object in the key-frame. The segmentation map usually takes one of three forms — a precise label map, a *trimap* or a *probabilistic* map.

Examples of such maps are demonstrated in Figure 2.4 for two objects - object one (white pixels) or object two (black pixels). The precise label map and trimap are discretised representations of the probabilistic map i.e. the map of our belief in the object location. The precise label map is an *optimistic* discrete belief map where each pixel only belongs to one object. The trimap introduces a region where the pixels are not given a membership so that only *trusted* pixels are used to generate the object representations. The probabilistic map represents, with a negligible degree of quantisation, the belief of the memberships of each object at each pixel. In such a map there is no provision for multiple objects per pixel, therefore a series of probability maps would be required for multiple object scenarios. Alternatively, a higher level top-down constraint could be applied where only the quantity of video objects is determined *a priori* by the human operator [134].

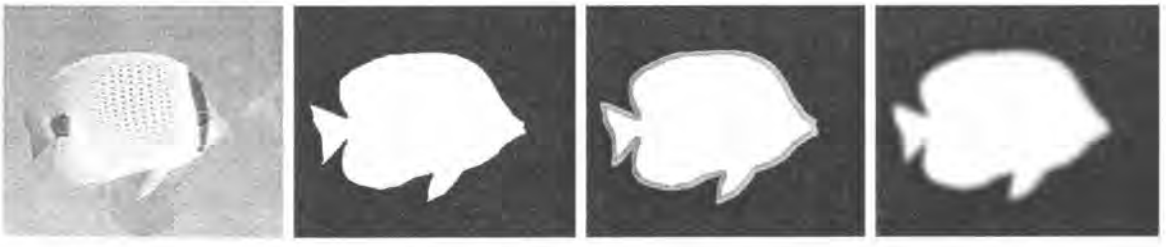


Figure 2.4: Three examples of segmentation maps. (Left) 'Bream' sequence, Frame 00000, containing two distinct objects (Centre Left) shows the precise label map (Centre Right) shows the trimap and (Far Right) shows the probabilistic map.

The labelling of bottom-up generated segments with object labels is a form of user input that is very efficient since it only requires sparse supervised samples to be located on the key-frame. A simple approach to this form of supervision is to label each region with the object label denoted by operator *scribbles* that fall within it (e.g. [91]). The objects are then found by taking the union of all regions that were scribbled by the operator and applying the appropriate labels, an example of this is shown in Figure 2.5. An alternative to this is to allow the user to define a coarse segmentation mask [95] (comprised of blocks) or approximate contour [27] from which the membership of the bottom up generated segments can be determined using set theory methodologies. The weakness of this approach is that the regions produced by the bottom-up process must contain at most one of the scene objects; if they do not then there will be contention with more than one label within a single region, further supervision is required to split these regions.

Figure 2.5: Examples of labelling bottom-up generated segments with object labels. (Left) 'Children' sequence Frame 00000 (operator input is shown in white). (Centre) shows the regions generated using the bottom-up segmentation algorithm (Right) shows the extracted objects formed from the union of the regions that were scribbled by the operator.

After performing a suitable supervision process a set of suitably accurate masks are generated for each key-frame. These masks delimit the video objects that are to be extracted from all the frames in the video sequence. For the remainder of the thesis the presented test video sequences start with a key-frame and contain all the frames upto, but not including, the subsequent key-frame in the sequence. The remainder of this chapter focusses on reviewing the state of the art techniques for extracting objects from video sequences.

2.3 Methods for Video Object Segmentation

Methods for video object segmentation are generally categorised by the technique used to represent and hence extract the video object. In this sense the segmentation methods can be divided primarily into region-based or boundary-based methods. These two distinct approaches attempt to locate an object based on the homogeneity of feature vector regions or by measuring gradient information in the feature space to locate object boundaries. Other divisions of the techniques divide the methods based on a mixture of the representational scheme and the feature space used; e.g. grouping object segmentation methods into three classes [121] — region based methods using homogeneous colour criterion, object-based approaches utilising homogeneous motion criterion and object tracking (or object-based homogeneous colour and motion criterion [6, 8]). As stated in the previous section, the video object segmnetation algorithm has many component factors and therefore taxanomic separation of such methods is only possible using marginal aspects of each algorithm. Three

broad taxanomic categories are defined within which various state of the art methods are described. The three categories are: *Morphological Operators*, *Image Plane Operators* and *Feature Space Classifiers*, these definitions and associated algorithms are discussed in the following sections.

2.3.1 Morphological Operators

Morphological operators are tools to extract image components that are useful for describing the regional structure in an image. Examples of these are boundaries, skeletons and convex hulls [81]. Morphological filtering can also be applied to pre- and post-process images e.g. pruning and thinning. In the context of video object segmentation there is generally only one such morphological method that is readily applied, this is the well known *watershed* technique.

Watersheds [169] are morphological processes commonly applied to the problem of image and video segmentation. To find the watershed segmentation a (smoothed) gradient image is first found, the analogy is that this gradient image consists of valleys and mountains. By placing a *marker* (or *seed*) point in the *flat-zones* of the valleys the watershed algorithm recreates the immersion process of flooding the valleys with water — where water from adjacent valleys meet a *dam* is placed, regions are then located by finding the areas enclosed by the dams in the image. The watershed algorithm generally oversegments images due to the abundance of local minima found by the gradient operators in generic sequences and is sensitive to image noise due to the lack of explicit noise modelling. The watershed algorithm also lacks a global analysis of the image which can lead to regions that have little semantic meaning; in areas of low image gradient the dams can be placed in arbitrary locations due to the requirement of localised 'flooding' by the algorithm. The watershed algorithm (and variants) has been applied to video object segmentation both spatially [134, 52, 183, 83, 110, 112, 33, 152, 76] or spatio-temporally [150].

2.3.2 Image Plane Operators

Image plane operators are determined to be the class of algorithms that are not morphological operators or feature space classifiers (introduced in the following section). In general the distinction between these methods lies in the methodology from which they were devised; methods that apply rule based approaches, region growing like methods or heuristics are

deemed to be the former whereas methods with a strong leaning to 'traditional' classification and feature spaces are chosen to be the latter. In essence, the image plane operators are those approaches that operate on (often raw) image data with the notion of pixels intact.

Region growing algorithms are a class of algorithms that form region segmentations of images by measuring similarities in the between some property of the region and adjacent pixels (the search step). The similarities measured as the region grows can be classified into three categories [28]:

- *Single-linkage.* Adjacent pixel similarity matching in search step — sensitive to noise.
- *Hybrid-linkage.* Neighbourhood level similarity matching in growing of regions
- *Centroid-linkage.* Region level statistics used in search step

The initial points for region growing are usually placed in the most homogeneous image regions, when all regions are found a *Region Merging* step (generally based on region statistics e.g. [126]) is often applied to reduce the total number of regions to a reasonable limit. Like the watershed algorithm, region growing algorithms generally suffer from a lack of global analysis and can be sensitive to noise in the image. The position of the initial region seeds can also greatly affect the location of the final image regions, region growing algorithms are commonly applied as sub-processes in video object segmentation schemes [132, 138, 187].

Split and Merge based algorithms (of which region merging is a specific case) attempt to find meaningful regions in an image by merging or splitting an initial set of regions based on some measure of similarity. A region merging algorithm generally starts with an *oversegmentation* of the current frame which is then proggresively merged based on similarity between regions — it can be seen that region growing can be thought of as a set of initial regions (seeds) that are then progressively merged with pixel sized regions. In general splitting and merging are both applied until some homogeneity criterion is met for the regions in the current frame. Region merging methods have been applied to spatiotemporal segmentation [124, 63] and for rule based merging of similar regions [138, 19, 149, 112]. Split and merge approaches have been applied to video object segmentation by [168].

Boundary based methods in the image plane attempt to find objects by locating the border pixels between adjacent pixels. Active contour models [122] (or *snakes*) allow a user

to specify an initial estimate of the location of the boundary of the object to be segmented; this boundary is subsequently refined to fit the local maxima in the gradient information. The effect of noise and quantisation errors generally make boundary based approaches an ill conditioned problem [28]. An extension to active contour models are *active surface* models [84] which can be used to extract the surface of the volume within which a video object lies. The *region competition* algorithm [162] presents a unifying algorithm for both active contours and region growing algorithms, and showed that these are both derived cases of the region competition approach.

If the camera is stationary a background model for edge pixels can be generated, this allows foreground edge pixels to be located; the objects are subsequently determined by looking for continuous sequences of horizontal and vertical pixels. Such approaches were shown in [119, 100, 99], although it remains unclear whether global warping could be successfully applied in the case of a moving background. Other boundary based approaches [58, 83, 120, 121] parameterise the boundary (e.g. as a polygon [172]) and use motion information to warp either boundary segments or the whole boundary between frames. The boundaries are subsequently corrected using colour and motion information [58] or a colour watershed algorithm [83] in a trimap-like uncertain zone around the warped boundary. In both these approaches the boundary correction results were only presented for relatively convex objects and may fail for non-convex shapes (where the search zones cannot be determined as clearly).

2.3.3 Feature Space Classifiers

Many researchers have turned the problem of object-based video analysis into that of *classification*; the problem of classification is more general than that of video segmentation and can be applied to many related research areas for example, signal processing, object recognition etc. An overview of the well known methods used in statistical pattern recognition is given by Jain *et al* [4].

Generally, the goal of classification is to generate efficient algorithms that have high accuracy when classifying previously unseen data; the classifiers may be supervised or unsupervised depending on the application. In the context of video object segmentation the application of classifiers requires specification of the feature space (i.e. the signal to be classified), the classifier (of which there are countless types) and the decision rule. The

decision function used is often closely tied to the classifier, although the basic ideas behind the decision functions are generally similar.

Classifiers can be combined into Mixtures-of-Experts [51] where several so-called experts are combined to form an overall classifier that out performs each of the individual component classifiers for a given test set. The classifiers themselves can be grouped into boundary and cluster (i.e. region) based approaches. The output of the classifier can also be grouped into deterministic (e.g. K-Means clustering [51]) and probabilistic (e.g. Gaussian mixtures models [51]) methods. A deterministic output results in a discrete partition of the feature space into the constituent classes. A probabilistic output takes into account the uncertainty of the classification, resulting in a 'soft' or 'fuzzy' partition of the feature space into the constituent classes.

Classification techniques for modelling video objects require three components to be defined — feature space extraction, video object representation and representational scheme update. In the following sections relevant background information and existing methods are reviewed for these components. Section 2.4 introduces a selection of feature spaces that are popular in video object segmentation. Section 2.5 reviews classification techniques (including decision rules) that can be used to represent the video objects within the specified feature space. Finally, Section 2.6 discusses existing methods that allow the representational models of video objects to be adapted to the evolving video sequence.

The work presented in this thesis performs video object segmentation by applying probabilistic methodologies; such techniques present a principled and formal solution to the problem of representing and extracting video objects from sequence.

2.4 Feature Extraction from Video Sequences

The multi-dimensional space in which classifiers are applied is known as the feature space. The choice of this feature space is an important consideration since the distribution of the feature vectors for each object should be discriminable within the space. All pattern recognition techniques will fail if there is not enough discriminatory evidence to separate such entities. The feature spaces commonly applied for video object segmentation use a combination of colour, texture, gradient and motion information to delimit the objects in the image plane. Implicit to this is a defined (linear or non-linear) mapping between the

raw image plane space (spatial and colour co-ordinates) and the feature space. The mapped pixels are represented as multi-dimensional feature vectors in the feature space. Within the feature space the concept of distance is required to allow the proximity of feature vectors to be measured, commonly applied distance metrics include the Manhattan, Euclidean and Mahalanobis distances [51].

Perhaps the simplest feature space applied in video sequence analysis is luminance (grey-scale) information. This information provides an analytically simple feature space, although this information is sometimes not sufficient to provide even the human visual system with enough information to locate objects within the scene. Luminance information is generally used when computational power or available data storage is limited.

The analysis of the colour distribution of a video object remains one of the fundamental ways to accurately delimit a video object from other objects within the scene. Colour has many desirable properties for video object segmentation, for example colour cues show robustness under partial occlusion, rotation in depth, scale changes and resolution changes [161]. The RGB colour space has been used in many image processing and video analysis applications, this space is split into three quantised channels - red, green and blue. The effect of the luminance on RGB measurements can be reduced by normalising the per channel signal with the combined signal strength over all the channels [130, 11].

Other video object segmentation approaches have applied the YUV colour space to analyse video sequence information, the YUV colour space has a similarity with the human visual system in that it separates the luminance and chrominance information into separate channels and it can be derived using a linear transformation from the RGB colour space. The main advantage of this space is that it allows the luminance (Y) and chrominance (UV) information to be used separately. A related colour space to YUV is YIQ, in this space U and V are re-aligned to match human perceptual color sensitivities. YUV, YIQ and the related space $YCbCr$ are used in broadcasting and digital media standards — YUV is found in the PAL, NTSC and SECAM broadcasting systems, YIQ is optional for the NTSC broadcasting system and $YCbCr$ is found in the JPEG digital image standard.

To give a more intuitive description of colour and add robustness to light changes within the scene the RGB space can be transformed non-linearly to the HS-family of colour spaces. This transform splits the luminance and chromaticity information allowing the modelling of a chromatic signal without the influence of brightness (as with YUV), this gives algorithms

raw image plane space (spatial and colour co-ordinates) and the feature space. The mapped pixels are represented as multi-dimensional feature vectors in the feature space. Within the feature space the concept of distance is required to allow the proximity of feature vectors to be measured, commonly applied distance metrics include the Manhattan, Euclidean and Mahalanobis distances [51].

Perhaps the simplest feature space applied in video sequence analysis is luminance (grey-scale) information. This information provides an analytically simple feature space, although this information is sometimes not sufficient to provide even the human visual system with enough information to locate objects within the scene. Luminance information is generally used when computational power or available data storage is limited.

The analysis of the colour distribution of a video object remains one of the fundamental ways to accurately delimit a video object from other objects within the scene. Colour has many desirable properties for video object segmentation, for example colour cues show robustness under partial occlusion, rotation in depth, scale changes and resolution changes [161]. The RGB colour space has been used in many image processing and video analysis applications, this space is split into three quantised channels - red, green and blue. The effect of the luminance on RGB measurements can be reduced by normalising the per channel signal with the combined signal strength over all the channels [130, 11].

Other video object segmentation approaches have applied the YUV colour space to analyse video sequence information, the YUV colour space has a similarity with the human visual system in that it separates the luminance and chrominance information into separate channels and it can be derived using a linear transformation from the RGB colour space. The main advantage of this space is that it allows the luminance (Y) and chrominance (UV) information to be used separately. A related colour space to YUV is YIQ, in this space U and V are re-aligned to match human perceptual color sensitivities. YUV, YIQ and the related space $YCbCr$ are used in broadcasting and digital media standards — YUV is found in the PAL, NTSC and SECAM broadcasting systems, YIQ is optional for the NTSC broadcasting system and $YCbCr$ is found in the JPEG digital image standard.

To give a more intuitive description of colour and add robustness to light changes within the scene the RGB space can be transformed non-linearly to the HS-family of colour spaces. This transform splits the luminance and chromaticity information allowing the modelling of a chromatic signal without the influence of brightness (as with YUV), this gives algorithms

based on HS (i.e. Hue and Saturation) colour signals a limited amount of robustness to appearance changes of an object.

The YUV, HSI and RGB colour spaces are non-uniform with respect to the human visual system, so that measured distances in the space may not always relate to the perceived colour difference by a human observer.

A family of uniform colour spaces that are popular in computer vision applications are those derived from the linear CIE-XYZ tristimulus values [180]. These colour spaces are modelled to have perceptual uniformity, this has the desired property that, in theory, a distance measured between two colours within this space will be equivalent to the difference perceived between the two colours. The CIE-$L*a*b*$ colour space is a non-linear mapping of the CIE-XYZ tristimulus values, having separated luminance and chrominance information.

In many applications there is a requirement for modelling of the spatial distribution of objects e.g. to aid the segmentation of multiple objects that exhibit chromatic homogeneity and spatial inhomogeinity. The simplest form of spatial modelling is to apply connectivity constraints to the segmentation labelling process, and example of this can be seen in Figure 2.6, an extension of this is to model the segmentation as a Markov random field and impose spatial connectivity via neighbourhood based probabilisitic label analysis.

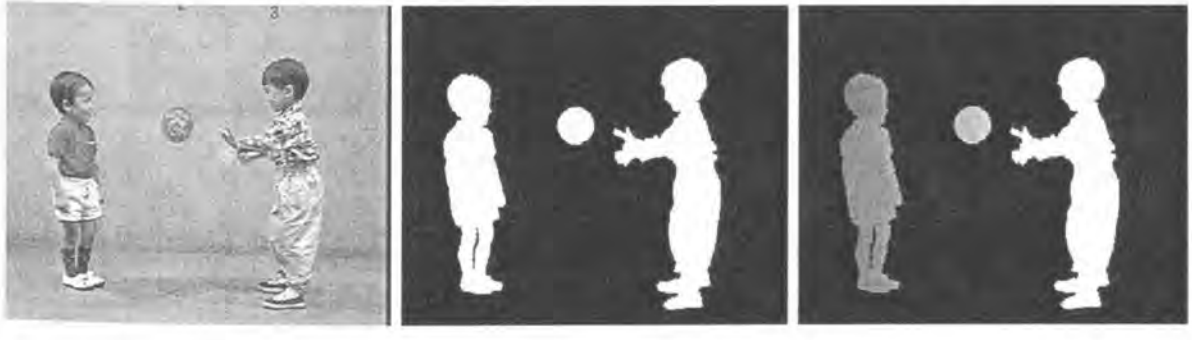


Figure 2.6: The effect of connectivity constraints on a foreground segmentation mask. (Left) Frame 00014 of the 'Children' sequence (Middle) object labelling result using only colour based segmentation (black is background, white is foreground) and (Right) modified labelling incorporating connectivity constraints (the foreground object is now split into three distinct objects, denoted by their grey-level).

Beyond the labelling constraints (to impose implicit spatial modelling), the use of spatial information in the feature space can be useful to define explicit structure. The integration of chromatic and spatial information often involves heuristic feature weighting to combine

THE FOLLOWING HAVE BEEN REDACTED AT THE REQUEST OF THE UNIVERSITY:

p16, figure 2.1

p32, figure 2.7

the features in a hybrid space [18, 154]. Spatial information can be incorporated at a higher level by using geometric or frequency domain models of objects to extract representative measurements e.g. region based fourier descriptors and bounding box [46].

Video objects of differing appearance can exhibit similar chromatic signals. In such cases textural information can be added to the feature space so that objects can be discriminated on the characteristics of the way the colour is distributed on the object. Generic methods for extracting texture do not exist, as the definition of texture is very much application dependent, a major problem is that the textures in the real world are often not uniform, due to changes in orientation, scale or other visual appearance. Figure 2.7 shows the problems of textural analysis of scenes — in the examples shown, the concept of texture is different and therefore prior decisions need to be made on which characteristics the texture is expected to exhibit in the scene. In many cases the measure of texturedness represents the variation of the chromatic signal in a finite neighbourhood.

Figure 2.7: Examples of texture encountered in image scenes. (Images courtesy (Left) University of Oulo Machine Vision Group, (Middle) the British Broadcasting Corporation and (Right) Corel)

Therefore, the simplest form of texture analysis is to measure the colour signal variance in pixel neighbourhoods. This provides a single dimensional feature space representing the energy of the image at each pixel. The windowed second moment matrix [75] extends this principle with descriptors of texture using edge/bar polarity based scale selection. Gabor filters [69] decompose images into multiple orientated spatial frequency maps from which amplitude or phase analysis can be used to form the feature maps.

Motion or temporal information about the video objects can also be introduced into

the feature space. According to the Gestalt "law of common fate" (an overview of Gestalt theory is given by Forsyth and Ponce [69], Chapter 14), meaningful regions are obtained if they are defined on the basis of temporal coherence. As objects move in the observed scene relative to camera motion there is a 2D vector field of velocity induced at each point in the image plane. This 2D vector field is termed the *motion field.* The goal of *optical flow* algorithms is to estimate the motion field by analysing the spatial and temporal variations of the image intensity within a sequence of images. This is achieved by modelling the link between the intensity and velocity to find the fundamental *image intensity constancy equation*[2].

A major drawback when utilising optical flow for video object segmentation is that, due to the neighbourhood based measurement of the flow and motion discontinuity, the object boundaries are not located accurately using only optical flow. It can also be unreliable in precense of occlusion and camera zooming, especially for non-rigid objects [91]; areas of homogeneous intensity in the image will also contain meaningless motion information. It is outside of the scope of this thesis to provide a comprehensive review of optical flow recovery techniques — well known algorithms for the recovery of optical flow include Lucas and Kanade [23], Horn and Schunk [89], Fleet and Jepson and Jenkin [68] and Black and Anandan [22]. An evaluation of several optical flow recovery techniques is given by Barron, Fleet and Beauchemin [17] and a treatment of motion models and motion field recovery techniques is given by Stiller and Konrad [160]. Barron, Fleet and Beauchemin arrived at the conclusion that the differential technique of Lucas and Kanade and the phase based approach of Fleet, Jepson and Jenkin give the most reliable optical flow computation. An alternative approach to extracting dense motion fields is to apply a block matching strategy (e.g. [171]) where pixels within a matched block are given the same motion label. Hierarchical block matching [20] improves the efficiency of a full search approach and the resulting motion field estimate can be more robust compared to optical flow when the scene has a high noise level [95].

A parametric model for 2-D motion fields can be derived from parametric models describing 3-D motion, 3-D surface function and camera projection geometry. The simplest motion model is a translational model, which displays feature space homogeinity only under similar object translations. The most commonly used motion model in video analysis work

[2]In many works (e.g. Trucco and Verri [164]) this is termed the image *brightness* constancy equation.

is an affine motion model [155], this model exhibits feature space homogeneity for object regions undergoing similar translation, rotation and zoom in scale. Motion models are often applied to generating a layered representation of video frames (e.g. [173]).

Some recent approaches to video analysis have used the concept of discrete time (i.e. the frame index) in the feature space [45, 150]. The use of temporal information is, at the time of writing, limited by the large computation and storage demands of multi-dimensional video processing. Fast moving objects within the scene may also exhibit discontinuities in the 3-D neighbourhood and there is little literature relating to how this may affect the final object segmentation result.

The choice of feature space can be treated as a selection problem where the most efficient and discriminating feature space is sought from a much greater pool of feature spaces. The selection of the 'good' features can be approached by either optimising the segmentation performance of the feature space over a range of test data (e.g. Guo *et al* [95, 96]) or use a learning algorithm to select a small number of critical features that best describe the objects of interest (e.g. Viola and Jones [170]). Viola and Jones presented the application of the Adaboost learning algorithm [71] to select 'good' image-based features from a much greater pool (i.e. image features that best discriminate a set of learning examples. A common problem with multiple feature space based approaches is that the multiple feature space transformations and analysis can make them computationally prohibitive and that the resulting (hybrid) feature space may contain illogical (and inefficient) combinations of features that represent the same visual characteristic.

Chapter 3 evaluates a selection of popular feature spaces that can be applied to the problem of video object segmentation.

2.5 Classification for Computer Vision

The learning of patterns and subsequent classification of unseen data is a cornerstone of computer vision research. In this work classification (also *pattern recognition*, *machine learning*) is defined as the process by which an algorithm can learn patterns from a training data set by modelling the data in a representive manner. These models can then be applied to classify previously unseen data and feedback this result to update the stored model representation. In the literature there is a great volume of work on the problem of estimating

(using a mathematical model) the underlying density function that describes the patterns of interest. The functional form used to represent the density varies from model to model, some models are used for their inherent simplicity while other models are applied for their ability to adapt to complex density functions. In this section methods are reviewed for general modelling of data in feature spaces.

When the homogeneity of data is not easily modelled parametrically or if modelling the data is analytically complicated beyond practical application it can often be more effective to model the partitions of heterogeneous regions within the image data. One of the simplest boundary-based methods is the k-Nearest Neighbour algorithm, kNN [50] is an algorithm designed to search for the k nearest labelled training data in a d dimensional feature space. From a prior generated labelled training data an extra feature dimension, or classification, can be assigned to the novel point on the basis of a weighted mean or mode calculated from the k-nearest training points. kNN explicitly defines boundaries from the classification labels of the training data and is an assignment algorithm rather than a modelling algorithm. Like all non-parametric methods kNN suffer from the *curse of dimensionality*, in that as the number of feature dimensions to be searched increases, the computational complexity increases in an exponential manner. Variations on kNN are often used in the assignment phase of other pattern recognition algorithms.

Support vector machines [167] is a supervised method for finding the optimal dividing linear hyperplane between classes that minimises the classification error on unseen data; this is achieved by performing a non-linear map of the input data to a high dimensional feature space. The support vectors relate to the feature vectors in the training set that efficiently define the boundary between the classes. Support vector machines have been found to perform well on high dimensional classification problems although the complexity can become prohibitive for large data sets.

The Adaboost learning algorithm [71] — popularised in computer vision by the work of Viola and Jones [170] — is a method for improving the accuracy of a pool of weak classifiers. This is achieved by learning a weighted combination of the weak classifiers to improve the classification accuracy over a range of training data. This algorithm is part of a wider family of algorithms known as Boosting. The advantage of Boosting algorithms is that a large pool of very simple, computationally efficient, weak classifiers can be refined into a smaller pool that can be combined linearly to create a classifier with good accuracy over

a range of learning examples. Adaboost requires a sufficient quantity of labelled examples during the learning phase, which can limit its performance in some applications where only small quantities of labelled data are available.

Decision trees are a well studied field for finding decision boundaries within data, a tree can be utilised for data generalisation, where a mapping is uncovered from independent (unlabelled) to dependent (labelled) values [179]. This can be used for predicting future dependent values. In boundary models the assignment of classes is straightforward once the boundaries between groups in the data have been defined. In *model* decision trees [70], data from the unlabelled set is followed from the *root* node down to the *leaf* node, where a smoothed linear regression function gives the predicted label assignment.

Neural networks [21] are computer learning methods inspired by the biological processes that occur in the human brain. In the neural network neurons (or nodes) are organised into single or multiple layers to allow general parameterised non-linear mappings between a set of input and output variables. The iterative learning phase of a neural network can be a supervised or unsupervised process. The successful application of a neural network is somewhat of a black art, with experimentation required to determine a good structure for the network for a given problem.

In the case where the data to be modelled exhibits modal or multi-modal homogeneity clustering algorithms can be applied to find an intrinsic classification or inherent structure in a data set using no prior information about the grouping [158]. Clustering is a very useful technique, especially in generalising large data sets into a more simplified form, and it can assist in [15, 62]:

- Formulating hypotheses about the origin of data.
- Data exploration and reduction.
- Describe data in terms of typology.
- Fit a model to the data.
- Predict future behaviour of types of this data set.
- Optimising a functional process.

There are also present many non-exclusive paradigms of clustering, classically clustering

is split into just two distinct methods — Partitional clustering and Hierarchical clustering[3] techniques. The main paradigms of current cluster techniques are [59] heuristic techniques, deterministic analysis, probabilistic analysis, hierarchical analysis and objective function techniques. An example of crisp probabilistic analysis is the iterative hard k-Means algorithm [51]. This classical clustering technique produces a crisp, or hard partition, as opposed to fuzzy c-Means which can produce soft boundaries between the clusters. The membership probability of a pixel to a cluster is equal to 1 or 0, hence each pixel is assigned membership to one cluster only, creating hard partitions.

General problems associated with clustering techniques are that the number of clusters is generally defined *a priori*, the solutions gained are local minima, the number of iterations are unknown and also that the algorithms can be computationally expensive in higher dimensions.

A related method to clustering is the graph theoretic Normalised Cut algorithm [156]. In this approach the (arbitrarily complex) feature vector set is viewed as a graph with nodes as data points and edge weights defined by a measure of similarity. In the application of this algorithm the criterion for cutting the graph can be thought of as a measure of the goodness of an image partition.

The underlying density function in the feature space can be estimated using probabilistic methods; these methods allow formal techniques to be applied to estimate and propagate the densities within the feature space of a video sequence. Modelling the feature space in this way allows analysis of the *a posteriori* probabilities to classify pixels as belonging to objects within the scene. There are two main categories for density estimation techniques — parametric and non-parametric. Parametric density estimation techniques attempt to define the functional form of the model to be fitted to the data, whereas in non-parametric density estimation the functional form of the model is the data itself. A third type of model, termed *semi-parametric* allows a number of parametric models to be adapted to the observed data in a systematic way. Both types of density estimation have advantages and disadvantages for modelling video sequence feature spaces, and can be applied to labelled or unlabelled data.

Perhaps the simplest (and computationally efficient) method for density estimation is the histogram [21]. In a histogram the range of data is binned (i.e. quantised) and samples are

[3] A tree-like description of clustering structure

accumulated to form a discrete estimate of the density function. The main drawbacks with histograms are the number/placement of the bins, the generalisation to higher dimensions and the problems of discontinuities in the estimated density. An advantage of histograms is that, unlike some non-parametric methods, the data can be discarded once the histogram has been constructed leading to an efficient representation.

Kernel density estimation (also known as the Parzen Window technique [21, 51]) is a non-parametric method that represents the observed data by centering a window function at each data point. This window function allows the density to be estimated for regions where no data was observed by performing a summation over all the windows that overlap the region of interest. The form of the window function can be discrete (e.g. a hypercube kernel [21]) or continuous (e.g. the Epanechnikov kernel [53]). Kernel density estimates share many similarities with histograms, and like histograms the estimate has discontinuities and does not scale well with dimensionality.

A recently rediscovered robust non-parametric iterative method for density estimation is the mean shift (or mode-seeking) method. The mean shift method was originally proposed by Fukunaga and Hostetler [74] and was revisited by Cheng [32] with generalisation and application to clustering and optimisation. The use of mean shift was further popularised in the computer vision community by Comaniciu and Meer, with works primarily focussing on image segmentation(e.g. [44]) and tracking (e.g. [41]). In the former work, the mean shift algorithm was applied to discontinuity preserving smoothing and image segmentation; in the latter work, the mean shift kernels were used to match objects using non-parametric appearance models (histograms) and the Bhattacharyya metric. Like many non-parametric techniques, the mean shift method does not scale trivially with dimensionality, and care must be taken to ensure the kernels contain enough information to find the high density pockets in the feature space. Like standard kernel density techniques determining the size of the kernel is a drawback of this method, it must be chosen a priori or determined using an additional module [43].

Gaussian based distributions are very useful for modelling data since they can be updated efficiently and allow a formal approach (i.e. probabilistic and model based) to video object extraction in contrast to other types of classifier. The simplest method for estimating the true probability density function (PDF) of observed feature vectors is to fit a single Gaussian density function to the data, which results in an estimated model that has few

parameters and is fast to generate from a set of observation samples.

The primary drawback of using a Gaussian distribution is, like all parametric methods, the functional form of the model is chosen *a priori*. If the underlying PDF does not match this form then the model is unlikely to be representative of the data. Hoti and Holmstrom [90] attemped to improve the applicability of Gaussian models (to retain the efficient representation) by transforming the data to separate the Gaussian and non-Gaussian data then using a combination of a Gaussian and a non-parametric kernel density model to represent the data; the resulting algorithm is computationally complex and may be too complex to apply to high dimensional data.

To overcome the drawbacks of parametric and non-parametric methods Gaussian mixtures models [51] represent a mixture of parametric models that are combined to form even more complex functional forms to estimate the true PDF of a data set. In statistical pattern recognition finite mixtures such as this allow a formal approach to the problem of unsupervised learning (i.e. clustering of data) or representation of arbitrarily complex PDF's. Gaussian Mixture Models, like clustering algorithms in general, have many drawbacks such as estimation of the number of components and convergence towards local minima and singular estimates in the feature space. Figueiredo and Jain [67] propose an unsupervised algorithm for finite mixtures overcoming many of these problems associated with mixture model initialisation and fitting. A principled technique for model order selection is the *minimum description length* criterion which was introduced for mixture models by Rissanen [146]. The maximum likelihood solution for fitting a Gaussian mixture model to a data set can be estimated using the Expectation Maximisation (EM) algorithm [1].

With an representative model of the feature space *decision theory* can be applied to create a *decision function* that minimises a *cost* associated with such a decision (and hence improve the classification accuracy). Therefore, a *classifier* consists of two fundamental stages — *density estimation* and a *decision function*. Decision rules are determined by the type of representative model that has been applied to the classification problem. For continuous probabily density models a commonly applied Bayesian decision rule is the *Maximum A Posteriori* (MAP) rule [51]. This rule minimises the expected (Bayesian) error of a decision by choosing the discrete labelling that has the maximum posterior probability. An alternative approach is to label points with the object label of the *nearest* cluster centroid — this decision rule is commonly used in non-probabilistic clustering methods

where the problem of labelling pixels can not be stated formally due to the heuristic nature of the algorithms. In many cases the labelling of the objects is implicit, for example in the watershed segmentation the region labelling is the representational form of the model.

Chapter 4 applies probabilistic representational models to the problem of video object segmentation. Three approaches to PDF estimation are implemented in the object segmentation framework — Gaussian density, kernel density and Gaussian mixture models.

2.6 Propagation of Video Object Representational Schemes

The methods for modelling video objects presented thus far are essentially supervised image segmentation algorithms. For these methods to be applied to the problem of video object segmentation methods are required to allow the representational schemes to *adapt* to the evolving video sequence. The update process for the representational models in the segmentation scheme can be thought of as an *estimation* process comprising three stages:

1. *prediction* — a model prediction made using previous observations.
2. *matching* — the model prediction is associated with the current observations.
3. *correction* — the current (corresponding) observations are used to correct the prediction, giving the estimated model.

This terminology is common in tracking literature (e.g. [16]). Using the framework presented in Figure 2.1 it can be seen that the prediction step is the inter-frame update step and the correction step is the intra-frame update of the representational model. In this section methods for updating representational schemes in the context of video object segmentation are reviewed.

2.6.1 Inter-Frame Prediction Strategies

The goal of the inter-frame prediction step is to determine the likely location and appearance of the object in the subsequent frame given the observations made in previous frames in the video sequence. Generally, this stage is used for updating the spatial component of the objects representation. The appearance of objects generally undergo relatively minor change between adjacent video frames (assuming the temporal resolution is high and the light

sources in the scene do not change dramatically). To allow the appearance representation of the object to adapt to the evolving video sequence it can be updated (along with the spatial representation) in the intra-frame update step. The inter-frame prediction step does not introduce new model components into the representational scheme [152].

It is common in video object segmentation schemes to propagate the representational models between frames by simply using the previous frame representation — unchanged — in the current frame. Of course, this type of inter-frame update methodology is based on the assumption that the object does not move a significant amount in between frames and is therefore a sequence dependent assumption. For many standard test sequences this is found to be a reasonable assumption and has been used in video object extraction schemes by several researchers e.g. [98, 61, 135, 171, 91, 11].

An alternative strategy is to use the motion information extracted from the video sequence to perform model-based compensation of the objects. The motion field for a video frame is extracted as raw optical flow; optical flow fields are generally noisy with many outliers resulting from uncovered/covered background or the aperture problem. Some approaches to video object segmentation warp (i.e. motion compensate) the representational schemes using per pixel motion information (e.g. [128, 111, 112, 27, 6]).

To limit the problem of unreliable motion information it can be beneficial to measure the motion within regions of the video sequence to estimate a parameterised model of the motion for that region. The parameters for the motion model are solved using an optimisation technique (e.g. least squares). The resulting model has a compact form that can be very useful in the context of video analysis. A parametric model for 2-D motion fields can be derived from parametric models describing 3-D motion, 3-D surface function and camera projection geometry (Stiller and Konrad [160] give a good introduction to the formation of motion models). These motion models assume rigidity within the region of interest, therefore objects undergoing non-rigid (i.e. articulated) motion require further processing to robustly estimate the multiple rigid motions.

The simplest motion model is a translational model, this is effectively the average motion vector measured over a region (or object) in the video sequence. It follows that an optical flow field can be thought of as a pixel level translational motion model. A translational motion model can model translational motions of objects, for objects undergoing non-translational motions the model will not be representative of the actual motion.

Perhaps the most popular motion model used in video based analysis is the affine motion model. This model is generally applied to sequences where the 3D scenes are sufficiently far from the camera to reduce the effect of perspective motions [94]. The affine model is generally seen as a trade off between model complexity and processing efficiency [125]. The affine model suffers from the problems common to many higher order motion models (and indeed modelling in general) in that a sufficient sized region must be used to allow reliable estimation of the motion model and that the simpler model may not sufficiently represent the motion of the video object.

In the case of projective motion it is neccesary to use a planar projective motion model. The difficulty of using such a model is that the regions must be sufficiently large enough to allow robust estimation of the eight parameter model. If the expected motion in the scene cannot be determined *a priori* then a hierarchy of motion models can be evaluated to determine the simplest model that can accurately warp a region of an image [95]. In such a scheme the residue error after motion compensation is measured using the simplest (i.e. translational) motion model and compared to the next most expensive model to check if there is any improvement in the accuracy of the result. This can be performed upto the eight parameter planar projective motion model. In the case of minor improvement in warping accuracy between adjacent models the simpler motion model would be chosen.

The prediction (and subsequent correction) of the video object representation can be performed using recursive filtering methods. Recursive filtering is used to estimate the current state of a model by combining the current observation with the previous observation history, essentially smoothing the estimate over a temporal window. Perhaps the most commonly applied recursive filter for tracking is the Kalman filter [97, 178]. The Kalman filter provides the optimal linear estimate of an unknown state by using known dynamics and the observable data. When the process noises in the system are Gaussian this filter will provide the optimal estimate. A Kalman filter can be applied to improve the estimate of, for example, affine motion parameters for a video region (e.g. [65]). In such a scenario the affine parameters are either assumed to be constant or to change according to some known model (e.g. constant change), and that any deviations from this model are due to the Gaussian process noise.

Particle filtering [12] provides an alternative strategy to the Kalman filter when the process noise in the system are non-Gaussian or the state of the tracked object is non-

Linear. Particle filter based tracking works by representing the posterior density (i.e. how likely a state given an observation) by a set of randomly sampled particles with associated weights. By resampling particles from a 'proposed' density (which should resemble the expected density) and injecting new ones, the posterior distribution can be evolved over time to track an objects state. A potential drawback when applying particle filters is the tradeoff between computational complexity and drawing a sufficient number of samples to adequately describe (and hence propagate) the underlying density.

2.6.2 Intra-Frame Matching Strategies

The goal of the intra-frame matching step is to correspond the predicted object-based representation with the observed objects in the current frame. In video object segmentation the observed objects are generally discovered using the predicted video objects by labelling the frame, therefore an explicit matching step is not required. The matching step is important when the object segmentation can be performed *independently* for each video frame i.e. using motion information [64]. In such a scenario mechanisms are required to correspond the video object representations from the previous frame with the newly discovered objects in the current frame [186, 187]. This can be performed using the appearance information (e.g. [171, 77, 76]) or filtered motion information for each object (e.g. [64]).

2.6.3 Intra-Frame Update Strategies

The goal of the inter-frame update step is to correct the estimated video object representation by updating the representational model for the object using the current frame data. The dynamic nature of video sequences makes this a challenging stage in the segmentation process, a balance must be sought between the adaptibility of the model and the robustness to noise in the underlying signal. For specific video sequences the assumption of constant object appearance [186] may result in the intra-frame update of the representational being a trivial step (e.g. [30]). This assumption is generally only valid for short video sequences.

A common per frame update methodology for the object representational models is to find the region of support for the object in the current frame using the propagated models and then reinitialise the object representation based on this new found object region (e.g. [61]). For parametric representational schemes the per frame reinitialisation can be an expensive and time consuming technique for intra-frame updating of the models. For

non-parametric methods — where the functional form of the model is the data — this type of update can be computationally cheap. A better strategy for updating parametric models is to use the previous frame model as a seed to guide the reinitialisation of the models to the newly observed data. This type of approach has been applied by [27, 135, 145, 111].

Alternatively, the object representation can be adapted to take into account changes in the object appearance due to lighting, pose or motion changes etc. One possible technique to achieve this on a per frame basis is to apply a recursive filter to smooth the adaption of the model so that predicted object parameters are less susceptible to noise. For Gaussian mixture models the update equations can be expressed to allow sequential, as opposed to batch, processing. The incremental (or online) EM algorithm [49, 39] is a form of recursive filter such that the the model parameters are recursively updated by each newly observed feature vector, weighted by the probability that the feature vector belongs to that model. This allows the model estimate to slowly adapt to changes in the true PDF.

2.6.4 Spatio-Temporal Representation

If the video data is available for processing in batch mode then joint spatial and temporal grouping [45, 87, 150, 103, 104, 139] can be applied to perform grouping in the spatio-temporal video volume. Using the temporal information explicitly in the object representation can alleviate the problem of how to update the video object on a per frame basis, as this is included in the representational model. These type of schemes have drawbacks such as fast moving small objects exhibiting discontinuities in the space, the large data storage requirements of the video and the limited scalability of the approach.

Chapter 5 and Chapter 6 introduce methods for propagating representations. Chapter 5 details and evaluates methodologies for propagation and innovation of video regions and Chapter 6 extends these methods to propagate video object representations.

2.7 Performance Evaluation of Video Object Segmentation

It is important in any scientific discipline to define an evaluation scheme that characterises the algorithms in a clear and unbiased manner. There is limited work in the literature relating to the evaluation of video object segmentation algorithms. Since video object segmentation is the extension of image segmentation (a review of which can be found in

[131]) to video sequences, then it follows that image segmentation evaluation strategies can be extended to quantatively evaluation video object segmentation algorithms.

Zhang [184] defines three main groups of analysis — analytical, empirical goodness and empirical discrepancy. Analytical analysis of segmentation considers the effectiveness of the algorithm itself, based on measures of the principles, requirements, and complexity. These methods fall short due to the lack of a general theory for image segmentation from which a true analytical comparison can be derived.

The empirical methods aim to judge the quality of a segmentation based on quantitative evaluation. Quantitative evaluation is represented by either a *goodness* factor or a *discrepancy* measure. The empirical goodness method computes the goodness of a segmentation without the *a priori* knowledge of a reference segmentation.

Empirical discrepancy uses a reference segmentation to allow mathematical discrepancy evaluation of the output segmentation. This is very useful when the algorithm is complex and fully automated. The evaluation procedure for depth content generation was derived from this technique. Using the set of ground truth[4] segmentation maps, the output depth map was quantitatively evaluated using a discrepancy measure between the output and the ground truth.

In the realm of video object segmentation techniques have been developed to evaluate segmentation quality using measures of empirical goodness and empirical discrepancy. Villegas *et al* [142] suggest the use of perceptive weights to attempt to quantitively evaluate desirable properties of a segmentation mask. This approach consists of a per frame spatial quality measure and a second measure of temporal stability that is measured from two frames. In his thesis, Giaccone [79] suggests that the perceptual weightings have no grounds in any subjective studies. In post production the segmentation mask of a video object should exhibit accurate edge location and mask *density* i.e. the degree to which connected pixels forming an object in a ground truth segmentation are represented by pixels forming object regions in the outputted segmentation mask exhibiting the same connectedness.

The evaluation methodology adopted by the ISO/MPEG-4 [165], COST 211 [5] and ACTS/MoMuSys [110] projects was presented by Mech and Wollborn [118] where the spatial accuracy and temporal stability of the segmentation mask is compared to a ground truth segmentation mask; although the temporal stability is poorly evaluated by this mea-

[4]In medical work these are often termed gold standards

sure since it does not accomodate object movement. Another problem with spatial based discrepancy measures are that the majority of the errors detected are from a few, larger, regions. In an attempt to capture information about the many, smaller, estimation errors at the object boundary Mech and Marques [115] suggest a measure of distance between the object contours and a ground truth contour, supplemented with temporal coherency measuring the variation of the gravity centres of the video objects and the variation of the spatial accuracy. This approach has many practical drawbacks due to the complexity and discrete nature of object boundaries, making distance measures between two contours a difficult and somewhat inaccurate process.

Erdem and Sankur [54] evaluate three distinct approaches to video object segmentation. To achieve this they evaluate four penalty measures with respect to a ground truth segmentation. A combined penalty measure is formed from a weighted average of the misclassified pixels (weighted by proximity to boundary), a shape penalty based on turning angle functions and a motion penalty based on motion trajectories. Erdem, Tekalp and Sankur [55, 56, 57] build on this work to suggest empirical goodness metrics to evaluate video object segmentation without ground-truth. This work uses three *a priori* assumptions that the object boundaries coincide with colour boundaries, that the colour histogram of an object is stationary and that the colour histogram of the background is different to the object (although not necessarily stationary). To provide per frame quantitative goodness analysis colour and motion differences are analysed along the boundary of the video object. To measure whether the object is tracked correctly in each frame the colour histogram differences are observed between the video object in two successive frames so that the introduction of background information into the object mask will increase the distance between the histograms. Due to the prior assumptions made, this method can not be readily applied to generic video object segmentation evaluation; in generic scenes the assumptions may be violated. This work is further integrated into a video object tracking algorithm [58] that uses a feedback of performance evaluation measures to evaluate the goodness of the segmentation.

Correia and Pereira [36, 37] show a set of evaluation methodologies for both empirical goodness and empirical discrepancy methods. The empirical goodness based methods are split into two major categories — intra-object homogeneity measures and inter-object discrepancy measures. Empirical discrepancy measures are also split into two major categories

— spatial accuracy and temporal accuracy. They propose the addition of a further metric, Criticality, that combines the spatial and temporal information into a spatio-temporal evaluation of video sequence complexity. These metrics are applied to video object segmentation evaluation for both individual objects and overall evaluation. Many of the performance metrics shown in this work have perceptual weight terms based on informal subjective tests, this could affect the objectivity of the evaluation work.

Performance evaluation is performed in all chapters in this thesis. It is crucial that existing and proposed methods are evaluated to give a more thorough understanding of the characteristics of the algorithms.

2.8 State of the Art Video Object Segmentation

In this section the state-of-the-art approaches to video object segmentation and related fields are reviewed. There have been some recent developments in spatio-temporal based approaches to video object segmentation. Ahmed *et al* [3, 2] evolve the previous work of Greenspan *et al* [87] to perform automated segmentation of video objects using spatio-temporal Gaussian mixture models. Ahmed *et al* extend the basic approach to account for the relatively poor representation of object shape by the multi-dimensional GMM representation. To improve this, they analyse the spatial distribution to generate a uniform density based spatial model using the concept of *chords* passing between the object boundary and the object centre. It is unclear whether the resulting PDF representation of the object is similar to that which can be achieved by kernel density estimation (i.e. a binary mask with a 'fuzzy' boundary). Ristivojevic and Konrad [147] present an alternative approach to analysing video volumes using the concept of object tunnels. Their approach is interesting in that it only applies motion information (affine motion models) within a volume competition framework (a generalisation of region competition [162]). Explicit occlusion reasoning is applied using occlusion volumes within the 3D space of the video volume. The approach is only demonstrated for scenes with stationary backgrounds, and further work is required to apply the technique to general image sequences.

Mezaris *et al* [171] present a per-frame approach to automated region segmentation using colour and motion features. Regions are tracked using frame differencing to locate changed pixels and then normalised histograms are used to classify the pixel to the neigh-

bouring (unchanged) regions. New regions are detected using a rule-based approach based on the homogeneity of the region colours and motion models are applied when merging neighbouring regions with similar appearance. A limitation of the new region detection step is that misassociation can occur with previous (extinct) regions in the case that their appearances are similar. Kolmogorov *et al* [102] present two novel approaches to the problem of segmenting video layers when using a stereo camera. The two approaches (based on layered dynamic programming and layered graph cuts) both fuse stereo information with colour and contrast, captured by a stable probabilistic model. It is demonstrated that the fusion of stereo and colour/contrast is a more powerful descriptor than either alone and that good quality stability can be achieved without imposing temporal constraints.

In the related field of video matting, Li *et al* [108] build on the work of Chuang *et al* [35, 34] to present a system for extracting smooth object maps from video sequences. In their approach they apply a novel 3D graph cut based approach on the spatio-temporal video volume. The 3D graph cut approach partitions a per-frame watershed segmentation into foreground and background regions, preserving the temporal conherence. Feature tracking between key-frames is used to refine the segmentation using a local colour-based 2D graph cut. A user interaction step allows the operator to refine the boundary where necessary. A further area of work is the extension of such techniques to extracting alpha mattes for multiple objects. In a similar appraoch Wang *et al* wang2005ioa, wang2005ivc extend a per-frame watershed segmentation that is extended over time using graph cut. The user interacts via a novel spatial-temporal manipulation tool.

Apostoloff and Fitzgibbon [10] present an automated approach to spatio-temporal object segmentation using sparse features in the 3D video volume. These features (spatiotemporal T-junctions) are used as indicators of occlusion edges, which are learned by an occlusion edge model and a foreground/background appearance model. Finally, the segmentation is solved using a graph cut MRF that combines appearance and occlusion edge terms to give a global solution. Han *et al* [85] propose a related approach based on sequential clustering of sparse edge and corner points. Sparse motion layers are extracted using a joint spatio-temporal linear regression method. Finally, dense motion layers are create by using MRF to assign the remaining image pixels using colour and spatial proximity to the local sparse features. Xiao and Shah [181] apply the graph cut algorithm within a spatio-temporal video volume to generate a motion-based segmentation. A general occlusion constraint is used to

determine the object and occlusion segmentations. Finally, Wang and Ji [176] introduce a method to integrate contextual constraints into object based segmentation, with spatial and temporal dependencies unified within a probabilistic framework. The segmentation method combines intensity and motion cues with video frame history and spatial interaction of the data. This combination of features was shown to improve the accuracy of video object segmentation.

An alternative strategy to extract video objects is to use *a priori* 3-D models to determine the presence of objects at each frames. Everingham and Zisserman [60] demonstrate such an approach for detecting and extracting people from video sequences. The first step of this approach is to match the pose of a (tracked) face region using the 3D model for all the faces in the training set. Once the pose is estimated, the target can be identified by matching the proposed faces from the training data. The use of 3D textured training models may not be suitable for tracking articulated objects, since extra free parameters will be introduced into the matching process. This may make the resulting algorithm computationally prohibitive. An alternative approach to using high quality training data is to attempt to train object categories using features extracted from a large pool of labelled images (e.g. pictures of cars etc). Liebe *et al* [105, 106] presented a method combining the capabilities of both object categorisation and segmentation within a common probabilistic framework. Features extracted from an image frame are first matched (via a codebook) to those extracted from the training set. The hypothesis of the object type / location is then determined using localised probabilistic voting. This allows the backprojection of the object hypothesis into the image, which is refined into a category specific segmentation mask using the local image appearance. Such an approach is of great interest for the future of video object segmentation, since it allows labelled imagesto be used to segment images on a per-pixel basis. It is not unfeasible that such an approach could be trained used an internet based search engine (e.g. Google Image Search [82]) leading to *near*-automated methods for semantic video object segmentation / categorisation.

In the field of video object segmentation evaluation, Gelasca *et al* [78] presented an automatic evaluation framework incorporating quantitative measures chosen to incorporate perceptual factors associated with the end application. A drawback with this approach is that the perceptual factors have to be attained by subjective application-specific tests using human subjects, which may limit its wider application.

Chapter 3

Feature Spaces for Video Object Segmentation

The choice of feature space and the extraction of it from the raw image data is a fundamental part of the video object segmentation process.A *feature space* is defined as being a multi-dimensional space that wholly encloses a finite set of *feature vectors.* An example of a 2D feature space is shown in Figure 3.1. In this example the set of feature vectors is divided into two classes — triangles and circles. In this simple example the classes are well separated — the feature space is said to provide enough *discriminatory evidence* to classify observations as belonging to one of the two classes. In this example it is noted that there are also two *outlier* vectors (i.e. pixels that have erroneous values); such outliers are often generated by extraneous signals in the measurement of the feature vector or from insufficient sampling of the data (i.e. not seeing the complete structure of the feature space).

A model-based representation can be built using the information contained within this feature space,allowing further observations to be classified as belonging to one of the two classes. The representation of the data can primarily be *region* or *boundary* based i.e. a model of intra-region homogeinity or inter-region heterogeinity. The actual classes associated with the feature vectors may be provided by a human operator or derived automatically, for many classifiers it is imperative that the training data is labelled *a priori.* The representation of this space is discussed in depth in Chapter 4. In the current chapter the choice and extraction of the feature space are focussed on with respect to video object segmentation.

