Video Object Segmentation using Phase-based Detection of Moving Object Boundaries

Submitted by

Long Thang To

for the degree of

Doctor of Philosophy

School of Information Technology and Electrical Engineering

University College

University of New South Wales

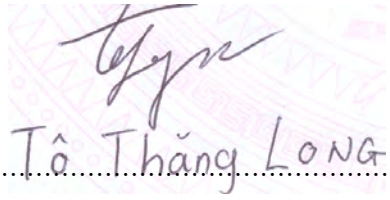
Australian Defence Force Academy

October 2005

Declaration of Originality

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues, with whom I have worked at UNSW or elsewhere, during my candidature, is fully acknowledged.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.



Signed .....Tô Thăng LONG.....

on this 4^{th} day of *November* in the year 2005

Acknowledgements

This work would not have been possible without the help that I have received throughout my stay at the school of Electrical Engineering. I would like to thank my supervisors, Dr. Mark Pickering, Professor John Arnold, and Associate Professor Michael Frater, for the attention, support and encouragement during my study at the school, and for proof reading and providing helpful feedback on the drafts.

Special thanks to Mark for the early suggestion of using the same phase information in formulating the difference image. Also, I would like to acknowledge and thank Getian Ye for the work on the super-resolution object-based video codec, which contributed to the results in chapter 6.

Finally, I am very grateful for the financial support from the University College and the school, which made it possible for me to carry this project to a completion.

Abstract

A video sequence often contains a number of objects. For each object, the motion of its projection on the video frames is affected by its movement in 3-D space, as well as the movement of the camera. Video object segmentation refers to the task of delineating and distinguishing different objects that exist in a series of video frames.

Segmentation of moving objects from a two-dimensional video is difficult due to the lack of depth information at the boundaries between different objects. As the motion incoherency of a region is intrinsically linked to the presence of such boundaries and vice versa, a failure to recognise a discontinuity in the motion field, or the use of an incorrect motion, often leads directly to errors in the segmentation result. In addition, many defects in a segmentation mask are also located in the vicinity of moving object boundaries, due to the unreliability of motion estimation in these regions.

The approach to segmentation in this work comprises of three stages. In the first part, a phase-based method is devised for detection of moving object boundaries. This detection scheme is based on the characteristics of a phase-matched difference image, and is shown to be sensitive to even small disruptions to a coherent motion field. In the second part, a spatio-temporal approach for object segmentation is introduced, which involves a spatial segmentation in the detected boundary region, followed by a motion-based region-merging operation using three temporally adjacent video frames. In the third stage, a multiple-frame approach for stabilisation of object masks is introduced to alleviate the defects which may have existed earlier in a local segmentation, and to improve upon the temporal consistency of object boundaries in the segmentation masks along a sequence.

The feasibility of the proposed work is demonstrated at each stage through examples carried out on a number of real video sequences. In the presence of another object motion, the phase-based boundary detection method is shown to be much more sensitive than direct measures such as sum-of-squared error on a motion-compensated difference image. The three-frame segmentation scheme also compares favourably with a recently proposed method initiated from a non-selective spatial segmentation. In addition, improvements in the quality of the object masks after the stabilisation stage are also observed both quantitatively and visually. The final segmentation result is then used in an experimental object-based video compression framework, which also shows improvements in efficiency over a contemporary video coding method.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A video sequence contains projections of real-world observations into a series of two-dimensional representations. The capability of the human vision system to recognise and distinguish between different objects and features, whether from a direct observation of the environment or through the view of a recorded video, seems almost spontaneous and effortless, aided no less by the fact that much of this process is facilitated by complex interactions between 10^{10} or so neurones in the human brain [1].

A desirable goal for researchers working in the area of digital video processing is a realisation of the associations between the digitised data stream and the individual objects, via a means of algorithmic reasoning. While the general objective of analysis and understanding of video content may appear as more pragmatic, the difficulty in reaching this goal in practice has generated a number of discussions on the adequacy of the perspectives in use to approach the problem [2–12]. On one hand, a reconstructive regime is advocated for a generic approach with a self-guided ability to model and reconstruct a scene from the knowledge gained from the analysis. A purposive paradigm, on the other hand, is argued for a more explicit account of the objective of a vision system at its inception. As was pointed out in these discussions, it would be difficult to imagine a vision system which does not entail any purpose, whereas in a reconstructive approach, such purposes may already be

masqueraded by the extra constraints being imposed on the data during processing. In fact, many of the outstanding problems in video processing are seen as relevant to both approaches.

The work in this thesis belongs to the area of moving object segmentation, a research topic which often serves as a prelude to further understanding of visual content and to applications in video representation. In the context of a video frame, object segmentation refers to an ability to accurately delineate and distinguish different objects that exist in the scene. Unlike object recognition, which is considered as a conscious process handled by the visual cortex, the task of locating an object position is delegated to another part of the brain, the *superior colliculus*, where it can even be carried out unconsciously [13]. In many video processing tasks, object segmentation can also be treated as an independent process. For example, many video compression experts see objects as the solution to a compact representation of temporal data, which promises a more efficient use of transmission bandwidth and improved object-based functionalities at both encoder and decoder; computer vision applications, on the other hand, may rely on such information as a step toward gaining the knowledge about the actual object structures and surfaces in three-dimensional space.

The problems associated with segmentation of moving objects stem largely from the reduced dimensionality of the video data. The absence of depth information in a two-dimensional representation requires an algorithmic approach to deduce the object separation and boundary based on correspondences amongst frames of a video sequence. In a segmentation method, the decision to assign each region (or pixel) in a video picture to an object is often made according to a partial or collective support received for the following conditions:

- Spatial coherency: A contiguous region with constant illumination (and/or color) is likely to belong to one object, and
- Temporal coherency: A moving object may be characterised by its motion, or its motion can distinguish it from others in the neighbourhood.

The first condition, while it may be directly verifiable using the pixel values of a region, is neither strictly sufficient nor necessary for object segmentation. A natural object may be comprised of multiple patches of different colors and textures, or two objects of a similar color may be moving past each other at a picture location, therefore allowing a constant-color patch to form across their boundary. In addition, an initial decomposition of a picture into spatially-coherent segments often results in a large number of candidate regions being classified. On the other hand, there is a strong incentive to involve temporal support in segmentation, as the number of moving objects in a scene is often limited, and classification may become less complicated if their motion patterns are properly identified.

The particular difficulty with using motion arises from the fact that initially it is an unknown and unregulated quantity. Subject to the requirement of constant image brightness along the motion trajectory, the local motion estimation is formulated as [14]:

$$I_x.v_x + I_y.v_y + I_t = 0 \tag{1.1}$$

with I_x, I_y and I_t being the partial derivatives for the video frame along the x, y and t directions, and v_x and v_y being the unknown x and y components of the local velocity. As one equation with two unknowns, it reflects the ill-posed nature of the estimation problem. Most forms of motion estimation therefore need to exercise additional constraints on the solutions, such as requiring all pixels within a region to follow the same motion pattern as in a region-based estimation.

Given an arbitrary video region, a question often asked during segmentation is whether the region contains a boundary between moving objects. Unfortunately, because motion is only an approximate quantity, the answer is usually less than forthcoming. To establish that a region is boundary-free, one often has to rely on the assumption that there exists a single motion which minimises the interframe difference on the given region. This difference is however affected by two major factors, firstly an ability to properly model this single motion, and secondly an absence of spatial undersampling as the reproducible condition for constant image

brightness along the motion trajectory. In other words, the interframe difference on a boundary-free region may still remain substantial if either of the above conditions is violated. In addition, the presence of a moving object boundary may be masqueraded in a region-based estimation under the following circumstances:

- With little variation in illumination, local estimation is unreliable and it is therefore difficult to realise any actual object boundary in the region, even if such a boundary exists.
- An additional single-motion or smoothness constraint may be incorrectly imposed on a region with motion discontinuities, resulting in an overfitting estimated motion.

A failure to recognise the presence of a moving object boundary, or a reliance on an incorrect motion in the object classification process, undermines the integrity of segmentation results.

The setting theme for segmentation in this thesis is a scheme to address the issues of boundary detection and motion reliability. It is argued that while both spatial and temporal supports should be considered for object segmentation, there is much less a need to resort to the spatial support in a video region where no moving object boundary is detected, because such an area is already likely to belong to the interior of a single moving object. The spatial support should however be relied on selectively to obtain more accurate local motion information in video regions where the support for temporal coherency is yet unfounded, due to either the presence of a moving object boundary, or an unreliable initial estimation. Instances where both image intensity and motion are unreliable for segmentation purposes, such as at the boundary between two objects of the same color, are seen as irregular, and the solution to such instances involves extending the domain of segmentation to a longer image sequence, beyond the local frames where the condition may exist.

Based on this framework, the thesis delivers its findings in the following arrangement.

Chapter 2 provides a brief overview of the array of segmentation techniques from the contemporary literatures. It also states the positions taken by this work with regard

to a number of issues involving the use of motion information in segmentation.

Chapter 3 introduces a new measure for detection of moving object boundaries, based on the characteristic of a phase-matched difference image. An experimental and theoretical framework is developed to demonstrate the effects of an object boundary, or an incorrect motion, on this phase-matched difference image. In particular, it is shown that a departure from a good estimation on a boundary-free region does not only result in a higher energy in the difference image, but is even more acutely characterised by a significant shift of its energy into the low frequency components. The latter property is then used to detect video regions which are straddling a moving object boundary and/or subject to an initially unreliable motion. The sensitivity of this detection criterion is also compared against other conventional measures, such as sum-of-squared errors on a motion-compensated difference image.

Chapter 4 begins with a spatial segmentation outside the video regions classified as boundary-free. It also features a region-based motion estimation and clustering algorithm to integrate these spatial segments and the boundary-free areas into corresponding objects. The initial spatial segmentation includes an allowance for unobstructed formation of segments over a region of low texture. The motion-based region clustering uses backward-estimated affine motions, and assigns a best-fit motion in the neighbourhood to a region. The best-fit criterion is also assessed using a forward projection of the estimated motion on the next frame, in order to reject candidate motions which may be the result of an overfitting or noisy estimation. The segmentation result is then compared against a method which relies on a combination of global spatial segmentation and subsequent motion-based region merging.

Chapter 5 proposes a mask stabilisation approach based on the assumption of rigid motions for object movements. Defects in a local segmentation are detected using a sequence-based mask-referencing scheme, from the observation that a segmentation mask corresponding to a legitimate object is also often well-referenced by its counterpart at other frames. A motion-based temporal averaging process is then used on

each local object mask to remove the effects of noisy segmentation around moving object boundaries, or compensate for the lack of contrast across such boundaries which may persist temporarily. It is also shown that the mask stabilisation scheme can be extended to long image sequences by implementing it on overlapping, fixed-length sections of a video clip, allowing a gradual transition from one set of results to another. At the end of Chapter 5, an objective measure of temporal consistency is adopted to demonstrate the improvement in the quality of the segmentation mask after the stabilisation.

Chapter 6 incorporates the segmentation result into an object-based video compression scheme. At the encoder, each moving object is represented by a single super-resolution sprite image, complete with a motion trajectory to enable its reconstruction at the decoder. The coding efficiency of this object-based compression approach is also compared against results for the contemporary H.264 video coding scheme.

Finally, Chapter 7 provides a summary of the major achievements of this work, and a reflection on a number of issues that the author considers as potential and worthy targets for future investigation.

Chapter 2

Object segmentation: An overview

2.1 Introduction

Computer-aided object segmentation refers to the task of grouping discrete pixels from an image into clusters, each of which corresponds in its entirety to one individual object or part thereof. According to the MIT encyclopedia of the cognitive sciences [15], they represent "attempts to construct algorithmic implementations of various grouping factors", derived from the principles of grouping under the Gestalt perception [16]. Similarity measures account for a large number of these principles, often observed from properties such as colors, sizes, locations (*proximity*) and motions. While there also exist other grouping criteria in addition to these, most computer-based segmentation techniques are constructed from a manipulation of one or more of these measures, to decompose an image into individual objects. The specific interpretation of a similarity measure may vary widely between applications. For example, a simple foreground extraction by including only pixels not conforming to a global motion could also be seen as using a measure of motion dissimilarity.

The following sections provide an overview into a number of the segmentation approaches which have been introduced both recently and in the past. Since the area is still evolving, no universal solution - and sometimes none at all - exists for seg-

mentation on every type of video or image. The main objective is to identify the distinctions between these different methods, as a prelude to the work then described in the subsequent chapters.

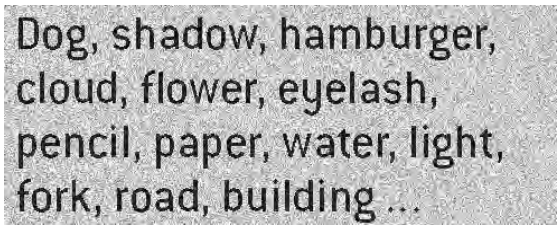
2.2 Segmentation of static images

Segmentation of still images has received extensive attention in the literature. A large number of contributions to image segmentation lend themselves well for video segmentation, as the semantics of an object are often unchanged in both cases.