For image and video analysis a feature vector set can be extracted for all the pixels

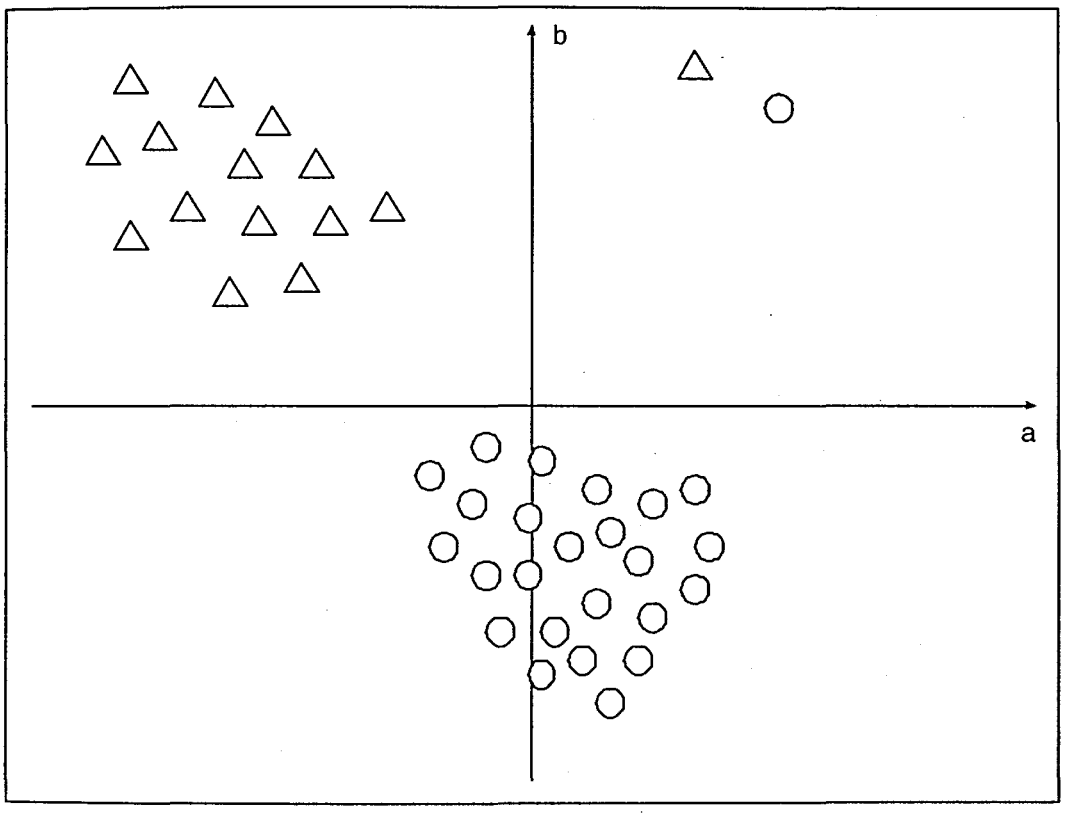


Figure 3.1: An example of a 2D feature space containing vectors representing two distinct classes.

available (*dense*) or a subset (*sparse*), the extraction process itself can be an operation on the individual pixels (e.g. colour) or the surrounding pixel neighbourhood (e.g. motion). It is possible that the pixel neighbourhood contains more than one object and hence features extracted in this region may not be representative of the individual objects. The feature vector $\mathbf{a}$ extracted at a pixel is a function of the spatial and temporal pixel location i.e. $\mathbf{a} = \mathbf{a}(x, y, t)$. This D-dimensional feature vector is formed from the D scalar values that are the extracted quantities at that pixel, such that:

$$\mathbf{a} = [a_1, \ldots, a_D] \tag{3.1}$$

The feature vectors extracted from an image or video therefore form a D-dimensional feature space. To perform model-based analysis of the feature space a *distance metric* needs to be defined which defines the measurement of the distance between two feature vectors and allows the distributions of feature vectors in the space to be described in a principled manner.

An example of feature space analysis in video sequences is presented in Figure 3.1. In this example, the feature vectors contain the translational motion information measured at sub-sampled pixels in a video frame. It is clear that there are two distinct *clusters* which represent two dominant motions in the frame. Using a model based representation of these clusters all the remaining image pixels can be classified as belonging to one of the two dominant motions and hence a *layered representation* of the scene can be found (e.g. Ayer and Sawhney [14]).

3.1 Distance Metric in the Feature Space

Within the feature space the concept of distance is required i.e. the *proximity* of two feature vectors. One of the simplest distance metrics is the *Manhattan* (or *city-block*) metric. The Manhattan distance between two D-dimensional feature vectors, $\mathbf{a}_1$ and $\mathbf{a}_2$, is given by:

$$\text{dist}\left(\mathbf{a}_1, \mathbf{a}_2\right) = \left|\mathbf{a}_1 - \mathbf{a}_2\right|_1 = \sum_{d=1}^{D} \left|a_{1,d} - a_{2,d}\right| \tag{3.2}$$

The Manhattan distance is relatively efficient to compute, although it could over-estimate what is termed the 'direct' or *Euclidean* distance. The Euclidean distance in the feature space between two D-dimensional feature vectors, $\mathbf{a}_1$ and $\mathbf{a}_2$, is given by:

$$\text{dist}\left(\mathbf{a}_1, \mathbf{a}_2\right) = \left\|\mathbf{a}_1 - \mathbf{a}_2\right\|_2 = \sqrt{\sum_{d=1}^{D} \left(a_{1,d} - a_{2,d}\right)^2} \tag{3.3}$$

The vector $\mathbf{a}$ can be a hybrid feature vector, that is, it can be comprised of several features combined into a multi-dimensional vector. Each dimension in the feature vector is characterised by a different scalar range and therefore the Euclidean distance may not be meaningful when using hybrid feature vectors. To accomodate the scale differences in the dimensions it is common to use the *Mahalanobis* distance [109]. In this distance metric the data is normalised using the covariance of the feature dimensions over the entire feature vector data set. These covariances are stored in a covariance matrix, Σ, such that:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} \left[\mathbf{a}_i - \mu\right] \left[\mathbf{a}_i - \mu\right]^T \tag{3.4}$$

$$\text{where } \mu = \frac{1}{N} \sum_{i=1}^{N} \mathbf{a}_i \tag{3.5}$$

Given the covariance matrix Σ, the Mahalanobis distance between two feature vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ is given by:

$$\text{dist}\left(\mathbf{a}_1, \mathbf{a}_2\right) = \|\widehat{\mathbf{a}}_1 - \widehat{\mathbf{a}}_2\|_2 = \left[\left(\mathbf{a}_1 - \mathbf{a}_2\right)^T \Sigma^{-1} \left(\mathbf{a}_1 - \mathbf{a}_2\right)\right]^{\frac{1}{2}} \tag{3.6}$$

The Mahalanobis distance is the same as the Euclidean distance if the covariance matrix is the identity matrix. It is common to assume that the feature vector dimensions are *independent*, that is, they are *uncorrelated*. Under this assumption the covariance matrix is block diagonal, containing only the intra-dimension variances of the feature vector data set. Therefore the uncorrelated Mahalanobis distance between two feature vectors, $\mathbf{a}_1$ and $\mathbf{a}_2$, out of a set of N feature vectors is given by:

$$\text{dist}\left(\mathbf{a}_1, \mathbf{a}_2\right) = \|\widehat{\mathbf{a}}_1 - \widehat{\mathbf{a}}_2\|_2 = \sqrt{\sum_{d=1}^{D} \frac{\left(a_{1,d} - a_{2,d}\right)^2}{\sigma_d^2}} \tag{3.7}$$

$$\text{where } \sigma_d^2 = \frac{1}{N} \sum_{i=1}^{N} \left(a_{i,d} - \mu_d\right)^2 \tag{3.8}$$

where μ_d is the mean value of the feature scalar a_d over the N measurements in the data set.

3.2 Previous Work

In Section 2.4 popular feature spaces for video based analysis were reviewed. In this section the feature spaces are discussed in the context of existing work on video object segmentation. The feature spaces commonly applied for video object segmentation use a combination of colour, texture, gradient and motion information to delimit the objects in the image plane; implicit to this is a defined mapping between the raw image plane space (spatial and RGB colour co-ordinates) and the feature space.

Several approaches to video object segmentation [138, 183, 52, 124, 120, 121, 116, 117, 135, 33, 6, 5, 134] use luminance (grey-scale) information to provide an analytically simple feature space component. The main advantage of using luminance information is that it simplifies the representation of object appearance compared to colour information. The main disadvantage is that the luminance information is often not sufficient to provide even the human visual system with enough information to distinguish objects within the scene.

The RGB colour space has been applied in many object detection and tracking algorithms [72, 19, 18, 145, 154, 128, 83, 13, 150, 96, 76]. The effect of the luminance on RGB measurements can be reduced by normalising the per channel signal with the combined signal strength over all the channels [130, 11].

Other approaches to video object segmentation [103, 104, 111, 98, 139, 96, 27, 31, 149, 112] have applied the YUV colour space to analyse video sequence information. The main advantage of this space is that it allows the luminance (Y) and chrominance (UV) information to be used separately and hence can be used to overcome some of the problems with the RGB colour space (e.g. [80]). A related colour space to YUV is YIQ, applied in [30] for video object segmentation. To give a more intuitive description of colour and add robustness to light changes within the scene the RGB space is often transformed non-linearly to the HS-family of colour spaces. This transform splits the luminance and chromaticity information allowing the modelling of a chromatic signal without the influence of brightness (as with YUV), this gives algorithms based on HS (i.e. Hue and Saturation) colour signals a limited amount of robustness to appearance changes of an object, although the problems of measuring colour value at the extrema of the luminance scale and the fact that different light sources emit different chromatic signals mean that the use of HS is often used in controlled enviroments with limited light sources (e.g. [143]).To give a more intuitive description of colour and add robustness to light changes within the scene the RGB space is often transformed non-linearly to the HS-family of colour spaces. When modelling the HSI colour space it is important to take into account the cyclic property of the hue (including a discontinuity in the space) and the relationship between the hue and the saturation. One way to achieve this is to convert the hue and saturation to cartesian co-ordinates to form cartesian Hue-Saturation-Intensity (XYI) space [42]. A similar technique has been applied to content based image retrieval [153], where it is proposed that this type of encoding caters for the fact that at small saturations (i.e. near the intensity axis) the hue differences are meaningless (since little useful colour can be measured). It does not, however, weight the hue as more relevant for larger saturations and intensities of colour i.e. at the widest part of the colour cone.

Several approaches to video analysis (e.g. [29, 61, 171, 87]) include CIE-$L*a*b*$ in the feature space, due to the preference of perceptual uniformity. CIE-$L*u*v*$, a uniform colour space similar to CIE-$L*a*b*$, has been applied to video based segmentation by

[186, 47, 95, 96, 91].

Several approaches to video object segmentation and tracking are based solely on colour (for example, [72, 143, 145]) although in many applications there is a requirement for modelling of the spatial distribution of the object. The use of spatial information in the feature space can be useful to define explicit structure. [30] combined spatial, textural and chromatic information to help delimit multiple objects with similar chromatic signals. There are many techniques that apply spatial cues to aid the segmentation of video objects [111, 61, 87, 28, 103, 104, 135, 27, 91, 11].

Textural analysis can be added to the feature space so that objects can be discriminated on the characteristics of the way the colour information is spatially distributed on the object. Texture descriptors extracted from the windowed second moment matrix have been applied to content based image retrieval [153], detection of repeated scene elements [107] and video object segmentation [61]. Gabor filters decompose images into multiple orientated spatial frequency maps from which amplitude or phase analysis can be used to form the feature maps [177, 11]. For video object analysis they have been found to offer neglible advantage over the simpler windowed second moment matrix scheme given the higher feature vector dimensionality and greater computational expense [61].

Motion or temporal information about the video objects can also be introduced into the feature space, describing the motion of pixels and regions between frames. Per pixel motion information can be applied in video object segmentation approaches as an additional discrimantory feature in the space (e.g. [27, 28, 30, 128, 171, 98, 8]), in this form the motion information is often weighted to account for the fact that it tends to be unreliable at the edges of scene objects and in areas of constant intensity in the image. The recovered per pixel motion field only displays homogeneity between pixels that are undergoing similar translation in the image plane; the optical flow information does not display homogeneity for pixels undergoing rotation, zoom or other complex movement.

A parametric model for per pixel (2-D) motion fields can be derived from parametric models describing 3-D motion under projective geometry. The simplest motion model is a translational model, which displays feature space homogeinity only under similar object translations. The most commonly used motion model in video analysis work is an affine motion model (used in [186, 124, 63, 52, 173, 120, 121, 66, 24, 58, 7, 46, 166, 134]) that exhibits feature space homogeneity for object regions undergoing similar translation, rotation

and zoom in scale. Motion models are often applied to generating a layered representation of video frames (e.g. [173]).

Some recent approaches to video analysis have used the concept of discrete time (i.e. the frame index) in the feature space [45, 150]. The use of temporal information is, at the time of writing, limited by the large computation and storage demands of multi-dimensional video processing. Fast moving objects within the scene may also exhibit discontinuities in the 3-D neighbourhood and there is little literature relating to how this may affect the final object segmentation result.

The choice of feature space can be treated as a selection problem where the most efficient and discriminating feature space is sought from a much greater pool of feature spaces. The selection of the 'good' features can be approached by either optimising the segmentation performance of the feature space over a range of test data (e.g. Guo *et al* [95, 96]) or use a learning algorithm to select a small number of critical features that best describe the objects of interest (e.g. Viola and Jones [170]). Viola and Jones presented the application of the Adaboost learning algorithm [71] to select 'good' image-based features out of a much greater pool. In this work, the Adaboost algorithm is applied to find the 'good' weak classifiers out of a much larger pool of weak classifiers that each depend on a single feature. In this way the selection of the 'good' weak classifiers will in turn select the 'good' features from the pool that best discriminate a set of learning examples. A common problem with multiple feature space based approaches is that the multiple feature space transformations and analysis can make them computationally prohibitive and that the resulting (hybrid) feature space may contain illogical (and inefficient) combinations of features that represent the same visual characteristic.

3.3 Feature Vector Extraction

Section 3.2 gave an insight into the myriad of features that can be applied to video object segmentation. In this section the feature space components to be evaluated are discussed. The raw video data is RGB and hence the feature spaces are derived using transformations of the RGB colour space and in Section 3.4 the discriminatory evidence gained or lost by each transformation is exploited using a clustering algorithm. Since there are many potential feature space components that can be applied to this problem the choice is limited

to feature components that are well known in video object segmentation or content based media retrieval work. Section 3.3.1 discusses the four principle colour spaces chosen — RGB, YUV, XYI and CIE-$L*a*b*$. Section 3.3.2 discusses the extension of these colour spaces with spatial information, Section 3.3.3 exploits motion features, and finally Section 3.3.4 shows the calculation of textural features that can aid the segmentation of video objects.

3.3.1 Colour

When extracting or recognising objects within a scene, colour is a powerful descriptor — one which is fundamental to the human perception of scene objects. There are two main classes of colour models predominant in computer vision — linear and non-linear spaces. Linear colour spaces are based on the principle of linear combination of the three primary colours.Non-linear colour spaces attempt to describe colour in more intuitive terms and are thought to most closely relate to human colour perception. There is also a class of uniform non-linear spaces where colour differences measured in the space relate to percieved differences by humans.

Linear colour spaces

The most common linear colour space, as mentioned previously, is the RGB space. This space is commonly used in computer vision since it is the space used by computers to display colour information. As the values of each channel of this colour space are bounded in the range 0–255, each channel can be represented by a single byte — an important consideration in early work on vision algorithms. Each channel of the colour space represents a primary spectral component (i.e. red, green and blue) and the space is formed in a cartesian co-ordinate system — the RGB cube. A colour in this space is defined by a vector extending from the origin $(0, 0, 0)$ (black) to $(255, 255, 255)$ (white), although the colour space is often scaled to the range (0.0–1.0) prior to conversion to other colour spaces. It should be noted that the RGB colour space is intuitively non-uniform, such that a proximity between two colours in the space may not necessarily represent two colours that are of similar appearance with respect to the human visual system. An RGB colour feature vector $\mathbf{f}_{rgb}$ is defined as:

$$\mathbf{f}_{rgb} = [\ r\ g\ b\ ]^T$$

A major disadvantage with the RGB space is that the luminance and chrominance

channels are combined such that an object colour will not appear to be uniform over time if there are lighting intensity changes in the scene. The effect of lighting intensity changes on a colour measurement is not to be confused with *colour constancy*, this is an area of image processing concerned with measuring uniform object appearance under differing lighting intensities and *temperatures* i.e. the perceived chromatic content of the light.

There are three colour spaces based on a linear transform of the RGB space commonly found in broadcast and picture systems, these are namely YUV, YIQ and $YCbCr$. The YUV colour space has one luminance(Y) channel and two chrominance channels (UV). The transformation from an RGB feature vector to the YUV vector $\mathbf{f}_{yuv}$ is performed as follows:

$$\mathbf{f}_{yuv} = \begin{bmatrix} y \\ u \\ v \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & 0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

The YUV colour space is advantageous over the RGB space due to the fact that the luminance and chrominance information is separated which can make the modelling of video objects more robust to lighting intensity changes in the scene. A major drawback with both the RGB and YUV colour spaces is that they are not well suited to describing colours in human terms. This consideration is important in video object segmentation if an operator is required to directly manipulate the colour information of the objects to be extracted.

Non-linear colour spaces

A more intuitive system is to describe colour in terms of the *hue*, *saturation* and *brightness*. The hue is the attribute of the colour that distinguishes, for example, blue paint from red paint. The saturation describes the quantity of the hue property — for example, varying quantities of red paint to white paint can be mixed such that the hue remains constant while the saturation (or amount) of red paint added to the white produces a range of colours encompassing pinks and reds. The brightness, or physical intensity of the perceived light, is subjective and is practically impossible to measure. This is often substituted for the *intensity*, a property that can be measured, although there are numerous examples in which an object of uniform intensity appears not to be of uniform brightness [141].

This description of colour leads us to the HSI colour space, a member of the HS-family of colour spaces. The HSI space is calculated by standing the RGB colour cube on it's

black vertex. To achieve this, the conversion from $\mathbf{f}_{rgb}$ to $\mathbf{f}_{hsi}$ is calculated as follows, with hue measured with respect to the red axis:

$$\mathbf{f}_{hsi} = \begin{bmatrix} h \\ s \\ i \end{bmatrix} = \begin{bmatrix} h \\ 1 - \frac{3}{r+g+b}\left[\min(r,g,b)\right] \\ \frac{1}{3}(r+g+b) \end{bmatrix}$$

$$\text{where } h = \begin{cases} 2\pi - \theta \ (b > g) \\ \theta \text{ otherwise} \end{cases},$$

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}\left[(r-g)+(r-b)\right]}{\left[(r-g)^2+(r-b)(g-b)\right]^{\frac{1}{2}}}\right\}$$

It is assumed that the RGB values are normalised in the range $[0,1]$ prior to application in this equation. The HSI space is often used in applications where intuitive colour descriptors are required. Like YUV space the luminance and chrominance is separated in the I and HS channels respectively.When modelling this space it is also important to take into account the cyclic property of the hue (including a discontinuity in the space) and the relationship between the hue and the saturation. One way to achieve this is to convert the hue and saturation to cartesian co-ordinates to form cartesian Hue-Saturation-Intensity (XYI) space [42], the X and Y co-ordinates are given by:

$$x = s\cos h$$

$$y = s\sin h$$

Converting HS to cartesian X and Y caters for the fact that at small saturations (i.e. near the intensity axis) the hue differences are meaningless (since little useful colour can be measured) and that at large saturations the hue value is more relevant (the arc length between hue values on the chromatic plane increase with saturation). In common with RGB and YUV spaces this space is non-uniform with respect to human perception of colour differences.

Uniform non-linear colour spaces

Uniform colour spaces are designed to be *perceptually* uniform — a distance measured in the colour space will be proportional to the subjective difference perceived between the colours by a human observer. The CIE-$L*a*b*$ colour model [180], proposed in 1976, provides an

approximately perceptually uniform colourmetric space in which colours that are perceived to be identical are encoded identically and that colour differences among various hues are perceived uniform. As the gamut of $L*a*b*$ contains the entire visible spectrum it is commonly used as a device independent space to convert colours between different systems. The $L*$ co-ordinate represents the luminance, whilst the $a*$ co-ordinate represents red minus green and $b*$ represents green minus blue. As with HSI and YUV, this space allows the chrominance and luminance information to be separated.

The conversion from RGB to $L*a*b*$ is achieved by first converting to the CIE-XYZ tristimulus values. This allows the CIE spectral primary colour co-ordinate system (P.65, Pratt [141]) to be described in a co-ordinate system where all tristimulus values are positive. The transformation from an RGB feature vector to the CIE-XYZ vector $\mathbf{f}_{xyz}$ is performed as follows:

$$\mathbf{f}_{xyz} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

The conversion from the XYZ tristimulus values to the CIE-$L*a*b*$ vector $\mathbf{f}_{L*a*b*}$ is subsequently performed as:

$$\mathbf{f}_{L*a*b*} = \begin{bmatrix} L* \\ a* \\ b* \end{bmatrix} = \begin{bmatrix} 116\ h\left(\frac{Y}{Y_W}\right) - 16 \\ 500\left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_W}\right)\right] \\ 200\left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{Z_W}\right)\right] \end{bmatrix}$$

$$\text{where } h(q) = \begin{cases} \sqrt[3]{q}\ (q > 0.008856) \\ 7.787q + \frac{16}{116}\ (q \leq 0.008856) \end{cases}$$

where X_W, Y_W and Z_W are reference white tristimulus values that are typically constants. The reference white commonly used in video object segmentation is that observed from a perfectly reflecting diffuser under a CIE standard $D65$ illumninant. $L*a*b*$ is commonly used in video object segmentation to provide an approximately uniform colour space, although the CIE standards on which it is based are usually regarded by professional colorimetrists as out of date [69].

In video object segmentation there is often little justification given for choosing one colour space over another. Extending this to general image segmentation, it is still an

unsolved and open issue as to which colour space will give the 'best' performance with respect to the segmentation criteria. Evaluating the four colour spaces presented in this section will allow one colour space to be chosen in a principled manner as being the 'best' for video object segmentation with respect to an evaluation methodology.

In video sequences where different video objects contain patches of similar colour, colour alone is often not enough to accurately and consistently segment video objects. In such cases spatial information can be appended to the feature vector to add spatial coherence to the video object representation.

3.3.2 Spatial Co-ordinates

As mentioned previously, colour alone is often not complete as a descriptor of a video object. To add spatial coherence to the object model spatial information can be appended to the feature vector. A spatial feature vector is given by:

$$\mathbf{x} = [\ x,\ y\ ]^T$$

where x and y are the spatial co-ordinates of the pixel in the video frame. Adding spatial information into video object segmentation algorithms generally decreases over-segmentation and leads to smoother regions.

3.3.3 Motion

Motion information is commonly used for segmenting moving video frame regions in video sequences. As objects move in the scene relative to the camera motion there is a 2D vector field of motion induced at each pixel in the video frame plane. This motion field is estimated by analysing the spatial and temporal variations of the image intensity within a video sequence.

The many methods for estimating the motion field can be roughly divided into two main classes — *differential* techniques and *feature-based* techniques [164]. Differential techniques produce a *dense* estimate of the motion field based on per pixel analysis of the spatial and temporal variations of the image intensity at each video frame. Feature-based techniques produce a *sparse* estimate of the motion field, restricting the analysis to image points that can be reliably *matched* and then *tracked* through an image sequence. Video object

extraction requires per pixel segmentation accuracy therefore the motion field estimated using a differential technique is preferential for inclusion in the feature space.

There are a multitude of differential techniques available for the estimation of the motion field. The differential technique of Lucas and Kanade [23] is used to estimate the motion field, principally because the gradient information can be reused in other aspects such as the texture measures presented in Section 3.3.4. This method is also applied to video object segmentation by Castagno *et al* [27] and Cavallaro [28] and has the added advantage that it is implemented in the Intel OpenCV image processing library [92, 93]. The technique of Lucas and Kanade is based on the standard differential formulation of the motion field estimation problem. This formulation itself is based on the assumption that the image intensity I is conserved over time:

$$\frac{dI}{dt} = 0 \tag{3.9}$$

where the image intensity $I = I(\mathbf{x}, t)$ is a function of the spatial and temporal co-ordinates at frame t. Using the fact that $\mathbf{x}$ is also a function of t, the differential chain rule can be applied to rewrite (3.9) as:

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \tag{3.10}$$

$\partial I/\partial x$ and $\partial I/\partial y$ are simply the horizontal and vertical components of the image intensity gradient i.e. ∇I, such that:

$$\nabla I = [I_x \; I_y]^T \tag{3.11}$$

Given the gradients of the image intensity ∇I and the motion field $\mathbf{u} = \mathbf{u}(\mathbf{x}, t) = [(dx/dt) \; (dy/dt)]^T = [u \; v]^T$ then (3.10) can be rewritten as the image intensity constancy equation:

$$(\nabla I)^T \mathbf{u} + I_t = 0 \tag{3.12}$$

where I_t denotes $\partial I/\partial t$, the gradient of the image intensity with respect to time. This temporal gradient is computed by considering a temporal window centered in the current frame and extending to the adjacent frames. To estimate the motion field efficiently the temporal window can be extended between the current and next frame. Since (3.12) is an

underconstrained equation with two unknowns further constraints are added by assuming the motion is constant within a local neighbourhood $\mathcal{N}$ to give the estimate of the motion field $\mathbf{u}$ at pixel (x, y) as:

$$\sum_{\mathbf{x}\in\mathcal{N}} \left[(\nabla I)^T \mathbf{u} + I_t\right]^2 \tag{3.13}$$

The approach of Lucas and Kanade [23] modifies this to be a weighted least-squares fit, such that:

$$\sum_{\mathbf{x}\in\mathcal{N}} W^2 \left[(\nabla I)^T \mathbf{u} + I_t\right]^2 \tag{3.14}$$

where $W = G(\mathbf{x}, \sigma)$ denotes a window function that weights the constraints at the centre of the window more than at the periphery. The solution to this least squares problem is found by solving:

$$A^T W^2 A\mathbf{u} = A^T W^2 \mathbf{b} \tag{3.15}$$

where, for the N points in the neighbourhood $\mathcal{N}$ (i.e. $N = \text{card}(\mathcal{N})$) at time t:

$$A = [\nabla I(\mathbf{x}_1), \ldots, \nabla I(\mathbf{x}_N)]^T, \tag{3.16}$$

$$W = \text{diag}[W(\mathbf{x}_1), \ldots, W(\mathbf{x}_N)], \tag{3.17}$$

$$\mathbf{b} = -[I_t(\mathbf{x}_1), \ldots, I_t(\mathbf{x}_N)]^T \tag{3.18}$$

The solution to this overconstrained linear system is therefore given by:

$$\mathbf{u} = \left(A^T W^2 A\right)^{-1} A^T W^2 \mathbf{b} \tag{3.19}$$

This is solved when the *windowed second moment matrix* $M = A^T W^2 A$ is nonsingular:

$$M = A^T W^2 A = \begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_x I_y & \sum W^2 I_y^2 \end{bmatrix} \tag{3.20}$$

Repeating this procedure at all the points in the current frame yields a dense optical flow with a value for $\mathbf{u}$ at each pixel. Dense motion estimation techniques using this approach suffer from the *aperture problem* — the optical flow estimated based on the constraint in

(3.12) only determines the component in the direction of the spatial image gradient i.e. the component *orthogonal* to the spatial image gradient is not constrained. Equation (3.10) is only true for small displacements, the optical flow method applied in this work extends this simple methodology with a laplacian pyramid to allow larger motions to be recovered. The reader is referred to Trucco and Verri [164] or Barron, Fleet and Beauchemin [17] for a more in-depth discussion on motion field estimation in video sequences.

The optical flow recovered can be appended to the feature vector at each video frame pixel. With the addition of this motion information meaningful regions (i.e. video frame regions with homogeneous motion) can be segmented from the video sequence using an unsupervised scheme. However, these regions often do not represent semantic video objects since objects often exhibit articulated or smoothly varying non-translational motion and are therefore inhomogeneous in the motion space. Worse, there are often sequences where a video object exhibits minor motion relative to other scene objects i.e. a static object against a static background. In such a scenario the optical flow will not add any discriminatory evidence.

A major drawback when utilising optical flow for video object segmentation is the fact that the calculation of motion is based on neighbourhoods.Motion discontinuities (i.e. the boundaries of moving objects) cannot be determined accurately when using motion information. However it is hoped that the motion information often complements the colour information present in the scene. Motion information is considered more 'reliable' in textured areas whereas colour information is more 'reliable' in textureless areas, and both features are 'unreliable' at the boundaries of moving video objects in the scene. Therefore when using optical flow in the feature space it is beneficial to use a feature weighting methodology (see Section 3.3.5) such that only 'reliable' feature vectors are used in the segmentation of the video objects.

3.3.4 Texture

Texture is added to the feature space of video object segmentation schemes to aid the discrimination between video objects that exhibit similar colour signals but differ in the way these signals are distributed in localised image regions. Whether constituent elements of images are described as texture or not depends on the scale at which that element is being viewed [69] and also the presence of neighbouring elements of similar size such that

the elements together appear distributed by a repetitive process. For example, in Figure 3.2 the pebble in the image on the left can be described as being an element of an area of texture (the pebbled beach). If the scale at which the textured area is viewed is changed then the interpretation may now be that the image contains textured surfaces on the pebbles, and the pebbles may be interpreted as individual video objects. Finally, the image can be viewed again at the original scale, removing the neighbouring pebbles in the textured area and replacing them by the 'water' texture. In the absence of neigbouring objects of similar size the pebble becomes more distinct and it may be interpreted once more as being a video object element in the scene.

Figure 3.2: Factors affecting texture description. (Left) The highlighted pebble is a constituent element of a textured image region (Middle) Closer up, the image contains textured surfaces on the pebbles, which is now a distinct scene object (Right) Replacing the neighbouring pebbles with a 'water' texture the pebble becomes more distinct at the orginal scale.

Based on these simple observations, the analysis of texture is an ill-posed problem and the *description* of the texture should be performed in a neighbourhood with an appropriate *scale* to the local structure being described. Texture descriptors (i.e. filter operations) are used to find certain properties of the texture e.g. one might look for *textons*[1] of a particular spatial frequency and orientation using a *Gabor* filter. The problem of scale selection (i.e. the finite sized neighbourhood over which the descriptors are integrated) requires that the region within which the texture is described should be large enough such that it is representive of the texture pattern. Carson *et al* [29] used this concept of representing texture in their seminal content based image retrieval (CBIR) work *Blobworld* using texture descriptors and scale selection techniques derived from earlier work by Leung and Malik [107]

[1]Textons are defined as regular subelements of images that form organised patterns.

and Garding and Lindeberg [75].

The texture descriptors applied in this work are based on those proposed by Carson *et al* [29]. The feature vector $\mathbf{w}$ extracted using this texture descriptor contains the *anisotropy* and *normalised texture contrast* which represent the number of dominant directions and 'strength' of the texture respectively for each image pixel. This approach uses the windowed second moment matrix presented in (3.20) and the scale of the texture is determined to be the integration scale of the window function in this procedure and this is determined separately for each pixel in the video frame.

To select the integration scale of the window Carson *et al* measure the *polarity* of the local texture and attempt to find the scale at which this measurement stabilises. This is achieved by first computing the gradient of the image intensity values, ∇I, where, as in (3.11):

$$\nabla I = [I_x,\ I_y]^T \tag{3.21}$$

The windowed second moment matrix, M, is calculated as shown in equation (3.20). The image intensity $I = I(\mathrm{x})$ is a function of the spatial image co-ordinates and the windowed weight function is a smoothing kernel such that $W = G(\mathrm{x}, \sigma)$. Therefore the windowed second moment matrix $M = M(\mathrm{x}, \sigma)$ within a local neighbourhood $\mathcal{N}$ is a function of the spatial co-ordinates and the variance σ^2 such that:

$$M = A^T W^2 A = \sum_{\mathrm{x} \in \mathcal{N}} W^2 (\nabla I)(\nabla I)^T \tag{3.22}$$

Leung and Malik suggest the use the eigenvalues λ_1 and λ_2 (such that $\lambda_1 \geq \lambda_2$) of M to determine the orientation of the texture at a pixel (x, y) for a fixed scale σ. If λ_1 is much larger than λ_2 then the local neighbourhood has a 1-D texture structure, such as an edge or a flow, specified by ϕ, the argument of the principal eigenvector of M. When the two eigenvalues are of neglible size the local neighbourhood is approximately constant, and when the eigenvalues are of comparable size then there is a 2-D texture structure, with no preferred orientation exhibited.

In order to determine the 'best' scale σ (for each pixel i.e. $\sigma = \sigma(\mathrm{x})$) at which M is computed the local image property known as *polarity* can be used; the polarity is a measure of the extent that the gradient vectors in a neighbourhood all point in the same direction.

The scale selection is described in more detail in [29]. Once the 'best' scale σ^* has been selected at a pixel three texture descriptors can be computed:

- The polarity p_{σ^*}
- The anisotropy, $a = 1 - \frac{\lambda_2}{\lambda_1}$
- The normalised texture contrast $c = 2\sqrt{\lambda_1 + \lambda_2}$

The last two measures are taken from the matrix M_{σ^*}. The anisotropy and polarity are modulated by the normalised texture contrast since it is meaningless in regions with low texture contrast. The use of texture analysis in the feature space aims to add discriminatory evidence about the video objects based on the patterned distribution of intensity within localised image regions. The texture descriptors used in the 'Blobworld' work are fairly simple, although they benefit from scale selection such that the finite regions in which they are calculated are chosen to be representive of the underlying texture pattern. In the texture feature space the measure of polarity is not used, since it is generally only large along the edges within the scene and therefore does not add discriminatory evidence in a feature space region descriptor [25].

In the segmentation of video objects the 2-D texture feature space therefore allows grouping across different polarity, orientation and scale of texture, since this information is not contained in the feature space. This measure has advantages over simpler texture descriptors based on the local, windowed, variance (as applied to video object segmentation by Cavallaro [28] and Castagno [27] amongst others) in that it uses the concept of scale selection whereas local variance based approaches commonly predetermine a global size for the window region using empirical means; the 'Blobworld' measure also includes the anisotropy, a descriptor of the orientedness of the underlying texture pattern.

It can be shown that the polarity descriptor is near zero when a neighbourhood contains 1-D bar texture with small gradient variations perpendicular to the dominant orientation axis such that the small variations are approximately symmetrical about the axis. In such a case, the neighbourhood will contain gradient vectors that are predominantly orientated in one of two directions, but this will not be reflected in the measure of polarity, with the possibility of causing an incorrect scale to be chosen. In practice this situation is rare, therefore the computation of the polarity is generally representive of the extent that the gradient vectors in a neighbourhood all point in the same direction.

3.3.5 Associating Confidence with Dimensions in Hybrid Feature Spaces

When motion, colour and spatial features are combined into hybrid feature vectors the knowledge about the reliability of the features can be imparted by weighting the influence of the feature dimensions [18, 154, 27, 28]. In generic video object segmentation schemes that contain motion and colour information the following can be observed:

- Motion information is more reliable in regions with high intensity variance.
- Colour information is the complement of the motion information, and is more accurate in regions of homogeneous colour.
- Neither colour or motion information is accurate at the edges of moving scene objects.
- Motion vectors of video objects are generally inhomogeneous.

To formalise these observations into a feature weighting methodology the work of Simoncelli *et al* [159] is followed. The eigenvalues ($\lambda_1 \geq \lambda_2$) of the windowed second moment matrix, (3.20), are used to determine the reliability of the motion information (i.e. the matrix is well conditioned). Simoncelli *et al* proposed that the sum of these eigenvalues provides a reliability measure for the motion estimates, and that the use of this reliability measure in optical flow computation can reduce the aperture effect. Barron, Fleet and Beauchemin [17] found that the smaller eigenvalue was a more reliable measure of the motion estimates. In their implementation of the Lucas and Kanade algorithm the velocity measured by solving (3.19) is deemed reliable if both eigenvalues are greater than a threshold τ.

Shi and Tomasi [157] also use the eigenvalues to find good features (i.e. textured areas, corners) to track in video sequences. They propose that for the motion to be deemed reliable for an image neighbourhood two requirements should be met — the matrix should be well conditioned and above the noise level for the video frame. The noise requirement implies that the eigenvalues should both be large, whereas the conditioning requirement means that they cannot differ by several orders of magnitude. Relating this to the texture measure in Section 3.3.4 two small eigenvalues represent an area of roughly constant intensity over the windowed area, one large and one small eigenvalue represent a 1-D flow of texture or an edge. Two large eigenvalues represent 'trackable' features — for example, corners or 2-D

texture stucture. Since the eigenvalues are bounded by the maximum allowable intensity value then neither of the eigenvalues can be arbitrarily large. Therefore Shi and Tomasi, like Barron, Fleet and Beauchemin, propose the requirement that the smaller eigenvalue should be larger than some threshold τ. The use of the smaller eigenvalue has the added advantage that it is generally small valued on the edges of video objects since these areas commonly contain 1-D texture flow with the result that the windowed second moment matrix will have one large and one small eigenvalue as per Section 3.3.4.

Incorporating the confidence of feature observations can be a beneficial step when the features are complimentary. To add this confidence into the motion-spatial-colour hybrid feature space (introduced in Section 3.5.1) a weight is assigned to each dimension of each feature vector extracted from the video, this weighting represents the belief that the measurement is reliable. This confidence weighting is subsequently used to bias classification algorithms to favour the more reliable feature vectors, with lower confidence vectors having a reduced influence on the representational model. For the motion and colour components the smaller eigenvalue λ_2 is used to represent the motion reliability confidence w_i^u at a pixel, scaled to the range $\left[0, \frac{1}{2}\right]$. The colour reliability is defined as the complement of this such that $w_i^f = \frac{1}{2} - w_i^u$. The spatial feature information is equally weighted with the combination of motion and colour information such that $w_i^x + w_i^f + w_i^u = 1$ at every pixel, this is similar to the confidence ratio suggested by Heisele *et al* [18].

The texture information is not included in this analysis of weighted hybrid feature spaces since there is little theoretical reasoning for the reliability/importance of this feature relative to the other features implemented. A potential weakness for this confidence measure is that in the edge regions the colour information will generally be given a higher reliability than the motion information where it would be preferential to give both measures a low reliability, this could perhaps be overcome by analysing a colour gradient map to determine the strong edges in the colour image where the colour reliability is reduced.

3.3.6 Pre/Post-processing

Video sequences, like images, can contain extraneous signals (noise) from several sources for example, electrical sensor noise, photographic grain noise or channel errors. The types of noise encountered can be primarily divided into two categories – *additive* noise and *multiplicative* noise. Additive noise includes *impulse* noise (i.e. salt and pepper noise) and

Gaussian noise, whilst multiplicative noise includes variable illumination

To limit the effect of impulse noise a rank-order filter can be used [81], that is, a filter that operates on a neighbourhood, $\mathcal{N}$, of pixels based on a ranking of the pixels. An example of an rank-order filter is the median filter [81]. The median filter replaces the pixel value at the centre of $\mathcal{N}$ with the median value of the N pixels that comprise that neighbourhood. This type of filter is often found in the pre-processing stages of video object segmentation since it forces points with distinct intensities to be more like their neighbours, yet attempts to preserve the edge information, an important consideration. In the case of multi-dimensional data the rank-order filter is applied separately to each dimension.

Filtering can also be applied to the output segmentation in an attempt to 'clean-up' the segmentation by removing artifacts from the masks. One way to achieve this is to apply a median filter to the extracted object masks. In the experiments the effect of both pre- and post-processing operations on the feature space will be demonstrated using a median filter.

In this section the feature spaces intended for video object segmentation evaluation have been presented. In Section 3.4 the evaluation methodology is described, this is used to evaluate these hybrid feature spaces formed from the constituent feature spaces presented in this section.

3.4 Performance Evaluation of Feature Spaces

The aim of this evaluation is to find the feature space that performs the 'best' with respect to some evaluation criterion. For video object segmentation the choice of feature space and its evaluation is non-trivial due to the multimodal nature of the objects themselves. Due to this it is difficult to perform evaluation of the feature spaces without having a representational model of the multimodal aspects of the object appearance in the feature space. For example, modelling the PDF distance between the objects requires that the PDF itself is estimated. To circumvent this difficulty the ground truth mask is used to build a compact representational model in the feature space. This model can subsequently be used to reclassify the image from which the performance of the feature space can be evaluated in terms of the segmentation quality when compared to the ground truth mask. An advantage of this procedure is that the result is more meaningful in terms of the effect it has on the segmentation and that the same evaluation procedure can be applied to other aspects of

the video object segmentation process.

Section 3.4.1 shows the data-sets used for the evaluation. Section 3.4.2 contains an overview of the procedure for the evaluation and shows the representation used as a model of the video object and finally Section 3.4.3 details the performance metric and experiments that will be performed.

3.4.1 Datasets

In any evaluation methodology it is important to use test datasets that are sufficient for the proposed evaluation. The datasets should be chosen such that they are representative of the challenges facing the algorithm. For video object segmentation the main challenges relate to:

1. Video object motion relative to the camera motion and other video objects.
2. The colour and texturedness of video objects.
3. The proximity, quantity and size of video objects (including inter-object occlusion).
4. Video objects entering or leaving the scene.
5. Video objects changing pose relative to the camera (including self-occlusion)

The results of any experiments must be statistically significant. To accomplish this the results of video object segmentation will be evaluated over multiple test sequences that encompass the challenges facing video object segmentation. To allow comparison against previous and future work only standard test sequences are used.

In this evaluation methodology feature spaces are analysed with respect to the effect on video object segmentation quality. Evaluation is consequently performed on individual frames of the test sequences (although motion information is derived from the surrounding frames in the sequence). To quantify the performance of the different feature spaces individual frames were chosen from several well known test sequences. The ten test frames used in the evaluation are shown in Figure 3.3 ('Akiyo' frame 00240) and Figure 3.4 (nine other well known sequences).

To reduce the computational burden of evaluating over so many test sequences and to reduce the ground truth requirement, a sub-region of the video frames is used (i.e. not using

the whole image); an example of such a sub-frame is shown in Figure 3.3 and the sub-frame is marked by a rectangle for each sequence in Figure 3.4. The addition of the sub-frame also, perhaps more importantly, allows the evaluation to be performed locally around video objects. This overcomes one of the major drawbacks in many video object segmentation evaluation methodologies; poor segmentation (with respect to the human visual system) can be reported as accurate when using a frame-wide quantitative measure in the evaluation. This effect is due to the focus of human attention to the boundaries of the objects, in many sequences the proportion of video frame pixels is large compared to the object boundary regions.

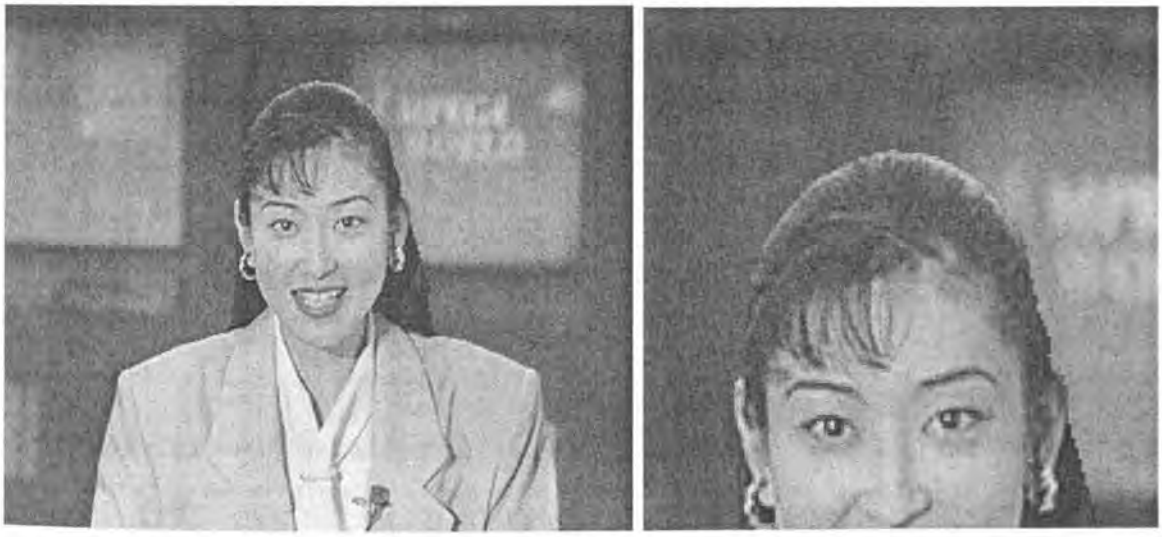


Figure 3.3: An example of a sub-frame extracted from a video frame. (Left) Frame 00240 from the 'Akiyo' sequence (Right) A 150×150 pixel sub-frame. This sub-frame is semantically important due to the perceived boundary between the hair and the dark background.

For each frame, ground truth is required. Video objects are segmented by a human operator, allowing segmentation results to be subsequently evaluated quantitavely against the ground truth. This ground truth takes the form of binary masks delimiting each video object. Binary masks are commonly used in the evaluation of segmentation due to the time consuming process of generating full alpha-mattes. In translucent areas such as hair or motion blur the crisp nature of the binary mask may not be truly representative of the soft video object boundaries. In the longer term recent developments in alpha-matte generation (e.g. Chuang *et al* [35]) could make 'soft' ground truth masks feasible for use in evaluation methodologies. The ground truth segmentation for each sub-frame is shown in Figure 3.5.

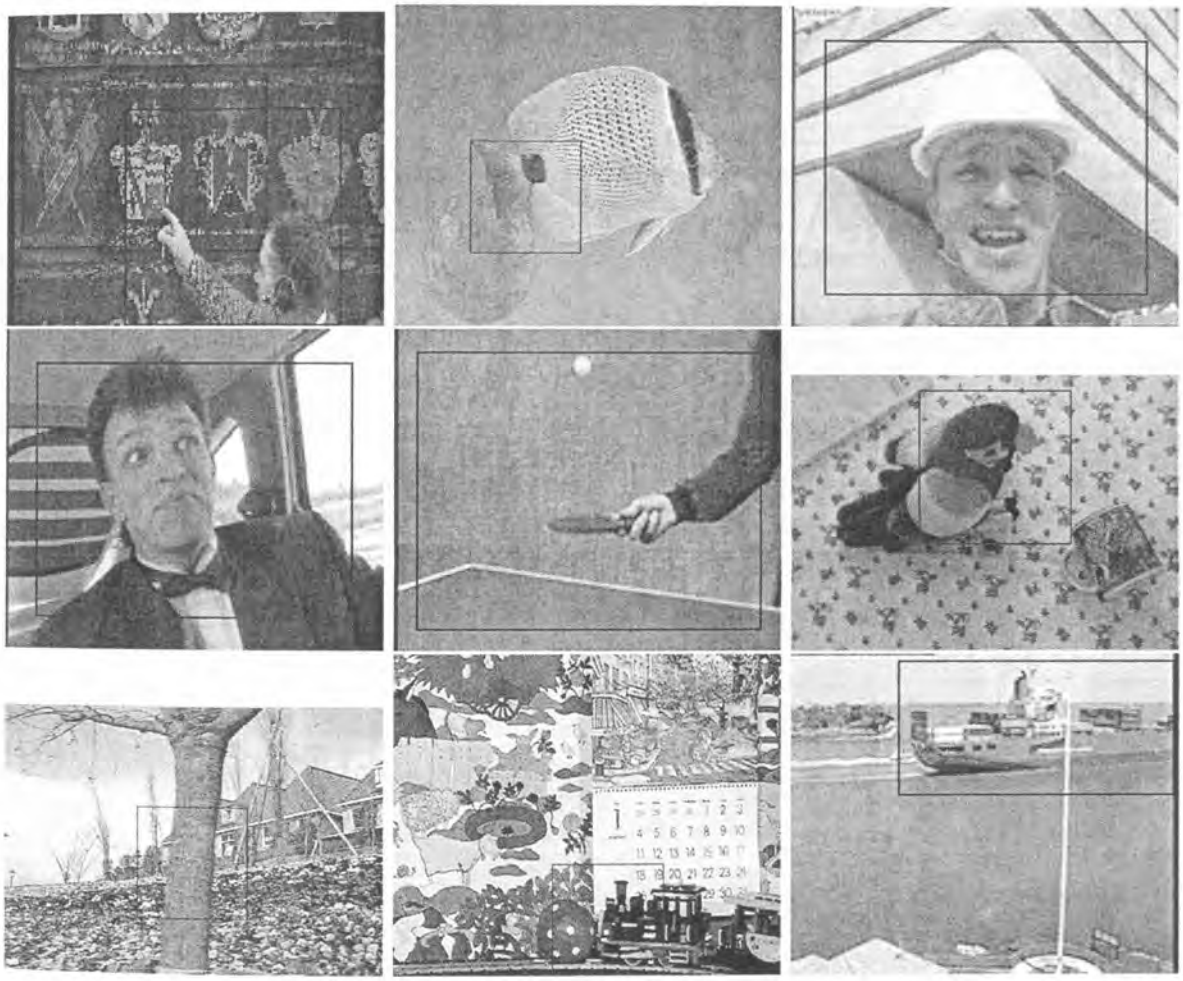


Figure 3.4: Tested video frames with the evaluated sub-frame marked by a rectangle. The frames are (from Top-Left to Bottom-Right) 'Shields' frame 00200, 'Bream' frame 00100, 'Foreman' frame 00021, 'Carphone' frame 00120, 'Ping-Pong' frame 00005, 'Parrot' frame 00010, 'Flower Garden' frame 00010, 'Mobile' frame 00090 and 'Container' frame 00220.

3.4.2 Feature Space Representation

To comparatively evaluate the different feature spaces over the test sequences a feature space representation is required to allow the video object to be modelled in the feature space. There are two main types: *boundary* models which model the extent of an object in the feature space or region based models which attempt to capture the density function of an object in the feature space. The evaluation procedure must be designed such that it does not exhibit bias towards any of the feature space configurations. In this section a cluster based representational form is presented, based around a K-Means clustering algorithm.

The proposed procedure (see Figure 3.6) has three stages — feature extraction, K-

Figure 3.5: Ground truth segmentation for the ten evaluated sub-frames.

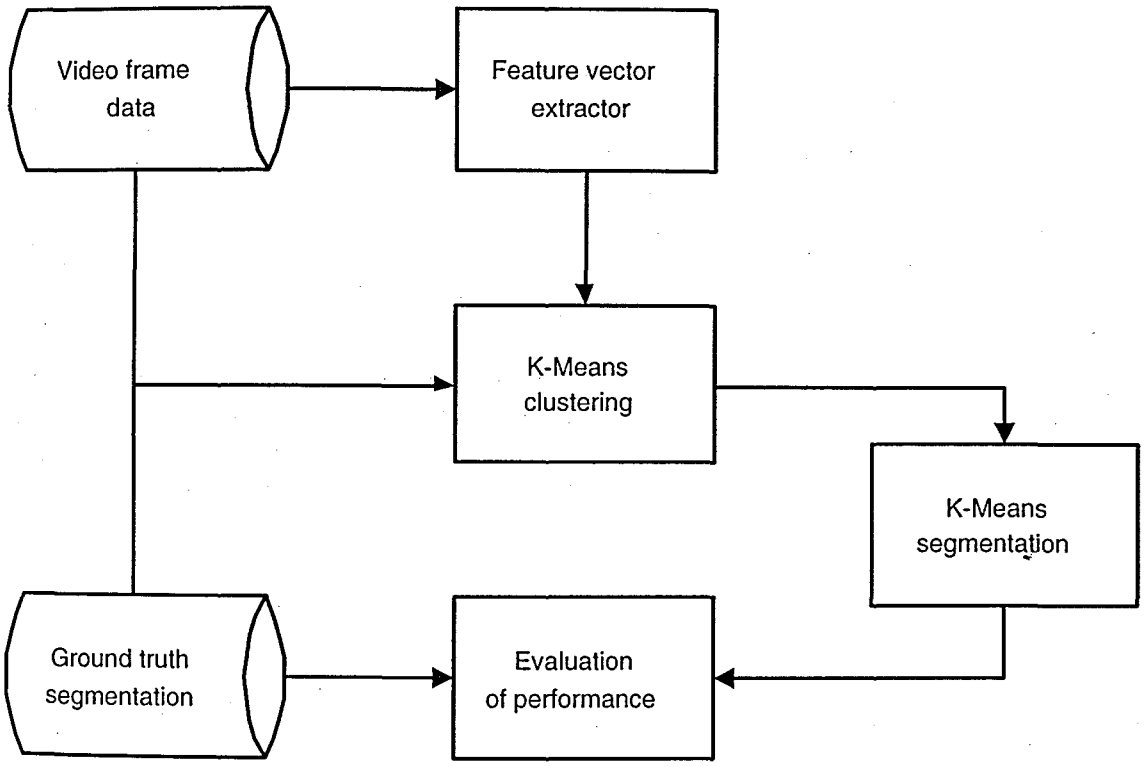


Figure 3.6: The evaluation procedure for a video frame at time t.

Means clustering and performance evaluation. The feature vectors from the video frame are used to form the initial clusters using the K-Means algorithm. The K-Means algorithm is an iterative optimisation procedure that minimises a squared-error criterion function. After applying the K-Means algorthm each pixel in the current frame can be assigned to one of the K clusters which results in a crisp segmentation of the frame. By mapping the K clusters onto the labels of the video objects (as shown in Figure 3.8) a segmentation mask of the video objects can be created. This can be compared quantitively to the ground truth mask. The metrics used to compare masks (and hence evaluate the feature spaces) are presented in Section 3.4.3. In this section the stages in modelling the video objects is described.

Feature Extractor

The feature extractor is the part of the evaluation procedure that takes the video frame and converts the raw data into the various feature spaces described in Section 3.3.

K-Means Clustering

The goal of clustering is to form the data into natural groups using a feature space distance metric as described in Section 3.1. The method of K-Means clustering, like all clustering techniques, employs three distinct procedures [62]:

1. A method for initialising the cluster prototypes.
2. A method for allocating entities to initialising cluster prototypes.
3. A method of reallocating some or all of the entities to other cluster prototypes as part of an optimisation process.

The cluster prototypes are themselves represented by K centroids, denoted by $\mu_1 \ldots \mu_K$, representing the mean vector of the cluster. Partitional clustering methods are designed to minimise an objective function, in the case of K-Means clustering, the objective function is shown in (3.23)

$$J = \sum_{k=1}^{K} \left(\sum_{i, \mathbf{a}_i \in S_k} \text{dist}\left(\mathbf{a}_i, \mu_k\right)^2 \right) \tag{3.23}$$

In this equation, S_k is the partition of the image feature space corresponding to cluster prototype k; J is therefore a sum of squares error for a given cluster prototype, the K-Means algorithm is used to minimise this for all the cluster prototypes within the feature space. To determine the mean vector centroid for a prototype:

$$\mu_k = \frac{\sum_{i=1}^{N} u_{i,k} \mathbf{a}_i}{\sum_{i=1}^{N} u_{i,k}} \tag{3.24}$$

The value of $u_{i,k}$ represents the membership of the feature vector, $\mathbf{a}_i$ to the cluster prototype k — $u_{i,k}$ takes the value 0 or 1 based on this crisp membership. (3.24) is therefore found by setting the gradient of J with respect to each μ_k to zero. Using these definitions, the K-Means algorithm takes the following form:

1. Choose a value for the number of clusters K.
2. Randomly initialise the K prototypes $\{ \mu_1, \ldots, \mu_K \}$.
3. Each feature vector $\mathbf{a}_i$ is assigned to the nearest cluster prototype.

4. End if no change occurs in cluster membership between iteration steps.

5. Re-calculate prototype centroids using membership and repeat from step 3.

When the iterative scheme is complete, K clusters define the partitions of data within the feature space which relate to homogeneous feature vector regions. The homogeneity of the partitions increase as the number of clusters increases and the number of feature vectors assigned to each cluster decreases.

Initialisation of Cluster Prototypes

To allow valid quantitative evaluation of the various feature space components the number of clusters for the K-Means algorithm is chosen such that it does not bias any of the configurations that we want to evaluate. To achieve this uniform sub-sampling of the sub-frame is proposed. This gives the initialisation positions and quantities of the K clusters. The rate of sub-sampling is chosen such that the number of clusters is much greater than that required to represent the video object in the feature space, thus generating an oversegmentation of the current sub-frame. An example of the pixels chosen for sub-sampling are shown in Figure 3.7, again for the 'Akiyo' sequence.

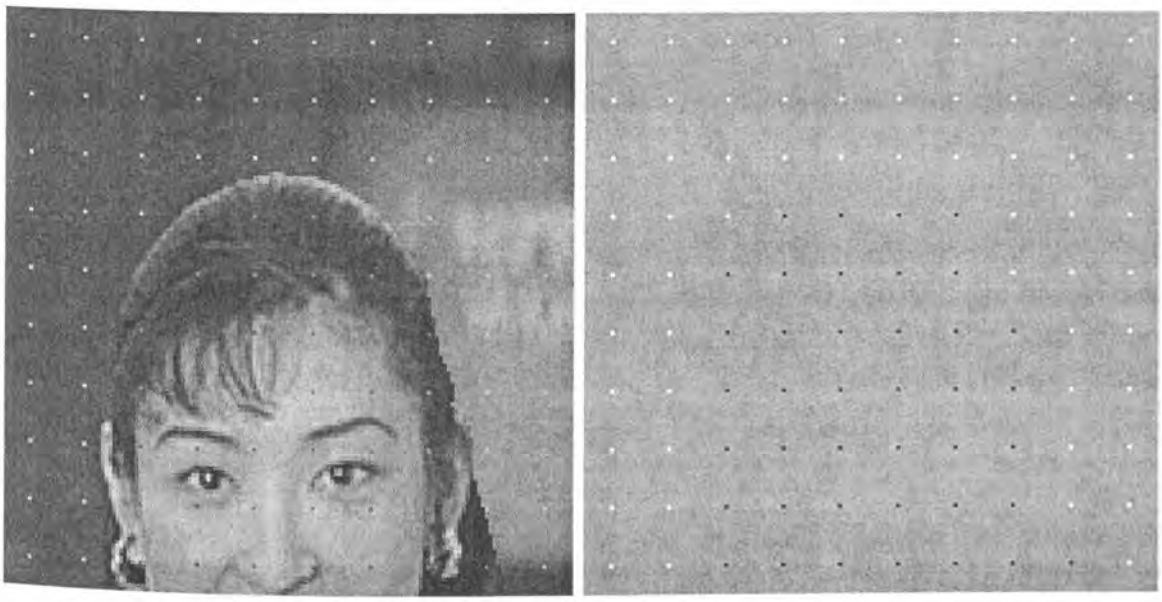


Figure 3.7: Sub-sampled seed pixels for the cluster prototypes. The seed pixels are uniformly sampled at 15 pixel steps, used to form initial seeds for the cluster prototypes for Frame 00240 of the 'Akiyo' sequence.

From the sub-sampled region K seeds are obtained from which the initial cluster prototypes are formed. The k'th cluster centroid mean can be set to the feature vector observed at the prototype seed such that:

$$\mu_k = \mathbf{a}(\mathbf{x}_k) \text{ for } \{0, \ldots, k, \ldots, K\} \tag{3.25}$$

Given that K is much greater than required to represent the feature space the resulting label map after applying the K-Means algorithm will over-segment the current scene, and the segmentation result will be a function of the feature space and distance metric used.

Distance Metric

The distance metric used in the K-Means algorithm is the Mahalanobis distance shown in (3.7). The reasoning for this is that in hybrid feature spaces the features can have different scales and ranges, the Mahalanobis distance provides a way to combine the different dimensional ranges in a principled manner.

For the weighted hybrid feature space (motion-spatial-colour information, Section 3.5.1) the method of Castagno *et al* [27] is followed to incorporate the reliability weighting of the feature dimensions into the clustering algorithm. The weighting strategy was discussed in Section 3.3.5. The Mahalanobis distance is modified to include the per feature weight w when computing the distance between a feature $\mathbf{a}_i$ and a prototype centroid μ_k, such that:

$$\vec{\text{dist}}\,(\mathbf{a}_i, \mu_k) = \left(\sum_{d=1}^{D} w_{i,d} \frac{(a_{i,d} - \mu_{k,d})^2}{\sigma_d^2} \right)^{\frac{1}{2}} \tag{3.26}$$

where $w_{i,d}$ represents the feature weight of the d'th dimension of the feature vector extracted at pixel i. It is clear that this distance function is no longer a symmetrical metric since the weighting implies that it is directed (i.e. the weight used is different if the reverse distance is computed). For the K-Means algorithm (and fuzzy C-Means [27]) this is sufficient since a membership function replaces the reverse distance computation. The feature weighting strategy is used when a hybrid feature space (including colour, spatial and motion information) is applied. For the feature space experiments where the feature weighting strategy is not used the weights are removed from the distance measure.

Segmentation of Objects

To evaluate the performance of the segmentation scheme the segmentation result is compared to a ground truth segmentation for each evaluated feature space. The labels $k \in \{1, \ldots, K\}$ of the clusters are mapped onto those of the ground truth segmentation by using the ground truth label at the initial cluster prototype seed points. An example of this is shown in Figure 3.8 where K region labels are mapped onto binary label values for the ground truth; this mapping allows quantitative measures of discrepancy to be made between the resulting segmentation and the ground truth mask.

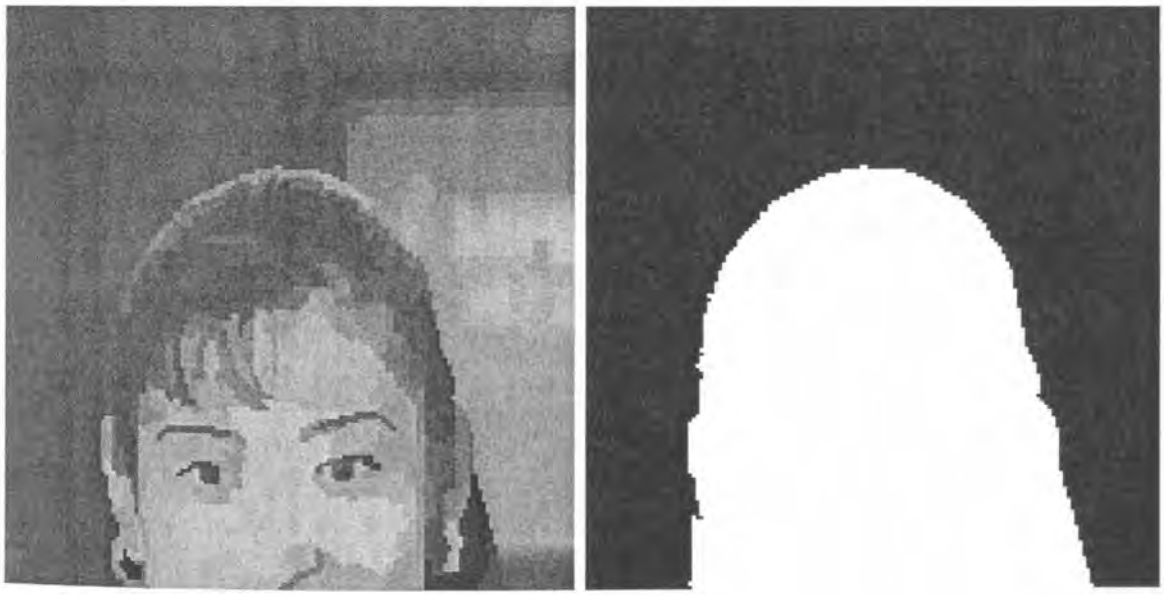


Figure 3.8: Mapping cluster regions to a binary mask. (Left) K mapped cluster regions, represented by their mean colour (Right) The resulting binary mask. This is shown for Frame 00240 of the 'Akiyo' sequence.

3.4.3 Evaluation Metrics

Using the ground truth reference and the algorithm output segmentation the performance of the system is quantitively evaluated by measuring discrepancy. Two measures are proposed — one measuring the quality of the output segmentation for the entire sub-frame and one measuring the quality of the segmentation at the boundaries between the video objects in the sub-frame. To evaluate the performance of the algorithm for a given feature space, the two measures are derived from the work of Villegas *et al* [142], who introduce the ideas

of *spatial accuracy* and *temporal coherency*. It was argued that a simple error measure between a ground truth reference and output segmentation is too limited since it does not take into account perceptual factors in the segmentation quality. It is suggested that boundary errors are more important with respect to segmentation quality. The influence of an error is weighted based on the distance from the edge. Giaccone [79] omits the perceptual factors introduced by Villegas *et al* based on the reasoning that the weights are based on the authors subjective opinion of what constitutes perceptual segmentation quality. Since only single frame segmentation is evaluated the measures of temporal conherency are ignored, concentrating on the spatial accuracy. In both the evaluation metrics the logarithmic signal to noise ratio form of the measures is not used as the evaluation measures of object segmentation should be meaningful to a human operator — signal to noise ratio measures are difficult to quantify and needlessly complicate the evaluation of algorithm performance. The first measure suggested is the *spatial quality of density* (SQD):

$$SQD = \frac{\sum_{r=1}^{R} (N_r - \Delta_r)}{\sum_{r=1}^{R} N_r} \tag{3.27}$$

This is a similar measure to that suggested by Giaccone, extended to be applicable to multiple scene objects. Δ_r is the number of pixels in the output segmentation that belong to an object label, r, but have been incorrectly labelled compared to a ground-truth reference segmentation. N_r is the number of pixels that have object label r in the ground-truth reference segmentation. This measure is therefore the proportion of labels that are correct for an evaluation frame.

Giaccone also suggests an edge based modification to this measure (the spatial quality of the edge, SQE), where the edge is defined as being the video frame regions that are predicted as transitioning from background to foreground (*covered*) or foreground to background (*uncovered*) using the optical flow information at that frame. A problem with this approach is that, in the case of a moving camera, the majority of covered and uncovered pixels tend to be located at the edges of the image, as new, unseen, frame information comes into view. Giaccone fails to take these artefacts into account in the ground truth reference segmentation and therefore in these circumstances the SQE will not be a reliable measure of edge based segmentation quality.

To measure the segmentation quality at the edges of the video objects a Gaussian region is defined, centered on the video object boundary within which the accuracy of the

segmentation is calculated. This approach is preferential to SQE since it does not suffer from the camera motion artifacts and it does not require understanding of optical flow terminology to produce the ground truth, hence, reducing the subjective reasoning required by the human operator when generating the reference frames. This type of measure also allows the evaluation of video object segmentation where the object is stationary in the scene, whereas the SQE measure would undoubtably fail in such a scenario since there would be no covered or uncovered regions located near the boundary of the video object.

The suggested measure, the *spatial quality of edge density* ($SQED$), is separated into three main stages — definition of the *video object boundary*, generation of the *smooth edge region* and finally computation of the $SQED$. First, the boundary of the video object is defined in the analysed sub-frame, this is achieved by applying a morphological filter to extract the boundary from a binary ground truth reference sub-frame image. Given the set $B(\Lambda_r)$ of ground truth reference pixels that encompass the r'th object Λ_r, then the boundary of the video object $\beta(\Lambda_r) = B(\Lambda_r) - [B(\Lambda_r) \ominus M]$ where M is a 3×3 structuring element and $\ominus$ is the morphological erosion operation [81].

This process gives the boundary of the video object as shown in Figure 3.9. The next step is to define the region of the sub-frame that contains the semantically important edge. This edge region contains the edge and an area of the sub-frame in which the edge is prominent (this area also contains edge pixels that are smooth due to motion blur or translucency). The edge region is defined as being a smooth region around the boundary. The smooth edge region for the r'th object is calculated by convolving the extracted object boundary with a Gaussian window, $G(\sigma)$, to form the edge region weight image $E(\Lambda_r)$ such that:

$$E(\Lambda_r) = G(\sigma) * \beta(\Lambda_r) \tag{3.28}$$

where $\beta(\Lambda_r)$ is the binary image containing the r'th object boundary. The final edge region is shown in Figure 3.9. For all scene objects, each having an edge region weight image $E(\Lambda_r)$, ground truth reference $B(\Lambda_r)$ and observed (i.e. output) binary segmentation mask $\tilde{B}(\Lambda_r)$, the spatial quality of edge density can be calculated as follows:

$$SQED = \frac{\sum_{r=1}^{R} \left(N_r^E - \Delta_r^E\right)}{\sum_{r=1}^{R} N_r^E} \tag{3.29}$$

$$\text{where } \Delta_r^E = \sum_{\mathbf{x} \in B(\Lambda_r)} \left[1 - \tilde{B}\left(\Lambda_r, \mathbf{x}\right)\right] B\left(\Lambda_r, \mathbf{x}\right) E\left(\Lambda_r, \mathbf{x}\right)$$

$$\text{and } N_r^E = \sum_{\mathbf{x} \in B(\Lambda_r)} B\left(\Lambda_r, \mathbf{x}\right) E\left(\Lambda_r, \mathbf{x}\right)$$

Comparing (3.29) and (3.27) the $SQED$ is weighted such that errors close to an object edge will be penalised more than errors further away. The SQD and $SQED$ both take values in the range $[0, 1]$ where higher values indicate fewer incorrectly labelled pixels and a value equal to 1 means that the output segmentation is identical to the ground truth reference for the given measure (i.e. scene or edge based). The edge region uncertainty in (3.28) is set empirically to $\sigma = 5$, this defines a meaningful region that encompasses approximately ± 15 pixels around the object boundary.

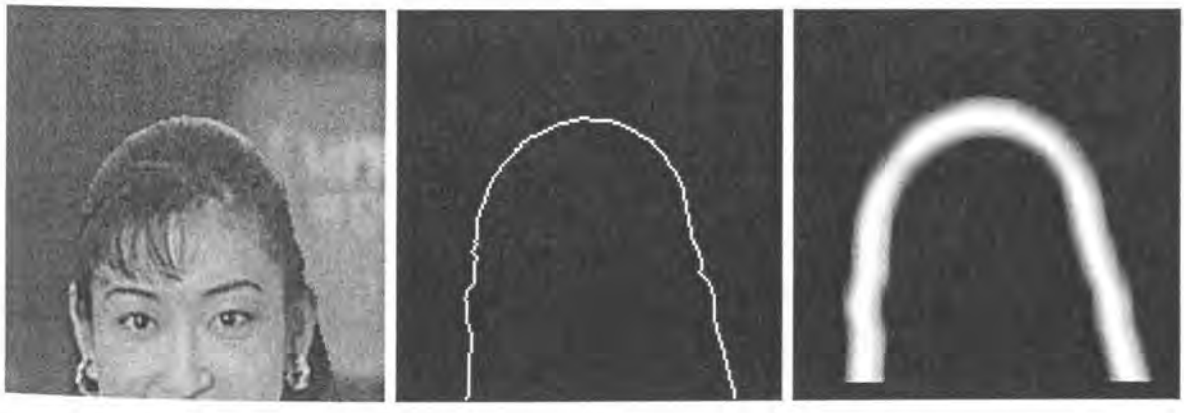


Figure 3.9: Object boundaries and smoothed edge regions extracted from a sub-frame. (Left) The sub-frame for Frame 00240 of the 'Akiyo' sequence (Centre) The extracted object boundary and (Right) The smoothed edge region.

3.5 Evaluating Feature Spaces

Feature spaces are evaluated in the same order as they were presented in Section 3.3. The colour features will be analysed first followed by the appended spatial information, motion and texture information, before looking at weighting and preprocessing the feature space. The feature spaces and symbols used are shown in Table 3.1.

Feature Vector	Space Name	Space Type	Dimensions
f_{rgb}	RGB	Colour	3
f_{yuv}	YUV	Colour	3
f_{hsi}	XYI	Colour	3
f_{L*a*b*}	CIEL*a*b*	Colour	3
x	Cartesian image co-ordinates	Space	2
u	Lucas-Kanade optical flow	Motion	2
w	Blobworld	Texture	2

Table 3.1: Evaluated feature spaces.

The feature spaces shown in Table 3.1 can be combined together to form hybrid spaces. To simplify the experiments all feature combinations are not exhaustively searched. A sub-optimal approach is taken, in which the 'best' colour feature (with respect to the evaluation metric) will give the 'best' performance when combined with other features. The 'best' spatio-chromatic feature space will subsequently be appended with motion and texture information since in generic sequences it is seldom practical to segment objects based solely on texture or motion. An exhaustive analysis of colour and spatio-chromatic feature spaces is used to validate these assumptions.

In this section a methodology has been presented for evaluating the performance of feature spaces for video object segmentation. This methodology implements the K-Means based procedure presented in Section 3.4.2 to evaluate the use of the feature space components presented in Section 3.3 for video object segmentation. The methodology has been designed such that the results are purely a function of the feature space, and the results capture the edge- and scene-based accuracy of the video object segmentation. In the following section the results of this quantitative evaluation are presented alongside qualitative discussion.

3.5.1 Results

The experiments presented in Section 3.5 were completed for the data-sets presented in Section 3.4.1. In this section the results of the experiments are presented, these show that some features can make significant differences to the quality of the output segmentation,

whilst other features may introduce undesirable side effects such as poor edge quality. Section 3.5.1 shows the results for a colour feature space, Section 3.5.1 appends this colour information with spatial information. Section 3.5.1 adds motion information and finally Section 3.5.1 shows the effect of adding texture information. Following these core experiments two further sections are added detailing the effect of the weighting methodology presented in Section 3.3.5 and the preprocessing stage discussed in Section 3.3.6.

Colour feature space

Referring to Table 3.1 in Section 3.4.3, four color spaces were evaluated for video object segmentation — RGB, YUV, XYI and CIE$L*a*b*$. Table 3.2 shows the mean (μ) and standard deviation (σ) of the SQD and $SQED$ for the ten test sequences in the test data.

a	μ_{SQD}	σ_{SQD}	μ_{SQED}	σ_{SQED}
f_{rgb}	0.8841	0.0795	0.7640	0.1069
f_{yuv}	0.9005	0.0702	0.7925	0.0972
f_{hsi}	0.8956	0.0708	0.7557	0.0944
f_{lab}	0.9019	0.0672	0.7931	0.0918

Table 3.2: Colour based feature space SQD and $SQED$ results. Mean (μ) and standard deviation (σ) results are computed over the ten test sequences.

The results demonstrate that all the colour spaces perform reasonably well for video object segmentation based on these scene and edge based error measures. The YUV and CIE$L*a*b*$ show notable improvements in accuracy when compared to the other two colour spaces, with CIE$L*a*b*$ being statistically the most accurate (highest mean) and least variable (lowest standard deviation) of the four colour spaces tested. Figure 3.10 shows the segmentation results for the ten test frames. Comparing these results to the ground truth segmentation shown in Figure 3.5 it is clear that the colour based segmentation lacks spatial coherence and that similar colours on the video objects cannot be differentiated even if they are spatially separated.

The RGB space shows sensitivity to changes in intensity, for example in the foreman sequence (fourth row) the RGB feature space results in poor segmentation of the shaded areas around the foreground persons eyes. Subjectively, the CIE$L*a*b*$ and YUV spaces

appear to give the best edge quality in the segmentation. This subjective analysis is confirmed by the quantitative results, where these two colour spaces are the most accurate in the edge region. Based on these results the 'best' colour space for video object segmentation from the four evaluated is the CIE$L*a*b*$ space, although the advantage between this space and the YUV colour space is marginal.

Spatial-Colour feature space

The lack of spatial coherency demonstrated by using only colour information is rectified by appending the spatial information to the feature vector. Table 3.3 shows a significant increase in the scene wide SQD accuracy compared to Table 3.2; due to the use of spatial information, pixels further away from the foreground video object are no longer assigned based on colour proximity since the colours are well separated spatially. The edge based ($SQED$) accuracy and variability is also improved for all of the colour spaces evaluated by appending the spatial information, again this is due to the spatial coherence given by this extra information, allowing segmentation based on more localised distributions of colour. Figure 3.11 shows the results for all the evaluated spatio-chromatic feature spaces, compared to Figure 3.10 the segmentation of the video objects are subjectively more pleasing, with less distracting artifacts around the borders and inside the video objects. The benefit of spatial information can be seen most clearly for the 'Shields' test frame (second row) where the foreground object is segmentated with improved accuracy compared to the extraction based only on colour.

a	μ_{SQD}	σ_{SQD}	μ_{SQED}	σ_{SQED}
$\left[\mathbf{f}_{rgb}^T, \mathbf{x}^T\right]^T$	0.9426	0.032	0.7977	0.091
$\left[\mathbf{f}_{yuv}^T, \mathbf{x}^T\right]^T$	0.9596	0.0264	0.8425	0.0726
$\left[\mathbf{f}_{hsi}^T, \mathbf{x}^T\right]^T$	0.9474	0.0317	0.8191	0.0734
$\left[\mathbf{f}_{lab}^T, \mathbf{x}^T\right]^T$	0.9574	0.0251	0.8331	0.0708

Table 3.3: Spatial-Colour based feature space SQD and $SQED$ results. Mean (μ) and standard deviation (σ) results are computed over the ten test sequences.

Based on these results the 'best' performing Spatial-Colour based feature spaces are $XYL*a*b*$ and $XYYUV$, mirroring the 'best' colour feature spaces. Again the differences

Figure 3.10: Extracted video objects using (Left-Right) RGB,YUV,XYI and CIE$L^*a^*b^*$ colour spaces.

between the two spaces are somewhat marginal, with $XYYUV$ performing better for both the scene wide and edge based measures, although the $XYL*a*b*$ measure is less variable. From these experiments $XYL*a*b*$ was chosen as the spatio-chromatic feature space to which additional features will be appended for evaluation. The choice of $XYL*a*b*$ is made over $XYYUV$ due to the fact that it is both subjectively more accurate for semantic object segmentation over the range of test sequence frames, and also because it is an approximately perceptually uniform space which may have benefits in subsequent work. It can be seen in some of the segmentation results — specifically, 'Shields' (second row) — that the inclusion of spatial information has highlighted a potential weakness with our evaluation methodology. On the right edge of the foreground object there lies a thin region of background that is not represented by any clusters in the initialisation procedure resulting in arbitrary assignment to foreground. As this problem affects all the feature spaces to different extents, it is anticipated that it does not invalidate any conclusions drawn from the comparative evaluation.

Motion-Spatial-Colour feature space

In this experiment motion information is appended to the $XYL*a*b*$ feature vector. The results show that motion information can both improve and reduce the segmentation quality, depending on the application semantics required. The quantitative results in Table 3.4 show that, compared to Table 3.3, there is a negligible improvement in scene based accuracy and variability, but a reduction in the edge based accuracy and variability. This confirms for video object segmentation the theory that a neighbourhood based operator such as optical flow estimators generally reduce the edge quality when used in the feature space of segmentation schemes.

a	μ_{SQD}	σ_{SQD}	μ_{SQED}	σ_{SQED}
$\left[\mathbf{f}_{lab}^T, \mathbf{x}^T\right]^T$	0.9574	0.0251	0.8331	0.0708
$\left[\mathbf{f}_{lab}^T, \mathbf{x}^T, \mathbf{u}^T\right]^T$	0.9578	0.0215	0.8073	0.0835

Table 3.4: Motion-Spatial-Colour based feature space SQD and $SQED$ results. Mean (μ) and standard deviation (σ) results are computed over the ten test sequences.

These measurements are subjectively reinforced by the resultant segmentations shown

Figure 3.11: Extracted video objects using spatial information, XY, appended to (Left-Right) RGB,YUV,XYI and CIE$L^*a^*b^*$ colour spaces.

in Figure 3.12. The addition of motion information in the feature space improves the differentiation of moving objects where the background and foreground contain similar colours and are spatially close. A good example of this is the 'Foreman' sequence where the background artifacts are reduced with the addition of motion information, creating a more cohesive object segmentation. However, the motion information significantly degrades the edge quality of the video object segmentation. This is demonstrated well by the 'Parrot' and 'Akiyo' results, where the segmentation result has unsightly artifacts due to poor motion estimation at the edges of the video objects (where motion discontinuities occur). It is observed that motion information, whilst perhaps still useful in the inter-frame update scheme for object models, generally adds unsightly artifacts to the segmentation mask that reduce the visual appeal of the segmentation.

Texture-Spatial-Colour feature space

The use of texture appended to the spatio-chromatic feature space also reduces the accuracy of general object segmentation when compared to a spatio-chromatic scheme. Table 3.5 shows a reduction in both the scene and edge based accuracy measures compared to Table 3.3. Subjectively, looking at the results in Figure 3.13, the addition of texture information does not show any significant benefit to the segmentation quality and in many cases reduces the visual appeal of the resultant segmentation.

a	μ_{SQD}	σ_{SQD}	μ_{SQED}	σ_{SQED}
$\left[\mathbf{f}_{lab}^T,\ \mathbf{x}^T\right]^T$	0.9574	0.0251	0.8331	0.0708
$\left[\mathbf{f}_{lab}^T,\ \mathbf{x}^T,\ \mathbf{w}^T\right]^T$	0.9499	0.0301	0.8191	0.0687

Table 3.5: Texture-Spatial-Colour based feature space SQD and $SQED$ results. Mean (μ) and standard deviation (σ) results are computed over the ten test sequences.

To place the results in context the use of texture information was investigated for a sequence containing animals exhibiting natural camouflage. This sequence, 'Leopard' is shown in the middle image in Figure 2.7. In this sequence a leopard walks against a background that has a similar colour distribution to the foreground object, but significantly different texture. As expected, the addition of texture information into the feature space provides improved discrimation between the foreground and background objects. Based on these

Figure 3.12: Extracted video objects using motion information, UV, appended to $XYL^*a^*b^*$ (Right Column), compared to $XYL^*a^*b^*$ spatio-chromatic feature space (Middle Column).

observations the use of this texture measure is not essential in generic video object segmentation, although it can be applied to more specific applications such as the segmentation of animals exhibiting natural camouflage.

Weighted Motion-Spatial-Colour feature space

The Motion-Spatial-Colour space results shown in Section 3.5.1 confirm that the use of motion information in the feature space can degrade the quality of the segmentation since motion is a neighbourhood operation and is not reliable at every point in the image. To counter this a feature weighting methodology was introduced in Section 3.3.5 that modifies the influence of motion and colour information based on a reliability estimate for the optical flow. The feature vectors used in this experiment consist of colour, spatial and motion information, i.e.

$$\mathbf{a} = \left[\mathbf{f}_{lab}^T,\ \mathbf{x}^T,\ \mathbf{u}^T\right]^T$$

Table 3.6 shows that the accuracy of segmentation for the weighted feature space is improved over the unweighted space. These results demonstrate that both the edge based and scene based measures show an increase in accuracy when a weighting methodology is used for the motion and colour information. However by comparing these results to the Spatial-Colour feature spaces (Table 3.3), it can be seen that even with feature weighting the addition of motion into the feature space shows little improvement and that the motion information still degrades the edge quality of the objects to a significant degree.

a	μ_{SQD}	σ_{SQD}	μ_{SQED}	σ_{SQED}
Weighted	0.9586	0.0237	0.8222	0.0745
Unweighted	0.9578	0.0215	0.8073	0.0835

Table 3.6: Weighted Motion-Spatial-Colour based feature space SQD and $SQED$ results. Mean (μ) and standard deviation (σ) results are computed over the ten test sequences.

Figure 3.14 confirms the results shown in the table. Whilst the weighted feature space segmentation results (right column) exhibit desirable improvements over the unweighted space (middle column) there are still noticeable artifacts when compared to the results of the $XYL*a*b*$ space shown in Figure 3.11. These artifacts are most apparent on the

helmet of the foreman (fourth row) and the beak of the parrot (seventh row). Most of the sequences show an improvement in these unsightly artifacts between the unweighted and weighted feature spaces. Based on these observations it is suggested that the use of motion information in the feature space (even when weighted based on reliability) does not significantly improve the results of the segmentation. The motion information also results in significant edge artifacts despite the fact that it can produce more cohesive interior segmentation when the object is exhibiting simple, relatively small, motions in relation to the camera.

Effect of median filtering

To analyse the effect of pre-processing colour information or post-processing the segmentation mask a median filter was applied to the Spatial-Colour feature space. The pre-processing median filter was only applied to the individual channels of the colour information, since motion and textural information are already generated using neighbourhood processes. The median filter used was empirically set to size 5×5, this size was found to be large enough to filter out image noise but small enough to not significantly degrade the appearance of objects in the video frames. Table 3.7 shows the effect of the median filtering. The main advantage is the removal of spurious artifacts in the segmentation mask, which improves the scene-based accuracy for both pre- and post-filtering. The edge-based accuracy for the test frames is reduced slightly by the median filter. This effect is less pronounced for the pre-process filtering.

Operation	μ_{SQD}	σ_{SQD}	μ_{SQED}	σ_{SQED}
Median Pre-Filter	0.9648	0.0205	0.8271	0.0642
Median Post-Filter	0.9683	0.0161	0.8221	0.0641
No Filtering	0.9574	0.0251	0.8331	0.0708

Table 3.7: Median filtered Spatial-Colour based feature space SQD and $SQED$ results. Mean (μ) and standard deviation (σ) results are computed over the ten test sequences.

Looking at the results in Figure 3.15, the use of a median filter does, in most cases, result in a segmentation that is smoother with fewer spurious artifacts in the mask. The pre-processing filtering appears to perform better than the post-processing — although this

is at the expense of having more artifacts in the segmentation masks. To improve the pre-processing filter results a further post-processing step could be applied, for example a connected components based algorithm could be used to filter out artifacts that are not connected to the video objects, followed by a median filter or morphological closing operation (which closes gaps in the contours of the objects). The size of the filter used is an important consideration. Choosing a filter that is ill-matched to the size of video objects sought can result in degraded segmentation quality. An example of this can be seen for the 'Container' sequence (bottom row), where the application of 5×5 median filtering has removed the relatively small, yet important, background region identifying the flagpole object.

The post-process filtering results appear to be subjectively less pleasing than the pre-process —perhaps due to small unconnected regions being smoothed into connected regions by the median filter. Based on these results it is proposed that pre- and post-filtering of segmentation results is a useful tool for generating object masks that appear semantically pleasing to the human visual system. However, the smoothness that this filtering imparts on the final segmentation result can reduce the quality of segmentation at the edges and may remove small objects. The filtering of the output segmentation is preferential to motion information for generating more cohesive object masks. In general the motion information significantly degrades the edges of the extracted objects such that it results in a semantically poor segmentation result.

In this section the results from the experiments to find the 'best' feature spaces for video object segmentation have been presented. It has been demonstrated both quantitatively and subjectively that the choice of feature space can make a significant difference to the quality of the output segmentation. It has been shown that the use of colour and spatial information is essential to accurately locate video objects, and that motion and texture information can degrade the edge quality of the segmentation and in generic sequences often make negligible difference to the output segmentation. The benefits of pre- and post-processing using a median filter have also been shown, this results in a cohesive object segmentation at the expense of edge accuracy, although the edges are not degraded as much as when adding motion information to the feature space.

Figure 3.13: Extracted video objects using texture information, AC, appended to $XYL^*a^*b^*$ (Right Column), compared to $XYL^*a^*b^*$ spatio-chromatic feature space (Middle Column).

Figure 3.14: Extracted video objects using weighted Motion-Spatial-Colour feature space (Right Column) compared to the equivalent unweighted space (Middle Column).

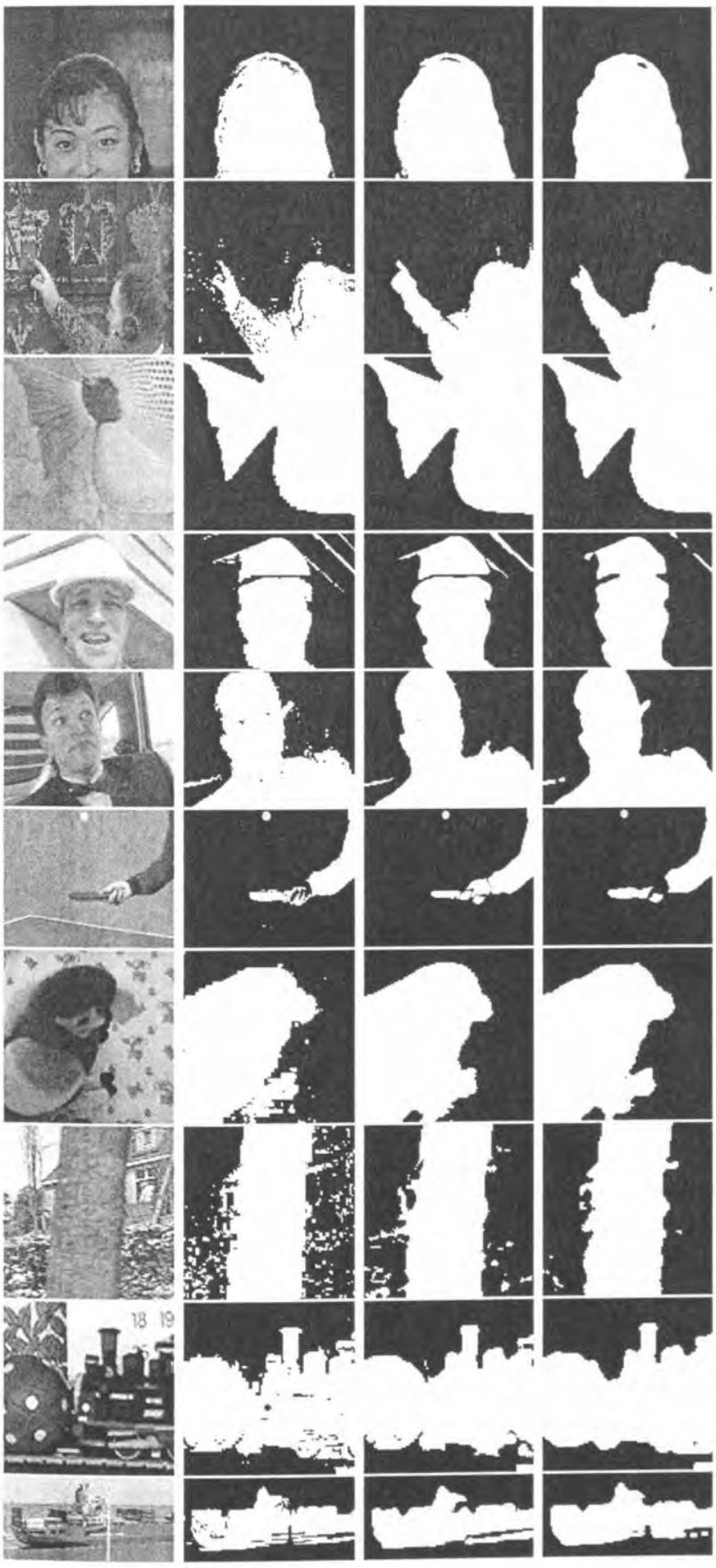


Figure 3.15: Extracted video objects using median filtering as a pre-process (Third Column) and a post-process (Fourth Column) compared to no median filtering (Second Column).

3.6 Conclusions

In this chapter a methodology has been presented for the evaluation of feature space performance for video object segmentation. This methodology implements a K-Means clusterer from which video object segmentation masks can be generated. These masks are a function of the feature space and distance metric used. Two measures for comparing the output segmentation to a ground truth segmentation were presented, and showed over a range of standard data-sets that these two measures are sufficient for describing the performance of a feature space with regards to scene-based and edge-based accuracy.

Using the proposed methodology spatial information appended to a colour feature vector has been demonstrated to be a powerful descriptor allowing a representational scheme to segment an object with sufficient accuracy. The comparison of different colour spaces for object segmentation showed that the YUV and CIE$L*a*b*$ colour spaces show notable improvements in accuracy when compared to the other two colour spaces, with CIE$L*a*b*$ being statistically the most accurate and least variable of the four tested. The addition of spatial information to the colour descriptor was demonstrated to improve the scene and edge based segmentation accuracy for all the colour spaces evaluated. The 'best' performing combined spatial and colour feature spaces were shown to be $XYL*a*b*$ and $XYYUV$, mirroring the 'best' colour feature spaces.

Motion information was shown to be beneficial to the feature space for some scenarios (although the main strength of motion information is perhaps in the inter-frame update scheme for the video object representation). It was shown that weighting the motion information does improve the scene and edge segmentation accuracy compared to unweighted motion, and that it can create more coherent object segmentations if the object is moving. However, the addition of weighted motion information to the spatial-colour feature space was found to decrease the scene and edge segmentation quality.

Texture information was shown to make negligible difference for generic object segmentation and that it is best applied to specific applications. Finally, the effect of pre- and post-processing the data using a median filter was presented. From the results this appears to be a superior method for generating coherent object segmentations since it does not degrade edge quality as much as the use of motion information in the feature space.

In summary, the following contributions have been made in this chapter:

- Proposed an evaluation methodology for video object extraction.
- Proposed two quantitative measures that give both scene and edge based measures of video object segmentation accuracy.
- Proposed and evaluated a feature weighting methodology that is based on the optical flow reliability.
- Evaluated a sufficient range of feature spaces on test sequences using the methodology.
- Demonstrated that spatial and colour information together forms a powerful descriptor for generic video object segmentation.
- Demonstrated that the inclusion of motion information can give more coherent objects at the expense of edge-based segmentation accuracy.
- Demonstrated that feature weighting improves the accuracy of the segmentation when using motion and colour information. However, the inclusion of weighted motion still decreases the segmentation accuracy at the object edges when added to a spatial-colour feature space.
- Demonstrated that the inclusion of texture information is suited to specific applications rather than generic object segmentation.
- Demonstrated that pre/post-processing filters are a superior method to using motion information for improving the cohesiveness of the extracted objects.

Analysing the results gained from the evaluation methodology it can be arguably stated that the assumptions and decisions made in this chapter are valid. Many of the features commonly used in video analysis work have been shown to be either valid or optional for the extraction of video objects, combinations of these features were formed into hybrid feature spaces and evaluated. The chosen data-sets were sufficient for testing the challenges facing generic video object segmentation. The main challenges include variations in the objects motion, colour, texturedness, size, proximity to other objects and quantity; as well as objects entering/exiting the scene or changing pose relative to the camera. The results demonstrated that some features only contribute to the final segmentation in specific applications.

The method for initialising the K-Means based video object representation has a potential weakness in that the initial cluster positions may be too sparse and that minor regions of video objects may be arbitrarily assigned to another object. However, this weakness affects all the feature spaces equally and it is envisaged that the effect is negligible for comparative evaluation if the cluster spacing is chosen such that the affected regions are semantically insignificant. A better solution to this problem would be to find an optimal spacing of the clusters for a given video frame by analysing statistical information about the video objects spatial distribution, or an alternative seeding method (e.g. sampling strategies [163]).

The application of a modelling algorithm in the evaluation scheme was used due to the multimodal nature of the objects. An interesting topic for future work would be to measure how disjoint (e.g. distant) the objects are in the feature space using only the ground truth segmentation and the input image frame i.e. with no modelling step. This could be achieved using, for example, by treating the objects as sets of points in the feature space, between which a metric such as the Hausdorff distance [48] can be computed. Retaining the modelling algorithm, the K-Means algorithm could be replaced by a probabilistic representation. This would allow the distance between the object PDF's to be measured using the Kullback-Leibler distance [51]. An advantage of using the object segmentation in the evaluation procedure is that the results are more representative of the problem. Further work would be required to determine whether the disjointedness of objects in a feature space is representative of how that feature space will perform when applied to the problem of video object segmentation.

The proposed evaluation metrics allowed a quantitative comparison of the feature spaces that was confirmed by subjective analysis of the results. The metrics were also easy to interpret unlike the logarithmic signal to noise ratios [79] where it can be difficult to relate the results to the observed segmentation masks. A more principled manner is required to choose the width of the Gaussian window used to find the edge region. Perhaps plotting the edge-based accuracy against the window size would clarify the nature of the convergence between the edge-based and scene-based measures. A more justifiable method for weighting spatial and texture information in a hybrid feature space is also required — although the suggested spatial weighting appeared to give good results. Further investigation is required to determine a suitable size for the median filters used in the pre- and post- processing stages. The size of this filter may have an optimal value similar to the optimal size of the

Gaussian window used in the edge-based evaluation metric.

The choice of feature space and it's extraction is a fundamental part of the segmentation process for video objects. In this chapter an evaluation methodology has been developed that has enabled a more reasoned choice to be made based on the characteristics exhibited by the hybrid feature spaces presented. It has been shown that the feature space can have a detrimental effect on the accuracy with which the object edges are extracted. As a consequence neighbourhood based descriptors in the feature space will not be used. Whilst the presented results are valid for the feature spaces and data sets shown, it is also important to remember that there are application domains where different features to those shown here may provide more powerful discriminatory evidence for the extraction of semantic video objects. The choice of feature space to be used in the representation of the video objects has been evaluated in a justified manner and the results of this evaluation can be used to determine which feature spaces give the 'best' performance for generic video object segmentation. The following chapter describes the application of probabilistic representational schemes to video object segmentation.

Chapter 4

Probabilistic Representation for Video Object Segmentation

The previous chapter gave an insight into the myriad feature spaces that can be applied to video object segmentation. Several hybrid feature spaces were comparatively evaluated to find the 'best' performing space with regards to scene and edge based quantitative measures of segmentation quality. Within the chosen feature space there exists a partition of the feature vector set $\mathcal{I}$ into the R objects in the current video frame such that:

$$\mathcal{I} = \cup\, \mathcal{I}_r \text{ where } r \;=\; 1, \ldots, R \tag{4.1}$$

Therefore, $\mathcal{I}_r$ is the r'th partition of the set of feature vectors, $\mathcal{I}$, relating to object r in the current frame. The partition $\mathcal{I}_r$ therefore corresponds to a region of the current frame that delimits the semantic object r from the other scene objects.

The sub-set of feature vectors extracted for each object are generated by an unknown process (i.e. the *density function*). If the form of the density function that generates this observed data (either from prior knowledge or by estimation) was known then *decision theory* can be applied to create a *decision function* that minimises a *cost* associated with such a decision. Therefore, a *classifier* consists of two main stages — *density estimation* and a *decision function.*

To apply this to video object segmentation indices $r \;=\; 1, \ldots, R$ are taken to be object category (or class) labels and estimate the true density of the object in the feature space based on feature vector observations. Using an estimate for the density unseen feature

vectors can be assigned to the *most likely* class based on the decision function. For a new feature vector observation $\mathbf{a}$, belonging to object class r^*, the decision function $h_{r^*}(\mathbf{a})$ is chosen such that:

$$h_{r^*}(\mathbf{a}) > h_r(\mathbf{a}) \quad \text{for all } r \neq r^* \tag{4.2}$$

The decision function is therefore maximised over the R possible object labels to find the most probable category label r^* for a feature vector observation $\mathbf{a}$. Using this decision function it is also possible to locate the *decision boundary*, which is a boundary in the decision space such that observed values of $\mathbf{a}$ on the boundary are equally likely for two or more categories.

In generic video object segmentation the distributions of feature vectors are often complex and can overlap between the different classes (since objects can exhibit similar feature patterns), so methods must be employed which minimise the expected error from the classification. The true density functions of video objects can be estimated using either *supervised* or *unsupervised* methods. In the *supervised* case, where a human operator provides training data, the number of classes and initial partition of the feature space into component objects are known *a priori*, in the *unsupervised* case this information is not known and natural groupings of the entire data are found using a *clustering* algorithm.

The type of model used to estimate the density function can be *parametric*, *non-parametric* or *semi-parametric*. In parametric methods a given form for the density function is assumed *a priori* and the parameters of the function are found by fitting this model to the observed data set. In non-parametric methods the functional form of the true density is not specified and is the density estimate is driven directly from the data. Semi-parametric methods specify a functional form for a component part of the true density. A sufficient quantity of component models are fitted to the set of data observations using an iterative optimisation scheme which results in a global estimate of the density. In Section 4.1 an overview of representational schemes applied to the problem of abstracting video into component regions/objects was given.

4.1 Previous Work

A video object, when decomposed into a feature space, can be thought of as a density function that has been generated by some (unknown) process. Classification techniques (an overview of which is given in Section 2.5) can be applied to generate representative models of the video objects, allowing them to be found in subsequent frames by classifying regions of the video frame with the labels of the scene objects. In this section the previous work on the use of classifiers for video based analysis is reviewed.

Clustering techniques have been applied in many object based applications in computer vision. Heisele *et al* [18] showed a method for using crisp K-Means clustering for tracking objects over a series of video frames. They found that clustering within a spatio-chromatic feature space allows robust modelling (and hence tracking) of non-rigid objects without the need for heuristics. Similarly, Schiele [154] implements a method to extract and track homogeneous image regions within a video sequence. This method again uses a crisp K-Means cluster analysis and they find the method is adaptable considering the fact that no *priori* assumptions are made about the data to be modelled. Naturally, clustering techniques have also been applied to image segmentation [131], where the unsupervised grouping of homogeneous image pixels into regions is the main objective. The K-Means algorithm (and variants) have been applied to the problem of video object extraction and spatiotemporal segmentation by [186, 175, 128, 7, 58] where it is used to determine the homogeneous feature space regions within the video data. A natural extension to using the crisp K-Means algorithm for video region extraction is to apply the *fuzzy C-Means* algorithm, which allows a pixel to have a *soft* membership of all the clusters in the model. The fuzzy C-Means algorithm was integrated into a multiple feature video object segmentation scheme by Castagno *et al* [27]. Another extension to the K-Means approach is the K-Means with connectivity constraint algorithm (KMCC/KMC), this was applied by [171, 103, 104] in the context of unsupervised video region segmentation.

A segmentation scheme applying *graph clustering* to perform spatiotemporal region merging was presented by Moscheni *et al* [124]. In such a scheme regions from an *oversegmentation* of the video frame are merged based on affine motion and spatial similarity until some merging threshold is reached. A similar clustering based region merging approach is taken by Dufaux *et al* [63] where regions from an initial luminance-based oversegmentation

(using the K-Means algorithm) are sequentially merged into homogeneous motion regions using a K-Medoids clusterer (essentially a *median* based K-Means algorithm).

Histograms remain a popular technique in computer vision for density estimation and have been applied to aspects of video object segmentation and tracking algorithms [61, 41]. The discontinuities in the histogram density estimate can be reduced by using smoothed Gaussian bins (e.g. [61]). Mezaris *et al* [171] use normalised histograms to perform Bayesian classification of pixels in uncertain regions of the video frame by assigning them to neighbouring (certain) regions to maximise the *a posteriori* probability.

Everingham and Thomas [61] showed a Gaussian kernel shape model to be a valid technique for fine modelling of an object's boundary in a joint distribution with a coarse spatio-chromatic mixture of histograms. The problem associated with separation of the colour and spatial information (i.e. spatially variant colour distributions) was alleviated by including the spatial information into the mixture of histograms. The discontinuities between the histogram bins were smoothed using a Gaussian window centred at each bin. Khan and Shah [98] also applied kernel density models to estimate the spatial distribution of regions of homogeneous motion in the video sequence.

The mean shift algorithm has been applied to the problem of video object segmentation by Hsu and Hsieh [91]. In this approach the prior model for the object in a new frame is determined by iteratively estimating the centroid of the object followed by the membership of pixels to that centroid. A weakness with this approach is that the kernel used is evaluated in a colour and spatial feature space resulting in the centroid of the object being implicitly assumed be a representative colour for the video object. In the case of objects with multimodal colour appearance, it is uncertain whether such a method could be applied. DeMenthon [45] suggests a spatiotemporal segmentation scheme in which a 7-D spatiotemporal chromatic feature space is created; within which hierarchical mean shift clustering is perform to extract homogeneous regions.

Parametric Gaussian distributions are generally used to model regions that display homogeneity in the feature space where the intra-region variation of the feature vectors is Gaussian in the feature space. Fablet *et al* [66] modelled colour regions using a Gaussian distribution within a Markovian framework to provide a coarse to fine segmentation of a video sequence. In addition to using kernel density models Khan and Shah [98] also used Gaussian distributions to model the colour and motion components of video regions.

These regions were initialised using a Gaussian mixture model and subsequently split into individual component Gaussians for each region.

Clustering using Gaussian mixture models for segmentation and tracking of semantic scene objects has been explored in many papers. Raja *et al* [143, 113] and Marlow and Connor [111] showed how a Gaussian mixture model, combined with a Bayesian decision rule, could be applied to the problem of segmentating video objects from generic sequences. Rares and Reinders [11] applied a spatial-colour-texture Gaussian mixture model to the problem of analysing objects in video sequences, with a goal of tracking regions as opposed to pixel level segmentation. Oliver *et al* [130] applied Gaussian mixtures to the problem of face recognition, where they are used to find general descriptors of face and mouth characterisitics which are then used in a Hidden Markov Model [51] to recognise individual expressions. Another common application for Gaussian mixture models is to model per pixel colour or greyscale background observations over a temporal period in the case of a static camera [40, 72]. This model of observed pixel values allows newly observed pixels to be classified as foreground and background using a probabilistic framework. Chalom and Bove Jr. [30] propose a method for choosing the number of clusters based on the relative improvement in entropy between running EM for a range of values of the number of clusters.

Apostoloff and Fitzgibbon [9] take a different approach to the problem of representing and extracting video objects. The principle behind their work is to first find the background layer of the current scene using a layer extraction technique such as that by Wang and Adelson [174] or Irani *et al* [94]. Once the background layer is known *background subtraction* can be used to generate a coarse predicted segmentation, prior models trained on the spatiotemporal gradients of the image sequence and alpha mattes are then applied to regularise the solution. This smoothing requires that the gradients are significant, if not then the alpha-matte solution may be over-smoothed at a boundary between objects. Earlier approaches by Chuang [35] and Ruzon and Tomasi [148] do not require that the entire background be extracted and instead use estimated (local) colour distributions of the background and foreground to recover the alpha matte relationship. In later work Chuang *et al* [34] extended their method using optical flow to propagate the predicted maps throughout the video sequence.

Popular decision rules applied in classification based video object segmentation and tracking work are the MAP rule (e.g. applied in [61, 143, 113, 111, 130, 98, 66, 135, 30, 87,

171, 91, 11, 76]) and the nearest cluster rule (e.g. applied in [103, 104, 186, 128, 154]). For video object extraction the resulting segmentation from such a decision rule may exhibit discontinuities and holes due to noise or subtle changes in the underlying density function. To improve the subjective quality of the masks a post-processing step is often applied such as morphological operators (e.g. applied in [168, 27]), Markov random fields (e.g. applied in [66, 134]) or probabilistic relaxation of the segmentation result [137].