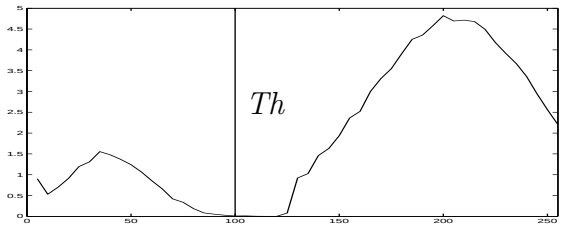
A basic image segmentation method most likely falls into one of four categories [17]: Threshold-based, edge/contour-based, region growing, or texture-based. In addition, some methods may be classified as hybrid techniques as they involve a combination of two or more categories.

2.2.1 Threshold-based

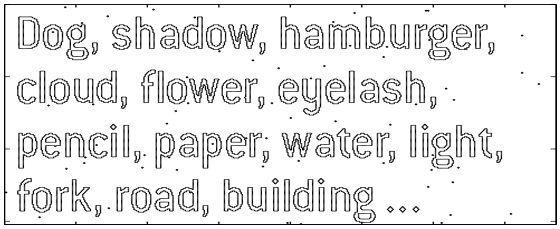
Both threshold-based and region growing methods aim to divide a gray-level image into spatial clusters according to the similarity of their pixels. In a method proposed by Otsu [18], and widely adopted later in optical character recognition applications, a threshold level is selected to partition the histogram of an image in such a way as to maximise the ratio of the between-class variance to the within-class variance, for the different groups of pixels which are separated by the threshold. The objective is to form spatial regions which have little or no fluctuation in their luminance, while keeping the separation between different regions. The method, however, requires that the number of regions are known at the input, and its application is most appropriate when there are two regions such as in a scan of a text-only document. If the objects are of more diverse textures, such a method would not be effective as the image histogram is spread out more evenly and its efficiency also reduces with an increased number of regions.



(a) A image with text



(b) Histogram and thresholding



(c) Segmented text

Figure 2.1: Segmentation by thresholding the image histogram [18]

A threshold may also be used indirectly as a means to reduce the number of features in an image as a pre-processing step. In using the HSV color space for segmentation in [19], a decision to select either the intensity level or the hue as a more appropriate descriptor for the image at each pixel location is made based on a threshold placed on the saturation level. Based on the observation that image features can be identified more easily using the intensity at low saturation, and the hue at high saturation, this processing leads to a reduced set of features which are then classified into objects by using a K-means clustering method.

For a number of specific applications, the objects of interest may be distinguished by their distinctive color characteristics. The selection thresholds can be devised according to such prior knowledge in order to extract an object from the scene. For example, [20] used the color characteristic of human skin to segment the face regions for presenters from head-and-shoulder sequences. Color properties also serve as a primary feature for road-sign detection in [21].

2.2.2 Region growing

Region growing schemes are typically implemented by identifying some pixels or groups of pixels as *seeds*, each of which is seen as the core to a region. Through an aggregation of pixels, such schemes allow the seeds to grow into a corresponding set of regions which eventually cover the entire picture.

An example is a split-and-merge strategy [17], which provides a direct way to enforce the spatial coherency of a region, without limiting the number of regions within an image. In a bottom-up approach, it usually at first searches for the basic image blocks where all pixels are of a similar value. These seeds can be found systematically by first partitioning the image into regular blocks, and on each block by sub-division into smaller blocks, if its pixels still show significant fluctuation (e.g. by having a large variance). The splitting step, which often produces many small blocks, is followed by a merging stage, where spatially-connected components are combined into one segment if they all correspond to a similar color. The segmentation stops when there is no further merging between adjacent segments.

An issue of some concerns with a region growing approach is that the segmentation results are often dependent on the selection of the seeds and the order in which the subsequent aggregations of pixels toward each seed are performed. A study of symmetric region growing in [22] provides a theoretical framework which suggested that a scheme can be made invariant to an initial selection of seeds if the operators assigned for the growing step (i.e. merging of two regions) are symmetric themselves. The accuracy of the result is, however, still dependent on how these growing criteria are designed and implemented.

Another stream belonging to region growing involves methods based on watershed segmentation [23]. The effectiveness of such methods is typically enhanced when used in conjunction with an appropriate pre-filtering. A watershed segmentation considers an image plane as a topography map with peaks and troughs. Selections of the initial seeds are therefore unambiguous, as they represent the local minimums on this surface. By gradually raising an (imaginary) water table, and constructing

a wall every time two adjacent seeds become connected by the water, the surface is completely partitioned into a number of regions. This result is also subject to oversegmentation, and further consolidation is often required by merging of adjacent watershed segments. To alleviate the effect of oversegmentation, the gradient image is often used at the initial segmentation instead of the intensity image itself,

2.2.3 Edge and contour-based

The shape of an object is also often delineated by a closed contour. Segmentation can be seen as identifying all the edge pixels associated with an object. In practice, edge detection usually contains discontinuities along the boundaries, due to the effects of noises and low contrast levels. Therefore a process of edge-linking is often needed before a closed contour can be established.

In a local processing, edge points within a small neighbourhood are examined for the likelihood that they belong to the same boundary, by comparing the difference in the contrast, and/or the direction of the gradient. Points with similar contrast level, belonging to the same neighbourhood, and lying on the same line, are more likely to be part of the same contour and should be connected. The Hough transform approach [24], on the other hand, proposed that the similarity can be identified globally from the parameter space, where lines along the edge of an object should converge at the same parameter coordinates. By locating these convergent points in the parameter space, it is then possible to deduce the connection between edge points in the spatial domain, hence reconstructing the discontinuities along a boundary. The method can also be used to detect if edge points belong to a higher-order curve, however this requires additional search dimensions in the parameter space.

Another approach to finding a contour representation is described as solving an energy-minimisation problem in [25]. The energy function is taken along the line of an active contour (snake), and design of this function can be made with preference for the contour to adapt to image features such as lines, edges and end points of an edge. More recently, in [26], a closed contour for a video object plane is completed by

a filling-in process, where iterative horizontal and vertical scans of an initial Canny edge-detection are performed until a connected edge map is formed.

2.2.4 Texture-based

While edges and lines are considered visual features in an image, texture is a statistically-defined property [27]. It is often possible to extend segmentation methods dealing with intensity images to cover textures, once the level of texturedness can be quantified locally within an image.

A texture pattern can be identified using a co-occurrence matrix [17], which is established on the basis of finding the number of pairs of pixels within each neighbourhood related by a position operator. Another method to quantify textures is to construct a feature vector from the means and standard deviations obtained from the coefficients of a Gabor wavelet filter [28]. In [29], the boundary between differently-textured regions is formed from a field of edge-flow vectors, whose directions are opposite at the boundary pixels. The use of various color spaces was also considered in [30].

Image segmentation can also be obtained by considering a measure of texturedness as part of the energy-minimisation equation in active contour methods, or factored in as a distance measure in a graph-based normalised-cut algorithm [31].

2.2.5 Pre-filtering in segmentation

Prefiltering is often a necessary step prior to segmentation, as it helps reduce the effects of noise and suppress spurious features on the results. Many developments in image filtering techniques therefore can also be regarded as contributions to segmentation.

In the simplest form, a mean or a Gaussian filter can be used to remove noise from an image. For example, a Gaussian-smoothed image I_σ is obtained by convolution

of the original image I with a Gaussian kernel

$$I_\sigma = I * \left(\frac{1}{2\pi\sigma^2} e^{-\frac{|x|^2}{2\sigma^2}} \right) \tag{2.1}$$

where σ specifies the standard deviation of the distribution. In scale-space, consider the image I also as a function of time t after each convolution, Gaussian smoothing can be seen as a solution of the linear diffusion equation [32]:

$$\frac{\partial I}{\partial t} = \frac{\partial^2 I}{\partial x^2} \tag{2.2}$$

The problem associated with this smoothing when applied iteratively in a pre-filtering stage is that due to the linear diffusion, the edges and boundaries between regions of different colors are also erased in the process. As the smoothing goes on, important structures in the image gradually become blurred, affecting the accuracy of segmentation. It appears that an ideal smoothing operation should not carry out the diffusion across a region boundary. The non-linear filter in [33] allows the diffusivity to change according to the local image structure by introducing an edge-factor into the diffusion equation:

$$\partial_t I = div(D.\nabla I) \tag{2.3}$$

where div is the divergent operator, D is the diffusion matrix and ∇I represents the image gradient. The matrix D is designed to suppress diffusion in the areas with a high level of edge activity, such as by setting:

$$D = g(|\nabla I|^2) = \frac{1}{1 + |\nabla I|^2/\lambda^2} \tag{2.4}$$

where λ is the contrast parameter.

As compared to linear filtering, a non-linear filter confines the smoothing operation inside the boundaries of image patches, therefore sharpening the distinction between different color regions. Another two popular non-linear filters which have often been used with segmentation applications are median and morphological filters. The median filter operates on the basis of replacing each pixel with the median (rather than mean) value of the pixels in its neighbourhood. On the other hand,

the morphological filter [34] is based on set theories and can be constructed using a number of different structuring elements.

A common weakness with most filtering techniques is that their usefulness is rather dependent on the spatial content of the image. On a heavily-textured image, for example, a non-linear filter may not function as effectively as it does on an image with few edges and color regions. The specific type of image is therefore also a major consideration when designing a segmentation algorithm.

2.3 Object segmentation from a sequence of video images

While the ultimate objective of a segmentation algorithm is the same whether it is aimed at a still image or a video sequence, the latter offers another temporal dimension where similarity between pixels and features can be assessed, in addition to comparisons made at the spatial domain. In this temporal dimension, the parameter most commonly used to distinguish objects is motion.

If two pixels, or groups of pixels, move with equal velocity and trajectory, then it is reasonable to deduce that both of them are parts of the same object. While spatial segmentation from a static video frame can group pixels into subsets of an object, such arrangements often result in many separate segments corresponding to one object, or *oversegmentation*; it is then the motion information which helps to further decide which pixel groups can be classified into the same object.

Given the extra dimension to compare similarities between object features, segmentation methods for video cover a diverse range of algorithms. Most methods targeting unsupervised object segmentation can be classified into four main groups, namely motion-based, spatio-temporal, dominant motion analysis, and statistically-based methods, which will be reviewed in the following section.

2.3.1 Motion-based segmentation

Motion-based segmentation refers to algorithms which attempt to locate the object shape masks primarily from motion information. Unlike the absolute value associated with each pixel, motions are usually represented by a hypothesised model and its parameters must be estimated from the image data. Even when a single displacement vector is properly estimated for every pixel location, grouping of these motion vectors into object correspondences is different from a grouping that would have been performed on pixels of a spatial image. While similarity of two pixels is usually translated as the absolute difference of their values, the similarity criteria for two motion vectors often involve comparing their conformity to a yet-unknown motion model. For example, the motion vectors at two sides of a rotating wheel may be pointing in opposite directions, but they are still part of one object. Apart from the accuracy of motion, the challenge to a motion-based method is how to arrange estimated motions into the best representation for moving objects in a scene. In all but the most trivial cases, additional constraints are required before such grouping can be carried out.

A commonly used assumption is that objects are transformed under a rigid motion in the Euclidean space, which produces a parametric representation for the object motion across video frames. In [35], Wang and Adelson proposed an approach based on clustering in the motion parameter space. The underlying assumption is that distinctive motions can be separated from each other by the differences in their respective motion model parameters. Under this approach, a dense motion field on a frame is first estimated using optical flow. From this motion field, a number of hypothesis motion parameters are generated by solving for the affine model equation:

$$\begin{aligned} a_1 x + a_2 y + a_3 &= \Delta_x(x, y) \\ b_1 x + b_2 y + b_3 &= \Delta_y(x, y) \end{aligned} \tag{2.5}$$

where $\Delta_x(x, y)$ and $\Delta_y(x, y)$ are the horizontal and vertical displacements from the estimated motion field, and $[a_1 \ a_2 \ a_3 \ b_1 \ b_2 \ b_3]$ represent the parameters of the

affine model hypothesis. By means of linear regression, the affine parameters can be derived on each sampled region P as:

$$[a_1\ a_2\ a_3]^T = \left(\sum_{(x,y)\in P} [x\ y\ 1]^T [x\ y\ 1] \right)^{-1} \sum_{(x,y)\in P} \left([x\ y\ 1]^T \Delta_x(x,y) \right)$$
$$[b_1\ b_2\ b_3]^T = \left(\sum_{(x,y)\in P} [x\ y\ 1]^T [x\ y\ 1] \right)^{-1} \sum_{(x,y)\in P} \left([x\ y\ 1]^T \Delta_y(x,y) \right) \quad (2.6)$$

In these equations, P denotes a region under consideration. The seed models are required before starting the clustering process. Initially, a frame is divided into regions of non-overlapping square blocks, from each block a 6-parameter affine model is estimated by the procedure described above. Under this arrangement, the number of initial models is usually much larger than the number of objects in a scene. The authors then proposed a K-means clustering process in the parameter space to group these motion hypotheses into a smaller set of representative motion models. The distance between any two models is measured as their Euclidian distance in the parameter space. After the clustering, the segmentation result associates each object with one of the remaining motion models An extension of this technique is found [36], which proposed an alternative update method for the affine model at each step of the clustering.