4.2 Probabilistic Representation

In this section the problem of classification using a probabilistic framework is investigated, that is, the density functions are *probability density functions* (PDF's), and the decision function is determined by applying Bayesian decision theory [51]. Bayesian decision theory is based on the assumption that the decision problem is posed in probabilistic terms and that the relevant probability values are completely known. Following this, attention is turned to the problem of estimating the true PDF of observed data that arises when the probabilistic structure is not completely known.

4.2.1 Bayesian Decision Theory

Bayesian decision theory is a fundamental statistical approach that has been applied to the problem of pattern recognition, the two main assumptions on which Bayesian theory are based are that the decision is posed in probabilistic terms and that all the relevant probability values are known. A probability density function obeys the *axioms of probability.*

Let $\mathbf{a}$ be an observed feature vector (i.e. a measurement) generated from some unknown probability density function. Object based video segmentation is an R-category problem (i.e. there are R objects) in which the measured feature vector may be generated by one of R probability density functions. The (joint) probability density of finding an observation that belongs to object r and has a feature vector $\mathbf{a}$ can be written in two ways [51]: $p(\Lambda_r, \mathbf{a}) = P(\Lambda_r|\mathbf{a})\, p(\mathbf{a}) = p(\mathbf{a}|\Lambda_r)\, P(\Lambda_r)$ where Λ_r denotes object r. Rearranging this leads to *Bayes theorem*:

$$P(\Lambda_r|\mathbf{a}) = \frac{p(\mathbf{a}|\Lambda_r)P(\Lambda_r)}{p(\mathbf{a})} = \frac{p(\mathbf{a}|\Lambda_r)P(\Lambda_r)}{\sum_{j=1}^{R} p(\mathbf{a}|\Lambda_j)P(\Lambda_j)} \tag{4.3}$$

The *a posteriori* probability $P(\Lambda_r|\mathbf{a})$ represents the probability that the object category

is r given the feature vector $\mathbf{a}$ has been measured. Bayes theorem provides a method for computing the posterior probability from the *prior* belief — denoted $P(\Lambda_r)$ — that the observed feature belongs to an object category r and the *class-conditional* probability — denoted $p(\mathbf{a}|\Lambda_r)$ — which is the likelihood that the observed feature $\mathbf{a}$ was formed by object r.

For a given feature vector observation the posterior probabilities for each object category can be computed and a decision can be made on the true state of nature (i.e. category) of the pixel from which the observation was made. Due to the fact that the true density functions for the different categories often overlap then this decision will also contain an average probability of error, equal to

$$P(error|\mathbf{a}) = 1 - P(\Lambda_{r^*}|\mathbf{a}) \text{ if we decide category } r^* \tag{4.4}$$

The Bayes decision function seeks to minimise this average probability of error by selecting the category that maximises the posterior probability such that:

$$\text{Decide category } r^* \text{ if } P(\Lambda_{r^*}|\mathbf{a}) > P(\Lambda_r|\mathbf{a}) \text{ for all } r \neq r^* \tag{4.5}$$

This decision function, known as the MAP (Maximum A Posteriori) rule, is formed by assuming that the cost[1] associated with making a decision is zero for a correct decision and equally costly (=1) for any errors[2]. In some applications it may be beneficial to weight some errors as being more costly than others, in which case a different loss function may be used.

Therefore if the true density is known function and the prior probability for each object category, the decision function shown in equation (4.5) can be calculated. This is evaluated at each pixel in the current frame to generate a crisp segmentation, delimiting the objects in the scene.

In video object segmentation (like many computer vision problems) only the observed data at each pixel is known, therefore representational schemes must be applied to *estimate* the density function in the feature space for each video object, allowing the decision function to be computed.

[1] Referred to as loss or risk.

[2] A *zero-one* loss function.

4.2.2 Probabilistic Estimation of Density Functions

Bayesian decision theory assumes that all the relevant probability values are known, which is often not the case. Typically it is neccessary to estimate the true conditional PDF's of the video objects in the feature space based on the available vector observations. Training observations — from which the object PDF's are modelled — may not actually contain the label information for different objects. In this case, natural groupings within the entire set of observations are found using a clustering algorithm. In the supervised case, the number of classes and initial partition of the observation data set into component object sets are known *a priori*.

There are three main strategies available for probabilistic density estimation [21] — parametric, non-parametric or semi-parametric methods.

In parametric methods a functional form for the density model is chosen *a priori*, containing a specific number of parameters that need to be optimised by fitting this model to the observed data sets. A problem often encountered with parametric density estimation is that the form of the model chosen might be incapable of providing a good representation of the true density.

With non-parametric methods the functional form of the model is determined directly from the observed data. In this approach, however, as the number of data points grows then the number of parameters can quickly become unwieldy.

Semi-parametric approaches attempt to take the best features of parametric and non-parametric methods. These methods allow the number of adaptive parameters of parametric models to be increased in a way such that ever more flexible models can be built. In this approach the total number of parameters is independent of the size of the dataset and is instead based on the complexity of the structure of the dataset in the feature space (i.e. how well the functional form of the model can represent the true density).

A problem affecting all approaches to density estimation is that in high dimensions the feature space is often sparsely populated by the observed data sets. This problem — known as the 'curse of dimensionality' — requires the quantity of training data required to grow exponentially with the number of dimensions. A possible solution to this problem is to first find strong correlations between the different dimensions of the data and therefore reduce the data to a lower dimension feature space (e.g. by applying principle component analysis

techniques [51]).

In the following sections the three flavours of probabilistic density estimation techniques are described in the context of modelling observed video object data within a feature space. In the process of density estimation a set of feature vectors, $\mathcal{I}$, are extracted from an initial 'training' video frame. Methods are sought to estimate the probability density functions of the data points in this space that are members of the individual objects within the scene. To achieve this, the feature space is split into the component sub-sets $\mathcal{I}_r$ (where $r = 1, \ldots, R$) and then use the above modelling techniques to estimate the true conditional density $p(\mathbf{a}|\Lambda_r)$ (i.e. the probability that the feature vector observation $\mathbf{a}$ was formed by the true density estimated by the object model Λ_r). With an estimate for each video object's PDF for each new video frame Bayesian decision theory can be applied to classify the new feature vector observations $\mathbf{a}$ into the most likely category r^*, giving the minimum error classification.

4.2.3 Gaussian Density Functions

The simplest and most widely used probabilistic parametric model is the Gaussian (or normal) distribution, which is so frequently applied due to it's suitability and mathematical tractibility. In one dimension the Gaussian probability density function is defined by

$$p(a) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(a-\mu)^2}{2\sigma^2}\right) \tag{4.6}$$

Therefore the distribution of the scalar a is determined by the two parameters of the Gaussian, the mean μ and the standard deviation σ. The Gaussian distribution is 'bell-shaped' with the peak of the bell occuring at $a = \mu$ and the distribution symetrical about this with a width proportional to the standard deviation σ.

The general form for a *multivariate Gaussian density function* (i.e. the multi-dimensional case of (4.6)) is given by

$$p(\mathbf{a}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{a}-\mu)^T \Sigma^{-1} (\mathbf{a}-\mu)\right] \tag{4.7}$$

Where μ is the mean vector parameter of the multivariate Gaussian model, and Σ is the covariance. Maximum likelihood techniques attempt to find the optimum values for these parameters by maximising a likelihood function that is derived from the observed data. Let the set of parameters for the Gaussian distribution be specified by the vector $\boldsymbol{\theta} \equiv \{\mu, \Sigma\}$

such that $p(\mathbf{a}|\boldsymbol{\theta})$ is given by (4.7). If the data set of N vectors $\mathcal{A} \equiv \{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$ are drawn independently from the distribution $p(\mathbf{a}|\boldsymbol{\theta})$ then the joint probability density of the whole data set $\mathcal{A}$ is given by

$$p(\mathcal{A}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\mathbf{a}_n|\boldsymbol{\theta}) \equiv \mathcal{L}(\boldsymbol{\theta}) \tag{4.8}$$

The function $\mathcal{L}$ represents the *likelihood* of the chosen parameters $\boldsymbol{\theta}$ for a (fixed) data set $\mathcal{A}$. The technique of maxumum likelihood parameter selection sets the parameter values by maximising the likelihood $\mathcal{L}(\boldsymbol{\theta})$ i.e. the most *likely* parameters given the observed data $\mathcal{A}$. In practice it is more convenient to consider the negative log likelihood such that

$$-\ln \mathcal{L}(\boldsymbol{\theta}) = -\sum_{n=1}^{N} \ln p(\mathbf{a}_n|\boldsymbol{\theta}) \tag{4.9}$$

Minimising this expression is equivalent to maximising $\mathcal{L}(\boldsymbol{\theta})$ since the negative logarithm is a monotonically decreasing function. In most cases of probability density estimation the optimum $\boldsymbol{\theta}$ will have to be found via an iterative optimisation procedure; in the case of Gaussian densities the maximum likelihood parameters can be found by differentiating (4.9). This involved differentiation leads to the following well known solutions

$$\widehat{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{a}_n \tag{4.10}$$

$$\widehat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{a}_n - \widehat{\mu})(\mathbf{a}_n - \widehat{\mu})^T \tag{4.11}$$

The covariance matrix in (4.11) is a symmetrical matrix and if the dimensions are *uncorrelated* such that they can be treated as *statistically independent* distributions then the covariance matrix is a diagonal matrix.

4.2.4 Kernel Density Functions

In non-parametric density estimation methods the functional form of the model is determined directly from the observed data. A possible direct estimate of the probability density function is [21]:

$$p(\mathbf{a}) \simeq \frac{K}{NV} \tag{4.12}$$

where K is the number of feature vectors out of a possible N that lie within a region $\mathcal{R}$ of volume V. This region (and hence PDF estimate) is localised around a feature vector $\mathbf{a}$.

A strategy for density estimation is to hold the volume constant and determine the number of feature vectors that fall within that volume from the observed data. To formalise this into a method for density estimation, suppose that the region $\mathcal{R}$ is a hypercube with sides length h centered at the feature vector $\mathbf{a}$ such that

$$V = h^D \tag{4.13}$$

Where D is the number of dimensions in the feature space. Therefore, the expression for K, the number of points falling within the region $\mathcal{R}$ can be defined by a *kernel function* (also known as a *parzen window*), $\varphi(\mathbf{u})$, such that

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_d| \leq 1/2; \quad d = 1, \ldots, D \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

hence $\varphi(\mathbf{u})$ represents a unit hypercube centered at the origin. For all training data points $\mathbf{a}_n$, $\varphi((\mathbf{a} - \mathbf{a}_n)/h)$ is equal to unity if the point $\mathbf{a}_n$ falls within a hypercube of side length h centered on $\mathbf{a}$ and zero otherwise. The number of points falling within this hypercube, K, can therefore be defined as

$$K = \sum_{n=1}^{N} \varphi\left(\frac{\mathbf{a} - \mathbf{a}_n}{h}\right) \tag{4.15}$$

Combining (4.15) with (4.12) and (4.13) the probability density function at the point $\mathbf{a}$ is estimated as:

$$\widetilde{p}(\mathbf{a}) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^D} \varphi\left(\frac{\mathbf{a} - \mathbf{a}_n}{h}\right) \tag{4.16}$$

This density estimate can be thought of as the superposition of N cubes of side h, with each cube centered on one of the data points. Due to the use of bins this estimate still has discontinuities, this can be smoothed by assuming a continuous form for the kernel, for example, a Gaussian kernel such that

$$\widetilde{p}(\mathbf{a}) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(2\pi h^2)^{D/2}} \exp\left\{-\frac{\|\mathbf{a} - \mathbf{a}_n\|^2}{2h^2}\right\} \tag{4.17}$$

where $h \equiv \sigma$ i.e. the standard deviation of the Gaussian kernel. To ensure the approximation of the probability density obeys the axioms of probability (i.e. $\tilde{p}(\mathbf{a}) \geq 0$ and $\int \tilde{p}(\mathbf{a})\, d\mathbf{a} = 1$) it is imperative that the kernel function satisfies $\varphi(\mathbf{u}) \geq 0$ and $\int \varphi(\mathbf{u})\, d\mathbf{u} = 1$.

4.2.5 Gaussian Mixture Models

Gaussian mixture models are by far the most popular type of semi-parametric approach to density estimation. To fit a mixture model to a set of observed data methods are required for [67]:

- inferring parameters of the component source models, and
- identifying which source model produced which observation.

Since mixture models are not based on heuristic principles, many of the issues associated with this type of model (e.g. cluster parameters) can be approached in a principled and formal way. In the non-parametric kernel-based approach for estimating the true density function of observed data (Section 4.2.4), the PDF is represented as a linear superposition of kernel functions with individual kernels centered on each data point. Mixture models are formed from a similar combination of parametric functions, except that K, the number of component functions, is a parameter itself that is typically much less than the number of data points N. Using this definition a mixture distribution can be written:

$$p(\mathbf{a}) = \sum_{k=1}^{K} p(\mathbf{a}|\boldsymbol{\theta}_k) P(\boldsymbol{\theta}_k) \tag{4.18}$$

The prior probabilities (or *mixing parameters*) of the component density functions are subject to the probabilistic constraints that $\sum_{k=1}^{K} P(\boldsymbol{\theta}_k) = 1$ and $0 \leq P(\boldsymbol{\theta}_k) \leq 1$; similarly, $\int p(\mathbf{a}|\boldsymbol{\theta}_k) d\mathbf{a} = 1$. Bayes theorem allows us to compute the posterior probability that a component k was responsible for generating the observation $\mathbf{a}$:

$$P(\boldsymbol{\theta}_k|\mathbf{a}) = \frac{p(\mathbf{a}|\boldsymbol{\theta}_k) P(\boldsymbol{\theta}_k)}{p(\mathbf{a})} \tag{4.19}$$

As with all probabilities $\sum_{k=1}^{K} P(\boldsymbol{\theta}_k|\mathbf{a}) = 1$, due to the scale factor $p(\mathbf{a})$. Each component in the mixture density therefore has an associated prior $P(\boldsymbol{\theta}_k)$ and the Gaussian parameters $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$.

The EM algorithm [1] is the usual technique for obtaining (local) maximum likelihood solutions for the mixture parameters. For Gaussian mixture models the convergence properties of the EM algorithm is a well researched topic that has led to many extensions and improvements to the standard algorithm. The generalised version of EM for Gaussian mixture models (with arbitrarily complex covariance matrices) is defined by the following iterative update equations:

1. **E-Step**: Evaluate the posterior probability for every mixture component k.

$$P(\theta_k|\mathbf{a}) = \frac{p(\mathbf{a}|\theta_k)\,P(\theta_k)}{p(\mathbf{a})} \tag{4.20}$$

2. **M-Step**: Update the model parameters to their *new* values using ML decision criterion

$$\mu_k^{new} = \frac{\sum_{n=1}^{N} P^{old}(\theta_k|\mathbf{a}_n)\,\mathbf{a}_n}{\sum_{n=1}^{N} P^{old}(\theta_k|\mathbf{a}_n)} \tag{4.21}$$

$$\Sigma_k^{new} = \frac{\sum_{n=1}^{N} P^{old}(\theta_k|\mathbf{a}_n)\,[\mathbf{a}_n - \mu_k^{new}]\,[\mathbf{a}_n - \mu_k^{new}]^T}{\sum_{n=1}^{N} P^{old}(\theta_k|\mathbf{a}_n)} \tag{4.22}$$

$$P^{new}(\theta_k) = \frac{1}{N}\sum_{n=1}^{N} P^{old}(\theta_k|\mathbf{a}_n) \tag{4.23}$$

3. **Convergence criterion**: Stop when log likelihood is improved by less than some threshold T from one iteration to the next.

Whilst EM is the *de facto* method for fitting mixture models to sets of observed data there still exists three well known problems:

1. *Estimating the number of components*: there exists some *optimal* choice for the number of component density models given the observed data.

2. *Sensitivity to initialisation*: Small groups of points close together can give rise to local minima in the error function.

3. *The boundary of the parameter space*: there exists parameter values for which the likelihood goes to infinity. This occurs when a Gaussian collapses onto a single feature vector such that the variance becomes zero.

In the following sections methods are described for overcoming these problems, although it is outside of the scope of this work to give an exhaustive overview of the different techniques.

Estimating the number of components

There are two main classes of model order selection algorithms for mixture models — *deterministic* and *stochastic* methods [67]. Deterministic methods evaluate a set of candidate models (from K_{min} to K_{max}) within which the optimal value for the number of clusters K^* exists. The number of components K^* is then found by minimising a model selection criterion function $f\left(\widehat{\boldsymbol{\theta}}(K), K\right)$ where $\widehat{\boldsymbol{\theta}}(K)$ is the estimated mixture model containing K clusters. A common form for the criteria is:

$$f\left(\widehat{\boldsymbol{\theta}}(K), K\right) = -\log p\left(\mathcal{A}|\widehat{\boldsymbol{\theta}}(K)\right) + \mathcal{P}(K) \tag{4.24}$$

where $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$ is the observed dataset. $\mathcal{P}(K)$ is a function that increases with K, therefore penalising higher numbers of clusters in the model. A broad review of deterministic methods for model order analysis can be found in McLachlan and Peel [114].

A class of deterministic methods use *information theory* to pose the problem in terms of a *minimum encoding length* criterion (e.g. Minimum Description Length (MDL) [146] or Minimum Message Length(MML) [129]). The idea behind these methods is that a dataset $\mathcal{A}$ that has been generated from a function $p(\mathcal{A}|\boldsymbol{\theta})$ which is then encoded and transmitted. From Shannon theory [38] the shortest code length for $\mathcal{A}$ is given by $-\ln p(\mathcal{A}|\boldsymbol{\theta})$ (measured in *nats*, or bits if the logarithm is base-2). If the model parameters $\boldsymbol{\theta}$ a not known a priori by the receiver then they also have to be transmitted, this leads to a two component message of $Length(\boldsymbol{\theta}, \mathcal{A}) = Length(\boldsymbol{\theta}) + Length(\mathcal{A}|\boldsymbol{\theta})$. The minimum encoding length schemes like MDL and MML find a parameter estimate by minimising the two component message length $Length(\boldsymbol{\theta}, \mathcal{A})$. The main issue with this approach to model order selection is that since the parameters are real valued then the message length required to transmit them is infinite unless the parameters are sent quantised $(\widetilde{\boldsymbol{\theta}})$ to some finite precision. Therefore there is a trade off between having a large $Length\left(\widetilde{\boldsymbol{\theta}}\right)$ (i.e. fine precision) but smaller $Length\left(\mathcal{A}|\widetilde{\boldsymbol{\theta}}\right)$ (i.e. close to shortest code length) or vice versa and it is this that the methods such as MDL and MML seek to formalise.

In the case of MDL applied to Gaussian mixture models, the value for K is chosen such that it minimises:

$$-\ln p\left(\mathcal{A}|\boldsymbol{\theta}\right)+\frac{l_K}{2}\ln N \tag{4.25}$$

where l_K is the number of free parameters needed in a K component mixture model. For Gaussian mixtures with full covariance $l_K=(K-1)+KD+K\left(\frac{1}{2}D\left(D+1\right)\right)$.

Sensitivity to Initialisation

The EM algorithm is highly dependent on initialisation — minor perturbations in the initialisation procedure can result in the algorithm converging to local maxima in the likelihood function. Solutions proposed to solve this method include multiple initialisation followed by sleection of model with highest likelihood, initialisation by clustering algorithms or the addition of split and merge operations to the mixture model optimisation criterion. The *random start* method for initialising mixture models assigns each cluster an uninformative (high entropy) prior value such as $P\left(\boldsymbol{\theta}_k\right)\approx 1/K$. This method has been found to give good performance [114] due to the *self-annealing* [144] properties of the EM algorithm i.e. the EM algorithm with automatically *anneal* without imposing a cooling schedule. The general idea behing annealing algorithms is to force the entropy of the model (i.e. the uncertainty of the covariance matrices) to decrease slowly over a time period to prevent it prematurely settling into a local maxima of the likelihood function.

The boundary of the parameter space

One of the main problems researchers face when implementing the EM algorithm is also one of the most infrequently mentioned in computer vision texts. When fitting a Gaussian mixture with unconstrained covariance matrices to an observed data set one of the component Gaussians can have a prior probability approaching zero and hence the corresponding covariance matrix can be close to being singular. To limit this problem soft constraints can be applied to the covariance matrices to enforce an annealing schedule [101] or to annihilate weak component models from the mixture.

4.3 Performance Evaluation of Video Object Representation

In this section a framework is presented to implement the representational schemes described in this chapter. The framework will be used to demonstrate the characteristics of these representational schemes when applied to the problem of segmenting objects from video frames and video sequences. In Section 4.3.1 the dataset that is used is introduced. In Section 4.3.2 the framework for video object segmentation is introduced. In Section 4.3.3 two different approaches to estimating the PDF of objects in an $XYL*a*b*$ feature space are discussed.

4.3.1 Datasets

Due to the lack of ground truth for many video sequences (a problem that has yet to be resolved by the research community), quantitative analysis is limited to publicly available sequences where ground truth is known for every frame. The representational schemes introduced in this chapter are evaluated for the 'Parrot' sequence (256 × 180, 18 frames, shown in Figure 4.1). This sequence contains a complex, yet colourful, foreground object moving approximately 30 pixels per frame against a cluttered, stationary, background. This sequence is useful for performance evaluation since it demonstrates problems such as the presence of similar colours in the foreground and background and moving foreground object without introducing the problems associated with moving backgrounds and object/scene innovation (i.e. new objects entering the scene etc).

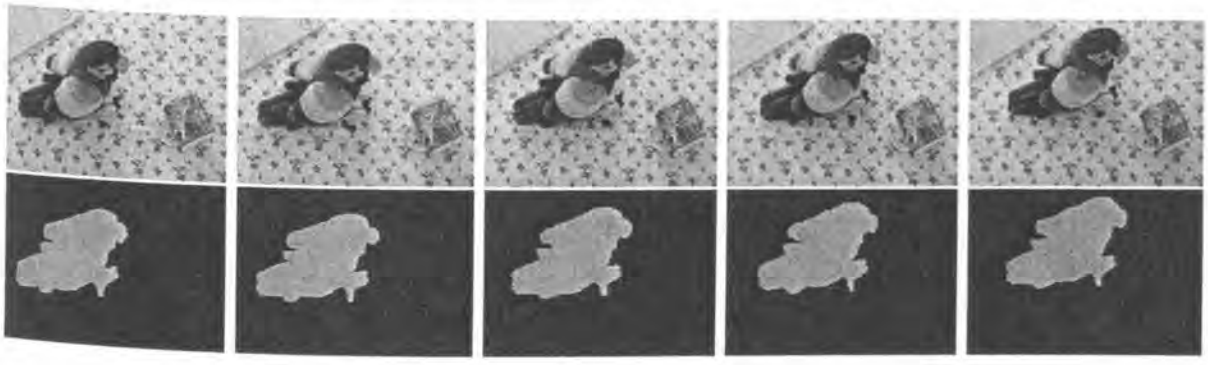


Figure 4.1: Frames with associated ground truth segmentation. (Top-Row) The 'Parrot' sequence Frames 1,5,10,14 and 18 (Bottom-Row) The associated ground truth segmentation.

The associated ground truth for the 'Parrot' sequence is shown in Figure 4.1, this will be used to quantatively evaluate the results of the segmentation using an extension to the

measures presented in Chapter 3. To evaluate the segmentation quality the spatial quality of density SQD and the spatial quality of edge density $SQED$ are computed at each frame (as introduced in Section 3.4.3). These measures provide an quantitative representation of the quality of the segmentation both scene-wide and at the edges of video objects. To characterise the temporal coherency of the object segmentation the variance of the SQD and $SQED$ measures is used (although the variance is meaningless without the accompanying SQD and $SQED$ measures, since a poor segmentation can exhibit frame to frame stability using this metric)

4.3.2 Framework

The framework applied to demonstrate the representational schemes is shown in Figure 4.2, this framework is based on the generic framework presented in Section 2.1. This framework consists of per object PDF models in an $XYL*a*b*$ feature space that are updated using a simple strategy. The inter-frame prediction is a 'null' step in that the object models from the previous frame are used — unchanged — as the estimated models in the current frame (assuming minor object motion between frames). The intra-frame update step consists of reinitialising the object models to the newly found regions of support for that object in the current frame. At each frame the 'final' segmentation is extracted by applying the MAP rule presented in equation (4.5).

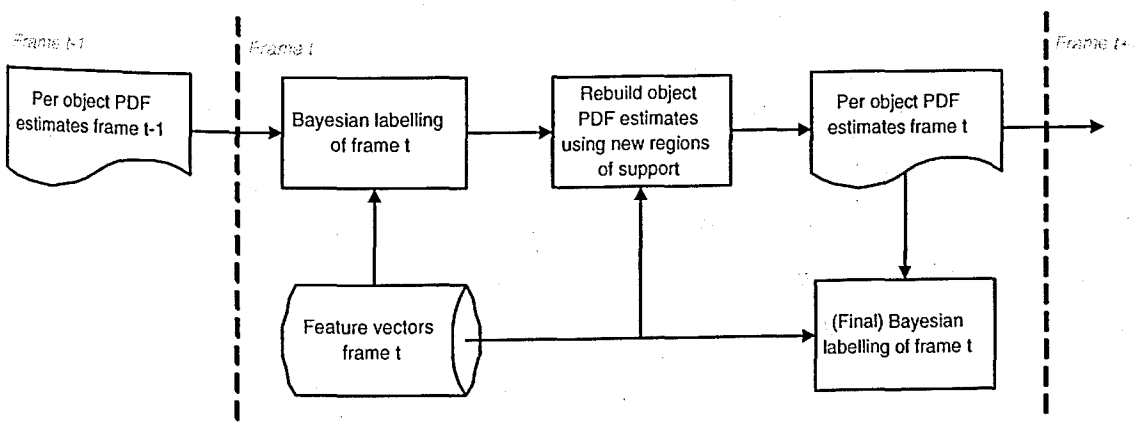


Figure 4.2: The framework used to demonstrate the representational schemes for video object segmentation.

Using this framework the characteristics of the representational schemes can be compared. The process shown in Figure 4.2 is initialised using the strategy presented Figure 4.3;

this is essentially the same as the intra-frame update procedure except that the initial object based segmentation is provided by the supervised user input (detailed in Section 2.2).

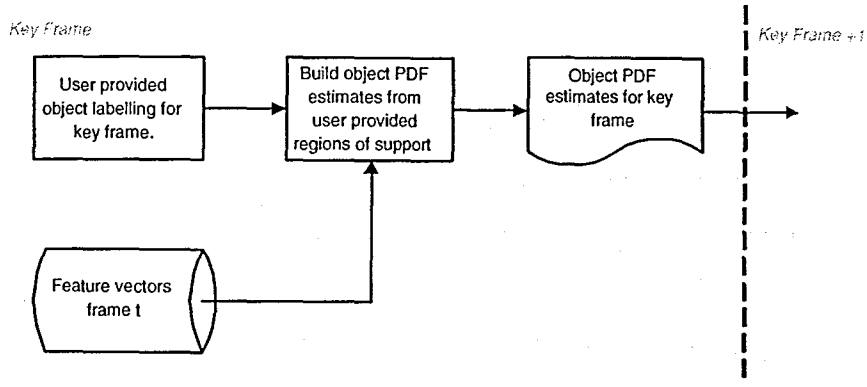


Figure 4.3: The key-frame based initialisation procedure for the representational models.

4.3.3 Independent Feature Space Representation

Applying density estimation techniques to the problem of video object segmentation imposes specific requirements such as whether the functional form of the estimated density gives a good representation of the true PDF; the ease of integration of the model into video object segmentation algorithms; the scalability of the model to higher dimensional spaces; and the computational complexity of the model.

The chosen feature space determines the nature of the true PDF of the video objects. In the spatial-colour $XYL*a*b*$ space the colour and spatial features can be combined into a hybrid feature space such that a single density estimate is made for the spatial-colour space. Alternatively, the problem of density estimation can be simiplified by assuming that the correlation between the colour and spatial information is neglible. Separating the feature space in such a way reduces the effect of the curse of dimensionality and allows for computationally cheaper algorithms, although this may reduce the generality of the model, especially when intra-feature type correlation is present.

Under the assumption of independence the PDF of the $\mathbf{a} = XYL*a*b*$ feature space can be estimated by independently estimating the $\mathbf{x} = XY$ and $\mathbf{f} = L*a*b*$ distributions such that:

$$p(\mathbf{a}) = p(\mathbf{f}, \mathbf{x}) = p(\mathbf{f})\, p(\mathbf{x}) \tag{4.26}$$

The advantage of modelling the data in this way is that different forms for the PDF estimate can be used, although generality will be lost and be unable to model spatially variation in the colour distribution (Everingham and Thomas [61] circumvented this problem using an explicit coarse spatial component in a colour/texture model).

The assumption of independence between the features is explored for the foreground object in the 'Parrot' sequence. For this object $XYL*a*b*$ feature vectors are extracted from which the correlation matrix between the spatial and chromatic information can be estimated.

To determine the correlation between the spatial and colour distributions the correlation coefficient is computed:

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}} \tag{4.27}$$

where σ_{ij} is element (i, j) taken from the covariance matrix Σ and ρ_{ij} is similarly an element of the correlation matrix $\boldsymbol{\rho}$. The correlation matrix is therefore equal to:

$$\boldsymbol{\rho}_r = \begin{bmatrix} \rho_{XX} & \rho_{XY} & \rho_{XL*} & \rho_{Xa*} & \rho_{Xb*} \\ \rho_{YX} & \rho_{YY} & \rho_{YL*} & \rho_{Ya*} & \rho_{Yb*} \\ \rho_{L*X} & \rho_{L*Y} & \rho_{L*L*} & \rho_{L*a*} & \rho_{L*b*} \\ \rho_{a*X} & \rho_{a*Y} & \rho_{a*L*} & \rho_{a*a*} & \rho_{a*b*} \\ \rho_{b*X} & \rho_{b*Y} & \rho_{b*L*} & \rho_{b*a*} & \rho_{b*b*} \end{bmatrix} \tag{4.28}$$

For the 'Parrot' sequence presented in Section 4.3, the covariance of the 5-D $XYL*a*b*$ feature space over the foreground object is given by

$$\Sigma_r = \begin{bmatrix} 915.4607 & -331.6967 & 45.1229 & 76.7905 & 183.7702 \\ -331.6967 & 563.5937 & -12.4783 & -113.1061 & -37.7940 \\ 45.1229 & -12.4783 & 316.0589 & -60.7345 & 196.4601 \\ 76.7905 & -113.1061 & -60.7345 & 215.7918 & 11.0012 \\ 183.7702 & -37.7940 & 196.4601 & 11.0012 & 426.7657 \end{bmatrix} \tag{4.29}$$

The correlation matrix computed from the covariance matrix is thus

$$\rho_{Parrot} = \begin{bmatrix} 1.0 & -0.4617 & 0.0838 & 0.1727 & 0.2940 \\ -0.4617 & 1.0 & -0.0295 & -0.3243 & -0.0770 \\ 0.0838 & -0.0295 & 1.0 & -0.2325 & 0.5349 \\ 0.1727 & -0.3243 & -0.2325 & 1.0 & 0.0362 \\ 0.2940 & -0.0770 & 0.5349 & 0.0362 & 1.0 \end{bmatrix} \tag{4.30}$$

From this matrix it can be seen that, even at a 'global' level, there exists correlations between the colour and spatial distributions. The strongest (anti-)correlation between the spatial and colour information occurs between the Y and $a*$ feature values. If the Y and $a*$ variables are plotted (see Figure 4.4) it can be seen that there indeed exists this correlation, although the data also exhibits local correlations due to the multimodal nature of the distribution.

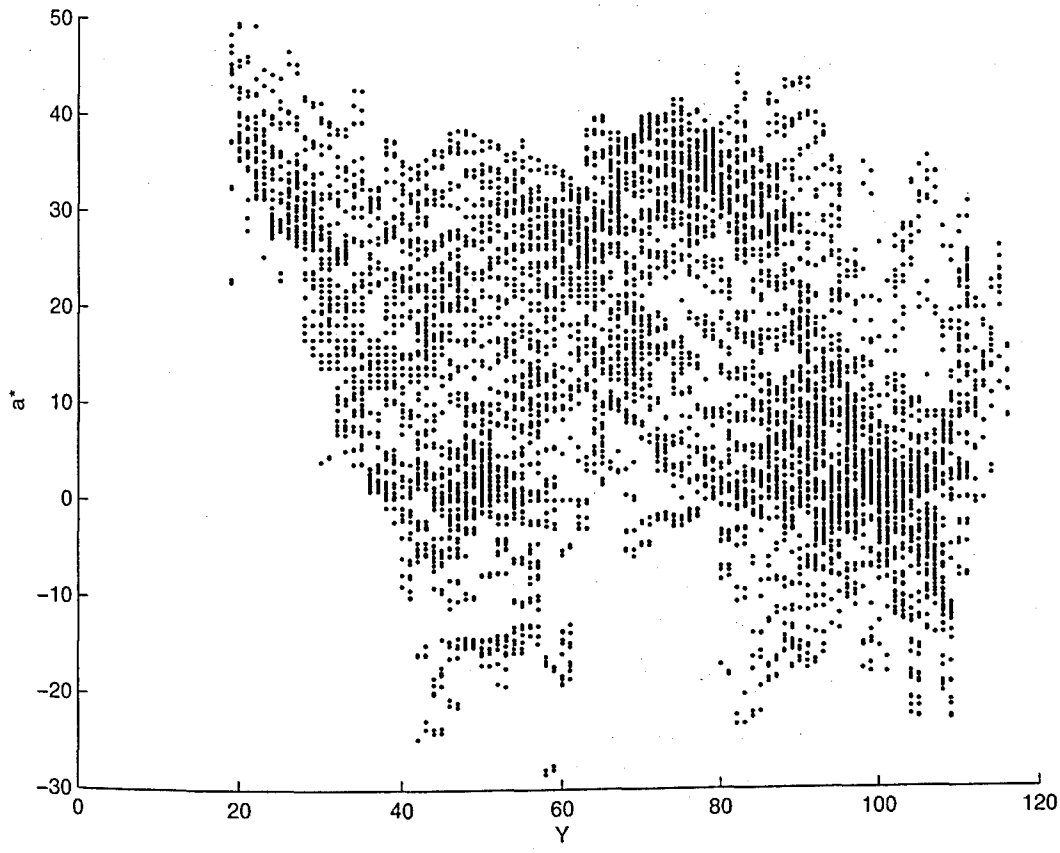


Figure 4.4: Y and $a*$ values plotted for the foreground object. An (anti-)correlation can be observed between the two features.

Not surprisingly, the assumption of independence between the spatial and colour infor-

mation in a video frame is likely to impact the segmentation accuracy since the correlation between the features is not explicitly modelled. There are many factors that affect the spatial and colour information observed over a video frame (e.g. camera, illuminant and object properties). The spatial and colour information observed for a given image region of finite size can only truly be viewed as independent if the region itself is homogeneous in colour from all viewing positions — as such the colour observed is independent from all factors that affect the 2-D shape. In generic video sequences intra-feature correlation is expected — homogeneous regions in the feature generally exhibit spatially and temporally varying appearance.

4.4 Results

In this section the three types of representational models presented in Sections 4.2.3, 4.2.4 and 4.2.5 are used to estimate the spatial-colour probability density function of objects in video frames and video sequences. In Section 4.4.1 and Section 4.4.2 the PDF of a video object is estimated in spatial and colour feature spaces to demonstrate the characteristics of the representational schemes. In Section 4.4.3 the framework presented in Section 4.3.2 is used to perform evaluation of the segmentation accuracy when using the different representational models. Finally, Section 4.4.4 repeats the evaluation with an independent representation of the spatial-colour feature space.

4.4.1 Spatial PDF Estimation in a Video Frame

The spatial distribution of a video object is often stored as a binary mask. An example of this is shown in Figure 4.5 for the foreground object in the 'Parrot' sequence. The first step in estimating the probability density of this object in the spatial feature space is to first assume that the spatial measurements are continuous and as such form a continuous probability density to be estimated. The high resolution of our video ensures that the quantised nature of the spatial samples is negligible.

It is assumed that the mask in Figure 4.5 has been generated by some unknown probability density function from which observations of the spatial information at each foreground pixel are generated. Therefore at each pixel a feature vector containing the spatial information is extracted i.e.

Figure 4.5: Distribution of spatial features extracted from the video object. (Left) 'Parrot' sequence Frame 00010 (Middle) The foreground object extracted as a binary mask (Right) The spatial feature vector distribution.

$$\mathbf{a} = \mathbf{x} = [\ x,\ y\ ]^T \tag{4.31}$$

First, a parametric Gaussian density model is fitted to the observed data for the foreground object. A map of the resulting class conditional probabilities can be seen in Figure 4.6; it can be clearly seen that the parametric form does not accurately describe the spatial distribution of the objects, although it does provide a unimodal descriptor for the object scale and position in the image, which may be useful for higher level object description e.g. in content-based image retrieval applications.

The spatial modelling of the foreground object is much improved by the use of the kernel density model (again shown in Figure 4.6), which in the 2-D form is equivalent to the binary image convolved with a Gaussian mask. This form of model captures the uncertainty of labelling around the edges of the video objects. The non-parametric form of the kernel density model appears to adequately represent the complex spatial PDF of the foreground video object.

Modelling the observed data as a Gaussian mixture density (applying the minimum message length based EM variant proposed by Figueiredo and Jain [67]) allows a more compact representation of the spatial distribution of the video object than the kernel density model (i.e. the model contains less parameters), although this is achieved at the expense of reduced modelling capability of the more complex spatial observations that are not well described by a parameterised mixture model. The conditional probability map is shown in Figure 4.6. The component Gaussian densities can clearly be seen in this image and whilst the number of component densities appears high (K=23), the overall (estimated)

PDF represents the mask efficiently with $K << N$.

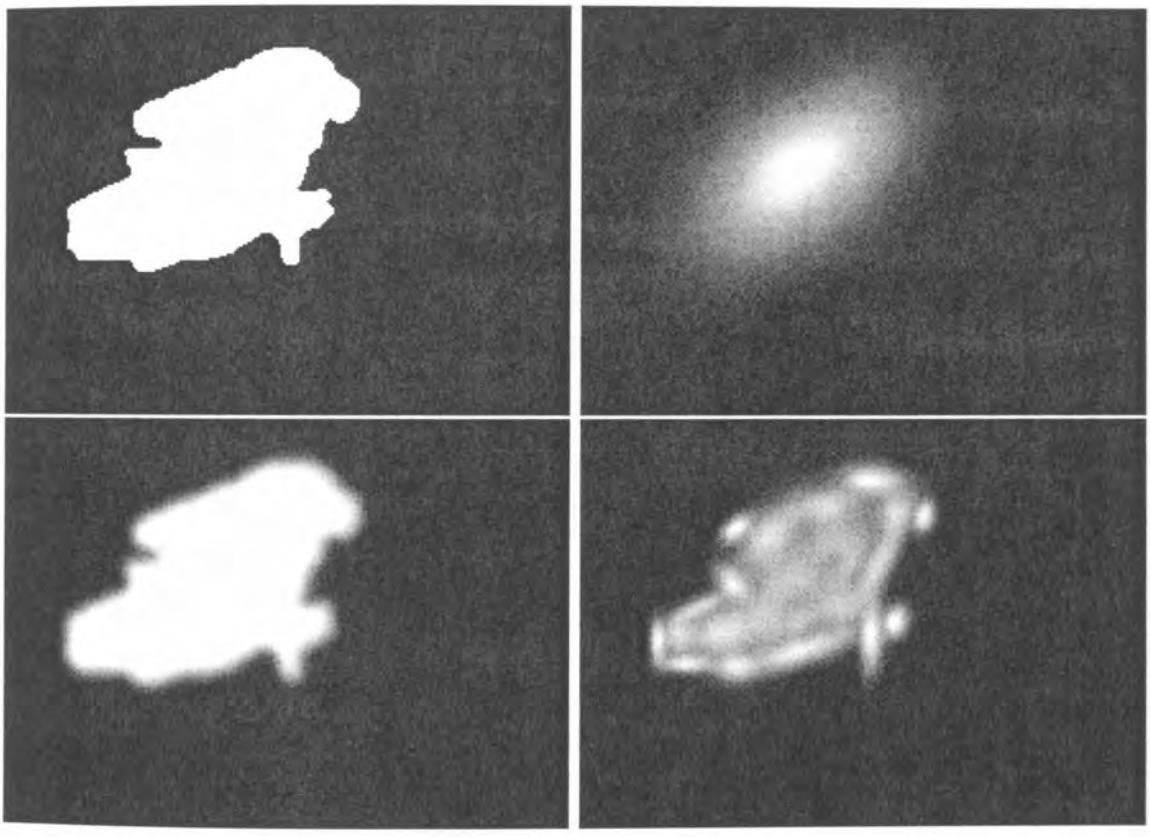


Figure 4.6: The conditional probability maps for the foreground video object spatial observations. (Top Left) Foreground object spatial observations for Frame 00010 from the 'Parrot' test sequence (Top Right) Modelled as a Gaussian distribution (Bottom Left) Modelled as a kernel density distribution (Bottom Right) Modelled as a Gaussian mixture model.

4.4.2 Colour PDF Estimation in a Video Frame

In the previous section the concepts of PDF estimation were applied to object shape representation. In this section the same methods are applied to model the colour information obeserved for the parrot object, shown in Figure 4.5. The observed feature vectors contain $L*a*b*$ colour information, i.e.

$$\mathbf{a} = \mathbf{f}_{L*a*b*} = [\ L*,\ a*,\ b*\ ]^T \tag{4.32}$$

It can be seen in Figure 4.7 that colour observations taken from the foreground object form clouds in the 3-D feature space; the distribution of these feature vectors are generally

multimodal and have a higher intra-class variance. A fundamental difference between colour and spatial distributions of video objects is that the colour distributions for different objects in the same scene can intersect in the feature space whereas the spatial distributions do not generally exhibit this property (except for quantisation errors at the object edges and translucent object regions). This effect will increase the Bayes error rate for colour based modelling compared to spatial based modelling.

A video object can also contain multiple pixels having the same extracted feature vector values. This effect will generate dense non-Gaussian regions in the PDF of the feature space, and commonly occurs when the observed colour is at the limits of the cameras sensitivity range. The dense regions in the PDF can be reduced by limiting the maximum number of observations that can be made for each colour value. In this work it is assumed that the quantity of pixels having the same colour value is small compared to the size of the image and hence this effect is negligible.

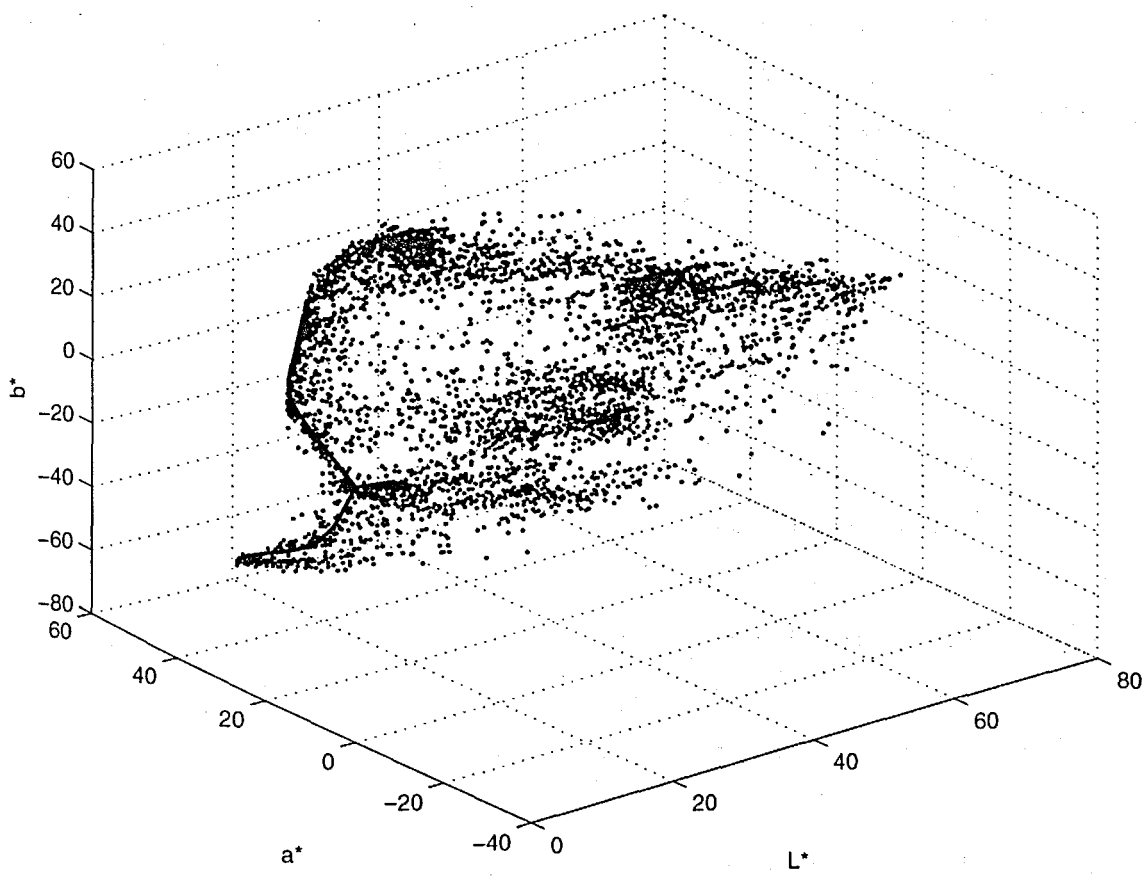


Figure 4.7: The $L*a*b*$ colour space distribution for the foreground video object. The object is extracted from Frame 00010 of the 'Parrot' sequence.

To visualise the modelling of the chromatic information models are built for both the foreground and background object for Frame 00010 of the 'Parrot' sequence. At each pixel in the video frame the posterior probability of the foreground object is computed using equation (4.3). Figure 4.8 shows the posterior probability maps for the three different techniques applied to estimate the object PDF's.

Modelling the (multimodal) foreground object using a single trivariate Gaussian distribution results in a reasonable segmentation quality. It is believed that this is due to the separation between the colour distributions of the two video objects, although errors are evident in the yellow and grey regions of the foreground object and the blue regions of the background object. Using kernel density models and Gaussian mixture models improves the discrimination between foreground and background colours, although there are still noticeable deviations from the expected posteriors due to the inevitable colour similarity between some foreground and background regions (e.g. the Parrot objects beak).

It can be seen that the three types of representational schemes all perform adequately at modelling the foreground and background colours in this video frame, and that using colour alone the Parrot object can be located and tracked in the video sequence. In the next section the application of the models to video object segmentation is demonstrated by performing spatial-colour feature space based video object extraction for the full 'Parrot' sequence on a per frame basis.

4.4.3 Spatial-Colour PDF Estimation in a Video Sequence

The performance of the PDF estimation techniques within the proposed framework was evaluated by segmenting the "Parrot" test sequence into foreground and background video objects. The first step of this evaluation was to find the 'best' performing value of σ (the uncertainty in the PDF representation) to be used for the kernel density model. To determine the value of σ it was varied in the range $0.5 - 9$ and for each value the average SQD and $SQED$ measures over the test sequence was computed. The results are shown in Figure 4.9.

It can be seen that σ has to be large enough to incorporate any temporal changes in the underlying PDF. The effect of this requirement is that for low σ values the accuracy of the video object segmentation drops dramatically. Above $\sigma \approx 3.0$ the segmentation (both scene and edge based) is reasonably stable. A value of $\sigma = 3.0$ was therefore selected.

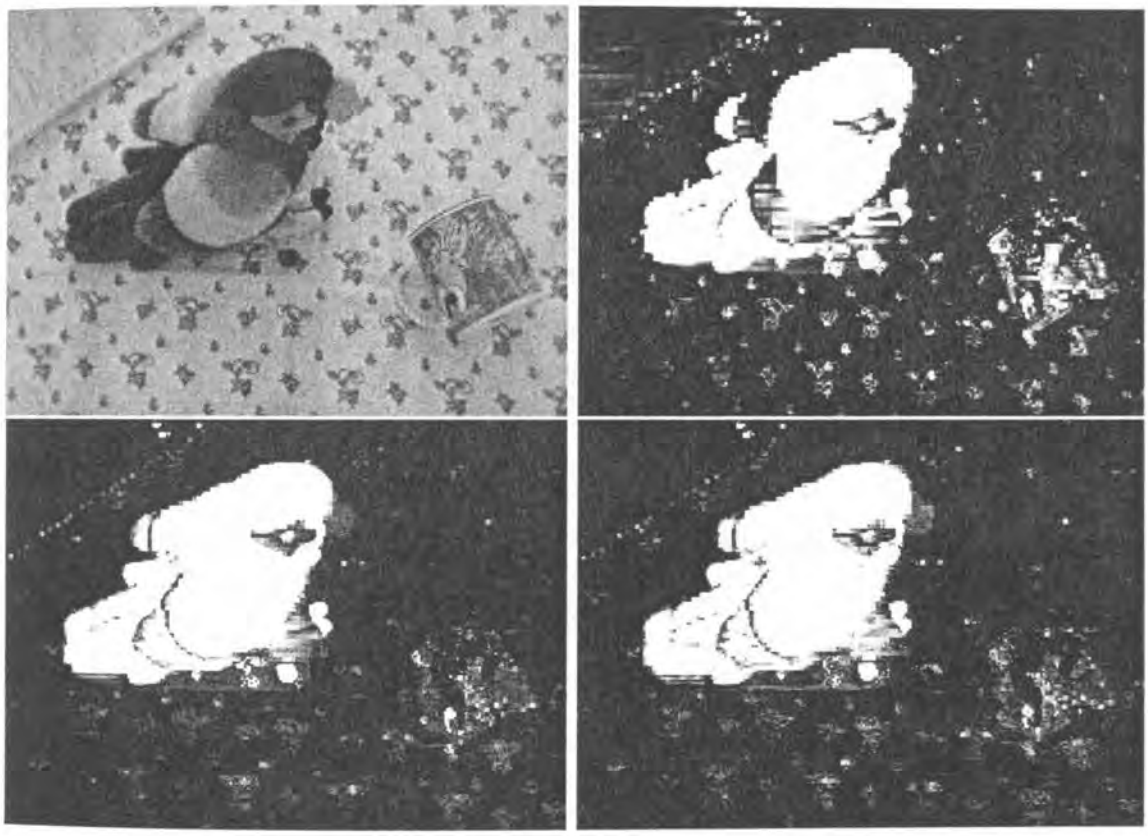


Figure 4.8: The posterior probability maps for the foreground video object chromatic observations. (Top Left) Colour observations for Frame 00010 of the 'Parrot' test sequence (Top Right) Modelled as a Gaussian distribution (Bottom Left) Modelled as a kernel density distribution (Bottom Right) Modelled as a Gaussian mixture model.

The two experiments above reinforce the message that scene based evaluation of object segmentation is generally not representative of the perceived segmentation quality (see Figure 4.11 for the segmentation result). For the case where $\sigma = 1$ the average SQD is measured at 94% yet the segmentation for these frames is visually poor. The edge based $SQED$ measure determines the average accuracy over the sequence to be 70% at $\sigma = 1$, which is more in line with a subjective assessment of the segmentation quality.

The Gaussian mixture model used in this evaluation contained ≈ 30 component densities per object, this number was determined automatically using an EM based algorithm with minumum message length criterion [67].

A comparison of the SQD and $SQED$ mean and standard deviation scores for the different representational techniques are shown in Table 4.1. It can be seen that kernel density model outperforms both the Gaussian mixture model and the Gaussian density model on

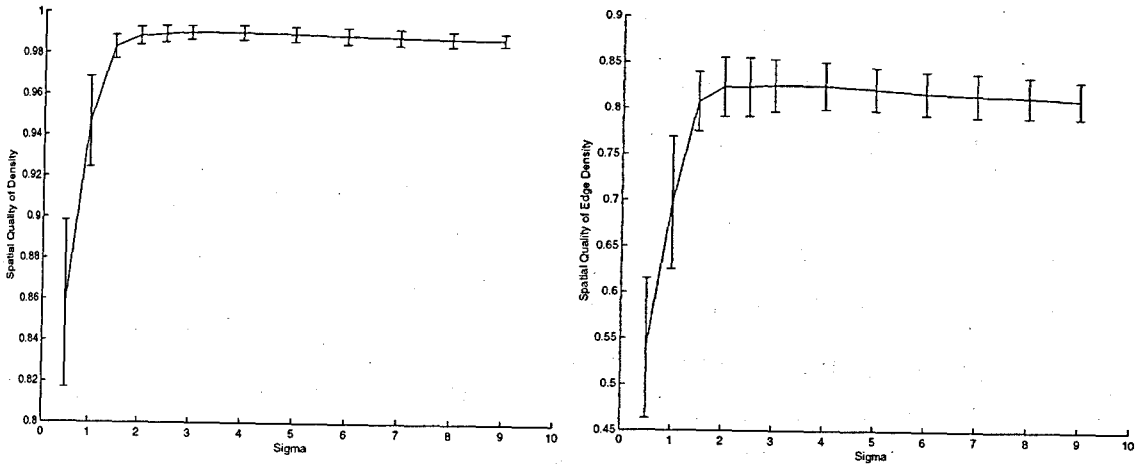


Figure 4.9: Plots showing the average segmentation accuracy for a range of σ values when using kernel density models to represent the video objects. (Top Left) SQD (Top Right) $SQED$. These results are averaged over all the frames in the 'Parrot' sequence.

the 'Parrot' test sequence. The Gaussian density performs much better than expected, perhaps due to the fact that a colourful object is being tracked which is generally well separated in colour space from the background and hence a single multivariate provides a reasonable representation of the foreground and background objects. The Gaussian mixture model and kernel density model report a similar performance at scene level, with approximately 98% of the scene pixels correctly labelled as foreground or background. At the edges of the video objects the $SQED$ measure demonstrates that the segmentation quality degrades to 82% for the kernel density and 79% for the Gaussian mixture model. This is probably due to the parametric form of the Gaussian mixture being unable to successfully represent the true spatial distribution of the Parrot object.

	μ_{SQD}	σ_{SQD}	μ_{SQED}	σ_{SQED}
Gaussian mixture model	0.9828	0.0065	0.7910	0.0336
Kernel density model	0.9871	0.0039	0.8210	0.0249
Gaussian	0.9588	0.0022	0.7031	0.0159

Table 4.1: Average SQD and $SQED$ results for the three probabilistic representational methods. These results are averaged over all the frames in the 'Parrot' sequence.

Whilst the results presented in Tables 4.1 show the avaerage performance of the rep-

resentational schemes over the 18 frames of the 'Parrot' sequence they do not characterise the relative performance of the algorithms on a per frame basis. To achieve this per frame charaterisation the evaluation measures for each frame are plotted. The resulting plots for the SQD and $SQED$ measures can be seen in Figures 4.10. In these plots it can be seen that the Gaussian mixture model based method not only gives a lower segmentation quality than the kernel density model, but that the segmentation degrades at a faster rate over the 18 frames largely due to the spatial mis-adaption of the model as the parametric form of the model inadequatly describes the spatial distribution of the foreground object. Interestingly, the Gaussian distribution appears to not degrade significantly over the course of the 18 frames although its performance overall is not as good. This result suggests that the model may be useful as a technique for approximately locating the video object on a per frame basis.

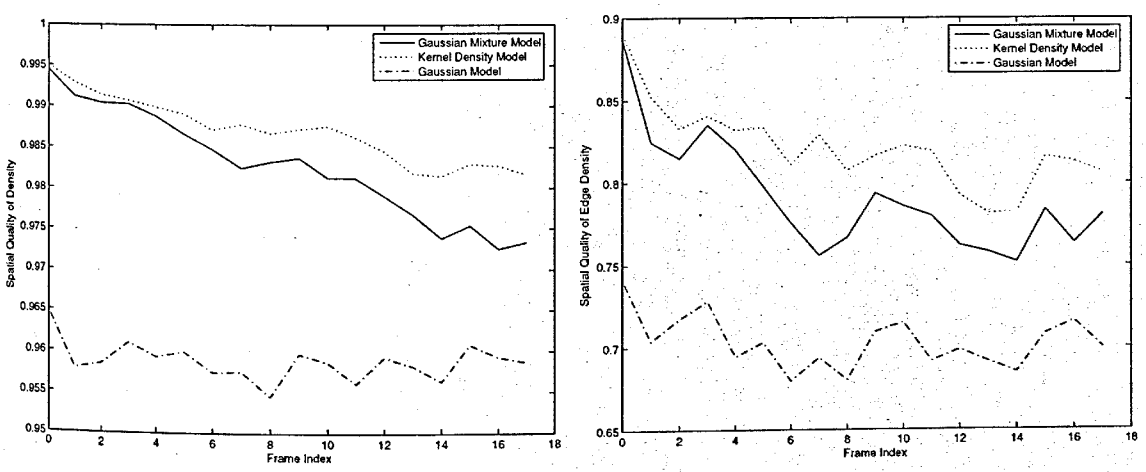
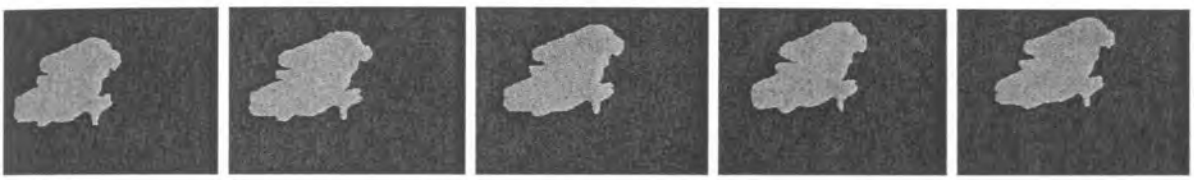


Figure 4.10: Plots showing the per-frame segmentation accuracy for the three representational methods. (Top Left) SQD (Top Right) $SQED$. The segmentation accuracy at each frame in the 'Parrot' sequence.

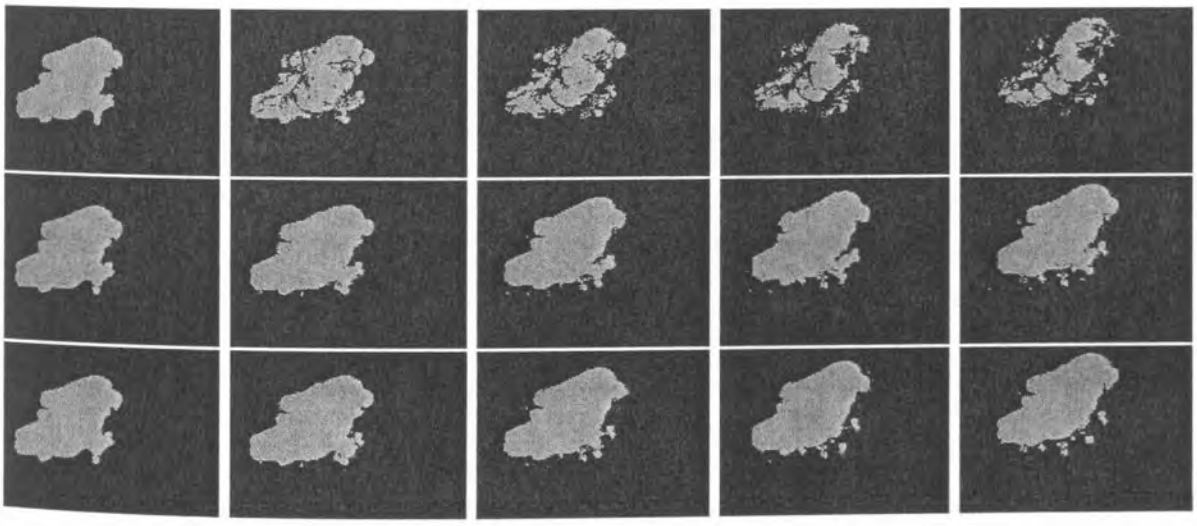
Figure 4.11 shows five resulting segmentation frames when using the proposed representational schemes. The kernel density model result is shown with various values of σ representing under smoothing ($= 1.0$), a near optimal smoothing ($= 3.0$) and over smoothing ($= 5.0$) of the estimated PDF. Subjectively the most accurate segmentation results are given by the Gaussian mixture model and the kernel density model. These two methods generally track the majority of the object accurately with few false positives/negatives detected due to misadaption of the representational models. With σ set too tightly the kernel

Ground-truth



Gaussian density model

Kernel density model for $\sigma = 1$ (top), 3 (middle) and 5 (bottom)



Gaussian mixture model

Figure 4.11: Video object segmentation and ground truth results demonstrating the different representational models. These are shown for frames 1,5,10,14 and 18 of the 'Parrot' sequence.

model cannot adapt to the temporal change in the spatial PDF of the foreground object, this has the effect that the region of support for the object diminishes over time. The Gaussian density model gives reasonable localisation, but subjectively poor segmentation, of the foreground object; this is due to the representation of the underlying multimodal PDF with 30 parameters, which is not enough to capture the nuances in the distribution.

4.4.4 Independent Spatial-Colour PDF Estimation in a Video Sequence

The effect of independent spatial and colour modelling of video objects is demonstrated for the foreground object in the 'Parrot' sequence (frame 1, shown in Figure 4.1). In the independent model the PDF of the colour observations is estimated using a Gaussian mixture model. It is reasoned that, in general, the colour distributions are adequately represented by a Gaussian mixture model distribution under the assumption that a mixture of observed colours on an object are distorted by Gaussian noise. The spatial PDF of the object is estimated using a non-parametric kernel density model, this representational model has been demonstrated to give good performance for modelling the spatial distribution around the edges of video objects.

The average SQD accuracy over the test sequence is measured to be 97%, which is marginally lower than joint PDF modelling using the Gaussian mixture model and kernel density model. The edge based $SQED$ accuracy is measured at an average of 74%, which is significantly lower than the joint PDF modelling using Gaussian mixture or kernel density models (shown in Table 4.1).

The resulting segmentation can be seen in Figure 4.12. Subjectively the segmentation result contains a coherent foreground object, although on closer inspection the mask contains a high proportion of false positives especially around the claw region. Comparing this result to Figure 4.11 the quality of the segmentation is between that of the Gaussian density model and the Gaussian mixture and kernel density models. This observation agrees with the measured SQD and $SQED$ accuracy.

4.5 Conclusions

In this chapter probabilistic models have been applied to the problem of video object segmentation. Three approaches to PDF estimation were described, implemented and eval-

Figure 4.12: Segmentation result using independent models.

uated in the context of video object segmentation. These three estimation methods were the Gaussian density, kernel density and Gaussian mixture models. It was demonstrated how these are applied to the problem of modelling spatial and colour based distributions. The performance of these three methods was evaluated using a framework for video object segmentation and the metrics presented in the previous chapter. Finally, the performance of the joint spatial-colour PDF models were compared to an independent model where the spatial and colour components were modelled individually.

The performance of the three types of PDF model was evaluated using a publicly available sequence. The models were implemented within a common framework to allow comparison between the measured accuracy for the three approaches. This framework used a 'null' prediction step such that the previous frame models were used as the initial estimate in the current frame. The models were subsequently adapted to the new observations using a per-frame reinitialisation strategy.

Modelling a spatial-colour feature space, it was determined that the kernel density model achieves the 'best' accuracy for segmentation around the edges of video objects. Kernel density models and Gaussian mixture models were found to give similar segmentation accuracy when measured at the scene level. It was also found that, as expected, the average error per pixel is greater at the edges of video objects than when measured over the whole image. The impact of different kernel density model smoothing values was also evaluated. This parameter was found to give a reasonably stable segmentation accuracy above a reasonable minimum value.

The independent modelling of the spatial and colour PDF's of video objects was also quantatively evaluated and was found to decrease the edge- and scene-based accuracy of the segmented video object when compared to the joint spatial-colour PDF models. This reduction in accuracy was due to the spatially variant distribution of colour over the surface of the video objects, which could not be well represented by the independent spatial-colour

PDF model.

In this chapter the following contributions were made:

- Kernel density models were found to give more accurate segmentation around the edges of video objects than Gaussian density and Gaussian mixture models.
- Kernel density and Gaussian mixture models were found to give similar segmentation accuracy when measured at the scene level.
- The smoothing factor in the Kernel density models was found to give a stable segmentation accuracy above a reasonable minimum value.
- The separation of colour and spatial modelling was found to decrease the accuracy of the resulting object segmentation.
- The $SQED$ edge-based segmentation accuracy was found to be closer to the perceived segmentation quality than the scene-based SQD measure.

The choice of representational scheme used to model a video object in the feature space has been demonstrated to have a significant effect on the resulting segmentation accuracy. Using a representational model that is ill-suited to the task will most likely degrade the accuracy of the segmentation of the object. From the evaluated representational schemes it appears that non-parametric distributions may be better suited to modelling the complex functional form of video objects.

A key issue when performing probabilistic modelling of video objects in multi-dimensional space is the computational efficiency of the algorithms. The computational requirements are reduced by dividing the modelling task into lower dimensional spaces that suffer less from the curse of dimensionality. Modelling marginal distributions of an object PDF is problematic due to the inherent spatially variant colour distributions of objects. In such circumstances the independent modelling of spatial and colour signals reduces the output accuracy of the video object segmentation algorithm.

The representational scheme for video objects has a significant effect on the resulting segmentation accuracy. In this chapter probabilistic methods have been evaluated for representing video objects in multi-dimensional feature spaces. A key advantage of using probabilistic techniques is that they allow a principled video object segmentation framework

to be built, so that different representational algorithms can be directly compared. Further work is required to evaluate the presented methods on more varied test sequences, although the creation of per pixel ground-truth is required for many standard test sequences. In the following chapter sub-regions of objects that exhibit homogeneity in both the colour and spatial feature dimensions are modelled and updated in video sequences. This property allows independent PDF estimates to be evaluated in localised image regions, leading to more efficient algorithms, overcoming the problems experienced when performing this type of independent feature space modelling.

Chapter 5

Propagation Strategies for Video Region Segmentation

In the previous chapter the concept of modelling video objects in a feature space was demonstrated using a selection of probabilistic representational schemes. It was found that the kernel density model achieved the 'best' accuracy for segmentation around the edges of video objects and that both kernel density models and Gaussian mixture models gave similar segmentation accuracy when measured at the scene level. The independent modelling of the spatial-colour PDF of video objects was quantatively evaluated and found to reduce the edge- and scene-based accuracy of the segmented video object when compared to the joint PDF models.

In this chapter the modelling is performed at a region-level, dividing each object (and hence each video frame) into homogeneous regions within the feature space. The application of region models overcomes some of the disadvantages of using the type of object-level model applied in the previous chapter:

- The objects to be defined have semantic meaning, and it is difficult to capture this semantic information at the object-level since it is commonly highly complex and exhibits a multi-modal character in the feature space.
- The probabilistic models required are generally computationally expensive when applied to the modelling of joint feature spaces.
- The separation of some feature components (e.g. spatial and colour information) is

not a valid assumption due to the intrinsic multimodal properties of the probability density distribution in the feature space.

Descriptions of regions and objects can be ambiguous in video analysis work. The incoming data in an video sequence is commonly quantised both spatially and temporally. The discrete unit of temporal quantisation is the image and the discrete unit of spatial quantisation is the pixel. A video-frame can be further divided into different levels of representation between the image-level (coarsest) and pixel-level (finest). Two such levels of representation are the aforementioned region- and object-levels. The definition of a region is chosen such that it is homogeneous due to some criteria (which is similar to the definition used by Deng and Manjunath [46]).

Based on this definition the modelling of regions may be more beneficial than modelling at the object-level since regions are homogeneous with respect to some criteria that can be defined mathematically. Therefore, the advantages of using an explicit region based modelling approach are that:

- The features can be assumed independently distributed between the taxanomic categories (e.g. motion, colour, space)
- The processing of regions is more localised, leading to faster, more efficient algorithms
- It is a well defined problem to innovate new regions in the scene
- Region-level based descriptors can be tracked between frames

Methods are introduced in this chapter for segmentating homogeneous regions of video sequences and a range of techniques for updating the region-based representation are presented. There exists a partition of the current video frame feature space $\mathcal{I}$ into S constituent regions $\mathcal{I}_s$ such that:

$$\mathcal{I} = \cup \mathcal{I}_s \text{ where } s = 1, \ldots, S \tag{5.1}$$

This mirrors the splitting of the feature space into constinuent objects in the previous chapter as shown in equation (4.1). The regions that are defined have parallels to the work of motion segmentation where sequences with abundant motion of objects are segmented into homogeneous regions in motion and colour feature space. However as demonstrated in

Chapter 3, the addition of motion information reduces the quality of the segmentation at the edges of the regions (although motion information can be beneficial in certain sequences). The video frames (and implicitly the video sequence) will therefore be decomposed into constituent regions by applying statistical modelling, adaption and innovation techniques. Applying a region labelling scheme it is possible to extend this work to the problem of extracting semantic video objects.

5.1 Previous Work

It is common in video object segmentation schemes to propagate the representational models between frames by simply using the previous frame representation — unchanged — in the current frame. Of course, this type of inter-frame prediction methodology is based on the assumption that the object does not move a significant amount in between frames and is therefore a sequence dependent assumption. For many standard test sequences this is found to be a reasonable assumption and has been used in video object extraction schemes by several researchers e.g. [98, 61, 135, 171, 91, 11].

Castagno *et al* [27] use the motion information to warp the current frame labelled segments to the next frame, from these warped segments markers are derived to allow guided initialisation of the cluster based representational models for the regions (and hence objects). Marlow and Connor [111] apply a similar method except that the statistics for the segments in the new frame are calculated from the motion projected partition of the regions. The approaches of [128, 112, 6] motion compensate regions forwards to find correspondances using set relationships between the projected regions and the homogeneous regions detected in the current frame. The regions in the work of Chalom and Bove Jr. [30] are defined by a set of training points provided by the user. In their work the inter-frame prediction stage warps the individual points to the next video frame from which pixel-wise classification of the scene is performed.

[149, 77, 76] apply translational motion models to warp regions between adjacent frames to find correspondances using pixel-wise set relationships between the projected regions and the current frame regions in an approach similar to [112, 128].

Wang [52] performs affine motion based region projection and merging, the projection error is used to resolve any conflicting regions. These projected regions are used to extract

watershed markers from which the new regions of support can be found. Fablet *et al* [66] apply the affine motion model in a slightly different approach. Instead of applying the model to warp the object representation they instead use it to determine the dominant motion in the video frame (assumed to be the camera). Colour regions that do not follow this dominant motion are found and assumed to belong to the moving objects in the scene. Deng and Manjunath [46] use affine motion models to warp homogeneous regions forwards through sequences between key framees. At the key frame the warped regions are matched to a colour and texture segmentation of the scene to determine the correspondance and hence allow long term tracking. Heuristic rules are subsequently applied to assign uncovered regions to the tracked regions in the scene. Salembier *et al* [152] peforms region tracking using affine projections with connectivity constraints of the regions at the previous frame, these are subsequently used as markers for a watershed based region growing algorithm. Patras *et al* [134] use an affine motion model to propagate watershed regions to be used as a *prior* temporal constraint on an iterative motion estimations / labelling process. [120, 121, 65, 64] applied the affine motion model with a geometric filter to allow the affine parameters of tracked video regions to be recursively estimated. Pateux [133] applies the affine motion model to perform backwards propagation of regions to the previous object segmentation map, the motion model estimate is smoothed using the motion estimates of neighbouring regions to give a more stable result.

Mech and Wollborn [117] apply a projective motion model to motion compensate adjacent frames to bring them into correspondance (in the case of a moving camera); after this a change detection is applied to find the regions of the image frame that belong to a moving object. Gu and Lee [83] apply the projective motion model to warp the object boundary between frames, they assume that the object itself is near-rigid does not undergo any significant non-rigid (i.e. articulated) motion between frames. To accomodate any non-rigid motion a boundary adjustment stage is used to correct the boundary estimate in the current frame.

If the expected motion in the scene cannot be determined *a priori* then a hierarchy of motion models can be evaluated to determine the simplest model that can accurately warp a region of an image. Guo *et al* [95] apply a hierarchy of motion models to determine the simplest model that can be used to warp the boundary representation of an object. The residue error after motion compensation is measured using the simplest (i.e. translational)

motion model and compared to the next most expensive model to check if there is any improvement in the accuracy of the result. This is performed upto the eight parameter planar projective motion model. In the case of minor improvement in warping accuracy between adjacent models the simpler motion model is chosen.

Posing the inter-frame prediction and intra-frame update schemes in the context of tracking allows recursive filtering methods to be applied to the problem of predicting and correction the video object representational schemes. Raja *et al* [143, 113] applied this methodology to estimate the bounding box location of a new object by recursively filtering the mean position of the component Gaussians in a mixture model. Once the bounding box location had been estimated a coarse-to-fine strategy was applied to find the object support on a per pixel basis.

Oliver *et al* [130] applied a zero-order Kalman filter to update the spatial parameters for each video object (e.g. face and lips); the state vector (blob centroid and bounding box dimensions) of this filter were updated in the MAP decision rule based on the newly observed object support. The object segmentation was found to be much more stable when using this type of filtering. A similar application of the Kalman was used by Chen and Huang [31] for tracking regions of interest in video sequences. Meyer and Bouthemy [120, 121] applied the Kalman filter with an affine parameter state vector to track the convex hull of spatio-temporal regions. The per region affine parameter measurements are made by solving for the affine motion model within the region from a dense motion field. The matching process is achieved by minimising the distance between the predicted and observed polygons by varying the translation and rotation of the predicted region polygon and allowing some vertices to be in an occluded state. The tracking of affine regions in this way assumes that 3-D motion terms are negligible in region tracking since accurate 3-D reconstruction is not required.

Similarly, Ziliani and Moscheni [65] investigated the use of Kalman filtering of affine motion information to track the location of spatio-temporal video regions, allowing accurate position and motion predictions to be made. The predicted and observed regions at a frame are corresponded using the affine motion parameters, moment based affine invariant spatial features and trajectory information. The use of the Kalman filter is proposed as being advantageous during occlusions since the object location can be estimated during such an event. In earlier work [64] a similar recursive estimate was used to both guide the motion

segmentation process and perform multi-hypothesis correspondance of regions.

Particle filtering [12] provides an alternative strategy to the Kalman filter when the process noise in the system are non-Gaussian or the state of the tracked object is non-Linear. Particle filtering has been applied to the problem of colour based region tracking in video sequences by [136, 127, 182]. In such approaches, objects are identified on a per-frame basis using some form of colour similarity (extended with edge based features in [182]) and then tracked between frames using a particle filter tracker. A potential drawback when applying particle filters is the tradeoff between computational complexity and drawing a sufficient number of samples to adequately describe (and hence propagate) the underlying density (which, if arbitrarily complex, may require a prohibitive number of samples).

The goal of the intra-frame update step is to correct the estimated video object representation by updating the representational model for the object using the current frame data. The dynamic nature of video sequences makes this a challenging stage in the segmentation process, a balance must be sought between the adaptibility of the model and the robustness to noise in the underlying signal. For specific video sequences the assumption of constant object appearance [186] may result in the intra-frame update of the representational being a trivial step (e.g. [30]). This assumption is generally only valid for short video sequences.

A common per frame update methodology for the object representational models is to find the region of support for the object in the current frame using the propagated models and then reinitialise the object representation based on this new found object region (e.g. [61]). For parametric representational schemes the per frame reinitialisation can be an expensive and time consuming technique for intra-frame updating of the models. For non-parametric methods — where the functional form of the model is the data — this type of update can be computationally cheap. A better strategy for updating parametric models is to use the previous frame model as a seed to guide the reinitialisation of the models to the newly observed data. This type of approach has been applied by [27, 135, 145, 111]. Meyer and Bouthemy [120, 121] track affine regions by imposing that the convex hull of the region always has the same number of vertices, if the distance between the predicted and observed objects is large the region representation (a polygon) is reinitialised to the most recent observation.

Alternatively, the object representation can be adapted to take into account changes in the object appearance due to lighting, pose or motion changes etc. One possible technique

to achieve this on a per frame basis is to apply a recursive filter to smooth the adaption of the model so that predicited object parameters are less susceptible to noise. Raja *et al* [143, 113] applied a recursive filter of this type to adapt the parameters of a Gaussian mixture model to the dynamically changing scene properties due to illumination and viewing conditions etc.

5.2 A Region-Based Segmentation Algorithm

In this section a region based segmentation algorithm is developed in the context of video sequences. Like video object segmentation this algorithm comprises three main components — feature space, representational model and model update. The feature space applied in this algorithm is the $XYL*a*b*$ spatial-colour space (introduced in Chapter 3). The representational model, presented in Section 5.2.1, takes advantage of the region based processing by combining both parametric and non-parametric models for each homogenenous region. Finally the model update methodology is presented in Section 5.4, Section 5.5 and Section 5.3. These sections detail the inter-frame prediction and intra-frame update for the regions on a per-frame basis, followed by a description of the innovation and termination criteria for regions and finally the special case of initialisation of such regions in the first frame of a sequence.

5.2.1 Representation of Spatial-Colour Regions

The segmenting of regions by estimating the PDF in a high dimensional feature space can be computationally prohibitive (and suffer from the curse of dimensionality). To efficiently model the joint spatial-colour PDF distribution of video regions the properties of region-based representation are exploited. The main advantage of using a region-based representational scheme is that some feature space dimensions can be assumed independently distributed between the taxonomic categories. Video frame labelling using region-based representational schemes can be more efficient than an equivalent object-based representation by implicitly localising the labelling process to areas around the regions (which also have less complex representational models than objects).

Object-level models do not take into account articulated motion i.e. significant differences in the motion of individual regions belonging to a parent object. The ability to model

the varying motion over an object can improve the prediction of the representational scheme (particularly the spatial distribution) between frames. In the proposed reprensentational scheme the modelled regions represent spatial areas with colour homogeneity in the video using independent colour and spatial models.

The joint distribution of the spatial $\mathbf{x}$ and chromatic $\mathbf{f}$ components of an $XYL*a*b*$ feature vector $\mathbf{a}$ is given by:

$$p(\mathbf{a}) = p(\mathbf{f})p(\mathbf{x}) \tag{5.2}$$

The form of this expression assumes an independence between the chromatic information of a region and its spatial distribution. The PDF $p(\mathbf{a})$ is modelled for each region using parametric and non-parametric models in each colour or spatial component respectively. The observed chromatic distribution for a region λ_s is modelled using a multivariate normal density of the form:

$$p(\mathbf{f}_i|\lambda_s) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_s|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{f}_i - \mu_s)^T \Sigma_s^{-1}(\mathbf{f}_i - \mu_s)\right] \tag{5.3}$$

$\mathbf{f}_i$ is the chromatic $L*a*b*$ observation at the i'th pixel and Σ_s is the covariance matrix and μ_s is the chromatic mean of the s'th region model. The *a posteriori* probability that this observation, $\mathbf{f}_i$, belongs to region λ_s is given by Bayes theorem.

The observed spatial PDF of a region λ_s is modelled using a Gaussian kernel density model of the form shown in equation (4.17) i.e.

$$p(\mathbf{x}_i|\lambda_s) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi\sigma^2)}\exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_n\|^2}{2\sigma^2}\right\} \tag{5.4}$$

where $\mathbf{x}_i$ is the spatial feature vector extracted at the i'th pixel and $\mathbf{x}_n$ represent the N spatial feature vectors extracted at pixels within the region λ_s. Evaluating this function will be expensive if the value of N is large. A faster evaluation of the spatial distribution can be achieved by forming a binary image for each region and convolving with a 2-D Gaussian kernel. Changing the value of σ in the model allows the uncertainty of the region shape to be changed, in scenes with large motion setting σ higher allows the spatial model to adapt to such motions.

The MAP decision rule is applied to find the per frame support for the regions as they propagate throughout the video sequences. At each pixel observation vector $\mathbf{a}_i$ the pixel i

is labelled by choosing the region index s^* such that

$$s_i^* = \arg\max_s P(\lambda_s|\mathbf{a}_i) = \arg\max_s p(\mathbf{a}_i|\lambda_s)P(\lambda_s) \tag{5.5}$$

where, from equation (5.2),

$$p(\mathbf{a}_i|\lambda_s) = p(\mathbf{f}_i|\lambda_s)p(\mathbf{x}_i|\lambda_s) \tag{5.6}$$

$p(\mathbf{x}_i|\lambda_s)$ is obtained from the spatial model of equation (5.4) and $p(\mathbf{f}_i|\lambda_s)$ is obtained using the Gaussian density model of the $L*a*b*$ colour distribution in the s'th region defined in equation (5.3). As the spatial model has a known uncertainty, σ, therefore equation (5.5) can be evaluated for regions *local* to the pixel that have a non-zero spatial probability. This leads to a very efficient algorithm for finding the region of support. The prior probability of the regions $P(\lambda_s)$ can be set as equiprobable for all regions or approximated by the relative area of the region to the video frame.

5.3 Initialisation of Regions

At the first key-frame in a video sequence, the region based representational scheme is initialised to provide models for the homogeneous regions in that key-frame. Once the models are initialised these regions are then propagated through the video sequence using inter- and intra- frame strategies, presented in Section 5.4.

The regions are initialised using a colour based representational scheme. Figure 5.1 provides an overview of this process. The initial image is first segmented based on $L*a*b*$ colour information by learning the natural clusters in the data (applying a variant of the expectation maximisation algorithm for Gaussian mixture models [67]). The segmented image consists of disjoint clusters, therefore connectivity analysis is applied to find the individual homogeneous colour regions within the scene. A minimum region size is enforced to remove insignificant regions. Each remaining colour region is subsequently analysed to generate the colour and spatial PDF representational models. The initial prior probability for each region-level representation $P(\lambda_s)$ is computed as $1/S$ where S is the number of regions created (i.e. equiprobable).

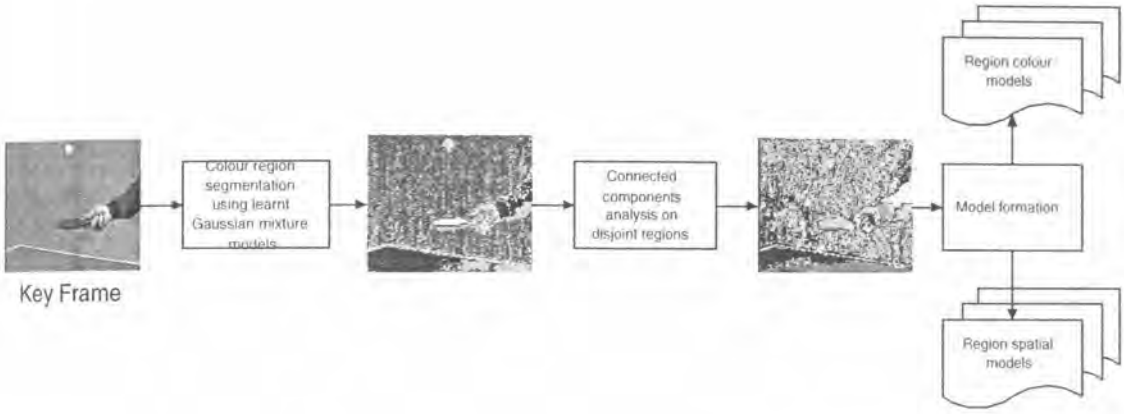


Figure 5.1: Overview of the initialisation process for video region segmentation.

5.3.1 Choosing a Minimum Area Threshold

The minimum area threshold N_{min} used to remove insignificant regions can influence how well the region models represent the PDF of the video frame. It is found that the majority of the regions generated by the initial connectivity constrained colour segmentation of the scene are insignificant and can be removed, improving the computational demands of the algorithm. This culling step is designed such that the removed regions are small enough to be absorbed by the remaining regions in subsequent update steps. Any unclassified pixels in the output segmentation mask at each frame are either left unprocessed or labelled using a mode filter, depending on the application.

To demonstrate, the number of initial regions per unit area is measured for different minimum size thresholds N_{min} for three different key-frames from well known test sequences — 'Table Tennis', 'Parrot' and 'Coastguard'. The 'Table Tennis' (25,344 pixel) and 'Coastguard' (25,344 pixel) key-frames are shown in Figure 5.10. The 'Parrot' (46,080 pixel) key-frame is shown in Figure 4.5. Figure 5.2 shows the decreasing exponential relationship between the minimum region size (scaled by the video frame area) and the number of connected regions per unit area (i.e. an area density function estimate). It can be seen that the majority of regions for all three test sequences are small regions, representing a small proportion of the overall image. The small offsets evident between the curves are a function of the texturedness of each video frame i.e. a scene containing large amounts of texture will be segmented into smaller regions. The trade-off between removing smaller regions and retaining the representativeness of the region-based models is an important

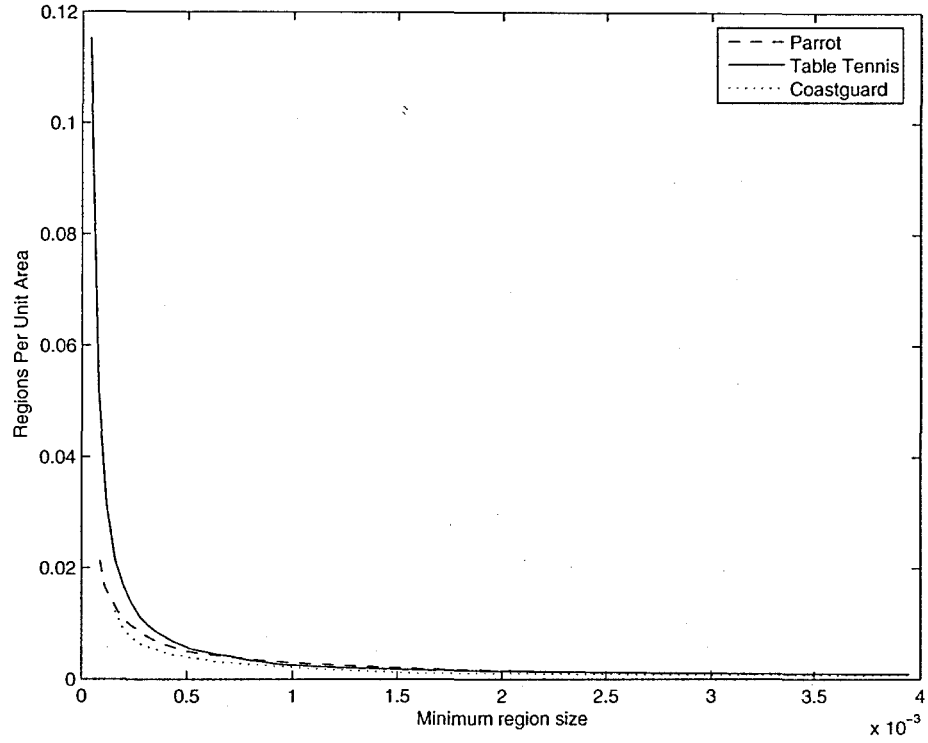


Figure 5.2: The number of connected regions plotted against the minimum region size for three video frame. The frames are taken from the 'Table Tennis', 'Parrot' and 'Coastguard' sequences. The minimum region size and number of regions are scaled by the area of the video frame.

factor. For the 'Table Tennis' it is found that a minimum region size of $N_{min} \approx 7$ pixels removes approximately 92% of the initial regions accounting for 15% of the video frame.

5.4 Propagation of Probabilistic Spatial-Colour Regions

To exploit the temporal information in video sequences, methods are required for *propagating* the homogeneous regions on a per-frame basis. In the case of the independent spatial-colour PDF's methods are sought that propagate the independent density models. Within a dynamic video sequence new, previously unseen, regions often appear in the video frame. The model adaption is therefore required to be constrained to prevent misadaption to new regions. The consequence of this constraint is that methods are also required to detect new candidate regions in the unassigned portion of the video frame.

Figure 5.3 shows a framework for a region-based video segmentation algorithm. The region models from frame $t-1$ are propagated in the inter-frame prediction stage to become

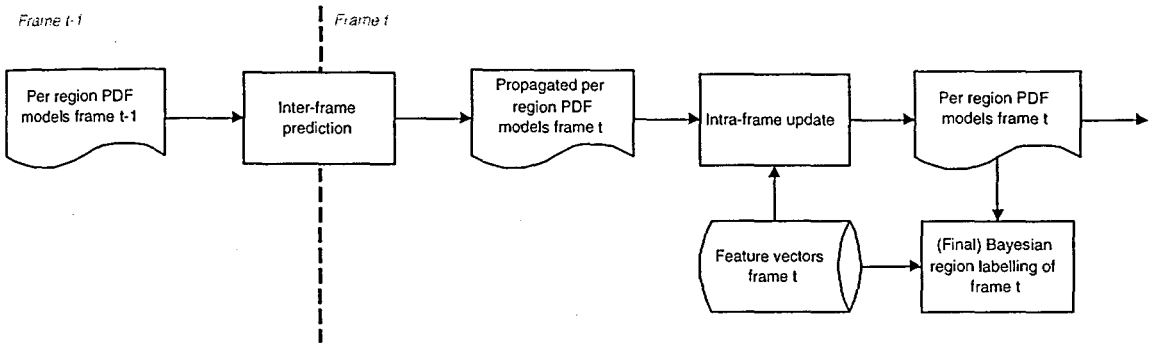


Figure 5.3: Framework for inter-frame prediction and intra-frame update of probabilistic region-level models. The regions are propagated on a per frame basis in this framework.

the initial estimate for the models in frame t. These models are subsequently updated in the intra-frame update step using the observed video data at frame t. New models are innovated in this intra-frame update step. This figure can be compared to the earlier object-based Figure 4.2 where the inter-frame prediction simply uses the models at frame $t-1$ without explicit propagation and the intra-frame update applies a MAP based decision rule followed by a stage of model reinitialisation.

Updating the regions on a per-frame basis is required to allow the temporal evolution of the representational models, and hence account for dynamic changes in the incoming video data and ensuring the representation of the video is meaningful. The goal when updating the regions is to seek a balance between the following factors:

1. Preserve existing region homogeneity i.e. *minimise* misadaption of existing regions
2. Minimise the innovation of new regions i.e. *maximise* lifespan and size of existing regions

A video region segmentation scheme that does not preserve region homogeneity will extract what are likely to be semantically inhomogeneous regions. Similarly, a scheme that has a large volume of short-lived new regions will be a difficult scheme within which to add object level labelling or temporal analysis. It is within this context that methods for adapting the PDF models of spatial-colour regions throughout the video sequence are introduced.

5.4.1 Inter-frame Prediction of Regions

The goal of the inter-frame[1] prediction step is to propagate the previous frame representational models to obtain a predicted model for the current frame. This predicted model is not expected to correlate perfectly with the observed feature vector information, but it is expected to be a better match than simply using the previous frame models. This difference becomes increasingly important for regions that move significant distances between the sampled time slots of the video sequence.

In this section three strategies are introduced that can be applied to the problem of propagating independent spatial-colour video regions. These techniques commonly analyse the spatial trajectory information of the regions to propagate the spatial model of the regions between frames. A related set of techniques for propagating regions are appearance analysis methods which make predictions of object appearance using the feature space PDF model information.

In this work the inter-frame prediction is performed in the spirit of trajectory analysis methods. This is based on the reasoning that the spatial motion of objects is more significant inter-frame than the appearance 'motion' of objects i.e. the colour information of an object is generally slow moving per-frame and can be adapted to in the intra-frame update step. In some sequences (e.g. with fast lighting changes) this may not be the case and recursive filters can be applied to help predict the lighting change per-frame. Due to the dynamic nature of many video sequences such techniques often only work well in constrained environments i.e. stationary cameras viewing a near-stationary scene. In scenes with moving objects and lighting changes it is an incredibly complex problem to classify changes in region appearance as being due to lighting change or, for example, partial occlusion.

In this section strategies for inter-frame prediction of video regions are introduced. The three strategies encompass the popular methods by which region models are propagated from frame to frame. The first method — 'Frame $t-1$ Models at Frame t' — is really a null step, in that the regions at the previous frame are not explicitly propagated between frames and retain the estimated spatial model from the previous frame. The second method — motion model based compensation — warps the video regions between frames by estimating a per-region motion model using pre-computed optical flow information. The final method

[1] Often referred to as tracking, warping, motion compensating in the literature.

— recursive filtering — assumings a trajectory model for the video region and attempts to predict the next frame model using the previously observed spatial information within a temporal window.

'Frame $t-1$ Models at Frame t' Strategy

This type of inter-frame prediction methodology is based on the assumption that the regions present in the scene do not move a significant amount in between frames (and is therefore a sequence dependent assumption). The propagated region PDF models at time t are the estimated PDF models from the previous frame $t-1$ i.e.

$$p'(\mathbf{a}|\lambda_{s,t}) = p(\mathbf{a}|\lambda_{s,t-1}) \tag{5.7}$$

where $p'(\mathbf{a}|\lambda_{s,t})$ is the propagated PDF formed from the model parameters belonging to the s'th video region λ_s at frame t. This type of 'null' prediction strategy is used as a control to compare the performance of other more complex strategies.

This method is only applicable to sequences with low motion between frames, in the case of high motion sequences the propagated models generally will not match the observed data. Figure 5.4 shows examples of this phenomenon for the differences between two consecutive object-level spatial representations (binary masks). The difference image for the 'Claire' sequence shows minor differences between the spatial distribution of the foreground object, for this sequence the update mechanism suggested in (5.7) is likely to be sufficient. For the 'Children' sequence, the majority of the moving ball region does not overlap between the consecutive frames presented therefore the propagated region model using this strategy is likely to reduce in size and diminish over a short period.

Motion-Model Based Compensation Strategy

This inter-frame prediction scheme projects the spatial component of the region models between frames such that the models are geometrically transformed to account for the motion of the regions between temporal observations within the scene. Figure 5.5 shows a general framework for motion based compensation of regions in the spatial plane. The motion-model is fitted to per pixel optical flow for each region to generate per region motion models. The optical flow is computed from two (or more) frames in an independent process

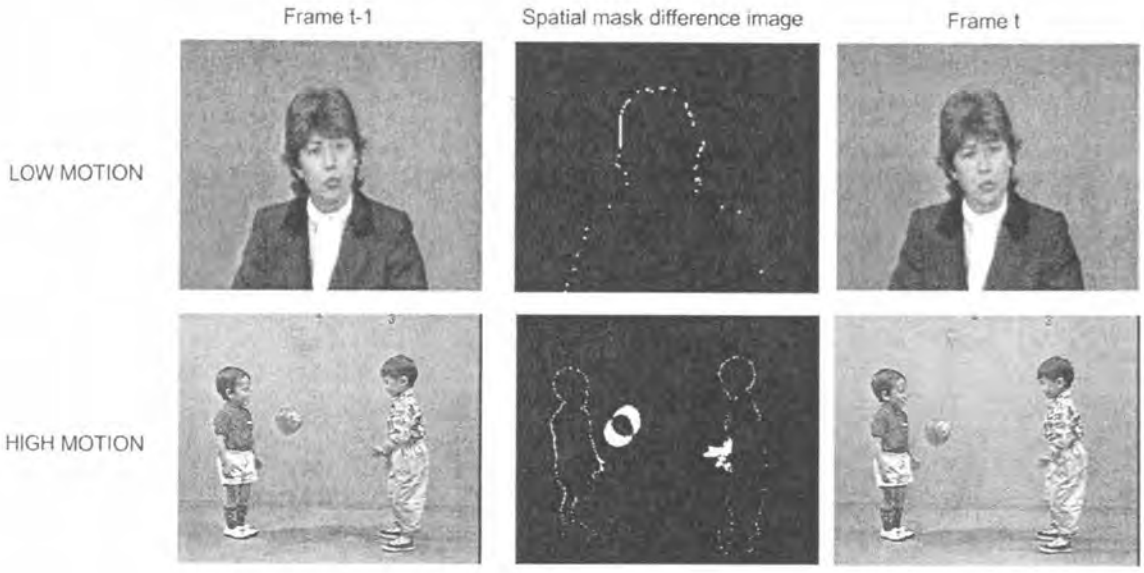


Figure 5.4: Examples of two qualitative levels of motion encountered in video sequences. (Top-Row) A lower motion sequence showing frames 00020 and 00021 from the 'Claire' sequence. (Bottom-Row) A higher motion sequence showing frames 00019 and 00020 from the 'Children' sequence. The binary masks show the difference in the object segmentation masks between these two frames.

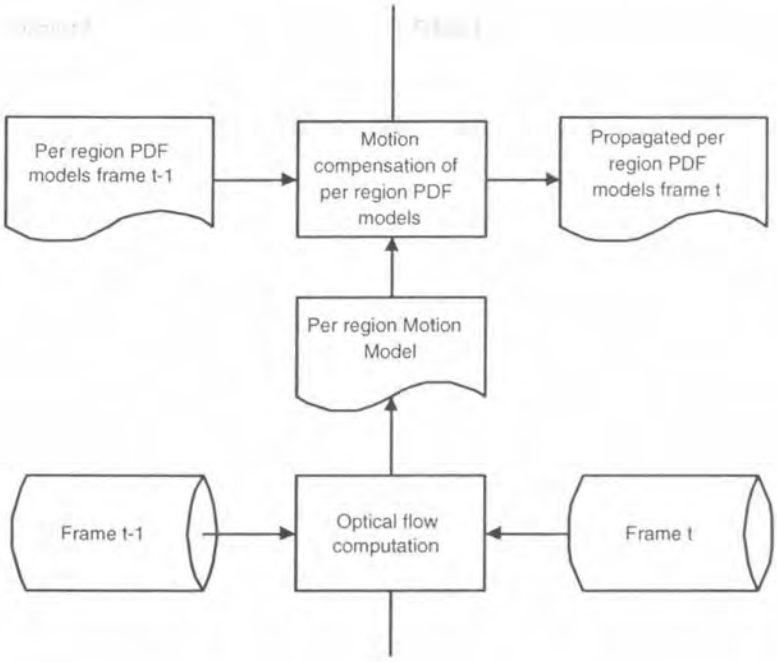


Figure 5.5: Inter-frame prediction of probabilistic region models using a motion model based scheme.

to the region tracking module. To generate this optical flow the Lucas Kanade method introduced in Section 3.3.3 is re-used. For each region the motion model parameters are estimated to describe the general motion of the region between adjacent video frames. This motion model can then be used to transform the spatial PDF model for each region.

A common motion model in the inter-frame prediction stage is the six parameter affine model — this type of motion model preserves parallel lines and allows for rotation, scaling, shearing and translational motions [155]. A general affine transform is defined by [155, 73]

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & u_3 \\ u_4 & u_5 & u_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5.8}$$

In the matrix A are the model parameters $u_1 \ldots u_6$. This transform displaces a point from position (x, y) in the source frame to the location (x', y') in the destination frame. This transform is defined around a spatial reference point, in (5.8) this is assumed to be at the origin of the co-ordinate system, if a different spatial reference point is used (for example, the spatial mean of a video region) then the relevant translation is made to the image co-ordinates to compute their position relative to the spatial reference point.

In general the affine transformation is defined by six parameters which are determined by fitting the transform to at least three (but often more) motion measurements using a least squares method. The per region affine motion model are estimated by fitting the transformation to the per pixel optical flow information in that region. Since the transform of discrete pixel locations often results in discontinuities in the transformed image a bilinear interpolation scheme is often applied to estimate the missing pixel values in the warped image.

Recursive Filtering Strategy

Inter-frame prediction of regions can be performed using recursive filtering techniques that are popular in the field of visual surveillance. In this work the Kalman filter is applied to the problem of inter-frame propagation of video regions i.e. 'tracking' the regions through the video sequence on a per-frame basis. A general framework for using a recursive filter in the inter-frame prediction step of video region segmentation schemes is shown in Figure 5.6. The per frame PDF models found at frame $t-1$ are first used to update the state information

of the recursive filter (e.g. the velocity based on a centroid measurement). This per region trajectory model is subsequently used to transform (i.e. propagate) the region spatial models based on a predicted trajectory between the two frames. Therefore the resulting region models after this inter-frame prediction have the colour model from frame $t-1$ and a propagated spatial model at time t.

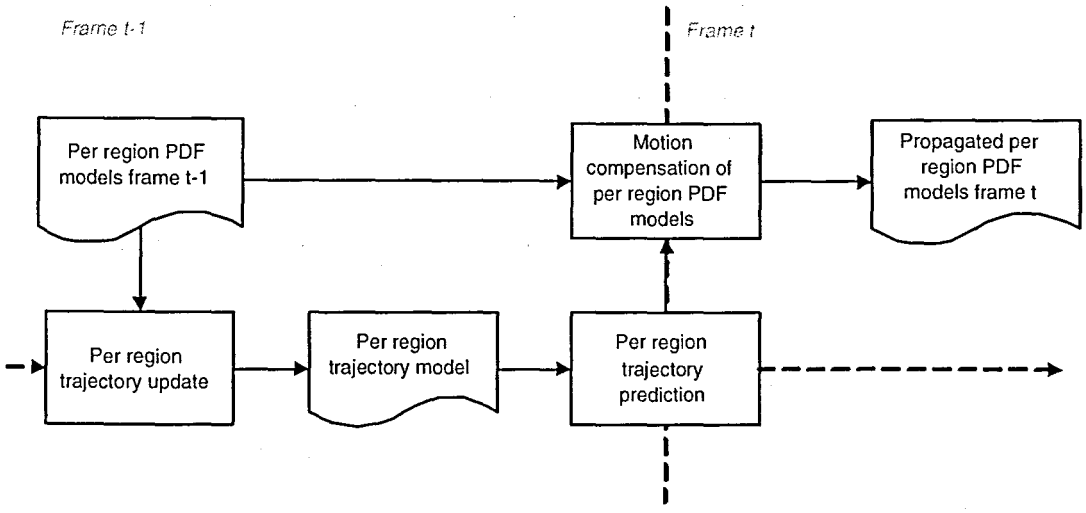


Figure 5.6: Inter-frame prediction of probabilistic region models using a recursive based scheme.

Let us define a D-dimensional state vector $\mathbf{q}_t$ for a region being tracked at time t. This state vector cannot be directly measured and the goal of all tracking algorithms is to make an estimation of this state based on noisy observations. At each discrete time instance a noisy observation vector $\mathbf{a}_t$ is made for each of the region being tracked; the contents of this vector are commonly trajectory related measurements describing the region e.g. bounding box information $[x_c, y_c, w, h]^T$. Tracking algorithms generally seek to estimate the state vector at a current time instance based on the set of observations and to predict future observations to constrain the subsequent region tracking, increasing the efficiency and reliability.

In a Kalman filter the probability of the state vector $p(\mathbf{q}_t | \{\mathbf{a}_1, \ldots, \mathbf{a}_t\})$ is modelled by a single Gaussian function i.e. $\{\mu_t, \Sigma_t\}$ where μ_t is the most probable state and the uncertainty is characterised by Σ_t. This Gaussian density is propagated over a period of time by fusing the parameters and the prediction of the observation with the actual observation $\mathbf{a}_t$. This fusion weights the predictions and the observations by their relative

uncertainty, stored in the Kalman gain matrix $\mathbf{K}_t$. The estimated state vector, $\widehat{\mathbf{q}}_t$, is given by:

$$\widehat{\mathbf{q}}_t = \mathbf{q}_t^* + \mathbf{K}_t \left[\mathbf{a}_t - \mathbf{a}_t^*\right] \tag{5.9}$$

the associated uncertainty of the state estimate is given by

$$\widehat{\Sigma}_{\mathbf{q},t} = \Sigma_{\mathbf{q},t}^* - \mathbf{K}_t \mathbf{H} \Sigma_{\mathbf{q},t}^* \tag{5.10}$$

where $\mathbf{K}_t$ is the Kalman gain matrix:

$$\mathbf{K}_t = \Sigma_{\mathbf{q},t}^* \mathbf{H}^T \left[\Sigma_{\mathbf{a},t}^* + \Sigma_{\mathbf{a},t}\right]^{-1} \tag{5.11}$$

and $\mathbf{H}$ is the measurement matrix that transforms a state vector to an observation vector i.e. $\mathbf{a}_t = \mathbf{H}\mathbf{q}_t$. The predicted observations and uncertainty are given by

$$\mathbf{a}^* = \mathbf{H}\mathbf{q}^* \tag{5.12}$$

$$\Sigma_{\mathbf{a}}^* = \mathbf{H} \Sigma_{\mathbf{q}}^* \mathbf{H}^T \tag{5.13}$$

If the state vector includes the first order derivatives i.e. $\mathbf{q} = [q_1, \dot{q}_1, \ldots, q_D, \dot{q}_D]^T$ then the prediction for the state at time $t + \Delta t$ is given by $\mathbf{q}_{t+\Delta t}^* = \mathbf{A}\widehat{\mathbf{q}}_t$ with uncertainty $\Sigma_{\mathbf{q},t+\Delta t}^* = \mathbf{A}\widehat{\Sigma}_{\mathbf{q},t}\mathbf{A}^T + \widehat{\mathbf{G}}_{\mathbf{q},t}\Delta t^4$. The $(2D \times 2D)$ state transition matrix $\mathbf{A}$ is given by:

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} & 0 & \ldots & 0 \\ 0 & \mathbf{B} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathbf{B} \end{bmatrix} \quad \text{where } \mathbf{B} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{5.14}$$

To predict the regions per frame the Kalman filter is configured using the following state vector (holding first order derivatives of the centroid):

$$\mathbf{q}_t = [x_c, y_c, \dot{x}_c, \dot{y}_c, w, h]^T \tag{5.15}$$

where x_c and y_c represent the centroid of a region's bounding box, which is of width w and height h. The observation vector for the filter contains the observable information about the bounding box:

$$\mathbf{a}_t = [x_c, y_c, w, h]^T \tag{5.16}$$

The bounding box dimensions are tracked to allow the system to adapt to limited changes in the object appearance due to motion. Alternatively, a set of affine parameters could be estimated using the filter (e.g. [65]). An affine parameter Kalman filter of this type is not evaluated in this chapter due to the reliance on the output of the affine motion-model estimation.