A segmentation approach can also be designed with an explicit goal of reducing the coding entropy under a minimum description length (MDL) framework [37]. Assume that there exist N objects between two adjacent frames I_k and I_{k+1}, with corresponding shapes and motions denoted as $b(n)$ and $\theta(n)$, $n = \overline{1..N}$. The inter-frame relation then can be expressed as:

$$I_{k+1} = \sum_{n=1}^{N} \left(f_{(\theta(n),b(n))}(I_k) + e_n^{k+1} \right) \quad (2.7)$$

where $f_{(\theta(n),b(n))}$ indicates a transformation of the shape mask $b(n)$ using the motion parameters $\theta(n)$, and e_n^{k+1} is the residual difference associated with object n after motion compensation. The description length, or ideal coding length, associated

with frame I_{k+1} is then:

$$DL(I_{k+1}) = \sum_{n=1}^{N} \left(-log_2 P_{\theta(n)}(e_n^{k+1}) + DL_{\theta(n)} + DL_{b(n)} + L^*(m(n))\right) + L^*(N) \quad (2.8)$$

where $DL_{\theta(n)}$ and $DL_{b(n)}$ are the coding length for the motion model and the shape boundary, respectively. $P_{\theta(n)}(e_n^{k+1})$ is a probability mass function and can be calculated analytically under the assumption that the residual error e^{k+1} follows a Gaussian distribution. $L^*(x)$ denotes the optimal coding length for an integer x, with $m(n)$ being the number of parameters in model $\theta(n)$.

The coding-oriented segmentation problem then becomes one of finding a set of motion models and corresponding object masks which minimise this description length. In the cited works [37], this is achieved by merging adjacent regions using a graph-based framework. Each initial region is represented by a node in a graph, and if two connected nodes have a sum of description lengths larger than the DL associated with the combined node, the two regions are merged and their representative motion model updated.

It is apparent that MDL does not necessarily result in boundaries which coincide with object semantics, since the constraint of a coherent motion within an object is not directly imposed. The K-means clustering, on the other hand, is subject to the accuracy of the dense motion field it inherited, as the field is treated as the groundtruth from which the parametric models are generated. It is well known that optical flow techniques usually impose a degree of global smoothness in the vector field, and therefore do not respond particularly well to abrupt changes at object boundaries [14], or in the presence of occlusion. In the vicinity of a motion boundary where two distinct motion patterns exist, the requirement for global smoothness may result in a distorted flow field which in turn is more likely to be approximated by an averaged model between the two motions. Because the success of the method is highly dependent on the ability to identify unique parametric models and assign them to objects, the ambiguity in estimation makes it difficult to correctly label pixels within the boundary regions to one object or another.

It is expected that a small number of motion models remain when the clustering/merging finally stops. In [35], it is driven by the threshold placed on the distance between any two motion models in the parameter space. However, as pointed out in [38], such comparisons are also subject to a number of limitations :

- A distance in the parameter space does not readily translate into a physical measurement in the spatial domain.
- The translational components are usually much larger than other parameters (such as shearing, zoom or rotation). Hence their effect over the parameter-based distance measure may be more significant than others, which might not be desirable.
- Distance between parameters of different model types cannot be accommodated.

2.3.2 Spatio-temporal segmentation

Spatio-temporal algorithms probably form the largest and most popular category for object segmentation in video. These algorithms often combine information from coherency of colors, which exists spatially in each video frame, with coherency of motions, which is detected across frames. The main advantage of such methods is the complementary nature of spatial and temporal information. For example, motion estimation would be unreliable on a region of constant intensity, but at the same time such spatial homogeneity makes it an ideal target for spatial segmentation.

Objects are not often found with a single intensity or color, except in the most simple cases. However, because neighbouring objects are also likely to exhibit different spatial characteristics, it is possible to assume that the true boundaries between such objects are a subset of all the spatial edges located within an image [39]. In other words, an object can be decomposed into a number of segments, each of which is coherent in color or intensity. For this reason, many advances from segmentation of stationary images are also important contributions to spatio-temporal segmentation.

A large number of segmentation algorithms can be categorised as spatio-temporal, on the basis that they all depend on and use the explicit knowledge of motion and spatial coherency. The characteristic which distinguishes one technique from another is the underlying algorithm which combines this knowledge to produce one single output - the object masks. It is these masks which provide the correspondence between separate spatial segments and the moving objects. Such outcomes are strongly influenced by how the motion and spatial coherence are integrated into a unified measure of coherency, which then forms the defining criteria for the segmentation process. At one end, if the spatial factor is dominant in the measure, the method would bear resemblance to a spatial segmentation; at the other end, if the motion-factor is dominant, the method would be similar to a motion-based approach. In fact, it is still largely an unanswered question as how to combine these two measures in the most efficient way for segmentation [40].

A spatial coherence, in most cases, refers to the similarities in colors and positions amongst pixels. On the other hand, a motion-based approach can be extended so that it treats each spatial segment, instead of individual pixels, as a building block for an object, such as in [39]. In a similar initialisation as [35], a number (K) of representative affine motion models (M_i) are selected from an optical flow field v between two frames, forming the set $M = \{M_i,\ i \in [1, K]\}$. A spatial segmentation is then subsequently performed in one frame, resulting in N color segments, $C = \{C_i,\ i \in [1, N]\}$. Segmentation is then a process of labelling each spatial segment with a motion model. Since both sets M and C are usually over-segmented with regard to the number of objects and motion patterns, object consolidation is also a necessary step. This is performed by iterative updating as follows:

- Update the model parameter M_k according to the latest segmentation label L_k

$$M_k = arg \min_{M_k \in M} \sum_{(i,j) \in L_k} |v(i,j) - v_M(i,j)|^2 \tag{2.9}$$

- Update the segmentation label for each spatial segment according to the latest

set of motion models

$$L(C_n) = arg \min_{k \in [1,K]} \sum_{(i,j) \in C_n} |v(i,j) - v_{M_k}(i,j)|^2 \tag{2.10}$$

A model M_k and the associated object label L_k is considered to have reached convergence if the changes in the motion model between successive iterations fall below a threshold, at which point the iteration stops. An alternative approach, region-based intensity matching, was also proposed in the above work, where the objective of updating the model parameters and motion labels is to minimise the residual squared difference over each segment, rather than the residual motion field. Both approaches appear to give similar performances when the number of objects, K, is properly set.

Tweed and Calway, in [41], proposed that the spatial segmentation is performed at the block level, instead of over the whole frame. For each frame block, two motion vectors are selected by correlation on the block itself and the overlapped neighbouring blocks. For any two adjacent spatial segments inside the block, a two-component measure of support was introduced to assign a motion vector to each segment, and to infer their relative depth ordering. Because all the segments are initiated at the block level, the algorithm also requires complex processing to merge these segments, and to synchronise the depth ordering amongst them.

A thorough spatial segmentation also provides a good starting point, as it reduces the number of initial regions which requires motion estimation and merging. In [42], a frame is first prefiltered using a morphological operation, then a multiscale morphological gradient operator is applied to produce a gradient image. A watershed segmentation is performed on this gradient image, followed by merging of adjacent regions in order to reduce the spatial oversegmentation. This merging is driven by the removal of *weak* edges, defined by a low gradient across pixels at the region boundaries. Adjacent regions with a weak common boundary are merged, ultimately resulting in a smaller number of spatial regions. A classification of moving objects is performed with the motion information from each segment, estimated using hier-

archical block matching and parameter fitting under a least-squares approximation. The distance between two neighbouring segments is quantified as the increment of mean-square motion-compensated error should they be merged, i.e. :

$$\Delta = \frac{E_{AB} - E_A - E_B}{N_A + N_B} \tag{2.11}$$

where A and B represents the adjacent segments, E_A, E_B and E_{AB} are the mean squared errors associated with A, B and the combined region AB, N_A and N_B are the number of pixels in each segment. A low value of Δ suggests there exist a motion model on the combined region which functions as well as it would on the individual regions, and the regions therefore should be merged. A large value of Δ, on the other hand, suggests that they probably belong to two different moving regions, and should remain separated.

More recently, the method in [43] employed a more elaborate spatial segmentation, combined with simpler motion-based postprocessing. Besides the intensity, color information was also used to improve the quality of spatial segmentation. Extensive non-linear filtering operations and a watershed transformation are first performed on a color image to produce the spatial segments. Object masks are then defined by grouping together segments with motion vectors in the same directions on a fixed co-ordinate system.

The most critical factor in spatio-temporal segmentation is the design of a merging criteria, which quantifies the similarity between regions. Relatively independent processing of spatial and temporal information can be seen in the previous method, as each play a dominant role in part of the segmentation. There have been attempts to create a *joint-similarity* measure which account for both domains in one comparison. One example is [40], where spatial similarity is formulated as the results of a statistical test on the pixel values along the common border of two adjacent regions, sampled across all pairs in the whole frame. The temporal similarity, on the other hand, is measured by a modified Kolmogorov-Smirnov test, which aims to characterise the difference in the distributions of the *residual* motion fields. The residual fields are created by subtracting from the original dense field, the parameterised

fields formed by either estimated models. The joint measure of spatio-temporal similarity between two regions, $Sim(r_1, r_2)$, is formulated as a hybrid function of the spatial similarity $S(r_1, r_2)$ and temporal similarity $T(r_1, r_2)$:

$$Sim(r_1, r_2) = T(r_1, r_2) - fT(r_1, r_2)(Max - S(r_1, r_2)) \tag{2.12}$$

The Max parameter is chosen as the highest value of spatial similarity in the neighbourhood of region r_1. The contributions from the spatial and temporal terms in this case are however not equal. The spatial information is of little help inside high-contrast regions, reflecting by a small Max which would also make the spatial term insignificant. Within low-contrast regions, i.e. regions with more constant color, the spatial term plays a more significant role. In either cases, the temporal factor has the driving role in this combined measure. The similarity measure, as described in [40], "only function as a corrective factor", as the impact of the term $(Max - S(r_1, r_2))$ only becomes noticeable when there exists spatial coherency within the neighbourhood, and even then its contribution still depends on the temporal measure. It should also be noted that this spatio-temporal measure is used on the basis that the frame has already been partitioned into a number of regions, presumably with a spatial segmentation.

In [44], a joint-similarity measure also unifies the motion and the intensity comparison in a single quantity, but assigns a linear weight to each criterion. Specifically, the similarity that a pixel at local (x, y) has with a region R is:

$$Sim(x, y; R) = \alpha T(x, y; R) + (1 - \alpha)S(x, y; R) \tag{2.13}$$

where α is the weight factor. The temporal similarity T in this case is synonymous with the residual difference at pixel (x, y) using the motion model associated with region R, and the spatial similarity is measured as the difference between the pixel itself and a value of a polynomial at the same location, approximated from the region R. The merging procedure is also preceded by a combination of morphological filtering and watershed segmentation steps.

As was mentioned previously, developments in the area of spatial filtering and segmentation have many direct benefits to the segmentation of moving objects. As

object classification becomes more complicated and unstable when the number of candidate regions increases, the main strength of a good spatial segmentation is to keep this initial number to a minimum. Labelling a region, rather than individual pixels, provides a degree of regularisation to the segmentation process, as it avoids having to accommodate overfitting motion fields over object boundaries.

However, there remain questions on the contribution from the temporal and spatial towards any overall measure of similarity. It would be relatively straightforward on sequences where the majority of objects, or object parts, can already be segmented using the constraint of spatial coherency . On the other hand, spatial processing may also pose a number of problems, when the initial assumption of the object boundaries as a subset of all spatial edges is not met:

- Spatial undersegmentation, which happens when regions belonging to different moving objects are inadvertently merged during a spatial segmentation, causes a loss of object boundaries which is rather difficult to recover from in subsequent processing. In addition, it may also lead to the assumption of a single motion being imposed on the undersegmented region, which results in an incorrect estimation of model parameters, hence preventing other legitimate region-mergings within the neighbourhood.
- While filtering is necessary to suppress the effects of noise and reduce oversegmentation, excessive pre-filtering may lead to structural changes and erasure of spatial boundaries in an image. It may also be an indirect cause for spatial undersegmentation.

In addition, video images with a high level of texturedness and intensity/color variance are usually subject to oversegmentation in any initial spatial processing. For the purpose of object labelling, the effectiveness of the spatial segmentation decreases as the number of regions to be classified increases.

2.3.3 Statistical approaches

The segmentation problem has also been stated alternatively under a statistical framework. As each object is labelled after segmentation, the labelled field can be modelled as an *a posteriori* probability function with regard to the algorithm inputs, such as estimated motions and image intensity.