5.4.2 Intra-frame Update of Regions

The goal of intra-frame adaption is to update predicted region models to the new observed feature vectors in the current video frame. This adaption can be performed for either the colour model, the spatial model or both. In this section two strategies for intra-frame updating of video regions are introduced.

Reinitialisation Strategy

A common update methodology for region-based representational models is to find the area of support in the current frame using the propagated models and then reinitialise the models to represent the new observations. Such a scheme is presented in Figure 5.7.

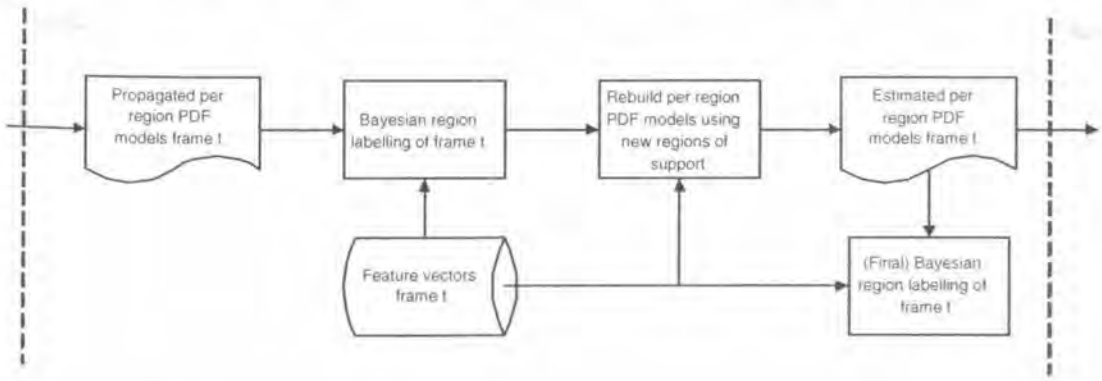


Figure 5.7: Framework for intra-frame update of probabilistic region models using a reinitialisation based scheme.

In this scheme the propagated (i.e. *predicted*) models from time index $t-1$ are used to generate a labelling of the newly observed frame t. The region models are *reset* and *initialised* to the newly observed feature vectors 'belonging' to each region. This process is a simple and effective technique for updating the feature space PDF models on a per frame basis.

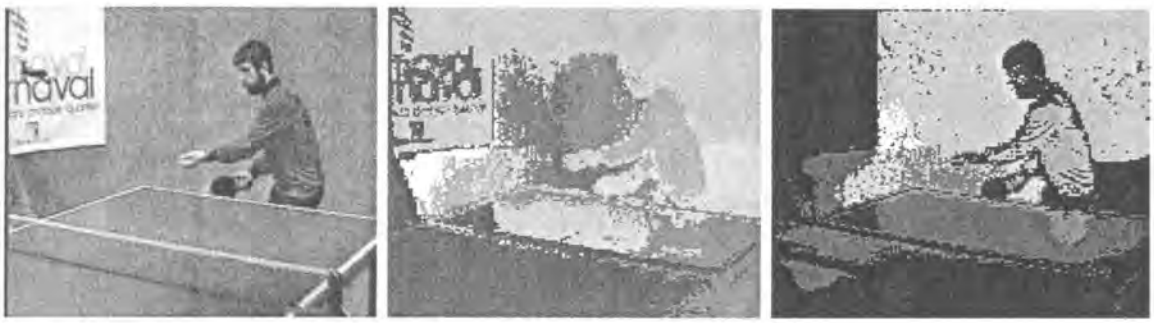


Figure 5.8: Propagated regions resulting from constrained and unconstrained intra-frame reinitialisation strategies. (Centre) Misadapted regions due to unconstrained intra-frame reinitialisation strategy for frame 00016 taken from the 176 × 144 QCIF sequence 'Table Tennis' (Left). (Right) A constrained intra-frame reinitialisation strategy. Notice that the regions in the constrained reinitialisation strategy do not 'spill-over' colour boundaries in the scene. For example, the background and poster objects are merged into a single inhomogeneous region using the unconstrained strategy. In the constrained strategy the poster object remains undefined, with no representative match found within the set of existing regions.

A major drawback of this method is that without constraints on the adaption the regions can 'drift' due to the "winner takes all" nature of the MAP decision rule. The effect of this misadaption is demonstrated for the 'Table-Tennis' sequence frame 00016 in Figure 5.8. It is clear that many of the regions are no longer homogeneous in colour. To reduce misadaption of the representational models a constraint can be introduced in the intra-frame update scheme. This constraint is a distance threshold on the MAP decision rule step. A pixel is only considered in the MAP update step if the colour feature vector is within a Mahalanobis distance threshold τ of the colour cluster mean belonging to the region under consideration. This has the effect that regions are only initialised to new observations that are inliers to the modelled colour PDF of that region.

The effect of this updated decision rule is shown (right) in Figure 5.8. This method preserves the homogeneity of the existing regions at the expense of having large areas of the final image unassigned since the Mahalanobis distance of these observed feature vectors from the mean feature vector of all regions has exceeded the threshold τ. For such a constrained technique to be applicable to video region (and object) segmentation a methodology is required to innovate new regions to model the unassigned areas of the video frame.

Recursive Filtering Strategy

For a newly observed frame in the video sequence the region-based model can be adapted to take into account changes in the object appearance due to lighting, pose or motion changes etc. One possible technique to achieve this on a per frame (i.e. sequential) basis is to apply a recursive filter to *smooth* the estimation of the model so that predicted parameters are less susceptible to misadaption caused by erroneous data. The observations are therefore assumed to be made from a slowly varying (non-stationary) signal.

Figure 5.9 shows one type of recursive strategy for estimating model parameters. This strategy is based on methods using recursive estimation methods. In such a scheme the current frame t is again partitioned using the models propagated from frame $t-1$ in a Bayesian decision process. New 'observation models' are created and fitted to these newly detected image regions. The predicted region models are subsequently updated using a recursive strategy to estimate the model parameters from the prediction and the observation. One such strategy is to regularise the estimated model parameters $\boldsymbol{\theta}_t$ using a rolling average of the form:

$$\boldsymbol{\theta}_{s,t} = \alpha \boldsymbol{\theta}_{s,t-1} + (1-\alpha)\widehat{\boldsymbol{\theta}}_{s,t} \tag{5.17}$$

$\boldsymbol{\theta}_{s,t}$ represents the set of model parameters for the region λ_s at time t. $\widehat{\boldsymbol{\theta}}_{s,t}$ represents the set of 'observation models' and $\boldsymbol{\theta}_{s,t-1}$ represents the set of model parameters recursively estimated in the previous frame. To reduce the effect of misadaption a constraint can be applied in the labelling scheme similar to that shown for reinitialisation.

A different form of recursive filtering is to use the region models in the previous frame $t-1$ as *seeds* to find the models in the current frame t. This approach is generally only applicable to clustering based algorithms. An example of this is the *incremental* (or *online*) EM algorithm where the model parameters are recursively updated by each newly observed feature vector, weighted by the probability that the feature vector *belongs* to that model, which allows the model to slowly adapt to changes in the true PDF.

The two techniques presented above for intra-frame model update allow online adaption to the unfolding scene in a video sequence. This leads to algorithms that can be used for representation of dynamic sequences. If constraints are imposed on the intra-frame adaption, a methodology is required to allow new region-based models to be generated to

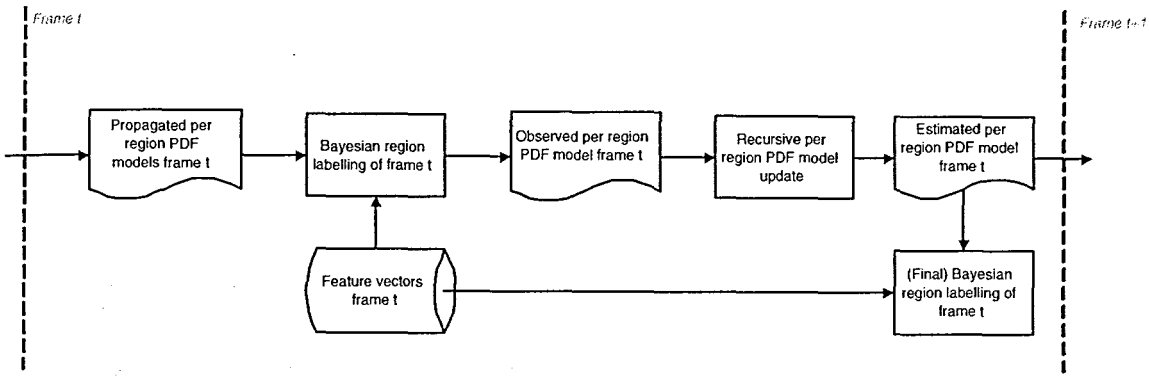


Figure 5.9: Framework for intra-frame update of probabilistic region models using a recursive strategy.

represent previously unseen video regions that are not well modelled by the existing regions. Such a methodology is presented in Section 5.5.

5.5 Termination and Innovation of Regions

The labelling of the current frame t with the propagated regions from frame $t-1$ may result in region models that are not supported by any observations or observations that are not supported by any region models. When regions are no longer supported by the observed data, a mechanism is required to terminate such regions since they are no longer active in the representational scheme. A constrained intra-frame update methodology results in areas of the current video frame that are not modelled by any of the existing regions. In such a scenario methods are required to innovate new regions to model these previously unseen areas of the unfolding video sequence.

5.5.1 Termination

In the case where an existing region model is not supported by the data in the current frame the membership of observations to the s'th region is a null-set. Two termination strategies can be adopted in this event. Either region λ_s is terminated at the frame at which there is no supporting evidence or the predicted trajectory information can be used for t' frames before deciding to terminate the track (if no further observations supporting the track are made).

In this work regions are terminated at the frame where there is no supporting evidence,

this leads to a more efficient algorithm since 'blind' tracking (i.e. without observation) often fails to recover a temporarily occluded regions due to the complex, dynamic, nature of video sequences. The total number of regions is updated after the termination stage as $S = S - S'$ where S' is the number of regions terminated.

5.5.2 Innovation

The constrained intra-frame update of regions results in pixels that are not well described by any of the existing region models. A method is required to innovate new region models that represent the newly formed (disjoint) unassigned region. This innovation is achieved by localised application of the initialisation strategy presented in Section 5.3. Let the disjoint unassigned region of video frame t feature space be denoted by $\mathcal{I}_\emptyset$. A method is required to partition this unassigned region into a set of S'' new homogeneous regions i.e.

$$\mathcal{I}_\emptyset = \cup\, \mathcal{I}_s \text{ where } s \;=\; 1, \ldots, S'' \tag{5.18}$$

The S'' innovated regions are found by applying the technique presented in Section 5.3 with a difference that the processing is confined to the unassigned region $\mathcal{I}_\emptyset$ (as opposed to the entire video frame feature space $\mathcal{I}$). This innovation step is reasonably efficient due to the relatively small size of $\mathcal{I}_\emptyset$ per frame. The total number of regions is updated after the innovation stage as $S = S + S''$ where S'' is the number of regions innovated.

5.6 Performance Evaluation of Region-Based Video Segmentation

When performing region-based segmentation of video sequences the type of segmentation extracted depends on the requirements of the application. For the proposed region-based segmentation algorithm the following factors are deemed important:

- *Fidelity*: Ensure that the region PDF models are representative of the video data.
- *Efficiency*: Preserve region homogeneity whilst maximising the area of each region.
- *Stability*: Minimise the innovation of new regions.

These criteria are used as the basis of the metrics by which the performance of the video region segmentation algorithm (and its variants) will be evaluated on a per-frame basis.

To measure the representativeness of the region model, an image-wide error metric is proposed based on the reconstruction error when regenerating the scene from the colour information of the regions. The error metric implicitly models region homogeneity in this context since inhomogeneous (and hence non-Gaussian) regions will not be representative of the underlying colour PDF. The innovation of new regions is measured by analysing the *innovation rate* and *termination rate* of regions. To improve the analysis, these two measures are supplemented by the average size and quantity of the regions per frame in the video sequence.

This section presents the test sequences, performance measurements and experiments that will be applied to evaluate the region-based segmentation algorithm presented in Section 5.2. The actual evaluation results are presented in Section 5.7.

5.6.1 Test Sequences

To evaluate the performance of region-based video segmentation a set of representative test sequences are sought. To allow comparison against previous and future work only standard test sequences that are available freely on the Internet are used. The region-based segmentation algorithm is evaluated for a representative set of four video sequences that present different challenges due to the content in the sequence (e.g. motion, spatial distribution of colour etc).

The first sequence in the test set is the 'Parrot' sequence (18 frames, 256 × 180) that was shown in Figure 4.1, this sequence contains a complex object moving approximately 30 pixels per frame against a cluttered background with a predominantly transitional motion. Frames from the remaining three sequences are shown in 5.10. The second evaluated sequence is 'Table Tennis' (88 frames, 176 × 120), this sequence contains a large zoom motion with many new video region candidates introduced as the scene changes. The third sequence is the 'Foreman' sequence (300 frames, 176 × 144), this sequence is shot from a handheld camera viewing exhibiting minor motion from the camera shaking followed by a major motion as the camera turns to view land adjacent to the individual. The final sequence tested is the 'Coastguard' sequence (100 frames, 176 × 144), this sequence shows two boats (one small, one large) passing each other on a stretch of river, predominantly exhibiting tranlational motion of the objects.

Figure 5.10: Three video sequences used in the evaluation. (Left) frame 00001 from the QCIF sequence 'Table Tennis' (Centre) frame 00001 from the QCIF sequence 'Foreman' and (Right) frame 00001 from the QCIF sequence 'Coastguard'.

5.6.2 Experiments

The experimental analysis of the region-based segmentation algorithm focusses on the inter-frame prediction and intra-frame update mechanisms and the innovation methodology to discover new regions as the video sequence unfolds. More specifically, the variations of the algorithm (introduced in Sections 5.2–5.5) to be evaluated are:

- Intra-frame update of regions:
  - Reinitialisation Strategy (Section 5.4.2).
  - Recursive Filtering Strategy (Section 5.4.2).

- Intra-frame update constraint and innovation of new regions (Sections 5.4.2 and 5.5).
- Inter-frame prediction of regions:
  - Motion-model based compensation strategy (Section 5.4.1).
  - Recursive filtering strategy (Section 5.4.1).

These variants will be evaluated in a sequential manner in that the previous "best" result will be used as a benchmark comparison for the current experiment. From these experiments the characteristics of the different model update strategies can be quantified in the context of video region segmentation (on a per-frame basis).

5.6.3 Performance Metrics

The factors to be evaluated relate to the representativeness, homogeneity, size and lifespan of the video regions. The metrics for the performance evaluation are designed to access these factors without a need for ground-truth segmentation of the image. Ground-truth is avoided since the definition and creation of a definitive region-based ground-truth segmentation is a difficult, if not impossible, task. A human abstracts an image into regions using information not available from a video sequence and as such the ground-truth regions would be unlikely to give a good performance measure of the proposed region based segmentation algorithm.

To measure the fidelity of the representative models and the homogeneity of the video regions a per pixel deviation is computed between the original frame and a frame reconstructed from the models. The reconstructed frame is formed by first segmenting the image into non-overlapping regions using the MAP decision rule per pixel. The pixel value for each segmented region is taken to be equal to the mean colour value of the colour model of that region. The deviation between a pair of observed and reconstructed video frames is measured using a image-wide euclidean RMS error in the colour space i.e.

$$\text{Error}\,(I, R) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\sum_{d=1}^{D}\left[R\,(i,d) - I\,(i,d)\right]^2} \tag{5.19}$$

where I is the source video frame and R is the reconstruction from the region-based models. The pixel index (i, d) represents the d'th colour space dimension of the i'th pixel (out of N total) in the image. The source video frame is converted to the $L*a*b*$ colour space to

match the feature space of the region-based representative models. This measure represents the deviation of the euclidean error between the reconstructed and source video frames.

The average size and average quantity of regions in the video sequences is accessed with the view that a segmentation strategy that generates fewer, larger regions is a more efficient representation of the video data. The average region size for a video frame is computed as N/S where S is the number of extracted regions in the video frame. This result is averaged over the video sequence resulting in the "average average region size" per frame, in the following this is simply referred to as the average region size per frame.

To analyse the stability of the system the innovation and termination rates of regions are included. These innovation and termination rates are determined to be the average number of regions innovated and terminated per frame in the sequence. For a given sequence the innovation and termination rates satisfy the following equality:

$$S_1 + \varphi T = S_T + \rho T \tag{5.20}$$

where φ is the average number of innovated regions per frame, ρ is the average number of terminated regions per frame and T is the number of frames in the sequence. S_1 and S_T are the number of regions in frame 1 and frame T respectively. Therefore, the innovation and termination rates *do not* include the regions 'innovated' and 'terminated' in the first and last frames of the video sequence. The effect of this is that the innovation and termination rates are not neccesarily equal over a video sequence. Inequalities between these two values can be used to determine if the representation is becoming more or less efficient during the sequence e.g. if the innovation rate is higher than the termination rate the number of regions representing the sequence will increase over time.

For region-based segmentation it is ideal to have regions that are mature with minimal innovation and termination of new regions relative to the amount of variability in the sequences.

5.6.4 Algorithm Parameters

To perform the experiments presented in Section 5.6.2 the set of algorithm parameters are defined. Many of these parameters are set empirically to values that gave reasonable performance over a set of training data. The representational model for the regions (described in Section 5.2) uses the default values presented in table 5.1.

Parameter	Description	Value
N_{min}	Minimum region size	7
σ	Kernel density shape model uncertainty	3.0

Table 5.1: The algorithm parameters used in the representational model in the region-based segmentation scheme.

The intra-frame update scheme for the regions (described in Section 5.4.2) uses the default values presented in Table 5.2. The constraint threshold $\tau = 3$ was found to be a good compromise between the representativeness of the model and the preservation of existing regions. Setting $\tau = 1$ allows only observations to be added to the models that will decrease the uncertainty. In practice this was found to remove too many pixels in the MAP update with only minor improvement in the fidelity of the model.

Parameter	Description	Value
τ	Mahalanobis distance threshold in constraint	3.0

Table 5.2: The algorithm parameters used in the intra-frame update strategies in the region-based segmentation scheme.

The inter-frame update strategies have no configurable parameters. In the next section the results of the performance evaluation are presented.

5.7 Results

In Section 5.6.2 the five variations of the segmentation algorithm to be evaluated were introduced. These variations incorporate intra-frame update strategies (reinitialisation and recursive filtering), intra-frame update constraint with region innovation and finally inter-frame prediction strategies (motion-model and recursive filtering). In this section the performance evaluation results are presented in a sequential manner in that the 'best' intra-frame update strategy will be used in the next step etc.

5.7.1 Intra-frame Update Of Regions

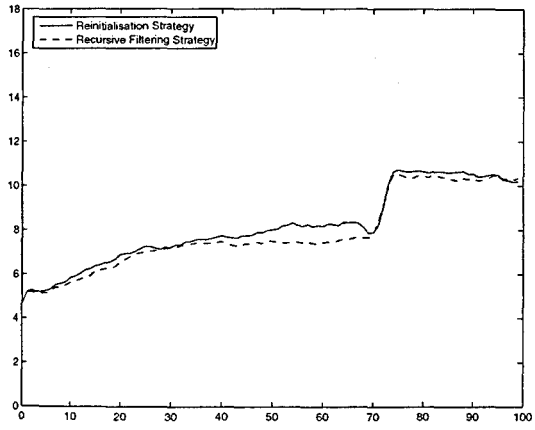
The intra-frame update strategies evaluated were the reinitialisation (Section 5.4.2) and recursive (Section 5.4.2) strategies that adapt the spatial and colour models using the region of support at each frame. The update method is unconstrained in both cases (see Section 5.4.2 for an explanation). An advantage of reinitialising the feature space models per frame is that the model is able to adapt to large appearance changes. For the recursive method the results may be less prone to noise since the models are estimated over a temporal window. For both these methods the 'frame $t-1$ models at frame t' inter-frame prediction strategy (Section 5.4.1) is used.

The RMS reconstruction error for the intra-frame update strategies are shown in Figure 5.11. The reinitialisation and recursive strategies are demonstrated to have similar reconstruction error over the test sequences. For three of the four sequences ('Coastguard', 'Foreman' and 'Parrot') the two strategies appear to offer little advantage over each other in terms of the fidelity and homogeneity of the region-based representation. For the 'Table Tennis' sequence the reinitialisation reconstruction error is lower, this is perhaps due to the motion of the camera zoom requiring a rapid adaption of the model to the previously unseen scene.

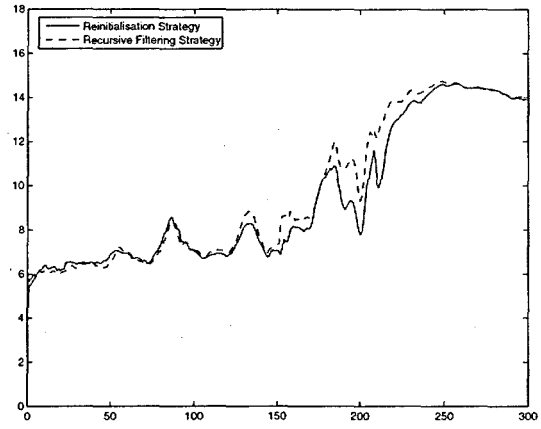
The mean and standard deviations of the RMS reconstruction error are shown in Table 5.3 for the two intra-frame update strategies. It is can be seen that reinitialisation based method of intra-frame update has the lowest reconstruction error for three of the four sequences, although the difference between the two strategies is marginal.

Sequence	Intra-Frame Update Strategy			
	Reinitialisation		Recursive Filtering	
	μ	σ	μ	σ
'Coastguard'	8.12	1.70	7.83	1.71
'Foreman'	9.40	3.13	9.70	3.27
'Parrot'	8.23	0.09	8.38	0.15
'Table Tennis'	11.16	3.26	12.35	4.00

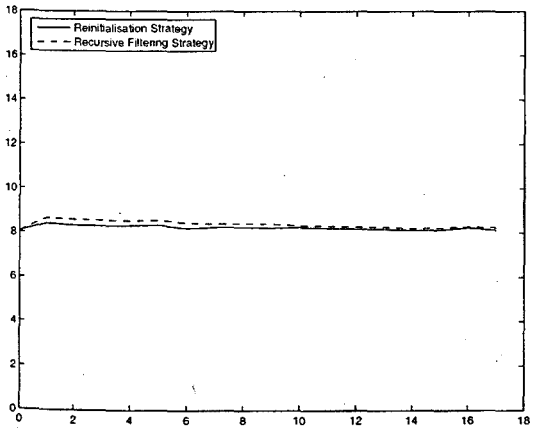
Table 5.3: Average RMS reconstruction error per pixel with unconstrained MAP labelling of regions and reinitialisation/recursive strategy for intra-frame update of the feature models.



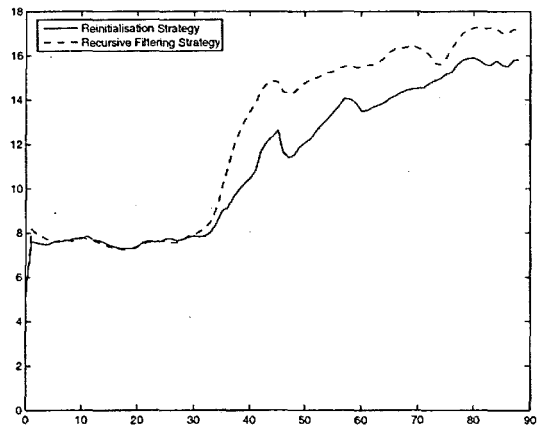
(a) Coastguard



(b) Foreman



(c) Parrot



(d) Table Tennis

Figure 5.11: Average per pixel RMS reconstruction error with unconstrained MAP labelling of regions and reinitialisation/recursive strategy for intra-frame update of the feature models.

The average size and quantity of regions per frame are shown in Table 5.4. For all the sequences the regions increase in size (and decrease in quantity) over the duration of the sequence, this is due to the unconstrained adaption and lack of innovation for the regions. For the 'Foreman' sequence both update strategies result in large poorly-fitted regions, the higher average size of regions reflects this. For the three of the sequences ('Coastguard', 'Parrot' and 'Table Tennis') the reinitialisation strategy results in fewer, larger regions. Given that the fidelity of the models was comparable for the two update strategies, the reinitialisation strategy has been demonstrated to result in a more efficient representation than the recursive update strategy.

Sequence	Intra-Frame Update Strategy			
	Reinitialisation		Recursive Filtering	
	Av. Size	Av. Quant.	Av. Size	Av. Quant.
'Coastguard'	305.36	87.18	296.62	89.57
'Foreman'	986.79	83.98	1339.95	93.34
'Parrot'	153.14	320.56	129.42	366.55
'Table Tennis'	437.91	59.63	398.30	70.33

Table 5.4: Average size and quantity of regions per frame with unconstrained MAP labelling of regions and reinitialisation/recursive strategy for intra-frame update of the feature models.

Reviewing these results, the reinitialisation intra-frame update strategy appears to outperform the recursive filtering strategy for the test data shown. It is proposed that this is due to the temporal variability in the test sequences being such that the recursive strategy does not efficiently adapt the region models to the changes within the unfolding scene.

5.7.2 Intra-Frame Update Constraint and Innovation of New Regions

The use of constraints and region innovation in the intra-frame update step (Sections 5.4.2 and 5.5) is demonstrated in this section by modifying the reinitialisation intra-frame update strategy that was found to give the 'best' performance when applied in an unconstrained update method.

The advantage of constraining the intra-frame update strategy is that previously unseen

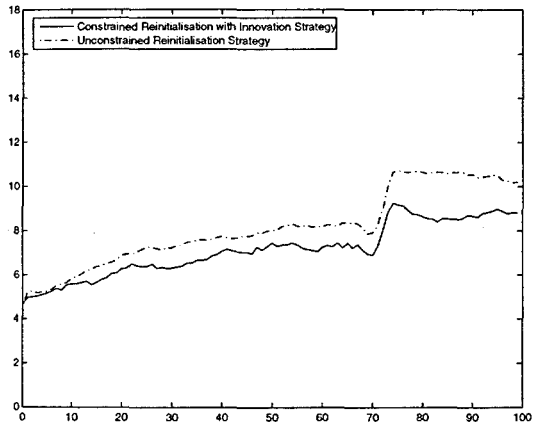
scene regions are no longer arbitrarily classified and subsequently used to update the model. The disadvantage is that previously unseen scene regions will be unassigned leaving portions of the image undefined, therefore a method is also required to innovate new regions to fill these unassigned 'null' regions.

The average RMS scene reconstruction error per pixel is shown in Figure 5.7.2, this demonstrates the improvement in the fidelity of the model that the constrained update provides. The RMS error measure is significantly lower for all the test sequences where there is significant changes in the content of the scene ('Coastguard','Foreman' and 'Table Tennis'). This trend can be seen clearly for the 'Foreman' sequence where new regions are innovated in the final ~100 frames to represent previously unseen scene regions as the camera pans to the right, resulting in a lower RMS reconstruction error.

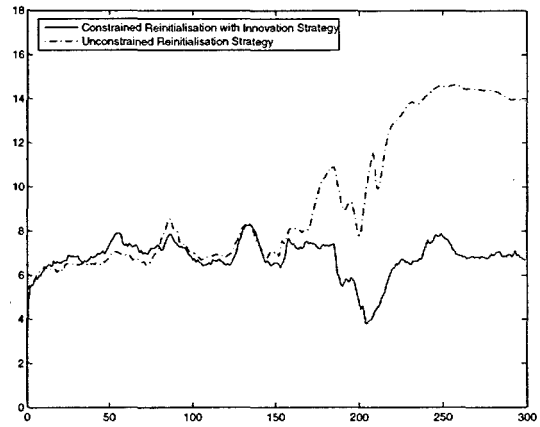
The RMS error appears to be slowly increasing over the duration of the sequences when using the constrained reinitialisation with innovation strategy. This increase in error is perhaps due to limited mis-adaption of the region models due to a conservative estimate of the constraint threshold τ. Alternatively, it is due to an increase in the texturedness of the video sequence resulting in a 'natural' increase in the RMS error. To confirm this a single gaussian was used to model the colour distribution of the 'Table Tennis' sequence as a single region, the average RMS error between the mean of this distribution and the observed pixels was found to increase from 19.3 to 29.0 during the sequence. This result implies that, as expected, the texturedness of the video sequence has a direct effect on the measured scene reconstruction error. To limit this effect it is possible that the RMS error of the region based reconstruction can be normalised by this 'base' RMS error measured when using a single region to represent the video frames.

For the 'Parrot' sequence the constrained and unconstrained update strategies are approximately equivalent with a maximum error discrepancy of 0.4 between the two approaches. Unexpectedly, the constrained method actually results in an increased scene reconstruction error, although the differences between the two methods are marginal.

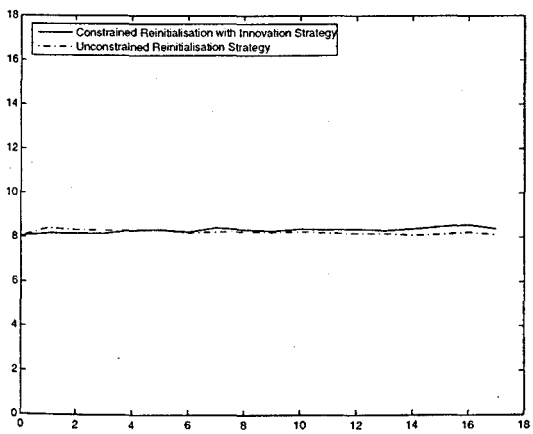
The mean and standard deviations of the RMS reconstruction error are shown in Table 5.5 for the two intra-frame update strategies. The table confirms the trends identified in Figure 5.12, with the constrained update strategy having a significantly lower reconstruction error that the unconstrained strategy over the test sequences. From these results it can be concluded that the constrained strategy results in a higher fidelity region-based



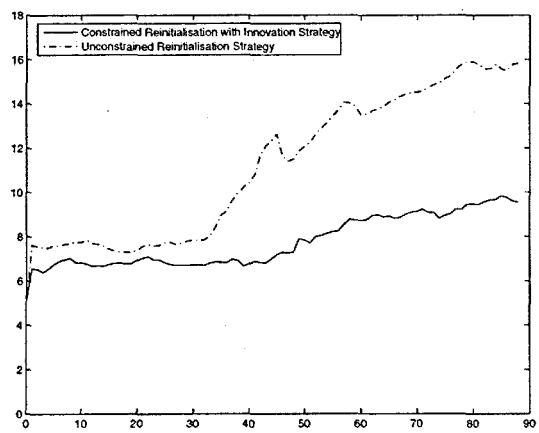
(a) Coastguard



(b) Foreman



(c) Parrot



(d) Table Tennis

Figure 5.12: Average per pixel RMS reconstruction error for the test data with constrained MAP labelling, innovation of regions and reinitialisation strategy for intra-frame update of the feature models. The comparison plot is the unconstrained MAP labelling reinitialisation strategy.

representation that is a better model of the underlying image data.

Sequence	Intra-Frame Update Strategy			
	Unconstrained		Constrained with Innovation	
	μ	σ	μ	σ
'Coastguard'	8.12	1.70	7.13	1.19
'Foreman'	9.40	3.13	6.78	0.81
'Parrot'	8.23	0.09	8.32	0.12
'Table Tennis'	11.16	3.26	7.75	1.15

Table 5.5: Average RMS reconstruction error per pixel for the test data with constrained MAP labelling, innovation of regions and reinitialisation strategy for intra-frame update of the feature models. This is compared to the unconstrained results presented in Table 5.3.

Table 5.6 shows the average size and quantity per frame of regions over the test sequences. It can be seen that the constrained MAP labelling strategy reduces the average region size and increases the quantity of regions when compared to the unconstrained strategy. The average size of the regions for the constrained MAP labelling strategy are all in a similar range between 100–150 pixels, which is an interesting observation given the differing nature of the test sequence content.

Sequence	Intra-Frame Update Strategy			
	Unconstrained		Constrained with Innovation	
	Av. Size	Av. Quant.	Av. Size	Av. Quant.
'Coastguard'	305.36	87.18	144.12	173.87
'Foreman'	986.79	83.98	113.51	224.28
'Parrot'	153.14	320.56	112.62	411.61
'Table Tennis'	437.91	59.63	129.82	163.04

Table 5.6: Average size and quantity of regions per frame for the test data with constrained MAP labelling, innovation of regions and reinitialisation strategy for intra-frame update of the feature models. This is compared to the unconstrained results presented in Table 5.4.

Table 5.7 shows the average innnovation and termination per frame of regions over the

test sequences. It can be seen that with the exception of the 'Parrot' sequence the number of regions created and lost per frame is relatively low when compared to the number of regions at each frame (shown in Table 5.6). This result demonstrates that the constrained MAP labelling strategy coupled with innovation and termination mechanisms produces a reasonably stable representation.

Sequence	φ	ρ
'Coastguard'	15.71	15.6
'Foreman'	28.33	29.0
'Parrot'	56.83	73.55
'Table Tennis'	7.55	7.94

Table 5.7: Average innovation (φ) and average termination (ρ) of regions per frame for the test data with constrained MAP labelling, innovation of regions and reinitialisation strategy for intra-frame update of the feature models.

The 'Parrot' sequence has characteristics not present in the other test sequences. The resolution is the largest of the four test sequences and the objects to be segmented are closer to the camera and are therefore more detailed. This has the effect that the repeated texture in the background takes many region elements to represent and hence while the localised processing is efficient, the algorithm uses more memory to store the region-based representation. The majority of regions that are innovated are terminated represent the smaller scene structures or quantisation/JPEG compression artifacts that are present in the original sequence, this type of noise requires pre-processing of the input image to reduce the effect on the segmentation algorithm.

The benefit of constraining the map update rule and innovating regions is demonstrated further in Figure 5.13. This figure shows the final frame for the 'Table Tennis' and 'Foreman' test sequences. Both of these test sequences undergo a large change in the video content between the first and final frames and hence in the unconstrained reinitialisation scheme the regions represent arbitrary patches of the image due to the nature of the MAP decision rule. With the constrained MAP labelling and innovation the resultant regions qualitatively better represent the data. From these examples the constrained MAP labelling and innovation criteria appear sufficient to model any sequence of arbitrary length, due to the relatively

small number of regions innovated in each frame the algorithm is reasonably efficient.

5.7.3 Inter-Frame Prediction Of Regions

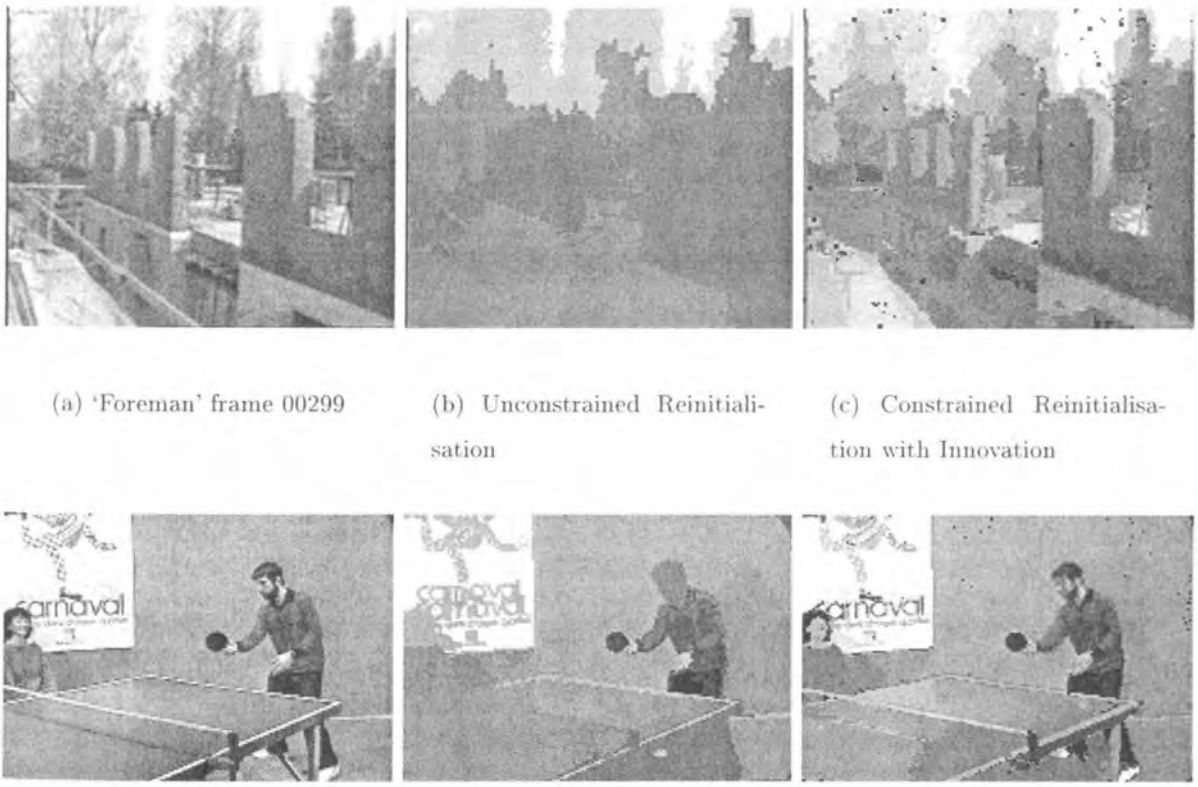
The intra-frame update scheme evaluated in the previous section was demonstrated to preserve the representativeness of the region based representation and effectively constrained the maximum region size to ensure region homogeneity. In the inter-frame prediction scheme a method is sought to predict the region spatial models to minimise the innovation of new regions i.e. improve the matching of regions to observations using prediction of the spatial location. The strategies evaluated were a motion-model based compensation (Section 5.4.1) and a recursive filter (Section 5.4.1). For comparison, the 'frame $t-1$ models at frame t' null prediction step is also evaluated.

The results demonstrate that the inter-frame prediction of the spatial representation of the regions is not a trivial matter. The mean and standard deviations of the RMS reconstruction error are shown in Table 5.8 for the two inter-frame update strategies. It can be seen that the two strategies do not have a major influence the fidelity of the representational scheme, the plotted results (not shown) confirm this with minor fluctuations in error between the strategies.

Sequence	Inter-Frame Update Strategy					
	Frame $t-1$ at t		Motion-Model		Recursive Filter	
	μ	σ	μ	σ	μ	σ
'Coastguard'	7.13	1.19	7.38	1.09	7.32	1.40
'Foreman'	6.78	0.81	6.61	0.71	6.89	0.72
'Parrot'	8.32	0.12	8.28	0.12	8.35	0.13
'Table Tennis'	7.75	1.15	7.56	1.11	7.66	1.17

Table 5.8: Average RMS reconstruction error per pixel for the test data with motion-model based compensation and recursive filtering inter-frame prediction of regions.

Table 5.9 shows the average size and quantity per frame of regions over the test sequences. Again, it can be seen that the two strategies for inter-frame update of video regions have only minor influence on the resulting size and quantity of the regions. Therefore, the inter-frame strategies are an unnecessary layer that does not improve the efficiency



(a) 'Foreman' frame 00299

(b) Unconstrained Reinitialisation

(c) Constrained Reinitialisation with Innovation

(d) 'Table Tennis' frame 00088

(e) Unconstrained Reinitialisation

(f) Constrained Reinitialisation with Innovation

Figure 5.13: Final region based segmentation for each test sequence. (Right) constrained MAP labelling and innovation of regions and reinitialisation strategy for intra-frame update of the feature models. This result is compared to the unconstrained reinitialisation strategy (shown Centre). The regions are represented by the average RGB colour inside the region. Insignificant regions are shown in black for the constrained strategy.

of the region-based representation.

Sequence	Inter-Frame Update Strategy					
	Frame $t-1$ at t		Motion-Model		Recursive Filter	
	Av. Size	Av. Quant.	Av. Size	Av. Quant.	Av. Size	Av. Quant.
'Coastguard'	144.12	173.87	147.04	171.69	141.13	179.12
'Foreman'	113.51	224.28	120.63	212.65	122.68	210.47
'Parrot'	112.62	411.61	116.70	399.11	115.30	404.33
'Table Tennis'	129.82	163.04	133.01	159.01	139.12	153.92

Table 5.9: Average size and quantity of regions per frame for the test data with motion-model based compensation and recursive filtering inter-frame prediction of regions.

Table 5.10 shows the innovation and termination rates of regions when using the motion-model and recursive filtering strategies for inter-frame region propagation. It can be seen that there is no general overall 'best' method from those evaluated for updating the regions in terms of the stability of the region-based representation (i.e. innovation and termination rates). The performance of the inter-frame prediction schemes is determined by the content of the video sequence, out of the three methods tested there is no generic scheme that improves the stability of the representation for all the test sequences. For example, the sequence 'Parrot' contains an object with well defined colour regions undergoing approximately linear trajectory, therefore it seems reasonable that the Kalman filter minimises the innovation and termination of regions.

The different methods have a relatively minor influence on the resulting region segmentation, with the change in innovation and termination rates relatively insignificant when compared to the number of regions in a frame. In general, it is difficult to draw conclusions about which is the 'best' method for updating homogeneous regions with respect to improving the stability of the region based representation. It has been shown that there is no general technique (out of the three tested) that efficiently updates the regions over all the test sequences, in lieu of *a priori* information about the sequence content the simpler 'frame $t-1$ models at frame t' strategy is preferable over the more complex motion-model compensation and recursive filter strategies.

Sequence	Inter-Frame Update Strategy					
	Frame $t-1$ at t		Motion-Model		Recursive Filter	
	φ	ρ	φ	ρ	φ	ρ
'Coastguard'	15.71	15.6	13.44	13.34	18.97	18.7
'Foreman'	28.33	29.0	23.98	24.62	25.49	26.21
'Parrot'	56.83	73.55	52.78	69.89	54.94	72.11
'Table Tennis'	7.55	7.94	7.33	7.63	7.48	8.15

Table 5.10: Average innovation (φ) and average termination (ρ) of regions per frame for the test data with motion-model based compensation and recursive filtering inter-frame prediction of regions.

5.8 Conclusions

In this chapter a region-based segmentation algorithm has been adapted to the problem of video representation. Developing on the probabilistic models introduced in the previous chapter, the per region PDF representation as estimated using an assumption of independence between the colour and spatial features. This region-based representational scheme was adapted to the problem of video-based representation by incorporating inter-frame prediction and intra-frame update mechanisms. The variants of the region-based segmentation algorithm were quantitatively evaluated on a range of representative test data using performance metrics that do not require ground-truth segmentations.

The representational scheme for the spatial-colour regions split the feature space representation into a spatial model and a colour model, using the assumption of independence between these features. The colour feature space PDF for the regions were estimated using Gaussian density models, taking advantage of the homogeneous colour distributions of such regions. The spatial feature space was modelled using an efficient implementation of kernel density models. The initialisation mechanism for the region models required a minimum region size threshold to remove insignificant regions, this minimum threshold was found to be a function of the texturedness of each video frame.

Variations on the intra- and inter-frame algorithms were evaluated on a range of representative test data. These variations incorporate inter-frame prediction strategies (motion-model and recursive filtering) and intra-frame update strategies (reinitialisation and recur-

sive filtering). The reinitialisation based intra-frame update strategy was modified with a constraint mechanism to prevent regions misadapting to form inhomogeneous regions. This mechanism presented a requirement for innovation of new video regions in the sequence to model new scene elements not well modelled by the existing representations.

In this chapter the following contributions were made:

- The intra-frame update constraint with innovation was found significantly change the fidelity and efficiency of the representative scheme.
- A selection of intra- and inter-frame strategies offered discernible benefit over the simpler strategies when updating regions on a per frame basis
- Introduced criteria for the decision rule constraint with innovation and termination procedures. This has been demonstrated to be well suited to the problem of modelling video sequences.
- Evaluated the region-based representation without ground-truth using measures of fidelity, efficiency and stability. Demonstrated the use of an RMS reconstruction error to measure the fidelity of the representation.

The variations of the propagation mechanisms for the region-based segmentation scheme were evaluated quantitatively without using ground-truth. The evaluation process measured the fidelity, efficiency and stability of the region-based representation. The more complex intra- and inter-frame strategies offered discernible benefit over the simpler strategies when updating regions on a per frame basis. This is due to the temporal variability of the regions, which makes the evolution of region-based representational models a challenging task. It was found that the intra-frame update constraint with termination and innovation of regions made the most significant change in the fidelity and efficiency of the representative scheme. By limiting the adaption of the regions the resulting region-based representation was more representative of the content of the video, at the expense of reducing the average lifespan of regions through termination and innovation of regions.

A key issue when performing region-based modelling of video sequences is how to predict the regions on a per frame basis to minimise the adaption, innovation and termination of regions. This problem is difficult due to the sensitivity of the region's spatial location to

the changing appearance of the video, this makes it difficult to propagate such regions on a per frame basis. A better strategy may be to predict the region-based representation by analysing higher-level information in the video sequence e.g. the motion of objects formed by grouping regions into higher-level entities with semantic meaning.

The update of region-based representational schemes in the context of video region segmentation has been demonstrated to be a challenging problem. It has been shown that the prediction and update of regions is not significantly improved by using more complex strategies. The constraint of the model update step was found to have the largest influence on the representativeness of the models. This constraint required mechanisms for innovation and terminating algorithms, which were demonstrated to retain the fidelity of the overall region-based representation. In the following chapter the inter-frame prediction of regions is performed using object-level information in an attempt to overcome the difficulties found when performing the prediction at the region-level. This is applied in a hierarchical framework to allow different levels of interaction between the regions and parent objects. The criteria for innovation and termination is extended to video objects, the problem of automatically innovating new video objects in an unfolding scene is one of the fundamental challenges in video object segmentation.

Chapter 6

Hierarchical Bayesian Framework for Video Object Segmentation

In the previous chapter a region-based segmentation algorithm was presented. This segmentation algorithm used the representational models presented in Chapter 4 and the 'best' performing feature space evaluated in Chapter 3. Popular methods for propagating the regions were investigated along with a methodology for innovating and terminating regions as the video sequence unfolds. It was found that while the innovation and termination components of the algorithm were adequate, more sophisticated methods for propagating the regions were found to give little advantage over simpler techniques. It is believed that this was due to the sensitivity of spatial-colour regions to the motion of objects within the scene, making such regions difficult to propagate on a per frame basis.

In this chapter methods are sought for propagating the region-based representation in an efficient way to perform video object segmentation. The membership between objects and regions can be determined for the first frame in the sequence using the mask information provided by the user. A key issue in the object-based segmentation algorithm is how to update the region-to-object memberships with the newly innovated regions, and how to innovate new objects in the process.

To solve these problems a hierarchical framework is proposed, within which relationships between video objects and regions are employed to improve the propagation of the representational models. It is envisaged that such a hierarchical framework allows variants of the region-based propagation strategies to be evaluated, some of which use object-based

tracking and/or motion compensation. Algorithms presented in the previous three chapters will be implemented in this framework.

6.1 Previous Work

There is limited prior work on hierarchical modelling of video objects. Such modelling can apply to any technique where the object is represented at a coarser or finer paritition of the video frame than object level. In some approaches the term 'hierarchy' is often used to describe algorithms where there is a label correspondence between a region-level representation and parent object e.g. Marques and Llach [112]. Salembier *et al* [152] introduced the concept of a parition tree to give a hierarchical description of the video scene where regions at a level in the hierarchy can be split to give regions at the next lower level.

For some approaches the hierarchical description is intrinsic to the algorithm used to represent the video objects. For example, Piroddi and Vlachos [138] applied a Recursive Shortest Spanning Tree [123] (RSST) region merging algorithm to grow intensity-motion-texture regions which are merged within motion boundaries using the RSST algorithm and then a set of rules are applied to merge regions across motion boundaries. Similarly Cooray *et al* [149] applied a RSST to find homogeneous colour regions from which a Binary Partition Tree was created to enable efficient browsing of video regions by a user. Tuncel and Onural [166] applied the RSST algorithm to find the motion segmentation of a video by estimating the parameters of an affine motion model while in earlier work Alatan, Tuncel and Onural [6, 5] applied the RSST algorithm with a rule-based approach to perform joint colour and motion segmentation.

A similar hierarchical clustering scheme is applied by Porikli [139] to group regions based on motion similarity and other constraints to allow the analysis of object properties using graph theory methods. Another advantage of such a structure is that the segmentation at the lowest (region) level does not require recomputation when new definitions of objects are introduced at the higher level. Content-based video retrieval work by Fu *et al* [73] explored the use of hierarchy to describe scenes using an interactive mapping of low-level motion features into semantic descriptors and also used co-ordinate system transforms to measure the temporal consistency of motion.

None of these previous approaches set out to describe a hierarchical Bayesian framework where distinct model levels are combined to perform video object segmentation and tracking.

6.2 From Regions to Objects

In the previous chapter techniques were presented for the segmentation of video using regions that may or may not have semantic meaning. In this chapter the original problem of video object segmentation is considered. Therefore it is important to define the set relationships between these layers. Objects are defined to be sets of regions and the video frame is defined to be sets of objects. The current video frame $\mathcal{I}$ decomposes into the set of video objects i.e.

$$\mathcal{I} = \cup\, \mathcal{I}_r \text{ where } r \;=\; 1, \ldots, R \tag{6.1}$$

Where $\mathcal{I}_r$ is the portion of the video frame assigned to the r'th object (with R objects in total). In the hierarchy the video objects have a membership to this parent video frame i.e.

$$M_{\mathcal{I}}^{\Lambda} = \{\Lambda_1, \ldots, \Lambda_r, \ldots, \Lambda_R\} \tag{6.2}$$

where $M_{\mathcal{I}}^{\Lambda}$ represents the member set between objects Λ and the parent image $\mathcal{I}$. Membership of regions to a parent object Λ_r can be written as:

$$M_{\Lambda_r}^{\lambda} = \{\lambda_1, \ldots, \lambda_s, \ldots, \lambda_{S_r}\} \tag{6.3}$$

where $M_{\Lambda_r}^{\lambda}$ represents the member set between regions λ and the parent object Λ_r. S_r represents the total number of regions that have membership of the r'th object Λ_r. Similarly individual pixels also have membership of the video objects within the scene. In the following section a hierarchical framework is introduced to exploit the sharing of information between object and region level representations.

6.3 Hierarchical Bayesian Framework for Video Object Segmentation

In this section a hierarchical Bayesian framework encompassing region- and object-level video object segmentation is proposed. Within the hierarchy two distinct layers are combined to perform video object segmentation. The higher layer in the hierarchy is the object-based representational model and propagation, whereas the lower layer contains the region-based representation. Implicitly, there are two extra layers (image and pixel level) in the hierarchical framework that can be defined by simple label memberships of the region and object layers. The proposed hierarchical framework is shown in Figure 6.1. The framework contains information flow between the two levels of the hierarchy which are termed the feed-downward and feed-upward links. The form of these information links is dependent on the implementation of the framework. For example the object-level prediction can be used to modify the region-level models prior to prediction. This framework provides the flexibility for the integration of new algorithms for video object segmentation.

For the case where the object- and region-level processes are independent there is no interaction between the representative models of the two levels i.e. the feed-down and feed-up links are removed from Figure 6.1. The two level processes could then be combined at the labelling stage to generate a partition of the current video frame into objects. An example of an independent hierachical framework is an object-level motion segmentation algorithm that uses label correspondence with a region-level colour segmentation scheme to improve the boundary accuracy based on the assumption that colour segments are subsets of motion segments e.g. Altunbasak *et al* [7].

In the scenario where the object-level representional model is the less accurate approximation of the underlying density and the region-level is the more accurate approximation then it would be expected that the region-level segmentation would generate a more accurate segmentation of the object (given per-object correspondence for the regions). In this case it is clear that the feed-upward of the region-level segmentation to update the object-level model would make the object-level a closer representation of the true object location.

Conversely, it is envisaged that the object inter- or intra-frame representation could also be used to improve the update of the regions, for example, object level descriptors

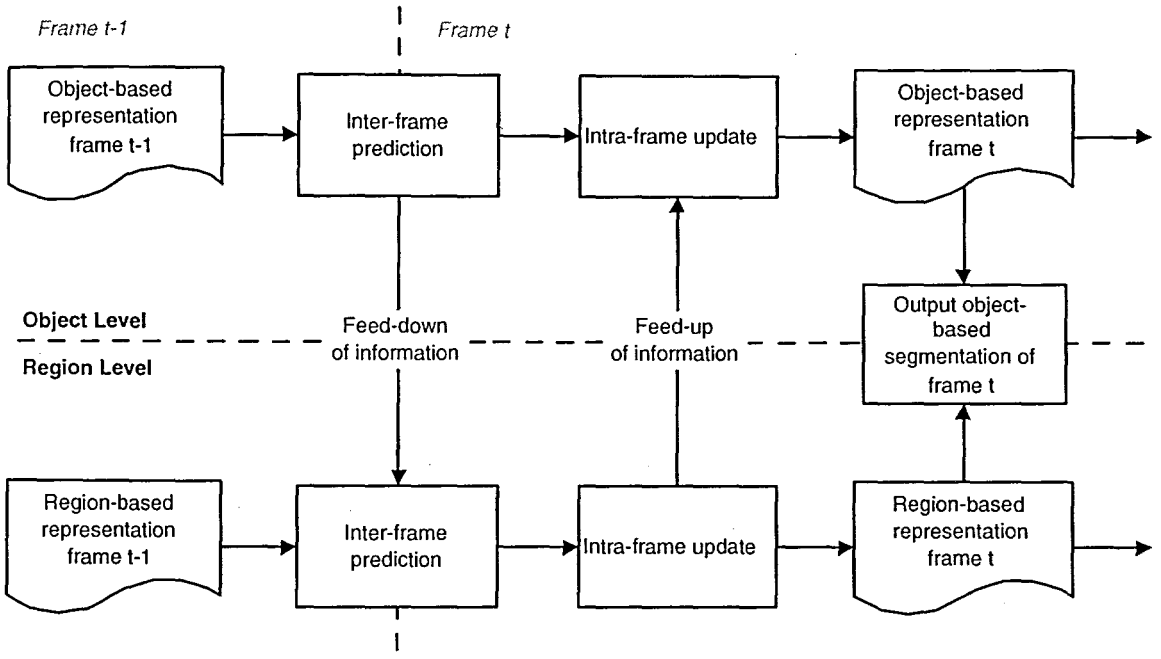


Figure 6.1: Hierarchical video object segmentation framework showing both feed-down and feed-up information links between the layers and joint labelling of the output result. The intermediate representational models and feature vector extraction process are not included for clarity.

(e.g. dominant colours) can be used to assign newly generated regions as belonging to that object representation. In such a scenario the sharing of information may be a two way process, with the possibility of optimisation schemes to perform iterative estimation of the representational models between the two layers.

The application of Bayesian methods to the framework allows the issues of video object tracking to be approached in a principled and formal way. At a frame in the video sequence an observed feature vector $\mathbf{a}_t(x, y)$ is measured at each pixel (x,y) in frame t of the sequence. For brevity the subscripts are dropped such that this referred to as $\mathbf{a}$ in equations.

By applying Bayes rule the probability that a pixel belongs to one of the R video objects can be computed for each level of the hierarchy. At the object-level (denoted by a subscript Λ), the probability of a pixel observation, $\mathbf{a}$, belonging to a particular object, Λ_i, is given by:

$$P_\Lambda(\Lambda_i|\mathbf{a}) = \frac{p_\Lambda(\mathbf{a}|\Lambda_i)P_\Lambda(\Lambda_i)}{\sum_{r=1}^{R} p_\Lambda(\mathbf{a}|\Lambda_r)P_\Lambda(\Lambda_r)} \tag{6.4}$$

where $p_\Lambda(\mathbf{a}|\Lambda_r)$ represents the conditional probability of the r'th object in the video scene.

At the region-level (denoted by a subscript λ), the probability that an observed feature vector at a pixel, $\mathbf{a}$, belongs to a particular region λ_i can be computed thus:

$$P_\lambda(\lambda_i|\mathbf{a}) = \frac{p_\lambda(\mathbf{a}|\lambda_i)P_\lambda(\lambda_i)}{\sum_{s=1}^{S} p_\lambda(\mathbf{a}|\lambda_s)P_\lambda(\lambda_s)} \tag{6.5}$$

Given that each region also has membership of an object (see section 6.2 for the set relationships) it follows that object-level probabilities can be computed by combining region-level probabilities. At the region-level (denoted by a subscript λ), the probability of a pixel observation, $\mathbf{a}$, belonging to a particular object, Λ_i, is given by:

$$P_\lambda(\Lambda_i|\mathbf{a}) = \frac{p_\lambda(\mathbf{a}|\Lambda_i)P_\lambda(\Lambda_i)}{\sum_{r=1}^{R} p_\lambda(\mathbf{a}|\Lambda_r)P_\lambda(\Lambda_r)} \tag{6.6}$$

where the conditional probability is computed using the total probability theorem:

$$p_\lambda(\mathbf{a}|\Lambda_i) = \sum_{s=1}^{S_i} p_\lambda(\mathbf{a}|\lambda_s)P_\lambda(\lambda_s) \tag{6.7}$$

where λ_s is the s'th region out of the S_i regions that have membership of the object Λ_i.

To combine the two layers in the hierarchy (denoted by no subscript) the joint probability that a pixel observation 'belongs' to an object can be computed. This is simplified under the assumption that the processes at each layer are statistically independent i.e.

$$p(\mathbf{a}|\Lambda_i) = p_\Lambda(\mathbf{a}|\Lambda_i)p_\lambda(\mathbf{a}|\Lambda_i) \tag{6.8}$$

Applying Bayes rule the probability that a newly observed pixel 'belongs' to an object is therefore computed as:

$$P(\Lambda_i|\mathbf{a}) = \frac{p(\mathbf{a}|\Lambda_i)P(\Lambda_i)}{p(\mathbf{a})} \tag{6.9}$$

where, as before:

$$p(\mathbf{a}) = \sum_{r=1}^{R} p(\mathbf{a}|\Lambda_r)P(\Lambda_r) \tag{6.10}$$

For all hierarchical layers the prior probabilities can be computed directly from the representational models, although it is more efficient to calculate these from the most recent observed label maps. In the next section the representational models and intra-/inter-frame update schemes are described.

In the next section implementations of the hierarchical framework are presented. These algorithms contain various configurations of the hierarchical framework. A mechanism for innovating new objects in the scene is also introduced. This mechanism allows the segmentation algorithm to run as an automated process and can be used in focus of attention strategies to alert the human operators that the segmentation process requires a new key-frame segmentation (e.g. a new object has appeared in the scene).

6.4 Variants of the Hierarchical Bayesian Framework

In this section several implementational aspects of the hierarchical framework are considered in the context of semantic video object segmentation. These variants focus on the problem of updating the representational models of the video objects so that they can be adapted to the changing object appearance over time. The four main implementational aspects to be addressed in this chapter are:

- Feature Space and Representational Models (section 6.4.1).
- Region-level representational models with object-level prediction (section 6.4.2).
- Interacting region-level and object-level representational models (section 6.4.3).
- Object-level innovation (section 6.5).

The feature space and representational models described in section 6.4.1 are chosen using the prior work in Chapters 3 and 4. The feature space and representational models are common to all configurations of the framework. The configurations of the framework described in sections 6.4.2 and 6.4.3 represent alternative techniques to achieve the same goal, that is, the segmentation of video into objects. The final configuration (section 6.5) deals with the innovation (i.e. creation) and termination of video objects during a video sequence, this is a difficult challenge for any segmentation algorithm where the objects to be segmentated are defined by semantics.

6.4.1 Feature Space and Representational Models

The feature space used is the $XYL*a*b*$ space. This combination of colour and spatial information was determined to give the 'best' object segmentation accuracy in Chapter 3. This feature space is modelled as a joint distribution of the chromatic signal, $\mathbf{f}$, and the spatial signal, $\mathbf{x}$ i.e.

$$p(\mathbf{a}) = p(\mathbf{f})p(\mathbf{x}) = p(L*, a*, b*)p(x, y) \tag{6.11}$$

In this chapter the assumption of statistical independence between the colour and spatial information is made for both the object-level and region-level models. For the test sequences used, this assumption has a negligible effect on the resulting segmentation of the video

sequence. It follows that in sequences with several objects of similar appearance and spatial location, such models for locating the objects may fail. In such a scenario extra processing (perhaps feature tracking methods e.g. [157]) may be required to separate the objects. The variants of the framework apply models to represent the appearance and location the object- or region-level components, although in some configurations these models are not required e.g. section 6.4.2 does not model appearance or location at the object-level of the hierarchy.

Object-Level

At the object-level of the hierarchy, a parametric model is used when describing the appearance and location of the object. For an object Λ_i the observed chromatic distribution at the object level is modelled using a Gaussian mixture model. The Gaussian mixture model is chosen since it allows an efficient representation of the multi-modal colour distribution of the object, with fewer parameters required than the equivalent kernel density model. Section 4.4.2 demonstrated that the two modelling techniques were both suitable for modelling the colour distribution of video objects. The chromatic GMM contains K trivariate densities of the form:

$$p(\mathbf{f}|\boldsymbol{\theta}_k) = \frac{\exp\left[-\frac{1}{2}(\mathbf{f}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{f}-\boldsymbol{\mu}_k)\right]}{(2\pi)^{\frac{3}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \tag{6.12}$$

where $\boldsymbol{\Sigma}_k$ is the covariance matrix and $\boldsymbol{\mu}_k$ is the chromatic mean of the k'th Gaussian cluster, $\boldsymbol{\theta}_k$. At the object level the class conditional probability of a pixel observation, $\mathbf{f}$, given the colour density model for object Λ_i is therefore given by:

$$p_\Lambda(\mathbf{f}|\Lambda_i) = \sum_{k=1}^{K_i} p(\mathbf{f}|\boldsymbol{\theta}_k)P(\boldsymbol{\theta}_k) \tag{6.13}$$

where K_i represents the number of clusters in the GMM that represents the object Λ_i. The spatial information of the object is represented using a single bivariate Gaussian distribution. This model is selected since it provides reasonable localisation of the object for the tested sequences, as demonstrated in Section 4.4.1. The conditional probability of a spatial observation, $\mathbf{x}$, given the spatial density model for object Λ_i is therefore given by:

$$p_\Lambda(\mathbf{x}|\Lambda_i) = \frac{\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]}{2\pi|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \tag{6.14}$$

where μ and Σ are the mean and covariance of the bivariate Gaussian distribution.

Region-Level

At the region-level of the hierarchy, a combined parametric and non-parametric representational model is used to achieve accurate pixel-wise segmentation. For each region the spatial distributions are modelled using non-parametric kernel density models and the chromatic distributions are modelled using a multivariate normal density. This type of combined representational model was applied to the problem of video region segmentation in the previous chapter. The observed chromatic distribution for a region λ_i, given a observation $\mathbf{f}$, is modelled using a trivariate normal density of the form:

$$p(\mathbf{f}|\lambda_i) = \frac{\exp\left[-\frac{1}{2}\left(\mathbf{f}-\mu_i\right)^T \Sigma_i^{-1}\left(\mathbf{f}-\mu_i\right)\right]}{(2\pi)^{\frac{3}{2}}\left|\Sigma_i\right|^{\frac{1}{2}}} \tag{6.15}$$

where Σ_i is the covariance matrix and μ_i is the chromatic mean of the representation of the colour distribution for the region λ_i. The probability density of a spatial observation $\mathbf{x}$ for a particular region λ_i is calculated using a bivariate Gaussian kernel density of the form:

$$p_\lambda(\mathbf{x}|\lambda_i) = \frac{1}{N_i}\sum_{n=1}^{N_i}\frac{1}{2\pi\sigma^2}\exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|^2}{2\sigma^2}\right\} \tag{6.16}$$

where $\mathbf{x}_n$ are chosen as N_i pixel observations made within the crisp partition of the region λ_i. Bayes theorem is applied to classifying pixels in the scene with the most probable object or region label using the MAP rule (presented previously in 4.5).

To initialise the representational models the user provides binary masks to define the semantic object regions at the first key-frame in the sequence. From the frame data and object masks the object-level models are formed. To initialise the regions a method based on that shown in the previous chapter is used, except that the regions are generated per object mask as opposed to over the whole video frame.

6.4.2 Region-Level Representational Models with Object-Level Prediction

In this section one variant of the hierarchical framework is described. In this variant the object-level does not have a representational model for the appearance and location of the object. Instead, the object-level spatial and colour representation is described using the

membership information that allows the conversion between regions and objects (described in section 6.3). In this way, the regions are aggregated into objects from which higher level analysis can be performed to extract other forms of object-level representation (e.g. the motion parameters of the object). The region level representation can then be predicted in the video sequence by applying this higher level information extracted at the object-level. In this section three distinct strategies are discussed for performing this type of prediction:

- 'Frame $t-1$ Models at Frame t' (COPY variant).
- Motion-Model Based Compensation (AFFINE variant).
- Recursive Filtering (KALMAN variant).

Figure 6.2 shows the implementation of the framework for these approaches. The object-level representation contains either nothing, the motion model parameters or the recursive filter parameters. This information is fed-downward to be used in the region-level prediction strategy. The region models are used to label the video frame at the region-level and at the object-level (using the membership information). The region-level segmentation is used to re-initialise the region-based representational models in the (constrained) intra-frame update stage. The object-level segmentation is used to re-estimate the object-level representation (if it exists).

The three strategies are based on the region-level prediction strategies described in section 5.4.1 in the previous chapter:

'Frame $t-1$ Models at Frame t' (COPY variant)

In this strategy the region models are not predicted. The propagated region PDF models at time t are the estimated PDF models from the previous frame $t-1$.

Motion-Model Based Compensation (AFFINE variant)

This prediction strategy projects the spatial component of the region models between frames such that the models are geometrically transformed to account for the motion of the objects within the scene. The motion-model is fitted to per pixel optical flow for each object to generate per object motion models. For each object the motion model parameters are

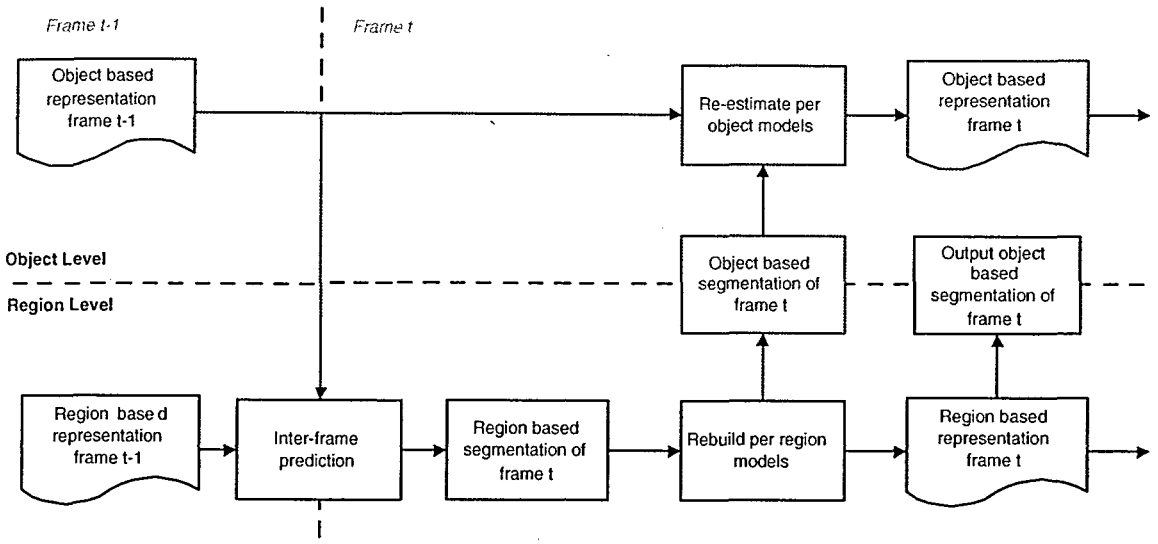


Figure 6.2: A hierarchical framework configuration for object-level prediction of region-level models. In the case where the prediction strategy 'Frame $t-1$ Models at Frame t' is used, the object-level prediction components (dark-grey in the framework) are not used.

estimated to describe the general motion of the object between adjacent video frames. This motion model can then be used to transform the spatial PDF model for each region using the location of the region relative to the parent object. A suitable motion model is the six parameter affine model(see section 5.4.1 for a detailed explanation of this motion model).

Recursive Filtering (KALMAN variant)

Alternatively, the prediction of regions can be performed using recursive filtering techniques. The Kalman filter is applied to propagate regions using object level state estimation. The object information found at frame $t-1$ are used to update the state information of the recursive filter (which contains centroid and bounding box information). The parent object trajectory model is subsequently used to transform (i.e. propagate) the region spatial models using the predicted object trajectory between the two frames. See section 5.4.1 for more information.

These three distinct strategies enable the region-level spatial models to be predicted using higher level information recovered from the object membership information. In the following section an implementation of the hierarchy is proposed where dual representational models at the region- and object-level are interacted to introduce higher-level information

into the region-level process.

6.4.3 Interacting Region-Level and Object-Level Representational Models (INTERACTING variant)

In this second implementation of the framework a combination of non-parametric and parametric models are applied with local co-ordinate systems to perform video object tracking. The implemented framework is shown in Figure 6.3. In this framework the feed-downward step of the algorithm uses the object-level models to compute a co-ordinate system for the region models that is localised on the object. The regions are therefore propagated by the object-level to a predicted location in the new frame assuming a rigid transformation. Finally, these regions are reinitialised in the intra-frame update and the resulting object-level label map is used to reinitialise the object-level models to provide a precise localisation of the object.

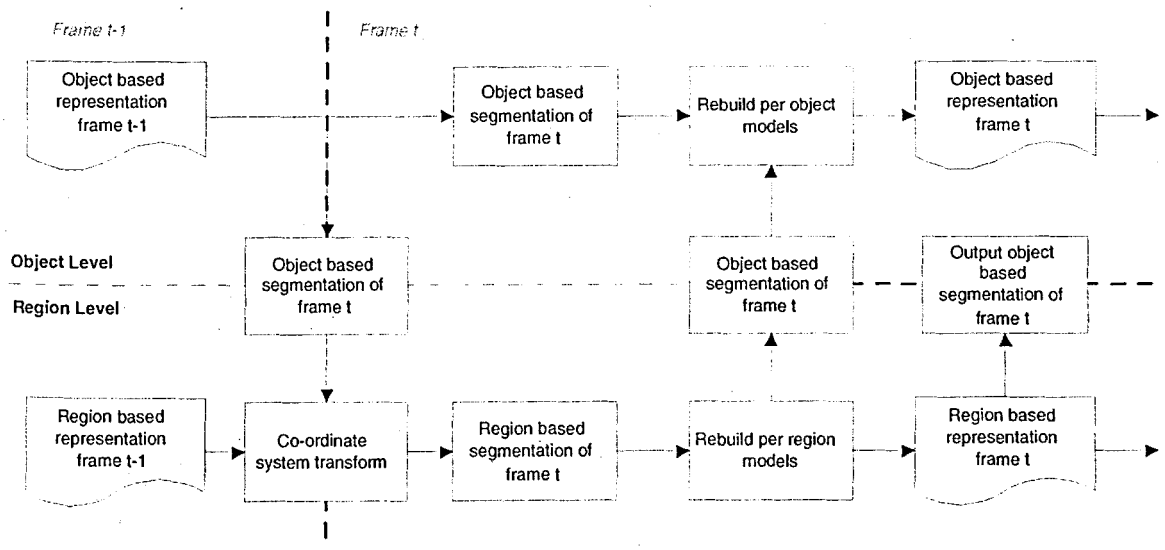


Figure 6.3: A hierarchical framework configuration for interacting object- and region-level spatial-colour models.

Co-ordinate System Transform

The inter-frame update step at the object level is a 'null' step, in that the objects at frame $t-1$ are used to locate the object in frame t with no explicit propagation. The result of this is used to label the current observed feature vectors in frame t. The spatial moments of

the video object can subsequently be estimated from the label mask, which can be used to derive a co-ordinate system with the co-ordinate system aligned with the major and minor axes of object. The spatial location of the feature vectors observed at the region-level are subsequently pre-processed using a Hotelling transform (the discrete form of the Karhunen-Loeve transform [51]) to locate them in the co-ordinate system of the moments of the spatial distribution of the parent object. This transformation provides invariance to rotation, scale and translation of the parent object. Figure 6.4 shows the effect of the transform on a video object extracted from the 'Parrot' sequence:

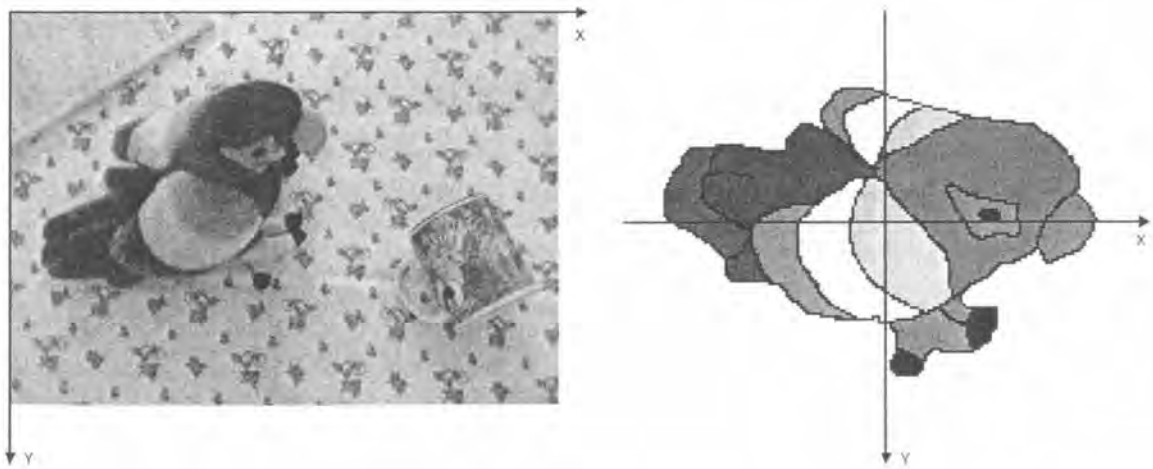


Figure 6.4: Hotelling transform on the foreground object's spatial-colour regions. (Left) A frame from the 'Parrot' seqeunce (Right) The result of the transformation.

The Hotelling transform is computed as follows. For an object Λ_i in the scene we calculate, at the object level, the mean vector and covariance matrix of the spatial distribution of the object. Because the covariance matrix, $\mathbf{\Sigma}_{\Lambda_i}$, is real and symmetric, finding a set of orthogonal eigenvectors is possible. Let $\mathbf{E}$ be a matrix composed of the eigenvectors of $\mathbf{\Sigma}_{\Lambda_i}$ in descending order of eigenvalue magnitude, then $\mathbf{E}$ is a transformation that maps a spatial vector $\mathbf{x}$ from the scene co-ordinate system to that of the object Λ_i as follows:

$$\mathbf{x}_{\Lambda_i} = \mathbf{E}\left(\mathbf{x} - \mu_{\Lambda_i}\right) \tag{6.17}$$

Equation 6.17 represents the Hotelling transform. The mean vector of the distribution of the spatial vectors resulting from this transformation is zero, and the covariance matrix is a decorrelated matrix whose elements along the main diagonal are the eigenvalues of $\mathbf{\Sigma}_{\Lambda_i}$.

The reverse of the Hotelling transform allows a pixel value to be transformed back into the scene co-ordinate system from the object coordinate system using the relation:

$$\mathbf{x} = \mathbf{E}^T \mathbf{x}_{\Lambda_i} + \boldsymbol{\mu}_{\Lambda_i} \tag{6.18}$$

Therefore the region models in the co-ordinate system of the object can be tracked, updated and innovated; this gives invariance to rotation, scale and translation of the parent object. Of course, it is clear than when objects do not have clearly defined major and minor axes (i.e. the object is near-circular in its distribution) then the co-ordinate transform may be unstable. To counter this the co-ordinate system can be 'frozen' if the minimum and maximum eigenvalues of the covariance matrix Σ are of a similar magnitude. This method of co-ordinate system localisation will also be unreliable in the case of severe occlusion of the object or rapid changes in object pose. The spatial distribution will not be representative of the shape of the object and hence the minor and major axes (defined by the eigenvectors) may not correspond to the object in a consistent manner.

Prediction of the Region-Level Models

With the models from frame $t-1$ (under the assumption that the objects do not move a significant distance between frames) an object level MAP rule is applied to calculate the object support in frame t. From the newly observed object moments the reverse Hotelling transform is performed on the region spatial models, from which a pixel-wise segmentation is determined in the image plane. From this partition of the frame the region and object level models are updated and both hierarchical level models are subsequently propagated to the next frame. The final segmentation result for the frame is taken from the sub-object layer since this is the more accurate representation of the video object (using the membership relationships between layers).

These variants of the hierarchical framework provide mechanisms for tracking video objects on a per-frame basis. A problem encountered is that with dynamic changes in the video content, the appearance of objects can change and objects can be removed from or added to the sequence. In section 5.5 a mechanism for region-level innovation and termination was described, this allowed the representation of the video frame to evolve to changes in the content. In the following section this mechanism is expanded to provide object-level innovation i.e. to discover new objects in the video sequence.

6.5 Termination and Innovation of Objects

In a video sequence objects are continually added and removed from the cameras field-of-view. This poses a problem for supervised approaches to object segmentation since the user is required to intervene to manually add and remove such objects. To reduce such intervention (or, in the longer term, remove it completely) mechanisms are required to *automatically* innovate and terminate objects during the sequence.

An additional problem in innovating and terminating objects is the problem of occlusion. An object undergoing total occlusion will be terminated and subsequently re-innovated within the scene. To prevent the creation of an extra object a long term tracking algorithm could be implemented to 'stitch' the new and old objects together into a continuous trajectory. The problem of complete occlusion is outside the scope of this work. The innovation algorithm presented below does provide limited robustness to partial occlusion. The region to object association algorithm allows occluded object regions to be correctly re-associated with the parent object when they become visible.

In the following, methods are presented to perform the termination and innovation of objects within the video sequence. These algorithms allow the video object segmentation algorithm to run on sequences of arbitrary length, although the extracted objects may only have semantic meaning in constrained scenarios.

Termination

An object is terminated when no child regions remain i.e. $M^{\lambda}_{\Lambda_r} = \emptyset$. The total number of objects is updated after the termination stage as $R = R - R'$ where R' is the number of objects terminated.

Innovation

A major difficulty with innovating objects is that the objects are defined to have semantic meaning, and as such may not exhibit any homogeneous properties in the feature space. The objects are segmented at key frames in the video sequence by human operators, described in section 2.2. It is currently impossible to develop a system to segment objects that correspond to those seen by the human visual system except for highly constrained scenarios. To achieve such segmentation the system would need greatly improved interpretation of the

scene, perhaps with complex 3D modelling and abstraction of object types. Without loss of generality, it is possible to circumvent many of these difficulties by making assumptions about the underlying properties of the objects. It is proposed that any newly visible regions of an object will match appearance with at least one other visible part of the object. This assumption is only valid when the object viewed has a hidden appearance similar to the appearance that is currently visible.

To apply this assumption in the framework the visible object parts are taken to be the region-level models (each with a parent object membership) such that new object parts are formed from newly innovated regions. These new regions are created using the technique described in section 5.5, where unlabelled areas of the image are segmented into new homogeneous regions. In this way an intra-frame mechanism is required to match (i.e. associate) newly generated regions to existing regions. Once matched, the new region will inherit the parent object membership information from the existing region. Any remaining unmatched regions are clustered into new objects.

A validation gate is used to associate newly innovated regions to existing regions by comparing the proximity of the regions in the multi-dimensional feature space. The effective size of the gate is determined by a threshold T on the normalised squared distance between two regions λ_i and λ_j (i.e. a new and existing region):

$$d^2(\lambda_i, \lambda_j) = \left[\mu_{\lambda_i} - \mu_{\lambda_j}\right]^T \mathrm{S}^{-1} \left[\mu_{\lambda_i} - \mu_{\lambda_j}\right] \tag{6.19}$$

where $\mathrm{S} = \Sigma_{\lambda_i} + \Sigma_{\lambda_j}$ is the combined covariance of the two regions. μ and Σ represent the mean and covariance of colour and spatial Gaussian models for each region. For the spatial distribution of a region, the mean and covariance of the kernel density model were computed from the observed kernel distribution. For the colour distribution the region's model was a Gaussian, hence the mean and covariance estimates were known.

An appropriate gate threshold for equation (6.19) can be determined from tables of the Chi-squared distribution, with the degrees of freedom given by the dimensionality (= 5) and required significance (= 0.95). A newly innovated region λ_i is matched with an existing region λ_j if:

$$\begin{aligned} d^2(\lambda_i, \lambda_j) &< T && \text{and} \\ d^2(\lambda_i, \lambda_j) &< d^2(\lambda_i, \lambda_s) && \forall s \neq j; s \in \{1, \ldots S\} \end{aligned} \tag{6.20}$$

This nearest neighbour matching strategy is evaluated between the newly innovated region λ_i and the S existing regions. The new region will inherit the parent object information of the closest region found to be within the gate (i.e. the validated nearest neighbour).

Regions not matched by this process are now clustered into potential new objects using the following suboptimal algorithm:

1. Unmatched regions are visited in descending order of size.
2. For the current unmatched region, a new object is created.
3. The nearest neighbour strategy (described above) is used to match any remaining regions to this new object.
4. Repeat until no unmatched regions remain.

The mechanisms described allow new (innovated) regions to be added to existing or new (innovated) objects. The total number of objects is updated after the innovation stage as $R = R + R''$ where R'' is the number of objects created.

The ability to innovate and terminate objects in the video sequence allows the segmentation algorithm to adapt to complex dynamic changes in the scene. The mechanism described here innovates new objects using co-homogeneity between regions, therefore the newly innovated objects may not relate to objects seen by the human visual system. Such an innovation strategy could be used to form the basis of a focus of attention strategy to alert the operator that further interaction is required to group the automatically generated objects into semantic entities.

6.6 Performance Evaluation of the Hierarchical Bayesian Framework

In this section the performance of the hierarchical framework is characterised over representative test data. The evaluation metrics applied at each frame are the spatial quality of density (SQD) and the spatial quality of edge density ($SQED$) as introduced in section 3.4.3. Following the evaluation approach of Chapter 4, the temporal coherency of the object segmentation is characterised as the variance of the SQD and $SQED$ measures. This variance should be viewed with the accompanying SQD and $SQED$ measures, since a poor segmentation can exhibit frame to frame stability using this metric.

6.6.1 Datasets

A hindering factor in the performance evaluation of video object segmentation is the requirement of pixel-accurate ground truth segmentations of objects against to which object segmentation performance can be compared. The number of standard test sequences available with ground truth is surprisingly low. From the MPEG-4 test sequences only 'Children', 'Akiyo', 'News' and 'Bream' are complete with ground truth, as is the 'Parrot' sequence used by Erdem *et al* [58]. The creation of ground truth for performance evaluation of video object segmentation is a time consuming task, and as a consequence, evaluation is limited to the test sequences with available ground truth. This has the advantage that the results reported in this paper can be compared to other works in video object segmentation. The 'News' sequence is omitted from the evaluation due to its similarity to the 'Akiyo' sequence (i.e. newsreaders against a stationary backdrop). A selection of frames from the chosen test sequences are shown in Figure 6.5 along with the ground truth result at each frame.

6.6.2 Experiments

The following variants of the proposed hierarchical Bayesian framework were evaluated over all the test sequences:

- Region-Level representational models with Object-Level prediction using 'Frame $t-1$ Models at Frame t' strategy (COPY variant, section 6.4.2).

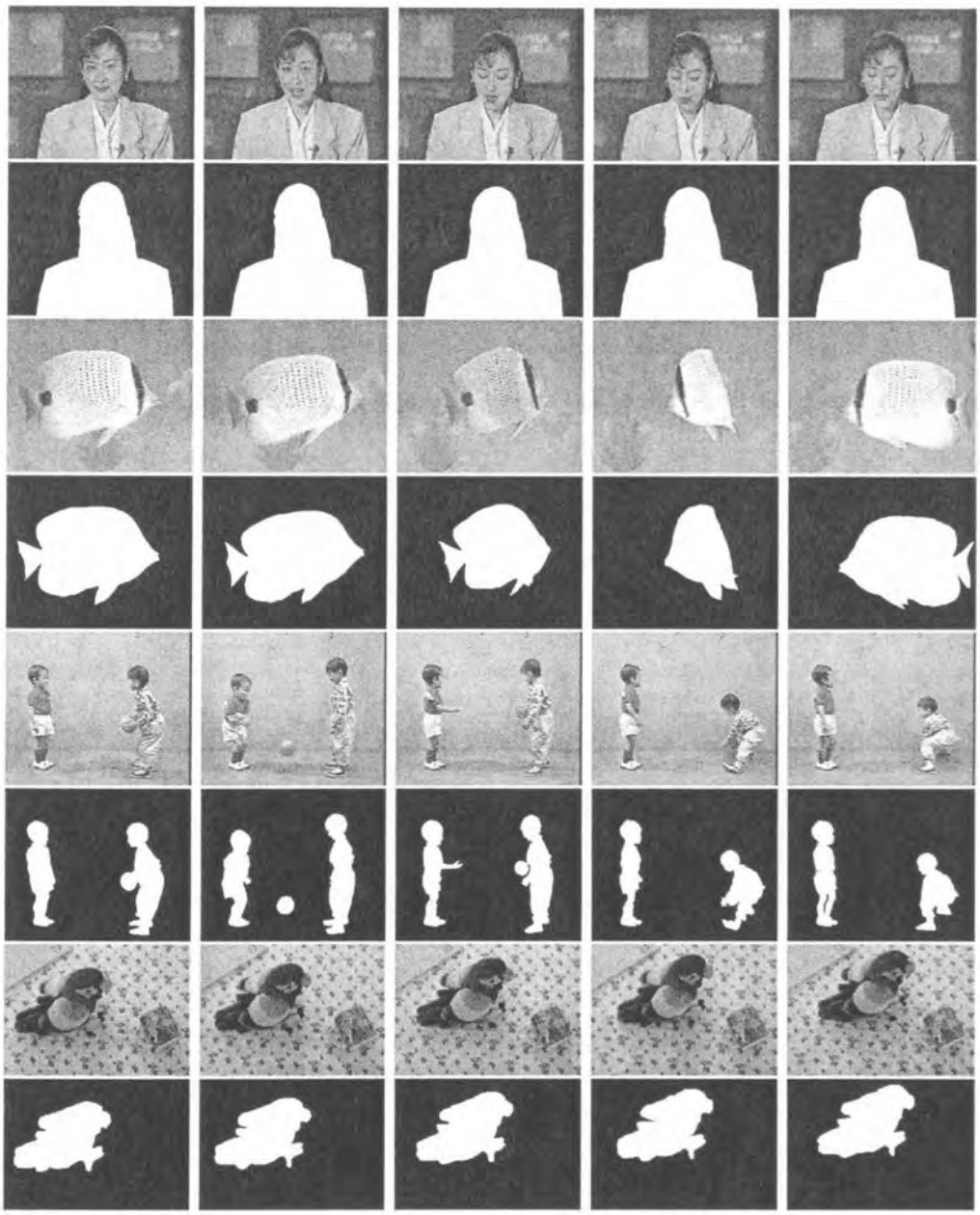


Figure 6.5: The test sequences with ground truth segmentation used to evaluate the performance of the hierarchical framework. Ground truth is shown as a binary mask. Top three sequences: frames 0, 30, 106, 120 and 134 from 'Akiyo', 'Bream' and 'Children'. Bottom sequence: frames 1,5,10,14 and 18 from 'Parrot'.

- Region-level representational models with object-level prediction using affine motion compensation strategy (AFFINE variant, section 6.4.2).
- Region-level representational models with object-level prediction using Kalman filtering strategy (KALMAN variant, section 6.4.2).
- Interacting region-level and object-level representational model (INTERACTING variant, section 6.4.3).

The termination and innovation of objects (section 6.5) was also evaluated using the region-level representational models with object-level prediction implementation. These experiments are designed to demonstrate the potential of the hierarchical Bayesian framework for the development and evaluation of a wide range of video object segmentation algorithms. To perform these experiments, the algorithm parameters are set to values that gave reasonable performance over a set of training data. Only the parameters introduced in this chapter are detailed. The innovation algorithm (described in section 6.5) uses the default value presented in Table 6.1.

Parameter	Description	Value
T	Squared distance threshold	15.09

Table 6.1: The algorithm parameters used in the object innovation algorithm for the performance evaluation.

6.7 Results

In this section the performance evaluation results are presented for various variants of the hierarchical Bayesian framework. These implementations demonstrate the potential of the framework to allow development and evaluation of a wide range of video object segmentation algorithms using both object- and region-level representations.

6.7.1 Region-Level Representational Models with Object-Level Prediction

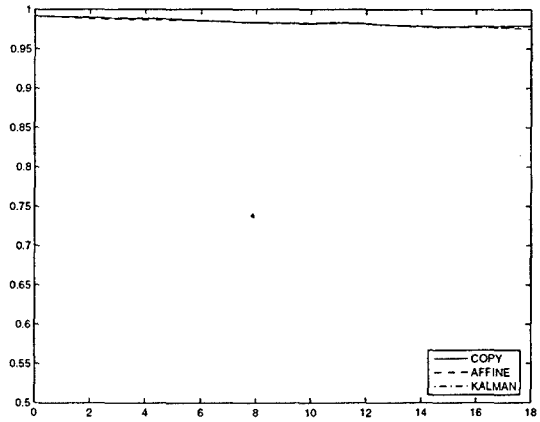
Three strategies were implemented to perform region-level representational models with object-level prediction (described in section 6.4.2):

- 'Frame $t-1$ Models at Frame t' (COPY variant).
- Affine motion-model based compensation (AFFINE variant).
- Kalman filtering (KALMAN variant).

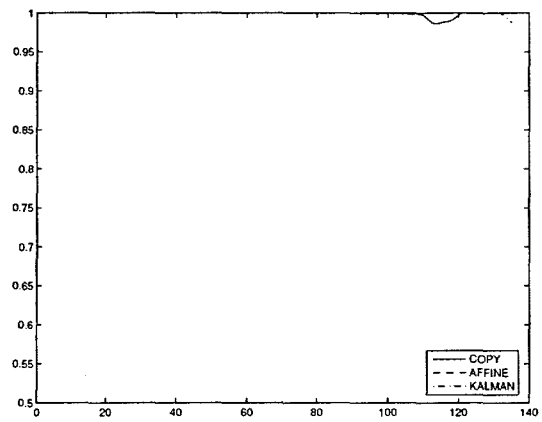
These three strategies were quantatively evaluated for the four test sequences (presented in section 6.6.1). At this stage termination or innovation of objects is not performed (see section 6.7.3 for the evaluation of termination and innovation strategies). The intra-frame update strategy for the region-level representation is the unconstrained reinitialisation update strategy (described in section 5.4).

The SQD and $SQED$ accuracy of the segmentation for the different strategies are shown in Figures 6.6 and 6.7. It can be seen that the prediction strategy generally has a minor effect on the SQD and $SQED$ segmentation accuracy for all but the 'Children' sequence. The 'Children' sequence exhibits a significantly lower SQD score when using the AFFINE strategy. On closer inspection this is due to the foreground object containing three objects — two children and a ball. The motion model prediction is not robust to multiple motions and hence the predicted affine motion does not represent the actual motion of the three individual objects, this results in segmentation error. The $SQED$ accuracy is less adversely affected by the inaccurate motion model since the majority of errors occur away from the object edges.

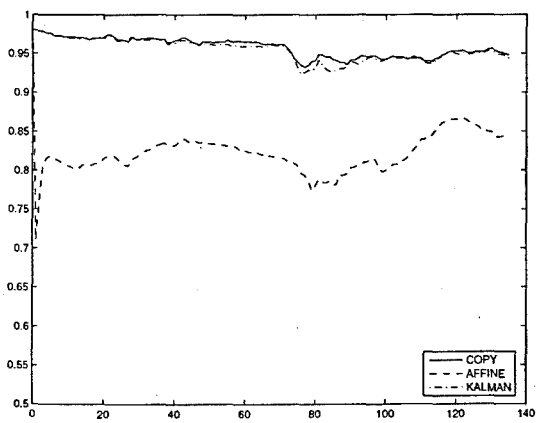
For the 'Bream' sequence it can be seen that the KALMAN variant also reduces the SQD and $SQED$ accuracy at key points in the sequence corresponding to the turning of the



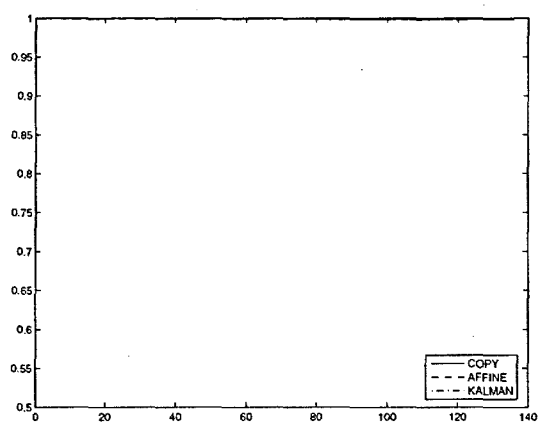
(a) Parrot



(b) Bream

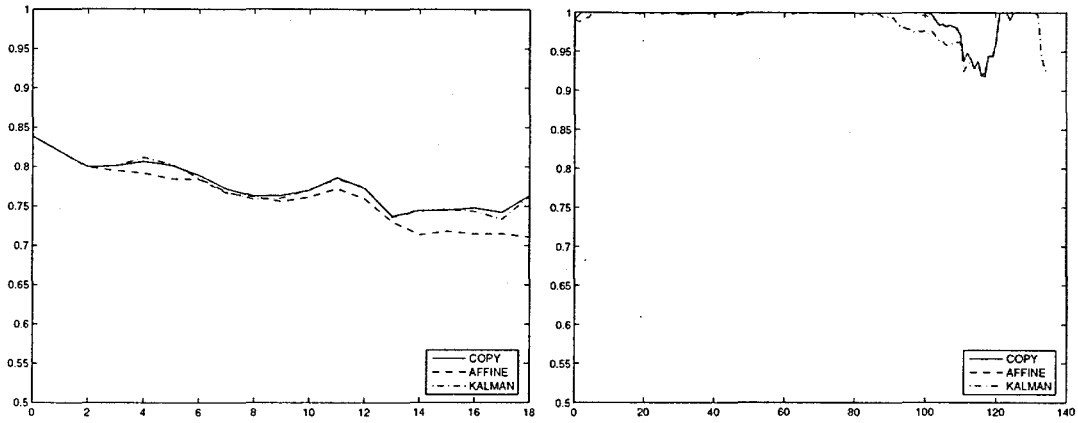


(c) Children



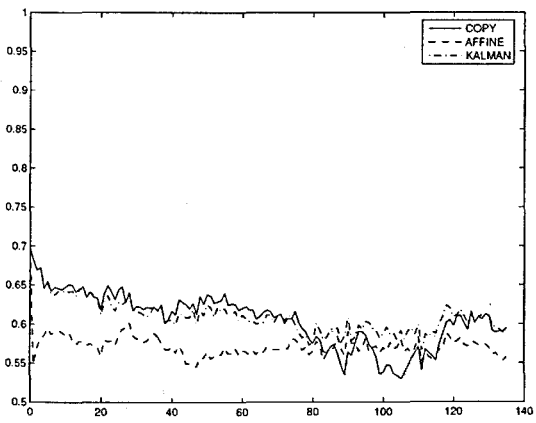
(d) Akiyo

Figure 6.6: *SQD* accuracy for the three region prediction strategies over the test data.

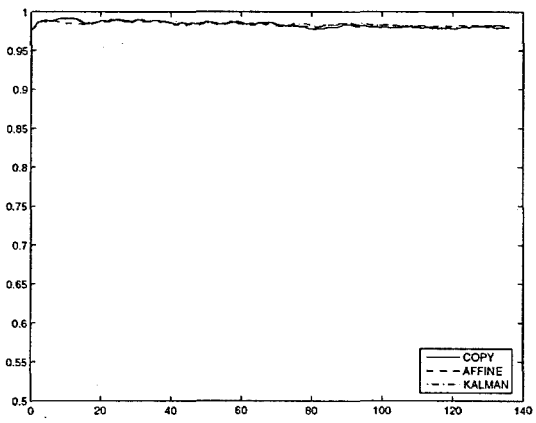


(a) Parrot

(b) Bream



(c) Children



(d) Akiyo

Figure 6.7: *SQED* accuracy for the three region prediction strategies over the test data.

fish. The event where the fish turns can be seen by a dip in segmentation accuracy (at frame ~ 110) due to an uncovered background object that is incorrectly classified as foreground. Due to the estimation process the KALMAN variant introduces a latency between a change of object direction and a state update, reducing the ability of the models to adapt to the changing video. The result of this latency is that the SQD and $SQED$ accuracy for the KALMAN approach is lower before the fish turn event and at the end of the sequence. The 'Akiyo' and 'Parrot' sequences show little difference between the three prediction strategies, although for the 'Parrot' sequence the AFFINE method causes a minor reduction in the segmentation accuracy. Again, this may be due to a lack of robust estimation in the model.

Table 6.2 shows the mean and variance SQD accuracy over the four test sequences. It can be seen that the prediction strategies generally have a minor effect on the output segmentation quality, especially for the sequences 'Parrot', 'Bream' and 'Akiyo'. The 'Children' sequence reflects the problems encountered when applying a single affine motion model to estimate the motion of three distinct objects. This results in a significantly lower segmentation accuracy when using the motion-model.

Sequence	Object-Level Prediction Strategy					
	COPY		AFFINE		KALMAN	
	μ	σ	μ	σ	μ	σ
'Akiyo'	0.9983	0.0001	0.9983	0.0001	0.9983	0.0001
'Bream'	0.9991	0.0028	0.9991	0.0028	0.9988	0.0030
'Children'	0.9577	0.0123	0.8222	0.0271	0.9549	0.0139
'Parrot'	0.9835	0.0042	0.9827	0.0046	0.9832	0.0046

Table 6.2: Average SQD error per pixel for the test data with motion-model based compensation and recursive filtering inter-frame prediction of regions at the object-level of the hierarchical framework.

Table 6.3 shows the mean and variance $SQED$ accuracy over the four test sequences. It can be seen that the quality of the segmentation at the edges of the objects is generally lower than the scene-wide measure, which is to be expected due to the difficulty of segmenting the edges between occluding objects. For the 'Bream' sequence the KALMAN variant has a lower average accuracy and higher deviation, due to the inability of the strategy to adapt to

the fast change in the object appearance. The segmentation accuracy for the 'Children' and 'Parrot' sequences are significantly lower than the accuracies for the 'Akiyo' and 'Bream' sequences. The 'Akiyo' and 'Bream' are created by merging a blue screen sequence with a new background whereas the 'Children' and 'Parrot' are the original recordings. Taking the foreground and background objects from separate sequences appears to reduce problems such as motion blur, translucency and sampling such that the edge accuracy is artificially higher. Even so, the intra-sequence patterns of the segmentation accuracy are expected to hold.

Sequence	Object-Level Prediction Strategy					
	COPY		AFFINE		KALMAN	
	μ	σ	μ	σ	μ	σ
'Akiyo'	0.9836	0.0041	0.9836	0.0024	0.9845	0.0030
'Bream'	0.9941	0.0167	0.9937	0.0168	0.9890	0.0213
'Children'	0.6060	0.0346	0.5720	0.0153	0.6103	0.0220
'Parrot'	0.7746	0.0276	0.7595	0.0366	0.7733	0.0291

Table 6.3: Average $SQED$ error per pixel for the test data with motion-model based compensation and recursive filtering inter-frame prediction of regions at the object-level of the hierarchical framework.

Finally, Figure 6.8 shows some segmentation results for the 'Bream' test sequence. It can be seen that all the segmentation results are on the whole similar to the ground-truth segmentation. It can be seen that there are a few false positive detections in frames 106 and 120 as the fish object moves to uncover a background object. To remove these false positives the mechanism for innovating and terminating objects needs to be added, this is evaluated in the following section. The KALMAN variant also introduces some false negative detections (in frames 106 and 134, marked in red) where the filter state cannot update to match the rapid motion of the foreground object.

From these results it can be seen that, in general, the use of KALMAN and AFFINE object-level prediction of regions has minor effect on the segmentation quality for the test sequences. At best, these more complex prediction strategies are no better than the COPY variant. It is likely that this is due to the complex spatio-temporal content of the test

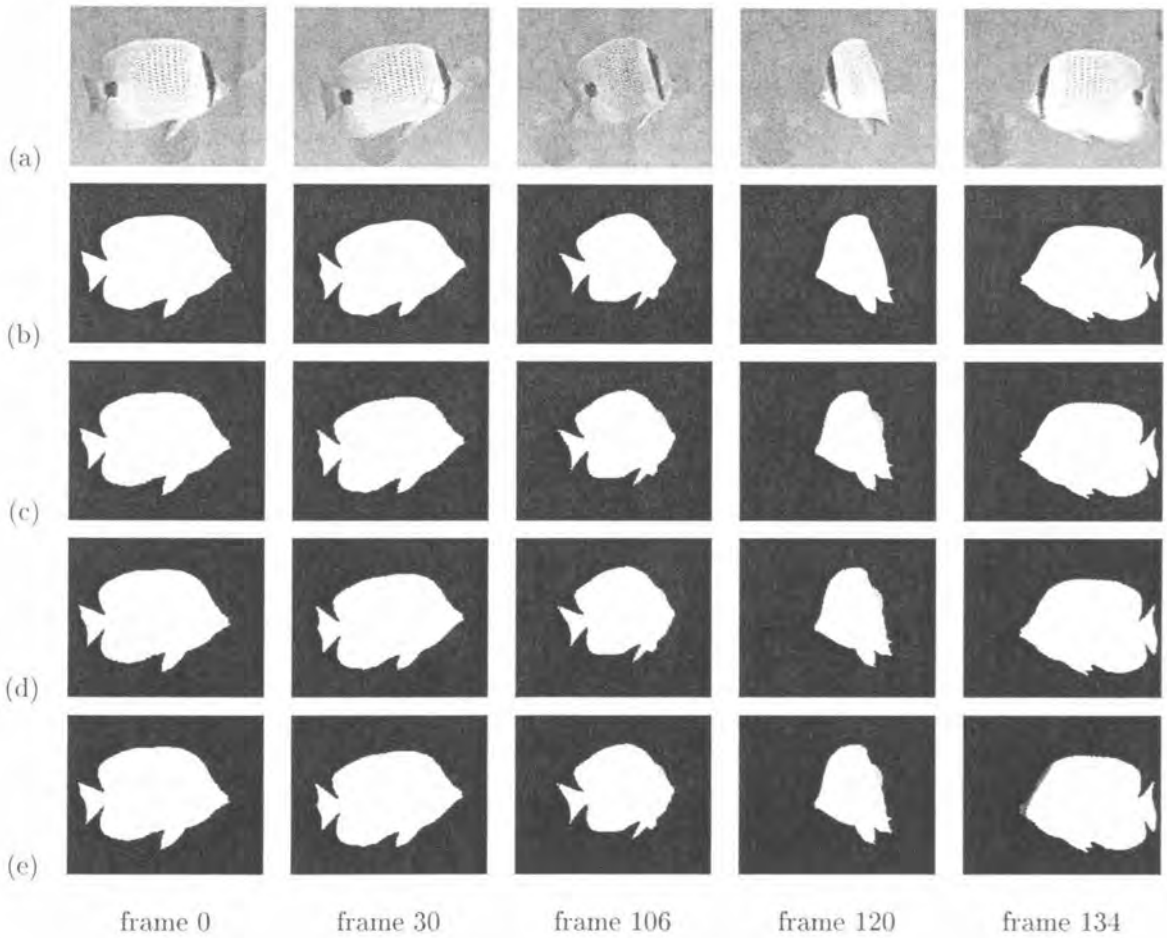


Figure 6.8: Segmented objects using object-level prediction of the region-level representational models. (a) Original video, showing frames 0, 30, 106, 120, and 134 of the test sequence 'Bream'. (b) Ground-truth object segmentation (c) COPY variant (d) AFFINE variant and (e) KALMAN variant. Red pixels represent under-segmented regions which are not detected. The cyan areas correspond to over-segmented regions (i.e. false positive detections).

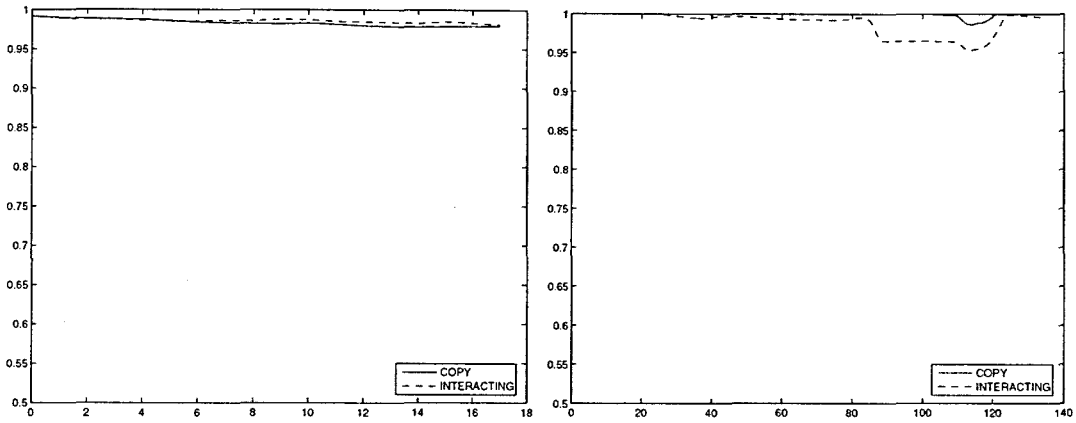
sequences, making the prediction of object regions difficult. In the next section the combined region- and object-level configuration is evaluated.