In [45], with an estimated optical flow field v supplied as the input, the probability of a segmentation label L is formulated using the Bayes rules as:

$$p(L|v) = \frac{p(v|L).p(L)}{p(v)} \tag{2.14}$$

Adhering to this formula, a properly segmented frame corresponds to a maximum of the probability $p(L|v)$. The problem then becomes one of finding the solution L to maximise the right hand side of this equation. Given that the flow field is obtained independently of any subsequent segmentation, the actual maximisation is carried out without the denominator $p(v)$. Amongst the remaining terms, $p(v|L)$ is the conditional probability of the models from the label field L being good approximation for the estimated field v, and $p(L)$ is the prior probability distribution function of the label L. The solution to the maximisation is located through simulated annealing, with $p(v|L)$ being modelled as a Gaussian distribution and $p(L)$ as a Gibbs distribution.

As motion estimation and segmentation are often considered inter-dependent problems, it would be possible to argue that the probability of the motion field $p(v)$ should not be made independent of the segmentation. The approach in [46] proposed a concurrent updating for both motion estimations and segmentation labels by reformulating the probability under optimisation as:

$$p(v, L|I_k, I_{k-1}) = \frac{p(I_k|v, L, I_{k-1}).p(v|L, I_{k-1}).p(L|I_{k-1})}{p(I_k|I_{k-1})} \tag{2.15}$$

where I_k and I_{k-1} represent the intensity image of the current and previous frame. With the denominator also omitted from the maximisation, the three terms in the numerator are all modelled as Gibbs distributions with their respective potential

functions reflecting on the residual frame difference, residual motion field, and spatial connectedness of segments. The optimisation is then performed using a highest-confidence first (HCF) method.

More recently, an approach to include more than two frames in the probability function was introduced in [47]. Instead of a dense motion field, it is initialised from a set of spatial watershed segmentation regions, with the probability for a segmentation mask at frame k defined as:

$$p(L_k|I_k, H_k, L_{k-1}, I_{k-1}, I_{k+1}) = \frac{p(I_k|L_k, H_k, L_{k-1}, I_{k-1}, I_{k+1}).p(L_{k-1}|L_k, H_k).p(L|H_k)}{p(I_k|H_k, L_{k-1}, I_{k-1}, I_{k+1})} \tag{2.16}$$

where I_k, I_{k-1} and I_{k+1} denote the current, previous and next intensity images, L_{k-1} is the mask obtained in the previous frame, and H_k is the set of motion parameters associated with objects in the current frame k.

Last but not least, in addition to spatial intensity and temporal information, color properties have also been factored in as part of the probability formulation [48].

2.3.4 Dominant motion analysis

A segmentation approach based on dominant motion analysis often separates objects from a video into two classes: those which conform to a globally-estimated motion, and those which do not. For the purposes of classification, the former is usually referred to as the background, and the latter as the foreground. The underlying assumption of these approaches is the background being subject to a unique global motion, which can be systematically estimated and compensated, such as camera panning on a planar surface. Foreground objects can be defined as any object which moves independently of the background. The reason these methods are not classified as motion-based is because the foreground segments are not often grouped according to a temporal similarity criteria. Instead, two deciding factors which assign two spatial segments into one foreground object are:

- Divergence from the background motion, and

- Spatial adjacency

In the simplest case, a scene may contain one foreground object, and the background is stationary. A common method to initialise the foreground objects is by first obtaining a change detection mask (CDM) between two adjacent frames. Because the background is stationary, the pixel values should only change in the area involving parts of moving objects, or parts of the background which become uncovered due to foreground motion. The CDM is created by thresholding the difference image of two adjacent frames, as the effects of sampling noise are usually present even on a stationary background.

In [49], the CDM is used in conjunction with a background registration scheme. A long-term background is obtained by monitoring the number of changes in the CDM, with the most temporally-consistent pixels assigned to a registered background. At each frame, this registered background is then used for comparison with the current frame, from which the foreground object would stand out as being different. A change detection mask is created between the registered background image and the current frame image, from which the foreground mask can be constructed. Post-processing is performed using morphological operators to remove isolated segments from the resulting mask.

In many cases where the background image is not stationary, because of its own and/or the camera movement, it is then necessary to compensate for this background motion before calculating a CDM, by registering one image toward the other before subtraction. Extraction of foreground objects may become more complicated due to (a) any inaccuracy of the estimated motion would lead to mismatches in the difference image, and (b) the interpolation errors from the registration. Nevertheless, there is a fundamental difference between changes due to object motions, and those induced on the background, as noted in [50]. The inter-frame differences observed within the background region are due mostly to camera noise and subpixel-inaccuracy compensation, which can be largely modelled as a Gaussian distribution. The error due to discrepancies between object motions are, on the other hand, *highly*

structured signals and therefore are not usually Gaussian. Foreground regions can then be detected by locating parts of the inter-frame difference whose underlying statistics represent deviations from a Gaussian distribution.

Another relatively different approach, based on tracking and updating of boundary pixels, is proposed by Meier and Ngan in [26]. While an initial mask may still be obtained by using a change detection mask, or an alternative procedure using morphological motion filtering, subsequent masks for a foreground object are reconstructed by a process of edge-detection and edge-matching. A Canny edge detector is used in each frame, followed by a boundary-matching algorithm using the Hausdorff distance measure to find the best match for the boundary pixels of the object mask from the previous frame. A filling-in algorithm is then implemented on the matching pixels to form a closed-contour shape mask for the current frame.

One of the strengths of a foreground/background approach is its ability to identify foreground objects without the need for an explicit motion model, because only a deviation from the global motion is required to trigger this foreground assignment, as demonstrated by the above algorithms. This may be a very attractive option for objects with hard-to-characterise movements. The methods however have some limitations, inherited directly from the classification strategy. While distinction between foreground and background objects is relatively straightforward, it is difficult to separate two foreground objects whose projections are overlapping. Its applications are therefore often limited to sequences where there already exists spatial separation between multiple foreground objects.

Another problem is associated with the estimation of the background motion. An accurate background motion is critical for a correct segmentation. Even when the assumption of a global motion is valid, however, the foreground objects acts as outliers which would affect the estimation accuracy, especially when their relative sizes and motions are significant when compared with those of the background. In addition, the background may be composed of objects of different planar and/or parametric surfaces, which may affect the fitness of a global motion.

2.3.5 Other segmentation methods

Some segmentation methods may not be classified strictly in one category, but in fact would be seen as inheriting features from two or more groups. For example, the method in [51] is not only relying on a dominant motion estimation, but also follows a motion-based scheme as it sequentially removes the dominant object after each step so that focus can be given to the next dominant object. Earlier papers such as [52] also advocated a sequential processing of dominant motions, although segmentation results were not explicitly shown. Work in [53], on the other hand, unified a spatio-temporal and a statistical labelling approach in one segmentation framework.

Last but not least, as is often the case, every segmentation method has its own advantages and disadvantages. A combined approach can alleviate some weaknesses and improve upon others. However, it is still true that none of the existing techniques come near the visual ability often taken for granted by human beings. For this reason, user-intervention is offered and supported as an additional cue in a number of techniques, which can be identified as an *interactive* category. In [54], inputs from a user help to group spatio-temporal regions into objects, when such decisions are semantically-correct but may appear ambiguous to an autonomous approach. Alternatively, users are required to help define the object contours at some key frames of a sequence, before handing it over to a segmentation and tracking stage [55]. A user-guided segmentation may also serve as a starting point for maximising an a posteriori probability of a segmentation label field, as in a statistically-based approach [56].

2.4 Motion estimation in segmentation

In the context of moving object segmentation, there are two primary types of defects, oversegmentation or undersegmentation, and both of them can coexist in the same shape mask. While the causes for such defects are many, and may originate spatially

or temporally, it could be argued that a motion-based solution is necessary for a stable performance along a video sequence.

2.4.1 Oversegmentation and undersegmentation

Oversegmentation is the failure of an algorithm to recover a complete shape mask for an object, and instead recognise some of its parts as independent objects. Undersegmentation, in comparison, happens when a segmentation shape mask inaccurately contains more than one object.

In the case of oversegmentation, the direct reason is often because the clustering process has been unable to merge oversegmented regions into the same mask, due to the merging criteria not being sufficiently satisfied. Undersegmentation, on the other hand, may be inherited directly from under-segmented regions at the initialisation, or due to erroneous merging of regions across an object boundary. While a merging criterion may be partially responsible in both cases, inputs to a clustering process also strongly influence the accuracy of its output.

At initialisation, a motion-based approach may treat each pixel as a segment with its own motion vector, whereas a spatio-temporal method may choose to perform a complete spatial segmentation on a picture before using any other similarity measures to regroup them. However, with a large number of initial segments to classify, not only does the complexity of an algorithm increase, but the stability of the result is also affected [35]. Since a clustering process often converges onto a final segmentation mask only after a number of iterations, any error associated with an initial segmentation, or committed during the merging process, is likely to have a propagating effect to subsequent clustering decisions. For example, undersegmentation compromises the spatial and/or temporal coherency of a region, making it difficult to decide on other mergings in the neighbourhood, therefore effectively creating a potential for oversegmentation elsewhere. Invariably, a large number of candidate regions translates to a higher probability that an error may occur, as well as the wider impact such errors may cause.

A conventional solution to the above situation is to try to reduce the number of candidate regions, often by an improvement in spatial preprocessing. The effectiveness of such processing is sometimes limited. Consider, for example, if two adjacent objects have similar colors across a section of their common boundary. A local spatial processing is ill-suited to recover this boundary. Likewise, spatial segmentation on a complex textured region would also be highly counter-productive, as it tends to produce an excessive number of spatial segments. While these two examples represent rather extreme cases that a segmentation algorithm might have to deal with, it also highlights the usefulness of temporal correlations under such situation. In the first case, assuming that the objects keep moving in their relatively different trajectories, there is likely a time where the obscured boundary section becomes more detectable, and it is then possible to project this boundary back onto those frames where it was not spatially visible. In the second case, implementation of an early motion-based prediction may detect that the entire textured region is moving with a coherent motion, hence bypassing any local spatial segmentation inside the region.

The need for a reduced number of candidate regions is also seen as necessary in this work, as a way to to lessen the burden on the clustering stage and to improve the stability of segmentation. However, the argument being made is that an attempt to achieve this reduction should not be made uniformly across the picture, but by confining the clustering process and initial segments to the spatial regions where they matter the most. Such regions are the neighbourhood that contains object boundaries, as the primary task of segmentation is to clearly identify object separation in these regions. Of course the knowledge of specific objects and their boundary location is an unknown at the beginning. Hence, what is needed is an early-detection scheme to locate those neighbourhoods which are the most likely to straddle an object boundary. Since the targets are object boundaries and not just any spatial edges, motion information would be invariably required.

2.4.2 Top-down and bottom-up

As it has been mentioned previously, correlations in the spatial and temporal domains form the basis for most segmentation method, and they can even be combined in a joint-similarity measure. While spatial correlation by itself is usually preconditioned by a strong spatial connectedness, a systematic assessment of temporal correlation may be carried out in either a top-down or bottom-up strategy.

A top-down approach involves first identifying the dominant object, often by an estimation of global motion. The boundary of foreground objects would then stand out as locations where the local motion does not follow this global pattern, usually indicated by a more significant residual on the motion-compensated difference image. This approach does not adapt well if the scene contains several moving objects of significant sizes and energy, since the accuracy of a global estimation would be affected under such circumstances. More specifically, if the variation between the different object motions are small, a global estimation usually produces a compromised model, making it more difficult to detect and reject non-conforming local motions at the boundaries. It should be pointed out that even though some proposals suggested that a dominant object can be sequentially removed after each analysis to accommodate a multiple-object scene, they still depend on the ability to accurately estimate the dominant motion in the first place.

In a bottom-up strategy, boundaries between objects are detected by observations within local neighbourhoods. A segmentation mask, in most cases, corresponds to a set of pixel-resolution object boundaries. Ideally, as pixels on opposite sides of a boundary between two objects are expected to show differences in motion and color properties, boundary detection by a pixel-based method should produce very accurate results. In practice, however, regions too small may be a poor target for motion estimation. The trade-off with using a larger base region for estimation, e.g. by using larger block sizes, is a reduced resolution at which object boundaries can be detected. This is, however, not viewed as an impediment, as it accomplishes the objective of locating the neighbourhood where object boundaries may pass, therefore

reducing the necessary aperture for any subsequent clustering process. In addition, a bottom-up strategy also allows individual objects to be distinguished from local constraints, which is the strategy favoured by the segmentation approach proposed in this thesis.

2.5 Summary

In this chapter we have looked at a range of techniques for segmentation of objects from still images and videos. In a single image, spatial coherency can be used to establish the local support for a region. Most image segmentation algorithms can also be extended to moving objects in a video by including the motion information, such as in a spatio-temporal approach. The link between objects and their corresponding motions across different video frames can be used to address the global support needed to regroup regions which belong to the same object, or to separate one object from another.

An efficient strategy to combine the spatial and temporal information from a video sequence is often cited as the most important factor contributing to the success of a segmentation algorithm. The difficulty in formulating such a strategy is primarily due to the unreliability of estimated motion, especially in the boundary region between two moving objects. While a global motion estimation approach can be used to detect and locate the regions under a dominant motion, the accuracy of such an estimation is subject to the significance of foreground objects and their movements. On the other hand, the support for a local region as being part of an object can be established if the region is certified as boundary-free, as well as having its motion accurately estimated.

The approach to segmentation in this work is based on the expectation that the presence of a moving object boundary would ultimately affect the choice of a segmentation strategy carried out on a selected video region. In the next chapter, a detection method for moving object boundaries is introduced within a framework

of block-based motion prediction. The objective of this detection scheme is to identify parts of a picture which belong wholly to one moving object, and which also have their motion estimated reliably. Being able to establish such regions in a video frame, it then allows the spatio-temporal segmentation efforts to be focused on areas which contain boundaries between the moving objects, a process which will be detailed in subsequent chapters of the thesis.

Chapter 3

Phase-based detection of moving object boundaries

3.1 Introduction

One of the most difficult issues facing a segmentation approach is how to combine the information available in the spatial domain, i.e. colors and illuminations, with the motion information, in order to identify objects in a video. While the value of all pixels in a captured video frame are readily known, their associated motion can only be described as an *estimated* quantity, usually derived from the spatial information across different frames, by an estimation process such as block matching [57] or optical flow [14]. Motion estimation is invariably affected by factors such as sampling noise, complexity of the motion, or a lack of illumination variations inside an estimation window [58]. In addition, ambiguity may arise from the fact that a two-dimensional motion representation may correspond to a number of different three-dimensional trajectories.

It is widely recognised that motion estimation is an inherently ill-posed problem, even under the assumption of a constant change in image brightness along the motion trajectory [14]. Further constraints are often required, usually in the form of an

assumption on the object motion. Although an estimation process almost always produces a set of motion parameters, their reliability depends on the estimation algorithm itself, and to a large degree on the additional constraints imposed on the estimation. The validity of such constraints serves as a prerequisite for accurate estimation results. For example, in block-based motion estimation, reliable motion can only be obtained based on the following two conditions:

- the selected area, for which motions are being estimated, does not contain a moving object boundary; and
- the parameters for the motion of the object are estimated correctly.

The first condition refers to the underlying assumption that all the pixels inside the selected region are subject to one motion. Only when this constraint is satisfied, does it become feasible to apply an algorithm to estimate the motion parameters. When this assumption is violated, i.e the block contains parts of different moving objects and the algorithm only produces one estimation, the presence of multiple motions cannot be accounted for. In many cases, the result corresponds to an "averaged" motion model, which is different from all the existing, true motions in the area. Similar effects can also be observed in an optical flow estimation when the smoothness constraint is applied indiscriminately over an object boundary.

In the context of moving object segmentation, accurate motion information is vital to the integrity of the object masks. When an object is formed by a cluster of regions with a similar motion, inaccurate estimation often leads to incorrect formation of the masks. Moreover, once a region has been committed as part of an object in segmentation, such defects are rarely reversible in post-processing. In order to maintain the quality of a segmentation algorithm, the motion accuracy should therefore be addressed early and carefully.

To ensure that a motion estimation produces accurate results, it is necessary to verify the validity of the assumptions. This chapter aims to resolve the question of how to identify blocks which satisfy the prerequisite constraint of not containing a motion discontinuity. A detection measure for moving object boundaries (MOB) is

introduced to classify each block from a video frame into one of the two following classes:

- *Single-motion*: assigned to blocks which do not contain a moving object boundary.
- *Multiple-motion*: assigned to blocks which contain a moving object boundary, or for which the motion cannot be estimated properly using the assumed model.

Note that the work in this chapter does not try to produce the final motion representation for every block of the video. Rather, it addresses whether such representation can be achieved *reliably* at each location. From this classification, different procedures are devised in later chapters to handle motion estimation in the two categories.

3.2 The single-motion assumption

When a block matching algorithm is performed between a current video frame and a reference frame, motion estimation produces a set of displacement vectors to match one frame to the other. If the motion is correctly identified for a block on the current frame, then a matching block can be located in the reference frame using this displacement. A residual difference is then defined as the difference between the original block on the current frame with its matching block in the reference frame. This motion-compensated difference can be considered as a by-product of the estimation process.

It is not difficult to realize that a motion-compensated difference image would rarely be an array of zero-valued pixels. It is affected by various factors such as interpolation errors from non-integer motions, sampling noise, and the accuracy of the estimated motions. Nevertheless, since this difference image indicates how well-matched the two blocks are, it is also usually used to measure the accuracy of the associated motion.

A common quantity to evaluate a difference image is the sum of squared difference (SSD), or alternatively the sum of absolute differences (SAD). These values would remain relatively small if the correct motion is compensated for. On the other hand, when there are two or more motions in the field, the residual difference cannot be minimised as much because attempting to compensate for one motion also creates a mismatch in the region under the other motion, hence increasing the difference in that region. A larger difference indicates that the motion has not been compensated for properly, which in turn suggests either an existence of multiple motions, or an inadequacy of the motion model in use. Works such as [59] used this quantity as a direct measure of motion reliability. Earlier, Anandan proposed that a confidence measure on the accuracy of an estimated motion can be deduced from the characteristics of the surface of the sum-of-squared differences [60], but also acknowledged that such measures should be accompanied by a concurrent ability to detect moving object boundaries.

A direct interpretation of the residual difference raises some issues of concern. Most motion estimation algorithms arrive at their results via the means of minimising an objective function, and in many cases this objective function *is* the same residual difference which is subsequently used to assess the accuracy of the motion. In seeking a minimum in the objective function, it may inadvertently lead to *overfitting* of the motion parameters, regardless of the multiplicity of motions in the area being considered. The result of such minimisation is usually a motion model resembling an average version of all the true motions in the scene, which has a two-fold consequence: a) the incorrect estimated motion, and b) the residual difference might be reduced to a level where it appears as if there is only one motion in the region. Under this ambiguous circumstance, a reliability measure based on the residual difference might lead to mis-classifications by labelling a region containing multiple motions as single-motion.

Such problems are much more likely to occur when the variation between different object motions is small. The following example illustrates one such case. In Figure

3.1-a, the block A on the left comprises two independent moving components, while the block B on the right is under one translation only. Figure 3.1-b shows the actual motions that are present in block A, whereas Figure 3.1-c shows the estimated motion in the same block using an affine motion model. The small difference between the two true motions, in conjunction with the assumption of an affine motion, results in the left field being approximated as a circular, single motion. Moreover, the residual difference associated with this estimation is even less than the difference on the right block, where the whole block is translated by a single (but fractional pixel) motion, whose motion is shown in Figure 3.1-d.

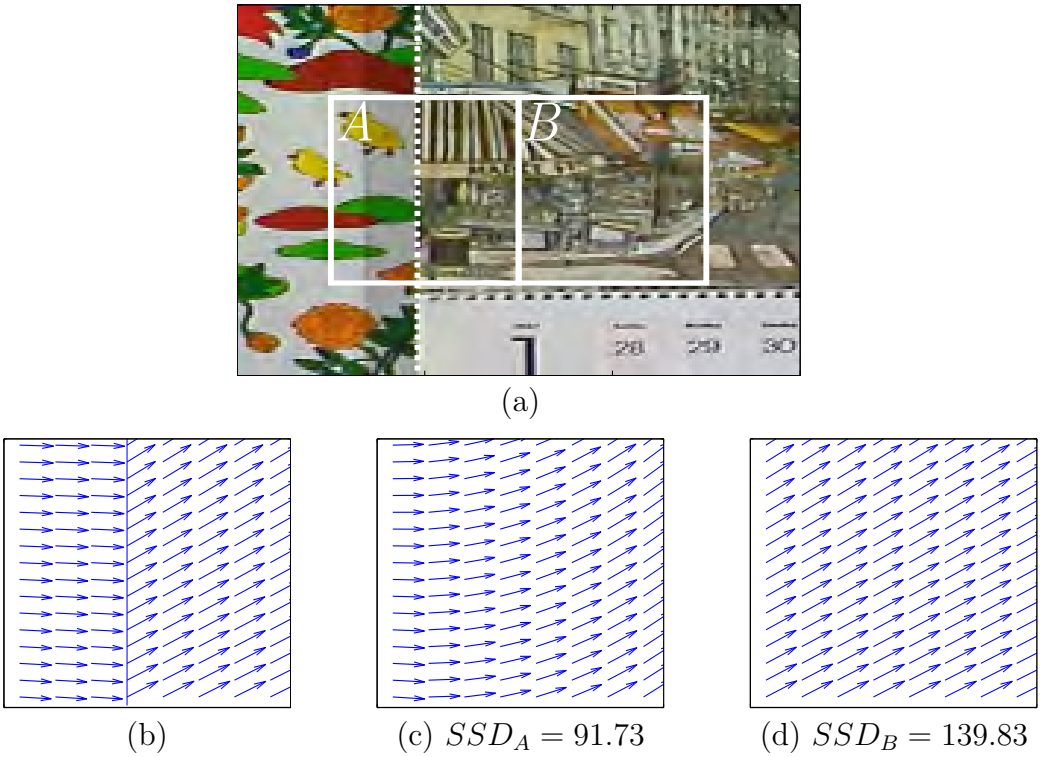


Figure 3.1: (a) Selection of blocks A and B; (b) The motion boundary and the two true motions present in block A; (c) estimated motion in block A and the residual difference; (d) estimated motion in block B and the residual difference. Motions in (c) and (d) are estimated using a 6-parameter affine model

From the magnitudes of the sum-of-squared differences alone, it would be difficult to tell that the left block has a moving object boundary, while the one on the right

does not. The problem is partly due to the blanket assumption of a single motion at the beginning. Furthermore, the subsequent attempt to verify the validity of this assumption also fails. This can be attributed to the fact that the means of verification and the means of motion estimation are inter-dependent.

The boundary detection measure proposed in the following sections aims to separate regions containing motion discontinuities from the rest of the picture. Instead of relying on the motion-compensated difference, the measure is based on a new phase-matched difference image. In addition, it does not rely on a direct quantity such as the sum-of-squared difference, but rather on the distribution of energy in the spectrum of this difference image. A phase-correlation method is implemented for the initial motion estimation, which is briefly reviewed in the next section.

3.3 Phase in an image

3.3.1 Motion estimation by phase correlation

Phase correlation was first used for motion estimation in [61]. Assume we have a translational motion between frame k and $k+1$, this motion can be modelled as follows:

$$i_k(x, y) = i_{k+1}(x + \Delta_x, y + \Delta_y) \tag{3.1}$$

where i_k and i_{k+1} represent the image intensities of blocks at two frames, (x, y) is the pixel coordinates, (Δ_x, Δ_y) is the displacement vector.

It is known that a linear shift in the spatial domain leads to a shift in phase, or

$$\mathcal{F}(f(x + \Delta_x)) = \mathcal{F}(w).e^{jw\Delta_x} \tag{3.2}$$

Therefore, by taking the Fourier transform of both sides of (3.1), we have

$$\mathcal{I}_k(w_x, w_y) = \mathcal{I}_{k+1}(w_x, w_y).e^{jw_x\Delta_x + jw_y\Delta_y} \tag{3.3}$$

The correlation product between I_k and I_{k+1} is formed by calculating their nor-

malised cross power spectrum as follows

$$\mathcal{C}_{k,k+1} = \frac{\mathcal{I}_{k+1}(w_x, w_y).\mathcal{I}_k^*(w_x, w_y)}{|\mathcal{I}_{k+1}(w_x, w_y).\mathcal{I}_k^*(w_x, w_y)|} \tag{3.4}$$

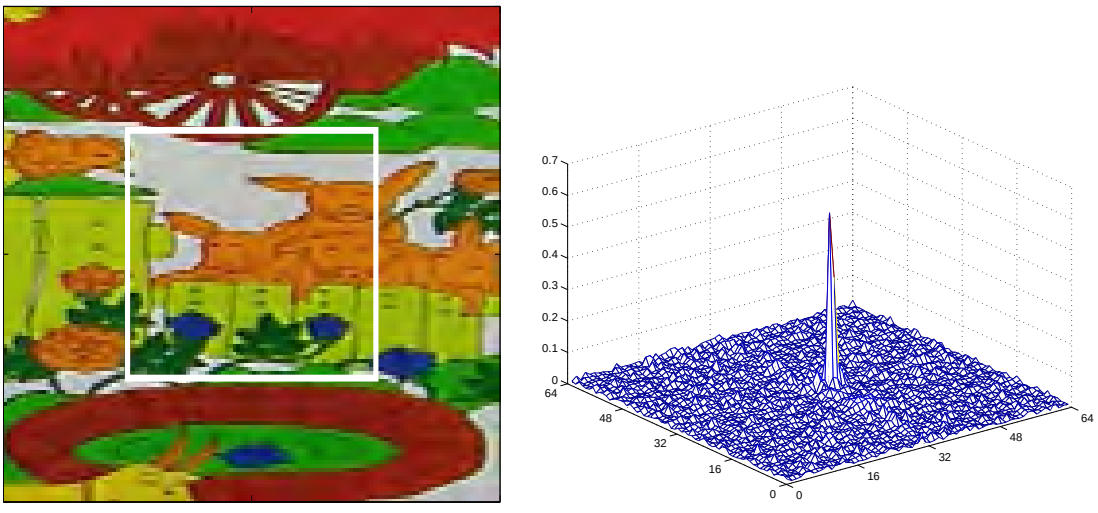
Substituting (3.3) into (3.4) results in:

$$\mathcal{C}_{k,k+1} = e^{-jw_x\Delta_x - jw_y\Delta_y} \tag{3.5}$$

By taking the inverse Fourier transform of this cross power spectrum, we have

$$c_{k,k+1} = \delta(x - \Delta_x, y - \Delta y) \tag{3.6}$$

where δ is the Dirac delta function. This phase-correlation surface in equation (3.6) corresponds to a single impulse at (Δ_x, Δ_y) and zeros at all other location. By identifying the position of this peak, one can deduce the relative shift, the displacement vector, between the two images. Following Parseval's theorem which states that the energy under a function is equal to the energy under its spectrum [62], and normalisation of the cross-correlation product in equation (3.4), an ideal phase-correlation surface is expected to have a single peak at the value of 1.



(a) (b)

Figure 3.2: (a) A block selected from frame 2 of the sequence "Mobile and Calendar", and (b) The phase correlation surface taken between this block between frame 2 and its reference at frame 0

Figure 3.2 shows an example of the phase correlation result taken at a block position between frame 0 and frame 2 of the sequence "Mobile and Calendar". The motion under this window is approximately translational, and with the origin at the center of the block, this phase correlation surface shows a peak at position $(0, 1)$. Its magnitude is affected by a number of factors [57], such as the existence of fractional motion in the block, and the non-cyclic nature of image data in the correlation window:

- The effect of fractional motion, also referred to as spectral leakage, is due to the discrete implementation of the Fourier transform. When the displacement is not an integer number of pixels, the phase correlation peak is degenerated into its surroundings. Instead of observing one single maximum, the surface may contain a wide lobe.
- Multiple motions in the block under consideration results in multiple peaks in the correlation surface. It has been argued that the positions of such peaks corresponds to the individual motions, and their magnitudes reflect their significance in the block. In practice, it is often found that multiple motions increase the ambiguity in deciding the dominant motion from this surface, especially when the difference between these motions is small. Under such circumstances, the peak may spread into a wider lobe, making the motions even more difficult to identify.
- The effect of non-cyclic motion: an assumption of the Fourier analysis of phase correlation is cyclic motion, meaning pixels moving out of the block at one side reappears at the other side. Since this is rarely the case in practice, the non-cyclic parts being correlated also reduces the magnitude of maxima on the phase-correlation surface.

3.3.2 Detecting the presence of multiple motions by phase correlation

A unique and strong maximum on the correlation surface supports the assumption that the blocks being correlated are only shifted by a single translational motion, while multiple motions would degrade the values of such maxima. In practice, however, such distinctions are much less straightforward on an actual video sequence, as a result of the following:

- Most real-life motions, even when closely approximated by a translational model, are fractional pixel in nature. A single translational motion of non-integer value reduces the magnitude of the peak.
- When two motions only differ slightly, say by an order of one or two pixels, the deterioration of the maxima takes place in the same neighbourhood with the peaks associated with the two motions. It is often difficult, if not impossible, to identify different motions from this observation.
- More complex motions do not respond in the same way as translational motions under phase-correlation.

If the peak of the phase correlation is relied on to determine the multiplicity of motion in the scene, an ambiguous interpretation of a less-than-perfect peak might result. The reduced magnitude of the peak may correspond to either a single-but-fractional motion, or a group of multiple-but-slightly-different motions, an example of which is illustrated in Figure 3.3. In this illustration, as in Figure 3.1, block A is a boundary block and contains two distinctive motions, while block B is boundary free. The phase correlation results, as seen in Figured 3.3-b and 3.3-c, however, are quite similar and therefore discrimination between them is difficult.

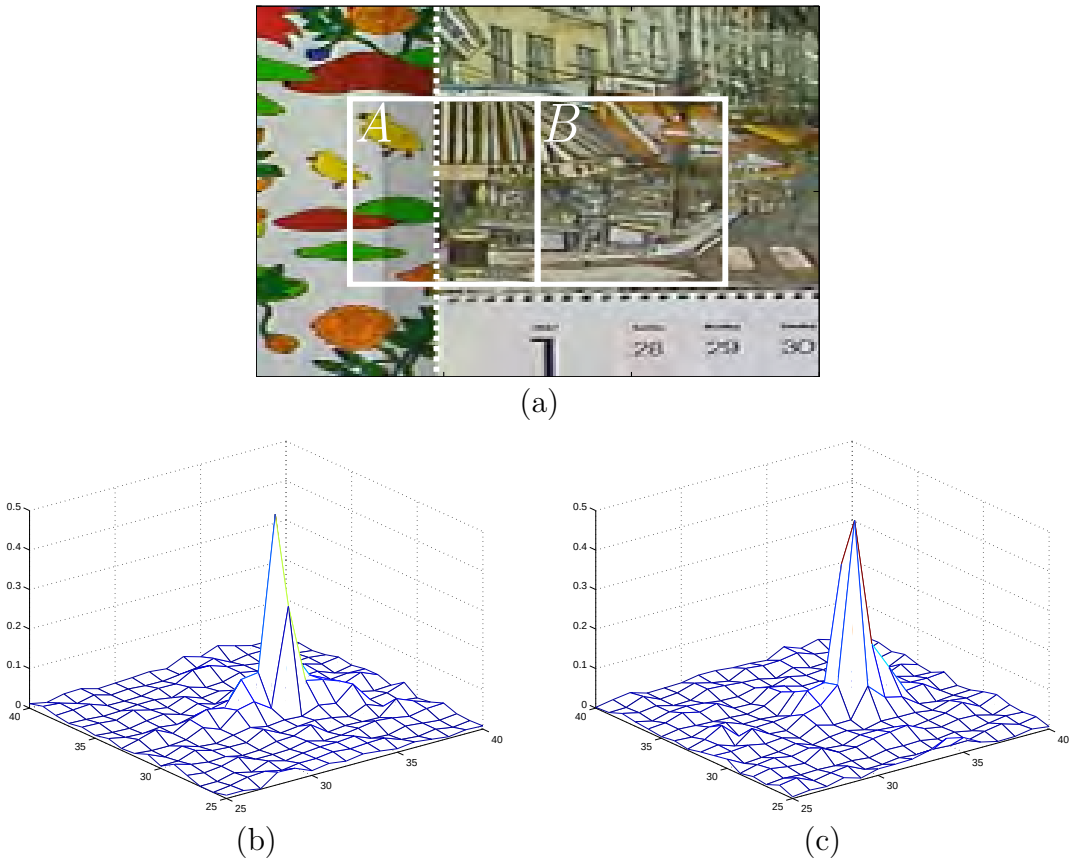


Figure 3.3: (a) Selection of blocks A and B; (b) The phase-correlation surface obtained on the boundary block A, and (c) The phase-correlation surface obtained on the boundary-free block B

3.4 Phase-matched difference and its characteristics

3.4.1 Motivation

One approach to verify the uniqueness of the motion inside one block is to see how well the estimated motion matches the current block to the reference block. As it was shown in the examples of Figures 3.1 and 3.3, one obstacle to such an approach is the inconclusive reply from the matching difference.

Aside from the fact that there might sometimes exist an ambiguous interpretation

of 3-D movements from a 2-D display, one problem affecting a direct correlation, such as in a motion-compensated difference or a phase-correlation method, is spatial undersampling, or aliasing, of video data. Under Fourier analysis, a translational motion is assumed to induce only a phase shift while the magnitude components remain unchanged, which theoretically allows one block to be reconstructed from the other by adding the appropriate shift into its phase component. However, if the matching blocks at two successive video frames are aliased and related by a sub-pixel shift, the correlation mismatch will increase due to the masqueraded high-frequency details. An attempt to reduce the matching difference by altering the phase shift components of one block is also often plagued by imprecise phase-wrappings, and non-linearities of the phase shift across the block.

In creating the phase-matched difference image for use in detection of moving object boundaries, it is proposed to bypass such problems by simulating the phase matching by *reusing the phase of one block on the other*, before subtracting one matching block from the other. For two matching blocks, the phase-matched difference is initiated in the frequency domain, with its magnitude being the difference between magnitude components of corresponding transforms, while its phase is assumed to be fully matched to the phase of the reference block. The phase-matched difference is realized in the spatial domain by an inverse transform of this product.

3.4.2 Creating a phase-matched difference

A block-based approach is implemented to calculate the phase-matched difference, which is composed of the local phase information and the difference in the magnitudes between the current and the reference frame. The block diagram in Figure 3.4 illustrates this process, which is outlined in the following section.

For each block in the current frame, phase correlation is performed using the current and reference frames to estimate the relative shift between the two blocks. Assuming that the motion is translational, the location of the peak in the phase-correlation surface is an approximation for the shift, denoted as (Δ_x, Δ_y). This estimation is

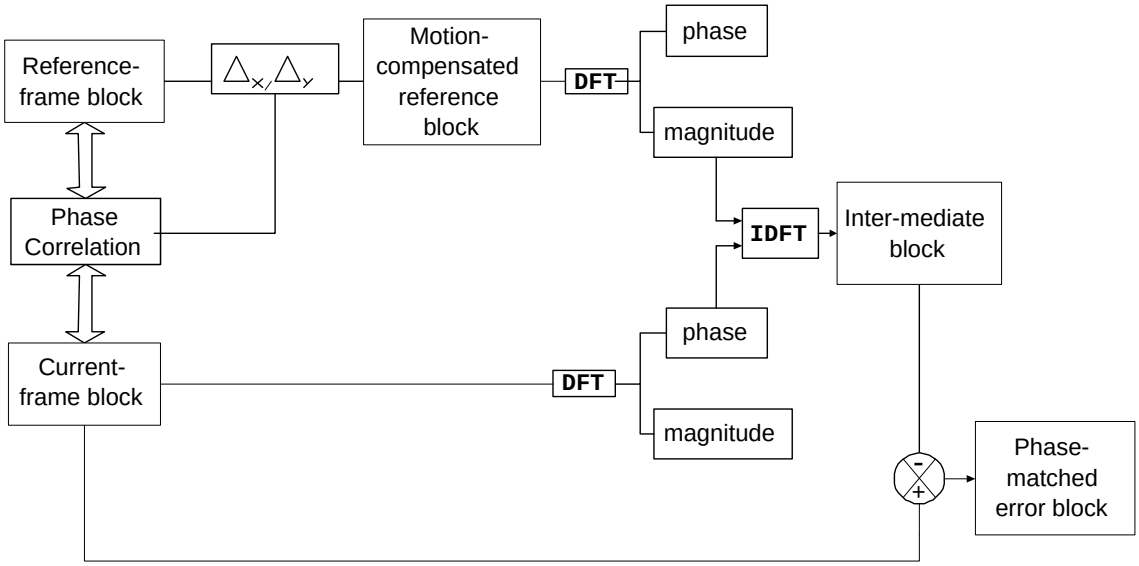


Figure 3.4: Calculating the phase-matched difference

then used to locate the matching block, i_{k-1}, at the reference frame for the current block, i_k. The block index (x, y) is omitted in this notation for simplicity.

The next step involves creating an intermediate block, denoted as i_{k_X}. In the transform domain, this intermediate block shares the same phase component with the current block, while its magnitude is inherited from the matching reference block. This intermediate block therefore can be written in terms of i_{k-1} and i_k as follows:

$$i_{k_X} = \mathcal{F}^{-1}(|\mathcal{F}(i_{k-1})|.e^{-j\theta_k}) \tag{3.7}$$

or

$$\mathcal{F}(i_{k_X}) = |\mathcal{F}(i_{k-1})|.e^{-j\theta_k} \tag{3.8}$$

where $\theta_k = \angle\mathcal{F}(i_k)$ is the phase angle in the Fourier transform of the matching reference block.

As the intermediate block now has its phase fully matched to the current block, the phase-matched difference is obtained by a subtraction in the following form:

$$E_{pm} = i_k - i_{k_X} \tag{3.9}$$

This calculation can be carried out independently for each block from the current frame. Two properties of the phase-matched difference are:

- It has the same phase component as the current block

$$\angle\mathcal{F}(E_{pm}) = \angle\mathcal{F}(i_k) \tag{3.10}$$

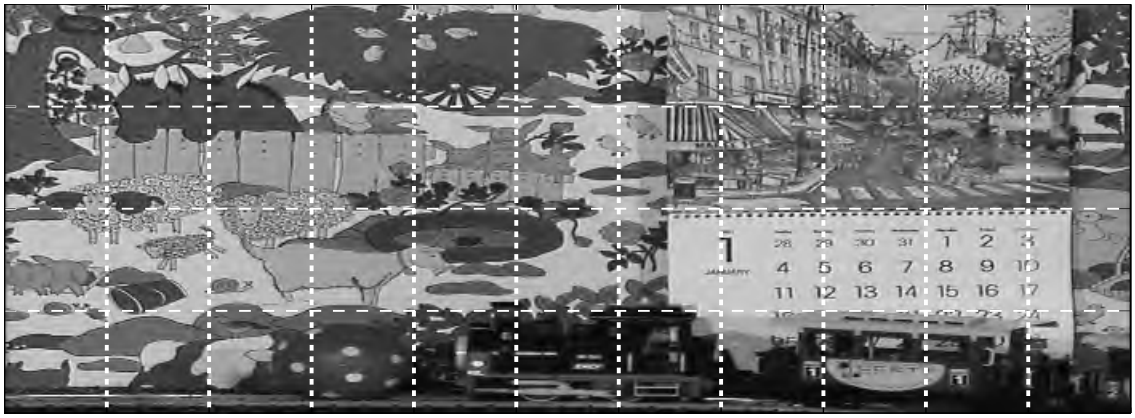
- Its magnitude is equal to the difference in the magnitudes of the current and reference blocks

$$|\mathcal{F}(E_{pm})| = |\mathcal{F}(i_k)| - |\mathcal{F}(i_{k-1})| \tag{3.11}$$

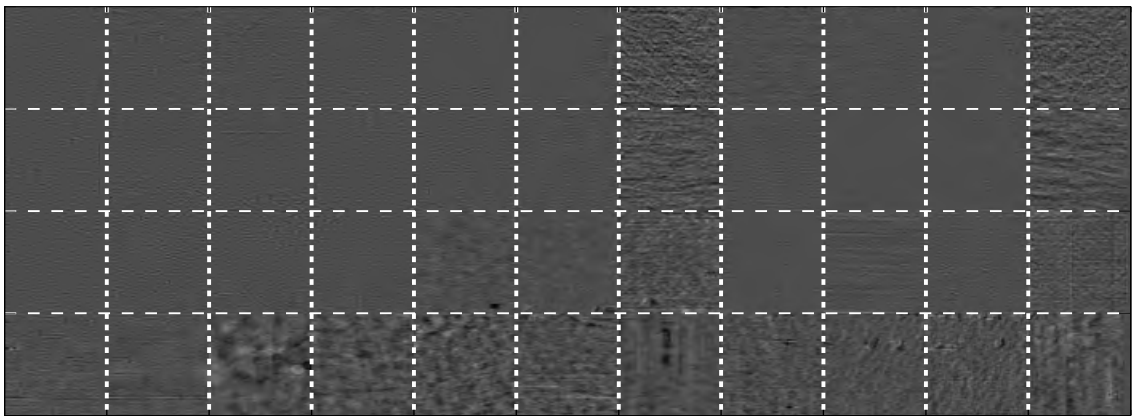
Both properties are self-evident from the derivation of the phase-matched difference. The significance of the method is in realizing the potential of the second property (3.11). It is often assumed that the Fourier transform of a shifted image only differs in the phase when compared to the original, while the magnitudes are taken to remain unchanged. If this is true then the implication of equation (3.11) would be trivial in the case of a translational motion. In practice, there is a difference between the magnitudes due to the effects from unmatched portions under the correlation windows (since most motions are fractional) and sampling noise. This difference, as shown later in the chapter, can be characterised as a function of the residual motion.

An example of the phase-matched difference over a video frame is shown in Figure 3.5, together with the corresponding motion-compensated difference for a visual comparison. The difference images are calculated for the sequence "Mobile and Calendar" using frame 2 as the current frame and frame 0 as the reference frame. The difference as shown between Figure 3.5-b and Figure 3.5-c is clearly visible. Most noticeably, the phase-matched difference does not display the residual signals usually associated with moving edges and object boundaries, which are clearly visible in the other motion-compensated difference image.

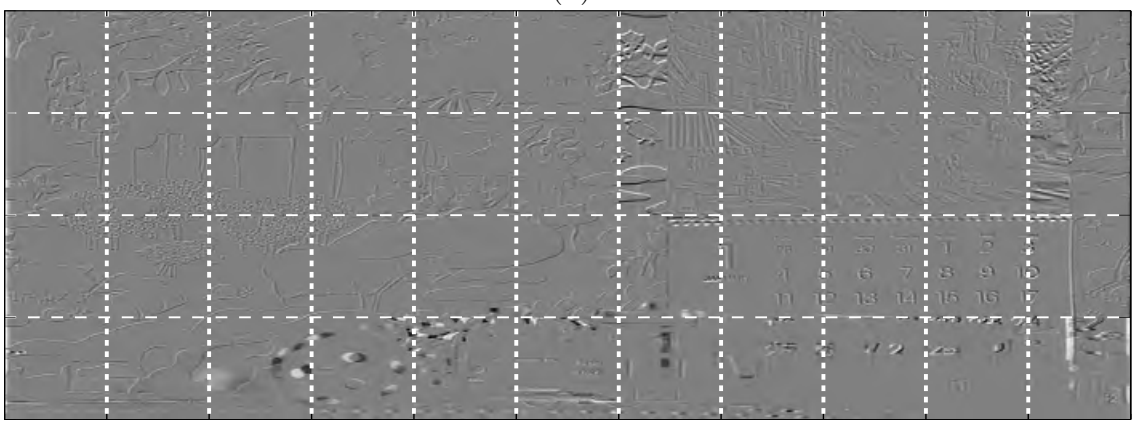
Previously, the phase component of a static image has been shown as carrying much of its visual content [63]. With reference to the proposed phase-matched difference image, observations show relatively flat regions in the boundary-free blocks in Figure 3.5-b. This suggests the effectiveness of *matching* the phase components in



(a)



(b)



(c)

Figure 3.5: (a) The odd field from frame 2 of "Mobile and Calendar", (b) the phase-matched difference image, and (c) the motion-compensated difference image. The grid lines indicate the blocks on which the differences are calculated.

creating the difference image. As it is often assumed that the magnitudes in Fourier transforms remain unchanged under a linear shift, using the difference between the

magnitudes in the phase-matched difference image at such location has the effect of suppressing those visual features that would otherwise be visible from a phase-only image. This also translates into a reduction in the energy level (or sum-of-squared differences) in each block.

3.4.3 The boundary detection criterion

Naturally, the next question which comes up is how to distinguish the difference due to a single motion, from the difference on a block which contains a moving object boundary. In particular, which properties of this phase-matched difference image can be exploited to make such distinctions?

Before introducing the detection measure, a more detailed version of the phase-matched difference in areas with and without a moving object boundary is presented in Figure 3.6. In this example, three spatially-adjacent blocks are selected such that two outer blocks are free of a moving object boundary. Each of these two blocks is transformed with a single motion translation. The middle block, on the other hand, is positioned right across a moving object boundary, with its pixels on one side corresponding to a translational motion different from the other side. The presence of multiple motions in this block is expected to affect the result after block matching, as one of the two motions would not be properly compensated. The difference images using direct motion-compensation, and the phase-matched method, are shown in Figures 3.6-b and 3.6-c respectively.

Instead of relying on direct quantities such as the sum-of-absolute, or sum-of-squared differences, this work suggests using the distribution of the energy within the phase-matched difference as the evaluation criterion for the boundary detection measure. This measure, calculated from the phase-matched difference E_{pm} and denoted as $\mathcal{R}_L(E_{pm})$, indicates the proportion of energy in the difference image contributed by its low-frequency components, and is formulated as follows:

$$\mathcal{R}_L(E_{pm}) = \frac{\sum (E_{pm}^{lowpass})^2}{\sum E_{pm}^2} \tag{3.12}$$

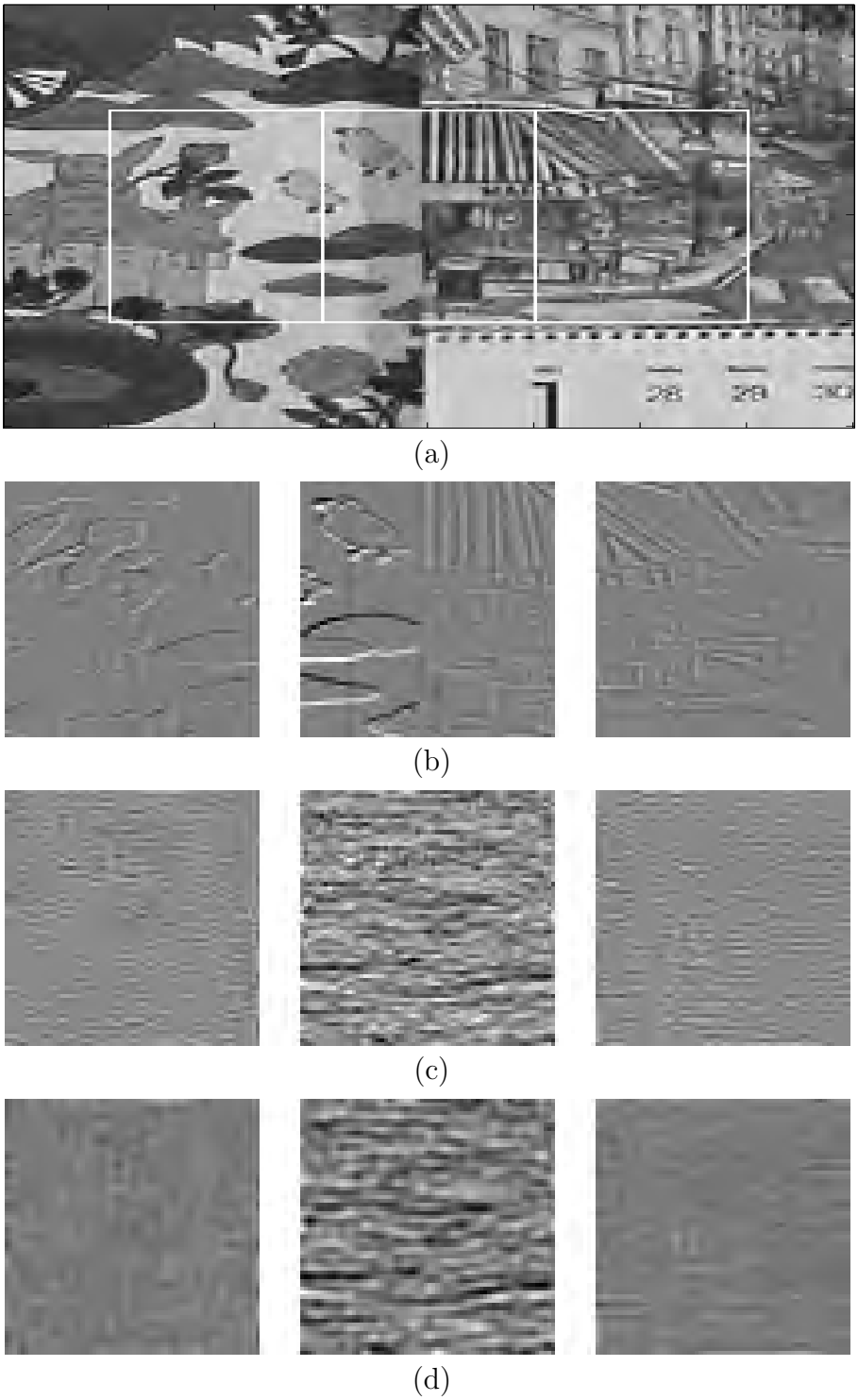


Figure 3.6: (a) Three adjacent blocks in "Mobile and Calendar", (b) Motion-compensated differences, (c) Phase-matched differences, and (d) Low-passed phase-matched differences

where $E_{pm}^{lowpass}$ is the low-pass version of the phase-matched difference.

Assuming that the phase-matched difference E_{pm} is an N-by-N matrix with indices $(x, y) \in \overline{0, (N-1)}$, the denominator of equation (3.12) is calculated directly as its sum-of-squared differences, or:

$$\sum E_{pm}^2 = \sum_{x=0}^{(N-1)(N-1)} \sum_{y=0} E_{pm}(x, y)^2 \tag{3.13}$$

For the numerator, the low-pass energy of the phase-matched difference is obtained using the DCT transform. Let C_{pm} be the matrix corresponding to the DCT transform of the phase-matched difference image, whose coefficients are calculated as follows [17]:

$$C_{pm}(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{(N-1)(N-1)} \sum_{y=0} E_{pm}(x, y) cos\left(\frac{(2x+1)u\pi}{2N}\right) cos\left(\frac{(2y+1)v\pi}{2N}\right) \tag{3.14}$$

where $(u, v) \in \overline{0, (N-1)}$ are the indices of the DCT transform image. While the energy of the difference image and the energy of its transform are the same, the DCT coefficients with smaller indices are associated with the image features of a lower spatial frequency. Therefore from this transform, the low-pass energy associated with the phase-matched difference in the numerator of (3.12) is then calculated as:

$$\sum (E_{pm}^{lowpass})^2 = \sum_{u=0}^{(N_0-1)(N_0-1-u)} \sum_{v=0} C_{pm}^2(u, v) \tag{3.15}$$

where $N_0 \leq N$ is used to define the range of low-frequency DCT coefficients being taken into account, which include $\{C_{pm}(u, v) | (u + v) \leq (N_0 - 1)\}$. In the transform image, these DCT coefficients constitute a triangle at its top-left corner.

There have been a number of previous studies on the distribution of DCT coefficients in images [64–66]. The work in [65], for example, suggested that the Laplacian distribution is applicable to both the DCT coefficients of video images and the inter-frame difference. Furthermore, the parameters for distributions associated with the difference signal were shown empirically to be more symmetrical in the transformed image. However, an analytical model for the distribution of the DCT coefficients

for a generic image remains a difficult problem. In order to decide on a value for N_0 in equation (3.15), the following experiment is setup to evaluate the response of the detection measure $\mathcal{R}_L(E_{pm})$ using a range of values for N_0, with reference to a known moving object boundary.

From Figure 3.6, imagine that instead of observing three discrete, non-overlapping blocks, the block on the left is gradually shifted toward the right, one pixel at a time, until it reaches the position of the right block. This transition is illustrated in Figure 3.7. Initially the block is associated with only one motion of the wallpaper object, on the left side of this figure. Since the boundary between the calendar and the wallpaper is almost vertical, under this transition the coherency of the motion field inside the square window is gradually disrupted as the window slides over this boundary. The proportion of the second object contained in the window is therefore increasing as the window moves further to the right. At some point along this transition, the calendar object then becomes the dominant object in the window. The moving object boundary then disappears from the observing window as the block moves in its entirety into the calendar.

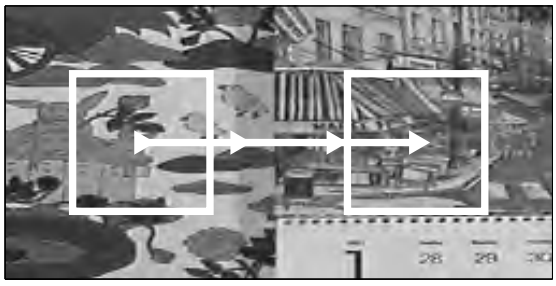


Figure 3.7: Window transition across a moving object boundary

The experiment is carried out for eight different values of $N_0 \leq N$, ranging from $\frac{1}{8}N$ to N. The value of $\mathcal{R}_L(E_{pm})$ is calculated at each window location along this transition, and plotted in Figure 3.8. In addition, the two vertical lines on each graph mark the *true* positions where the moving object boundary starts to appear or disappear from the square window. The line on the left indicates when the shifting window starts to move into the boundary region; the line on the right corresponds to

the last position when the motion boundary is still within the window. The region between these two lines therefore represents the boundary region, while the regions outside these lines are boundary-free. For each value of N_0, the mean values of $\mathcal{R}_L(E_{pm})$ are calculated separately for the boundary region and the boundary-free region. Table 3.1 contains these values, as well as the difference between the mean values from each test.

N_0/N	1/8	2/8	3/8	4/8	5/8	6/8	7/8	8/8
$\overline{R_L^{low}}$	0.0172	0.0428	0.0802	0.1271	0.1918	0.3017	0.4326	0.6520
$\overline{R_L^{high}}$	0.0379	0.1321	0.2411	0.3442	0.4678	0.5990	0.7131	0.8282
$\overline{R_L^{high}} - \overline{R_L^{low}}$	0.0207	0.0893	0.1610	0.2171	0.2760	0.2974	0.2805	0.1761

Table 3.1: Mean values of the detection measure at the boundary and boundary-free regions, at different values of N_0

As the results from table 3.1 show, the value of $N_0/N = 6/8$ provides the best separation (i.e. $\overline{R_L^{high}} - \overline{R_L^{low}}$) between the boundary and the boundary-free regions. Therefore in the experiments performed in the course of this chapter, N_0 is set equal to $\frac{3}{4}N$.

The effect of the low-pass filter operation on the phase-matched differences is illustrated in Figure 3.6-d. The association of the detection measure with energy distribution within the difference image is substantiated by the observation that the phase-matched difference appears to contain more low frequency components in the blocks which are affected by a moving object boundary. A more detailed analysis to support this argument is subsequently developed in section 3.5 in this chapter.

3.4.4 Comparisons to conventional measures

The objective of the proposed boundary detection measure, as stated earlier, is to detect whether a block is subject to only one object motion. The derivation of the phase-matched difference, and subsequently the phase-based boundary detection measure, is based on a motion estimation with integer-pel accuracy. Such an

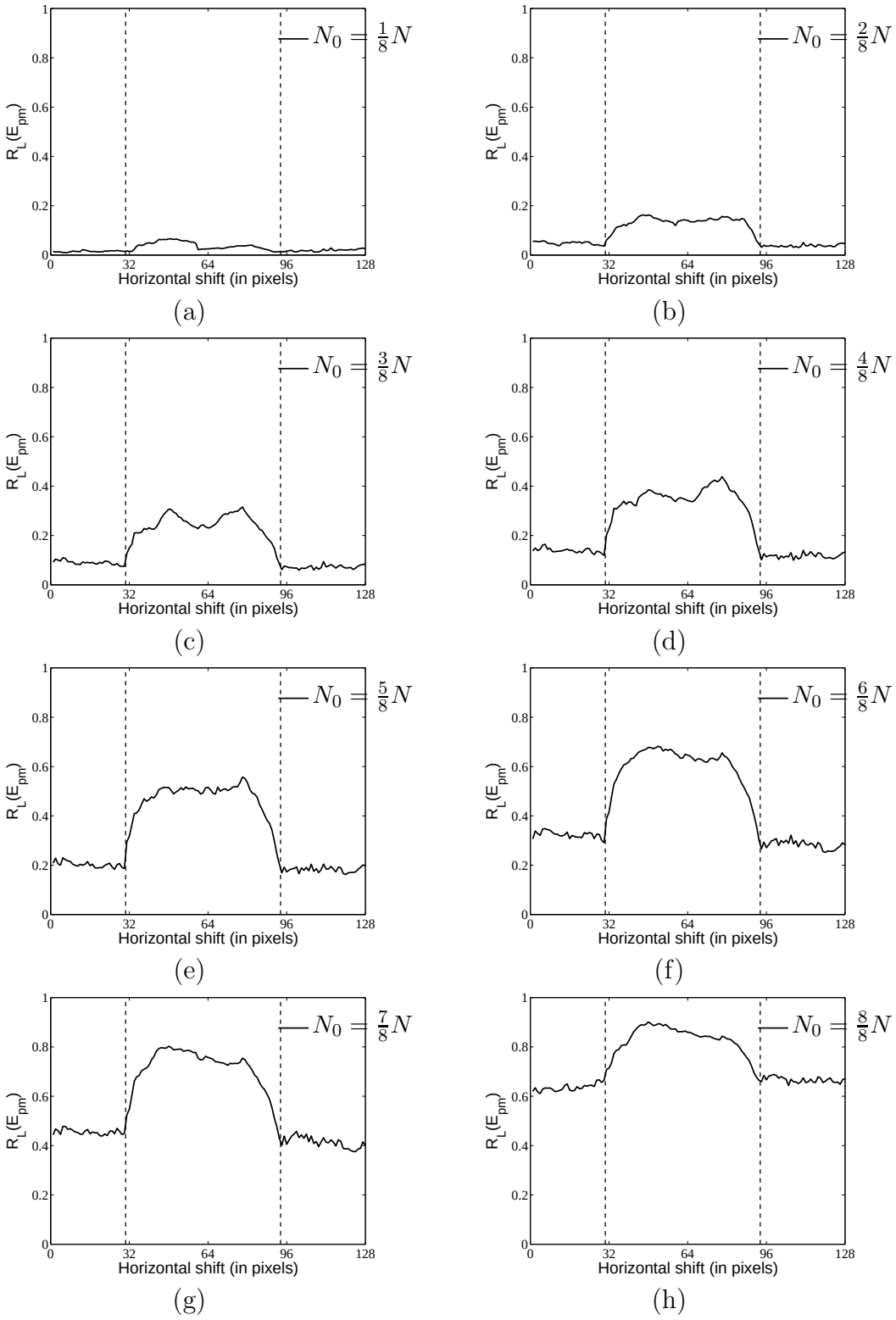


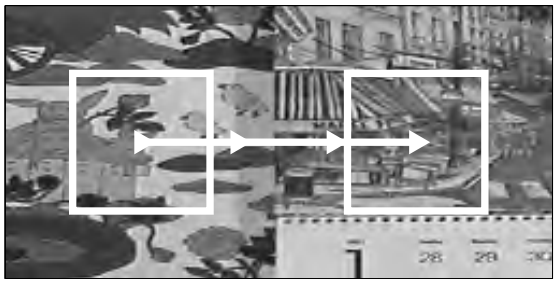
Figure 3.8: Performance of the boundary detection measure using different values of N_0 with block size $N = 64$

estimation is not likely to yield the most accurate motion, for example in the case of a fractional motion. The advantage, however, is that the process does not require interpolation. Once it is decided that the underlying block is only subject to a single motion, a more complicated measure can be implemented to improve the accuracy of the estimation.

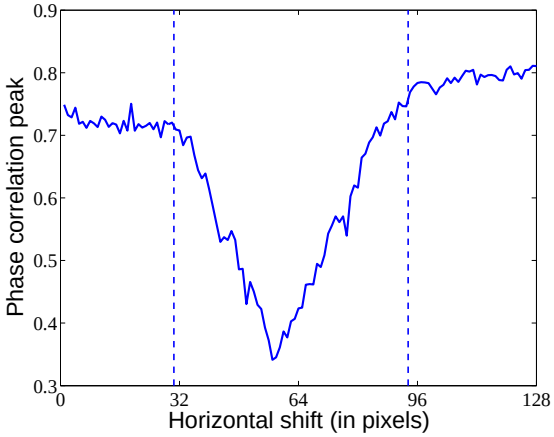
An important criterion to assess the performance of a detection measure is its sensitivity to outliers. In a block containing a moving object boundary, regions which do not follow the motion of the dominant object can be regarded as outliers. It is apparent that the larger the outlier, the easier it is to detect. It is, however, also equally important that the detection does not leave out cases where the outliers are of less significance. A missed detection often allows uniformity to be assumed over a discontinuous motion field, which leads to compromised motion estimation and object boundaries.

To demonstrate (experimentally) the sensitivity of the proposed phase-based detection measure $\mathcal{R}_L(E_{pm})$ in the presence of a non-dominant object motion, and compare its performance against other measures, the windows shown previously in Figure 3.7 are used. The transition of the viewing window achieves the same effect as introducing a second non-dominant motion into an original single-motion region, in increasing proportion. It will be easier to detect the existence of the object boundary, or multiple-motion, when the window is positioned right across the motion boundary, i.e when the areas occupied by the dominant and non-dominant motions are approximately equal. At the same time, it is also desirable to acquire a similar level of detection when the disruption occurs in a much smaller proportion, for example when this viewing window is just over the motion boundary. A sensitive detection measure should be relatively independent of the size and motion of the lesser object(s), in order to reduce the likelihood of a missed detection.

Figure 3.9 demonstrates the almost spontaneous response of the phase-based detection measure to the disruption to coherency of a single-object motion field. It is compared against two other measures, which are the changes in values of the phase-

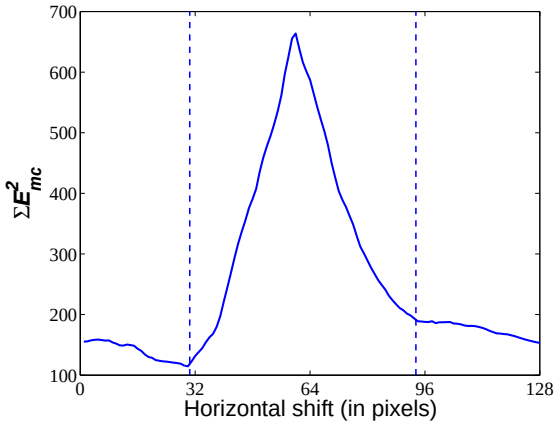


(a) The window transition

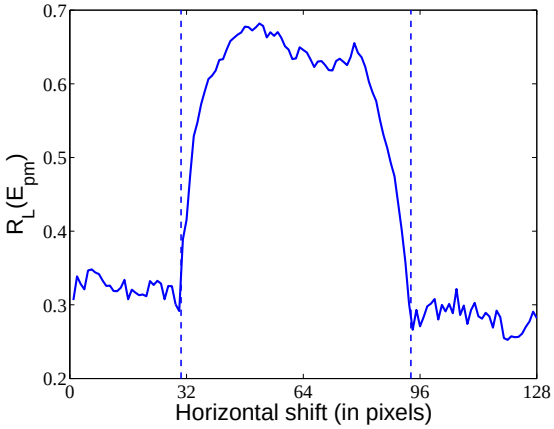


(b) Changes in the phase correlation peak