6.7.2 Interacting Region-Level and Object-Level Representational Models

In this section the performance of the INTERACTING strategy is demonstrated. This configuration of the framework (as described in section 6.4.3) contained an object-level model that was subsequently used to generate a local co-ordinate system for each object. This co-ordinate system implicitly modified the spatial location of the region-level representational models to match the newly discovered object in the scene. In this evaluation the SQD and $SQED$ segmentation accuracy is measured over the four test sequences. This is compared to the best performing variant found in the previous section, found to be the COPY prediction strategy. Again, no termination or innovation of objects is performed, the intra-frame update strategy for the region-level models is the unconstrained reinitialisation strategy described in section 5.4.

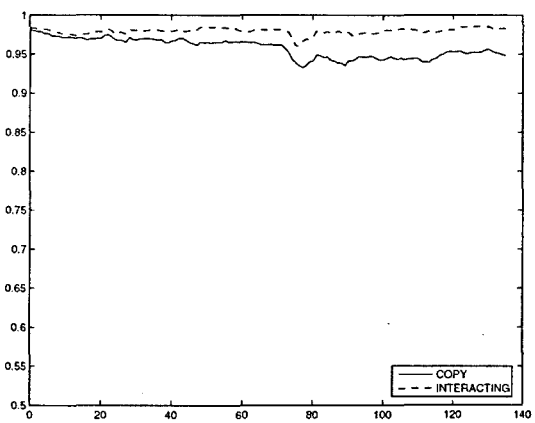
Figures 6.9 and 6.10 show the SQD and $SQED$ segmentation accuracy over the range of test data. It can be seen that for the 'Parrot' sequence the segmentation accuracy is generally higher when using the INTERACTING variant of the framework, when compared to the 'frame $t-1$ at t' object-level region prediction strategy. For this sequence, with a parrot undergoing translational motion, the use of a local co-ordinate system improves the adaption of the spatial-colour regions to the moving object with good localisation of the regions in relation to the parent object. For the 'Bream' sequence the COPY strategy outperforms the INTERACTING approach during the fish turn event. On closer inspection this is due to the changing shape of the fish object, causing instability in the predicted local co-ordinate system. This has the effect that the regions modelling the object are mis-located and a significant proportion are lost. To solve this problem the local co-ordinate system could be estimated in a more robust manner, perhaps by using the observed colour features of the object to locate the axes.

The 'Children' sequence shows an improvement in the segmentation quality compared to the COPY prediction strategy. This is misleading. The improvement appears to be due to a 'wobble' induced on the co-ordinate system by the moving ball object, which filters out some small false positive regions further from the centre of the co-ordinate system. The local

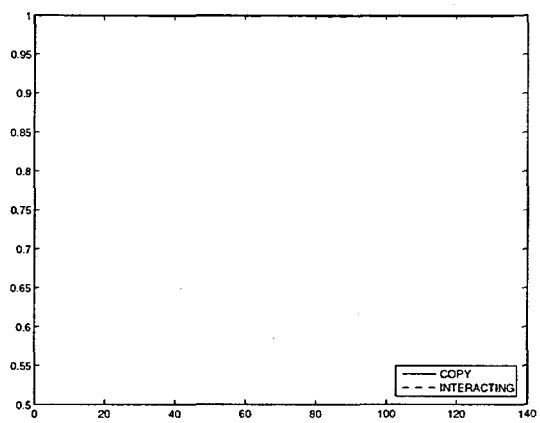


(a) Parrot

(b) Bream



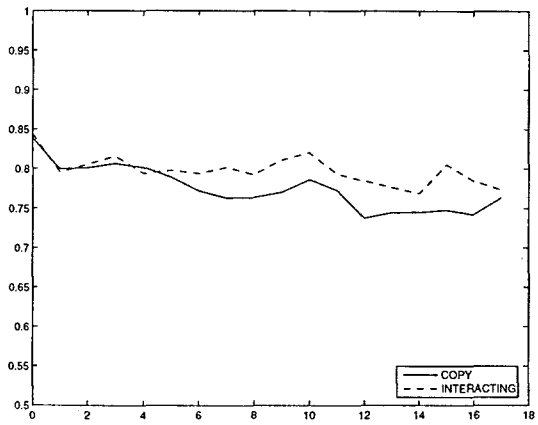
(c) Children



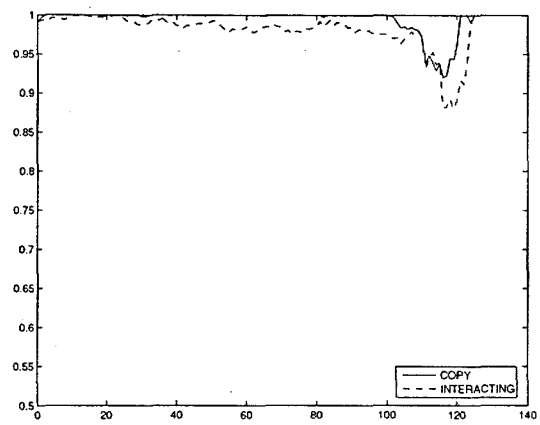
(d) Akiyo

Figure 6.9: *SQD* accuracy for the INTERACTING framework implementation compared to the COPY region prediction strategy over the test data.

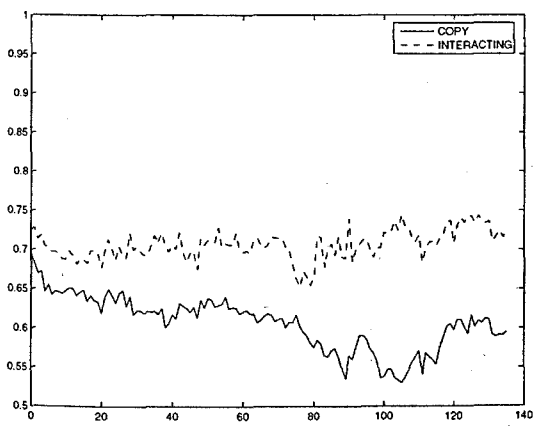
co-ordinate system is ill-suited to modelling multiple objects undergoing different motions. For the 'Akiyo' sequence the segmentation accuracy is similar for both two methods.



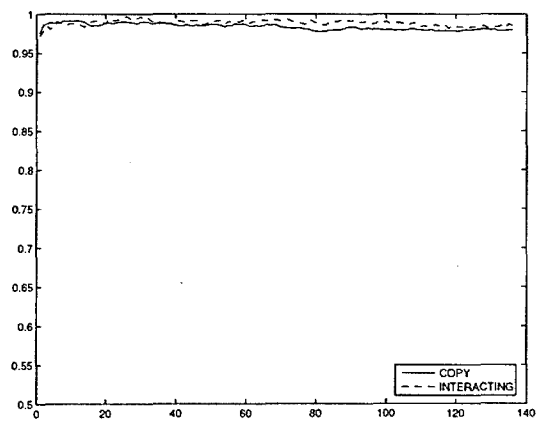
(a) Parrot



(b) Bream



(c) Children



(d) Akiyo

Figure 6.10: *SQED* accuracy for the INTERACTING framework implementation compared to the COPY region prediction strategy over the test data.

Table 6.4 shows the mean and deviation SQD accuracy over the four test sequences. These results confirm the trends observed in the graphical analysis. The INTERACTING implementation produces a higher SQD segmentation accuracy for the 'Children' sequence with similar accuracy noted for the 'Akiyo' and 'Parrot' sequences. The 'Bream' sequence suffers from reduced segmentation accuracy and increased deviation due to the instability of the local co-ordinate system.

Sequence	Strategy			
	COPY		INTERACTING	
	μ	σ	μ	σ
'Akiyo'	0.9983	0.0001	0.9992	0.0001
'Bream'	0.9991	0.0028	0.9877	0.0148
'Children'	0.9577	0.0123	0.9794	0.0042
'Parrot'	0.9835	0.0042	0.9861	0.0025

Table 6.4: Average SQD error per pixel for the INTERACTING and COPY prediction strategies over the test data.

Table 6.5 shows the mean and deviation $SQED$ accuracy over the four test sequences. These results demonstrate an improvement in segmentation accuracy at the edges of the objects for three of the four test sequences. Again, the 'Bream' sequence suffers due to the instability of the local co-ordinate system. As shown in the previous section the segmentation accuracy at the edge is higher for the 'Akiyo' and 'Bream' sequences, this is due to the way these sequences were generated. The 'Parrot' sequence is perhaps the most suited to this type of local co-ordinate system transform (an elongated object with translational motion) and this is reflected by the improved segmentation accuracy compared to the COPY strategy. The 'Children' sequence shows the greatest improvement when using the INTERACTING variant, the reason for this is most likely due to the filtering effect caused by the 'wobbling' co-ordinate system that removes smaller false positive regions.

Finally, Figure 6.11 shows segmentation results for the test sequences. Subjectively the best results are for the 'Akiyo' and 'Bream' sequences. The 'Parrot' sequence suffers false positive and negative detections around the parrot objects beak and claws due to the colour similarity between the foreground and background regions. Such a problem may be resolved

Sequence	Strategy			
	COPY		INTERACTING	
	μ	σ	μ	σ
'Akiyo'	0.9836	0.0041	0.9883	0.0037
'Bream'	0.9941	0.0167	0.9811	0.0242
'Children'	0.6060	0.0346	0.7066	0.0183
'Parrot'	0.7746	0.0276	0.7976	0.0182

Table 6.5: Average *SQED* error per pixel for the INTERACTING and COPY prediction strategies over the test data.

by using a contour based model and subsequently using local contour prediction to refine the estimate of the objects boundary. The 'Children' sequence is a reasonable result for the two person objects, although the ball is lost due to the changes in motion/appearance, partial occlusions and the instability of the moments of the foreground object (causing the local co-ordinate system to 'wobble').

In this section the INTERACTING framework variant has been demonstrated to improve the segmentation accuracy for some test sequences when compared to the COPY variant. These distinct implementations of the same framework demonstrates the flexibility of the framework to compare and evaluate different video object segmentation algorithms. The INTERACTING variant has been shown to decrease in segmentation accuracy when the prediction of the local co-ordinate system becomes unstable due to appearance changes or occlusions of the objects. To improve this the local co-ordinate system can perhaps be estimated robustly using the local colour features of the objects.

In the following section an object innovation and termination mechanism is added to allow newly discovered regions to be added to existing or new — innovated — objects.

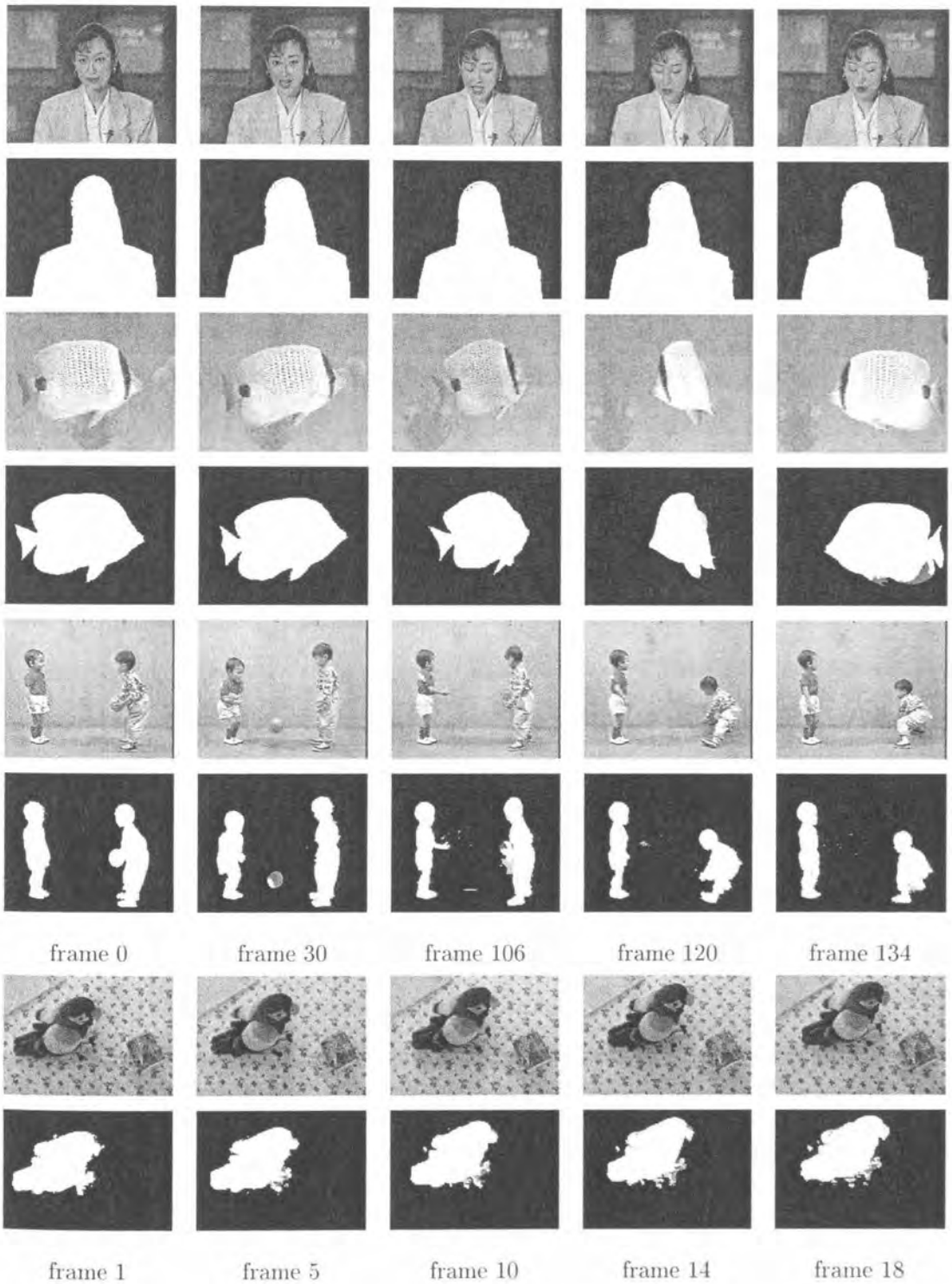


Figure 6.11: Segmented objects using the INTERACTING variant of the framework. (Top) Objects extracted for frames 0, 30, 106,120, and 134 from the test sequences 'Akiyo', 'Bream' and 'Children'. (Bottom) Objects extracted for frames 1, 5, 10, 14, and 18 from the test sequences 'Parrot'. Red pixels represent under-segmented regions which are not detected. The cyan areas correspond to over-segmented regions (i.e. false positive detections).

6.7.3 Termination and Innovation of Objects

In this section the performance of the object termination and object innovation mechanisms is demonstrated. The object termination and object innovation mechanisms are described in section 6.5. The innovation mechanism uses a validation gate to match newly innovated regions with the current set of video objects. If no match is found a new object is innovated. If an existing object has no support in the current frame then it is terminated. The innovation mechanism cannot be expected to match the performance of a human operator, and hence the newly innovated objects are those that cannot be matched to existing objects using spatial-colour information. The termination and innovation strategy is compared to a strategy with no termination or innovation, where unconstrained adaption of the existing models is used. In both cases the prediction scheme used is the COPY variant presented in section 6.4.2.

The termination and innovation mechanism is evaluated in a qualitative manner for five test sequences — 'Dinosaur', 'Ping Pong', 'Coastguard', 'Foreman' and 'Bream'. The 'Ping Pong' sequence is spatially and temporal subsampled to decrease the number of pixels to be processed. Ground truth is not used in this evalution due to the ambiguity of labelling new objects in the video sequences, and to allow a greater range of test sequences to be used to demonstrate the performance of the proposed approach.

The first sequence to be evaluated is the 'Dinosaur' sequence, shown in Figure 6.12. This sequence contains a model dinosaur spinning on a turntable against a blue background. The challenge in this sequence is to update the foreground and background representational models by adding and removing newly innovated regions to and from these objects. No new objects should be introduced by the innovation mechanism.

The results (also shown in Figure 6.12) show that the newly innovated regions (created as the dinosaur rotates) are mostly added to the existing objects. In this sequence the unseen parts of the objects are similar in colour appearance to the objects appearance in the first frame, therefore the colour and spatial based matching correctly adds most of the regions to the objects. A new object has been innovated in the shaded areas at the base of the dinosaur due to a significant difference in appearance compare to the rest of the dinosaur object. It is possible that further higher-level processing (using contextual knowledge) could be used to merge these two objects.

It can be seen that when not using the termination / innovation strategy the object is well segmented, with few false positive/false negative detections apparent. This segmentation is achievable due to the separation between foreground and background in the feature space, even though the foreground model has degraded, it is still representative of the foreground object relative to the background model. The COPY variant when not using the termination / innovation strategy is not as representative of the object in the final frame compared to the termination/innovation strategy.

It is noticeable that some pixels remain unassigned in the termination/innovation (shown as black pixels), particularly in the first frame. The reason for this is that the constrained labelling rule (presented in section 5.4) and removal of small regions leaves a small proportion of the image undefined by any existing region models. This problem is amplified for the first frame due to the relatively large number of small regions that are removed. This is a potential weakness of the region initialisation strategy (described in section 5.3). The problem is not rectified since for the first frame the key-frame segmentation is available. For subsequent frames the pixels tend to be isolated and as such could be labelled using a neighbourhood mode filter or Markov random field labelling process.

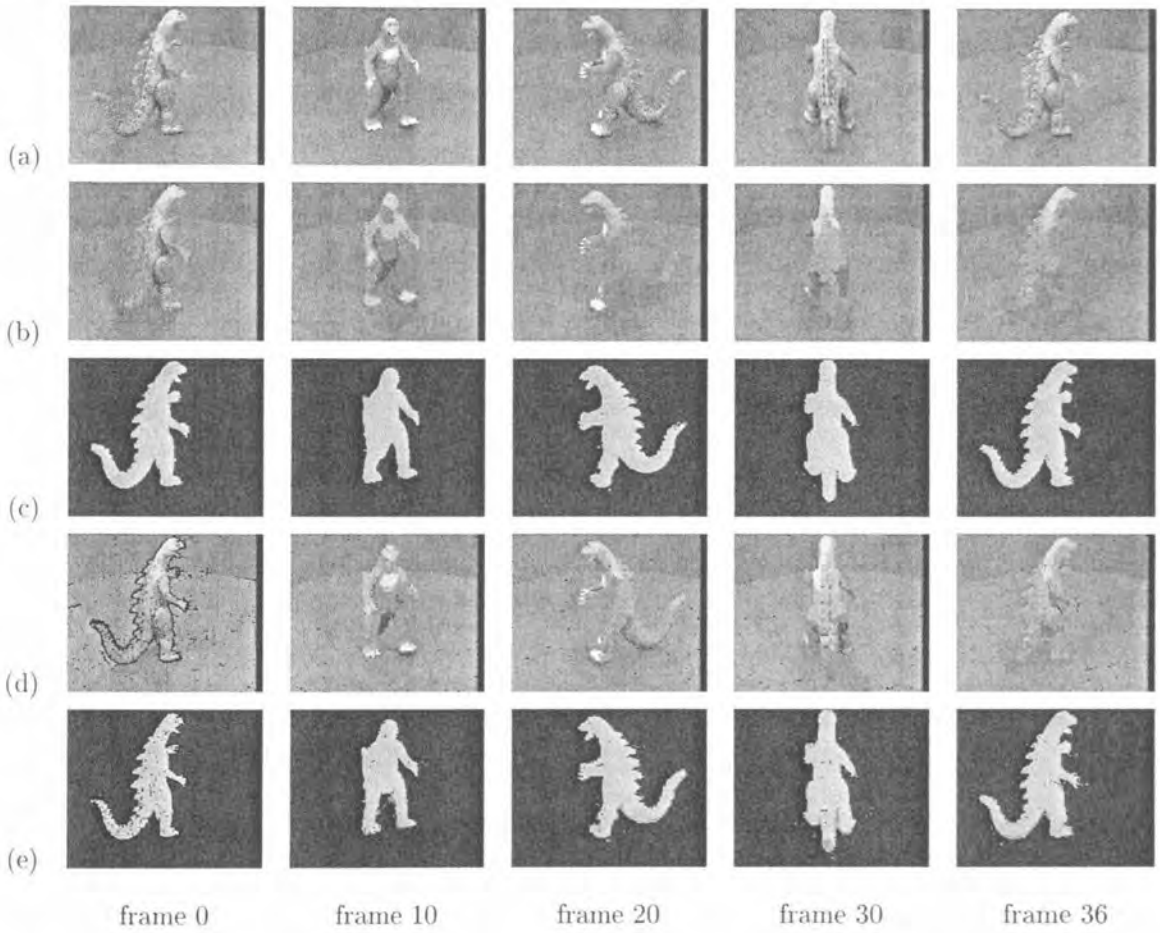


Figure 6.12: Segmentation results for region- and object-level innovation / termination strategies. (a) 'Dinosaur' sequence frames 0, 10, 20, 30, and 36. (b) Region-level representation without region- or object-level innovation / termination strategies (c) Segmented objects (false colour, representing object ID) without region- or object-level innovation / termination strategies (d) Region-level representation with region-level innovation / termination strategies (e) Segmented objects (false colour, representing object ID) with region- and object-level innovation / termination strategies.

The second sequence evaluated is the 'Ping Pong' sequence, shown in Figure 6.13. In this sequence the camera starts zoomed on the players hand, the camera then zooms out revealing more objects in the scene. The challenge in this sequence is to correctly identify the newly revealed objects including a poster. At the same time the player and table should be correctly innovated such that new regions are correctly added to the existing objects.

The results (also shown in Figure 6.13) show the limitation of using colour and spatial homogeneity as a criteria for object-level innovation. The newly innovation objects are sensitive to lighting variations in the scene (e.g. shadows and highlights) and as such the table object is split into three components. Encouragingly, the players clothing is correctly updated during the sequence, and the poster object is detected as a distinct object (although the lettering is detected as a separate object. The players face is merged with the background due to the colour similarity, although the hair and beard are detected as a further object. This sequence demonstrates the requirement for region- and object-level innovation in video object segmentation algorithms. Without termination and innovation, the representation of the scene degrades during this sequence due to the inability to adapt to the newly introduced elements. The final frame segmentation still gives a reasonable localisation of the players red jersey, although most other elements (the players head, the poster etc) are consumed by the background object representation. The representation of the final frame is higher fidelity (i.e. more representative) for the termination / innovation strategy when compared to the result without.

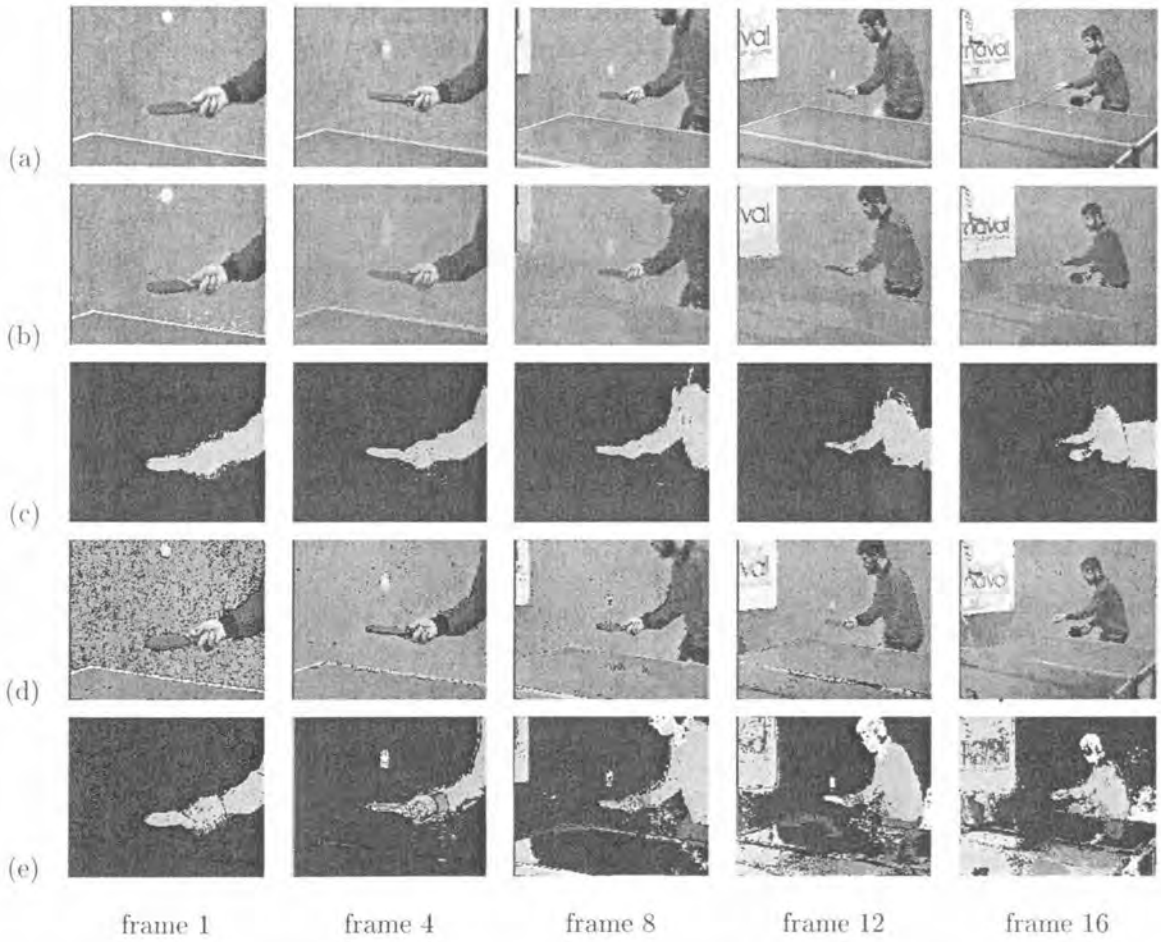


Figure 6.13: Segmentation results for region- and object-level innovation / termination strategies. (a) 'Ping Pong' sequence frames 1, 4, 8, 12, and 16. (b) Region-level representation without region- or object-level innovation / termination strategies (c) Segmented objects (false colour, representing object ID) without region- or object-level innovation / termination strategies (d) Region-level representation with region-level innovation / termination strategies (e) Segmented objects (false colour, representing object ID) with region- and object-level innovation / termination strategies.

The third sequence evaluated is the 'Coastguard' sequence, shown in Figure 6.14. In this sequence a boat is tracked along a river and a second boat appears in the scene and passes behind the first boat. The main challenge in this sequence is to correctly detect the second boat and detect it as it passes the first boat. The remainder of the scene undergoes relatively minor change in appearance.

The results (also shown in Figure 6.14) show that the second boat is not detected as a separate object. This is perhaps due to the similarity between the second boats appearance and some of the background object elements (e.g. the wake around the first boat). Looking at the first frame in the sequence, it can be seen that the second boat is partially visible and incorrectly identified as a background object, this error may be propagated throughout the sequence. This highlights a difficulty when using key-frame initialisation in that the human perception of the scene may be incorrect, in which case further processing may be required to correct the error (e.g. by detecting the second boat as a new object). Also, the representational model of the boat does not appear distinct, this is perhaps due to an under-constrained matching step at the region-level. The result when not using the termination / innovation strategy is subjectively similar to that with, this demonstrates that even when the innovation strategy fails the result is at worst similar to the result when not using the strategy.

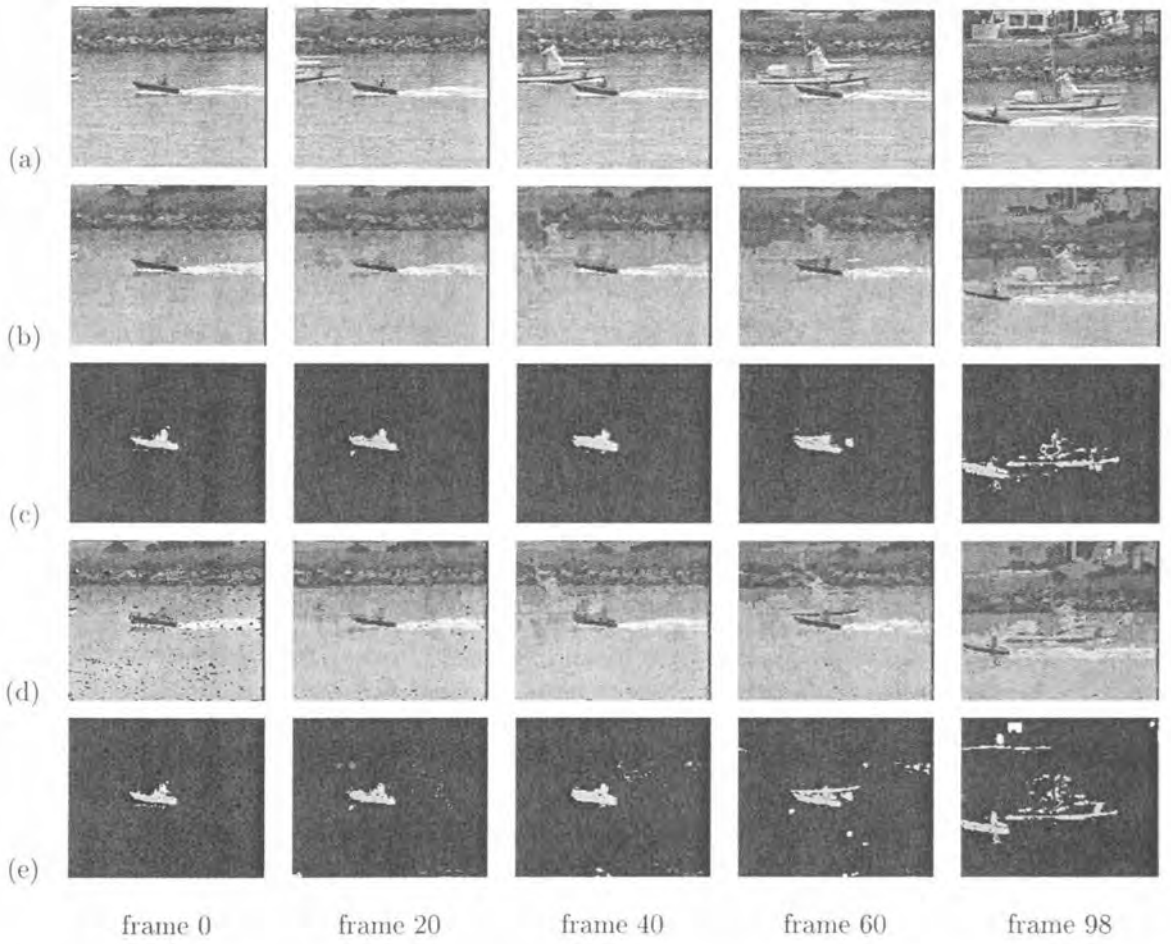


Figure 6.14: Segmentation results for region- and object-level innovation / termination strategies. (a) 'Coastguard' sequence frames 0, 20, 40, 60, and 98. (b) Region-level representation without region- or object-level innovation / termination strategies (c) Segmented objects (false colour, representing object ID) without region- or object-level innovation / termination strategies (d) Region-level representation with region-level innovation / termination strategies (e) Segmented objects (false colour, representing object ID) with region- and object-level innovation / termination strategies.

The fourth sequence evaluated is the 'Foreman' sequence, shown in Figure 6.15. This sequence contains an individual talking to a shaking camera which subsequently pans to the right to show a construction site. The challenge in this sequence is to not innovate any new objects until the point where the camera pans to the right

The results (also shown in Figure 6.15) generally show the result expected. The majority of the foreground person object is segmented except for the shoulders which are incorrectly added to the background object, due to appearance similarity with elements of the background. When the camera pans to the right the crane and sky regions (visible during the entire sequence) allow the background model to be propagated. When the construction site is visible it is correctly identified as a set of new, previously unseen, objects. Again, further processing could be used to determine if the component regions of the construction site can be merged based on feature space similarity, perhaps using more features in the space. This sequence again demonstrates the importance of using an innovation strategy for scenes where the content changes greatly. The result without the termination / innovation strategy provides a reasonable segmentation of the foreground person object, when the camera pans to the right the background representation dominates and consumes the foreground representation. For an application where the person object is to be segmented this result is adequate. For an application where the construction site is also to be segmented the termination / innovation strategy is required to correctly identify the new scene elements. Again, the representation of the final frame is closer to the scene content for the termination / innovation strategy when compared to the result without.

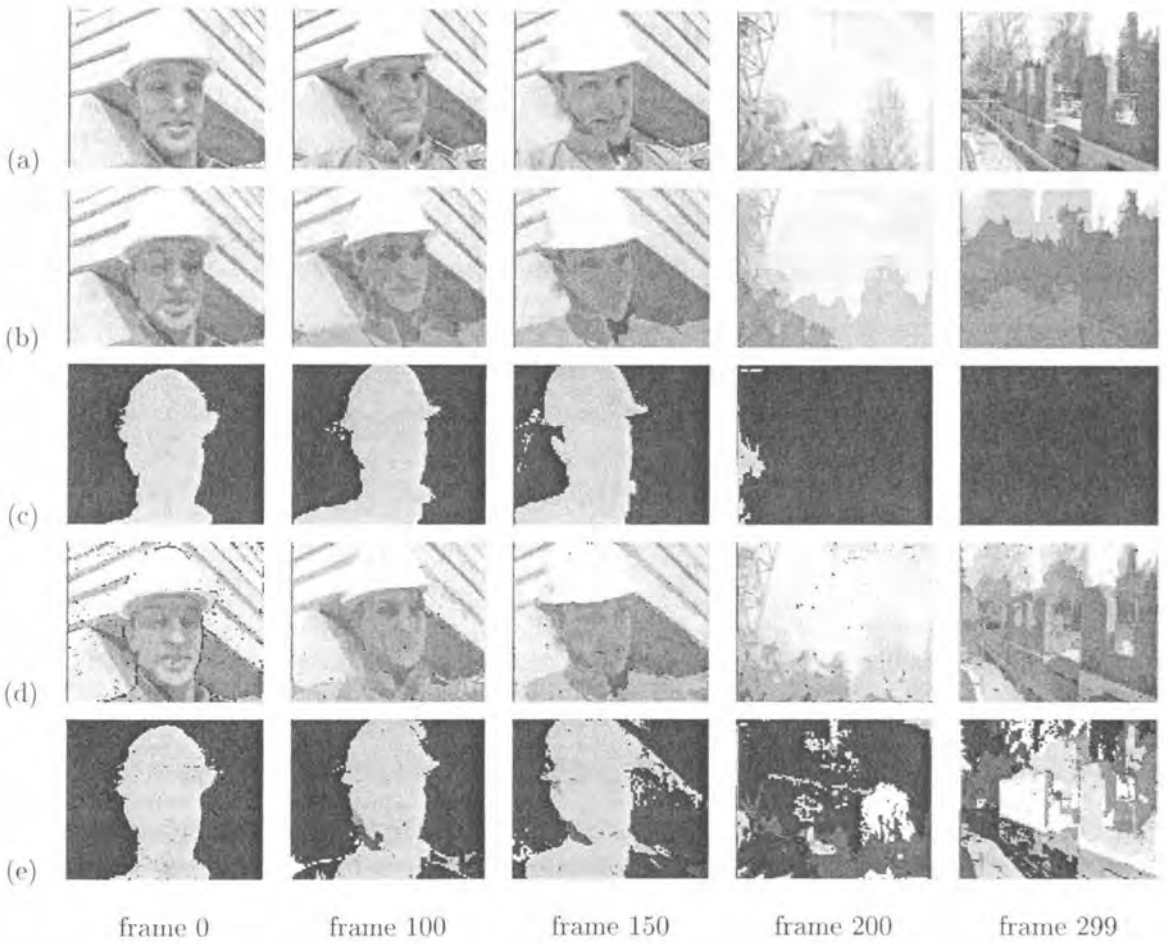


Figure 6.15: Segmentation results for region- and object-level innovation / termination strategies. (a) 'Foreman' sequence frames 0, 100, 150, 200, and 299. (b) Region-level representation without region- or object-level innovation / termination strategies (c) Segmented objects (false colour, representing object ID) without region- or object-level innovation / termination strategies (d) Region-level representation with region-level innovation / termination strategies (e) Segmented objects (false colour, representing object ID) with region- and object-level innovation / termination strategies.

The final sequence to be evaluated is the 'Bream' sequence, shown in Figure 6.16. This sequence contains an fish turning against a background containing moving planets. The challenge in this sequence is to correctly add the uncovered planets to the background object as the fish moves, and also retain the fidelity of the representation of the fish as it turns.

The results (also shown in Figure 6.16) show advantages and disadvantages introduced when using the termination and innovation strategy. The strategy correctly associates the newly innovated planet in frame 120 with the background representation, demonstrating explicit handling of problems associated with occlusion. Without the strategy this uncovered planet is incorrectly added to the foreground model. Another planet and portion of sky are also identified as a new object in frame 106, without using the termination / innovation strategy they are added to the background object. In general, it is incredibly complex to have semantic knowledge imparted on a segmentation algorithm, for this sequence it is ambigious whether or not the planets in the background constitute independent scene elements. In all cases the segmentation is application dependent, and as such the distinction between the different objects requires user intervention. The innovated objects in most cases can be merged into higher-level objects with semantic meaning. The representation of the final frame is closer to the scene content for the termination / innovation strategy when compared to the result without. This is especially noticeable around the tail of the fish, although a small region on the head of the fish has been incorrectly associated with the background representation even though it has adapted correctly to the appearance of the fish. As stated earlier, it is noticeable that some pixels remain unassigned in the termination/innovation (shown as black pixels), particularly in the first frame. This is again due to the constraint mechanisms in the termination / innovation strategy and can be fixed using local filtering of the segmentation result.

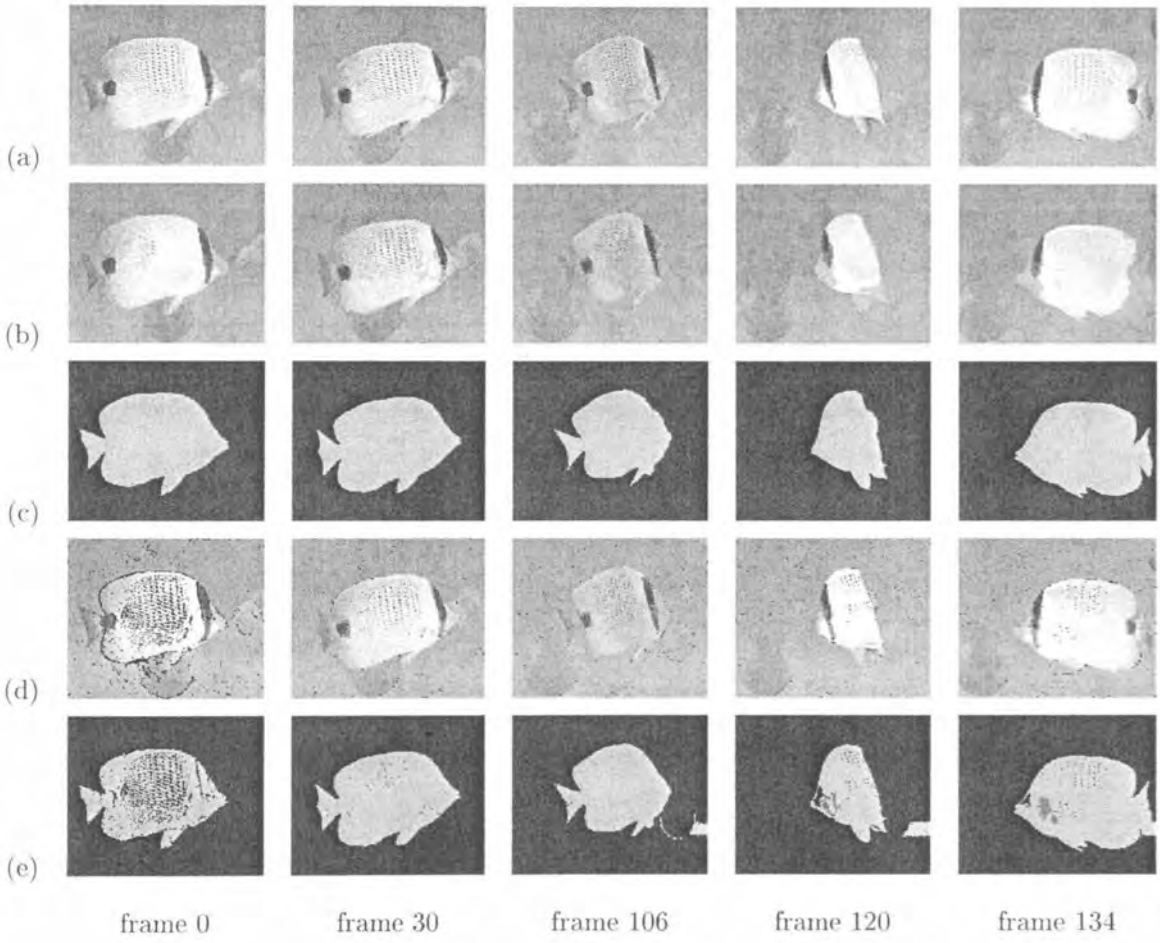


Figure 6.16: Segmentation results for region- and object-level innovation / termination strategies. (a) 'Bream' sequence frames 0, 30, 106, 120, and 134. (b) Region-level representation without region- or object-level innovation / termination strategies (c) Segmented objects (false colour, representing object ID) without region- or object-level innovation / termination strategies (d) Region-level representation with region-level innovation / termination strategies (e) Segmented objects (false colour, representing object ID) with region- and object-level innovation / termination strategies.

In this section a strategy for innovating and terminating objects has been demonstrated on a range of representative test data. The innovated objects are formed by detecting regions with homogeneous distribution in the spatial-colour feature space. As such, they do not represent semantic objects, but can often be grouped at a higher-level to give semantic meaning. This innovation and termination mechanism can be used as a focus of attention strategy in an application of the segmentation process. It also improves the ability of the system to handle partial (and even full) occlusions, uncovered background can be correctly associated with the existing background in many cases. In the case of total occlusion the hidden object may be redetected as a new object, in which case higher-level spatio-temporal processing is required to restore the original label. In the following section this chapter is concluded.

6.8 Conclusions

In this chapter a hierarchical Bayesian framework for video object segmentation has been proposed. This framework has been applied to the problem of updating the video object representational models on a per-frame basis, both at the region- and object-level of the hierarchy. Three region prediction strategies were implemented within the framework (COPY, AFFINE and KALMAN variants) and compared in a quantitative manner on well known test sequences. The hierarchical nature of the framework was further explored with the INTERACTING variant. An object termination and innovation strategy was subsequently introduced and evaluated over a range of test data.

The proposed hierarchical framework can be used to efficiently implement and compare object prediction strategies, interacting representational models at different hierarchical layers and termination and innovation strategies for video object segmentation. The Bayesian description of the framework allows further modules to be incorporated in a principled manner. Three distinct region prediction strategies were implemented within this framework to take advantage of the higher-level information in the scene. In these prediction strategies the object-level representation was limited to motion models or filtered spatial information (e.g. bounding box).

To explore the framework further the INTERACTING variant was implemented. In this framework a local per-object co-ordinate system was used to share information between the

object- and region-levels in the framework. The innovation and termination of regions at the region-level was extended to assign these regions to the video objects in the scene. If the region cannot be matched to any existing objects then new objects are innovated, this mechanism uses feature space homogeneity as the criteria for matching. The proposed implementations of the framework were implemented both qualitatively and quantitatively over a range of standard test data.

In this chapter the following contributions have been made:

- Introduced a hierarchical framework for video object segmentation.
- Introduced a Bayesian implementation of the hierarchical framework.
- Evaluated three methodologies for region prediction (COPY, AFFINE and KALMAN variants) over a range of test data.
- Evaluated a novel INTERACTING variant of the framework over a range of test data.
- Proposed an object termination and innovation strategy for video object segmentation.
- Evaluated the object termination and innovation strategy over a range of test data.

It was found that predicting the region-level models using the motion information at the object-level offered negligible benefit over using the simpler COPY strategy. The INTERACTING variant was found to be sensitive to changes in the local co-ordinate system of the parent object, which was used to locate the region-level models. The spatial-colour based innovation strategy was demonstrated to have the potential to be used as a focus of attention strategy for higher level processing, however the problem of finding semantic objects remains a fundamental challenge in video object segmentation.

This chapter has demonstrated the applicability of the hierarchical framework to video object segmentation. A key benefit to this approach is that many of the issues associated with video object segmentation can be approached using explicit modules in the Bayesian framework, simplifying the implementation and evaluation.

Future work includes implementing shape priors (built up over a temporal window) and graph-like structures to improve tracking of objects through occlusion (by allowing nodes to be hidden and predicted based on some local structural analysis). Occlusion makes

it difficult to predict and update region-based representative models of the video objects, explicit modelling of this is an important consideration in future video object segmentation work. The innovation of semantically important objects may never be solved completely, since the definition of semantics requires knowledge about the application which implies human intervention is required. In the following chapter the outcomes of this thesis are discussed along with the future directions for work on video object segmentation.

Chapter 7

Final Discussion

The aim of the work presented in this thesis was to segment semantic video objects from video sequences. Video object segmentation comprises three main components — feature space extraction, video object representation and representational scheme update. The work in this thesis presents algorithms and results that demonstrate the applicability of existing and novel algorithms to these three fundamental components. The research conducted during the course of this investigation is summarised in Section 7.1. The main contributions of this thesis are given in Section 7.2 and finally future research directions are discussed in Section 7.3.

7.1 Summary of Research

The research performed in this thesis is summarised in this section. The research is summarised in the order in which it was presented in the thesis. Feature spaces that can be applied to the problem of video object segmentation (Chapter 3) are discussed first. Following this the work completed on probabilistic representational models (Chapter 4) is reviewed. Chapter 5 presented methods for maintaining a region-based representation of a video sequence and finally Chapter 6 proposed a hierarchical Bayesian framework within which object-level prediction and innovation strategies were explored.

Feature Spaces for Video Object Segmentation

Using an evaluation procedure methodology it was demonstrated that spatial information appended to a colour feature vector is a powerful descriptor allowing a representational

scheme to segment an object with sufficient accuracy. It was demonstrated that motion information can also be beneficial to the feature space for specific sequences. Weighting the motion information was shown to improve the scene and edge segmentation accuracy compared to an unweighted strategy, creating more coherent object segmentations if the object is moving. However, the addition of weighted motion information to the spatial-colour feature space was found to decrease the scene and edge segmentation quality. Texture information was shown to make negligible difference for generic object segmentation and it is best applied to specific applications. Finally, the effect of pre- and post-processing the data was presented. From the results this appears to be a superior method for generating coherent object segmentation since it does not degrade edge quality as much as the use of motion information in the feature space.

Probabilistic Representation for Video Object Segmentation

Probabilistic modelling was applied to the problem of video object segmentation. Three distinct approaches to PDF estimation were implemented and applied to the problem of modelling spatial, colour and joint spatial-colour distributions. The performance of these three methods were evaluated within a common framework. It was determined that the kernel density model achieves the best accuracy for segmentation around the edges of video objects. The independent modelling of the Spatial-Colour PDF of video objects was subsequently evaluated and was found to reduce the accuracy of the extracted video object when compared to the joint PDF models.

Propagation Strategies for Video Region Segmentation

In Chapter 5 methods were introduced for segmenting video frames into homogeneous colour regions. An efficient spatial-colour region-based representative scheme was evaluated using a range of inter- and intra-frame model update strategies. The intra-frame update strategy was constrained to improve the homogeneity of the propagated regions, this required mechanisms for innovating and terminating the video regions. It was found that this constraint mechanism improved the representativeness of the models. The more complex inter- and intra-frame update strategies were found to offer little advantage over the simpler strategies due to the sensitivity of the regions to the appearance changes in the video.

Hierarchical Bayesian Framework for Video Object Segmentation

In Chapter 6 a hierarchical Bayesian framework for video object segmentation was proposed. The framework was used to propagate region-level models by using higher-level object information. Three implementations of the framework used object-level prediction of the region-level models. The hierarchical nature of the framework was further exploited in an interacting region- and object-level implementation where the region models where located in a co-ordinate system centered on the parent video objects. Additionally, methods were presented for innovating and terminating video objects to allow new scene elements to be modelled as the video content changes. It was found that the complex inter-frame prediction of region-level models offered little advantage over the simpler strategy. The interacting region- and object-level implementation was found to be sensitive to changes in the local co-ordinate system of the parent object. The video object innovation strategy was demonstrated to find objects in the scene using a spatial and colour homogeneity criteria, forming the basis of a focus of attention strategy to alert human operators to new scene elements.

7.2 Contributions

This section describes the work contributed in each chapter.

Review of Methods for Video Object Segmentation

Chapter 2 thoroughly reviewed the wealth of work that has been applied to the problem of video object segmentation. Three main types of approach to video object segmentation were identified — morphology based, image-plane based and feature space classifiers. Morphology based approaches are a popular subset of the image-plane based methods and were separated for clarity. The techniques for updating the representational models were also reviewed.

Evaluation of Feature Spaces

In Chapter 3 an evaluation methodology for video object extraction was proposed. This methodology incorporated two quantitative measures that give both scene- and edge-based measures of video object segmentation accuracy. The methodology was applied over a

sufficient range of feature spaces on test sequences to determine the 'best' performing feature space.

The 'Best' Feature Space

In Chapter 3 the 'best' performing feature space for video object segmentation was selected. It was demonstrated that spatial and colour information together forms a powerful descriptor for generic video object segmentation. Motion and Texture information was found to generally decrease the accuracy of the video object segmentation.

Video Object Representation

In Chapter 4 the use of existing probabilistic models for video object segmentation was researched. Three different representational schemes for video object segmentation were evaluated using the scene- and edge-based accuracy. Joint and independent modelling of the Spatial-Colour PDF of video objects was also evaluated using the same procedure. It was found that kernel density models gave the highest segmentation accuracy at the edges of the video objects.

Region Based Video Representation

In Chapter 5 aspects of region based modelling in video sequences were researched. An efficient independent spatial-colour region model was proposed. Following this existing methodologies were applied to the problem of inter-frame prediction, intra-frame update, termination and innovation of per-region representational models. These methods were evaluated using performance measures that did not require ground truth segmentation. It was found that constraining the intra-frame update of the models gave a significant improvement in the representativeness of the models.

Hierarchical Framework for Video Object Segmentation

In Chapter 6 a hierarchical Bayesian framework for video object segmentation was described. This framework was implemented using both object-level prediction of region models and interacting object- and region-level representative models. Methods for innovating and terminating video objects were described. It was found that the different methodologies

for updating the representative models all had comparable performance. The innovation strategy was shown to generate objects homogeneous in the colour space, which could be used for further higher-level processing to recover semantic objects.

7.3 Future Research

The problem of video object segmentation is complex, one which the human visual system can solve in many cases without difficulty. The future directions for research on this topic touch on many areas of computer vision — from learning to estimation, image analysis to geometry. The three main limitations of the work presented are the proposal of an update mechanism for regions, a method for innovating previously unseen semantic objects and the problem of tracking through severe occlusions. In this section some directions are proposed for this research.

Update mechanism for regions

The strategies evaluated for predicting region-based representations were generally found to have little improvement over simply using the previous estimate of the region location.

to limit potential matches. Finally, unmatched regions can be gathered into potential new objects using connectivity constraints within the scene. A part of this work would be the mechanisms for merging and splitting objects that are thought to belong the same physical object.

Occlusion

The problem of occlusion is present in many real-world applications of computer vision. For video-object segmentation it is important to explicitly handle the interaction of objects, specifically the covered and uncovered regions of such objects. It is proposed that this can be achieved by using mesh based methods where the regions are treated as nodes in a graph formed from the mesh. By allowing nodes to be covered or uncovered and using physical constraints on the warping of the mesh it may be possible to allow a region of the mesh to be tracked during occlusion using the partial observation. For full occlusions a mechanism is required to store descriptors of the video objects such that matching is possible on reappearance.

Prior Knowledge

Humans can efficiently see objects in images and video because they have *a priori* knowledge of objects that can exist in the world. This prior knowledge helps group together features and regions to perceive meaningful semantic objects. In video object segmentation it would be interesting to incorporate prior knowledge about objects to help constraint the solution for the segmentation. This could be achieved perhaps by starting with simple primitive shapes before looking at full object representations. Since the world is 3D it is logical that the prior knowledge would take the form of 3D models, developing a top-down paradigm for video object segmentation. Top-down modelling is often used for tracking and recognition in constrained scenes and the sheer quantity of objects (not to mention complexity e.g. articulated structures) would complicate the creation of a generic 3D database of objects. This requirement limits top-down modelling of video objects to specific constrained applications.

Chapter 8

Personal publications

The work described in this thesis has been presented in the following publications:

- D.J. Thirde and G.A. Jones, *H*

ERROR: nocurrentpoint
OFFENDING COMMAND: currentpoint

STACK: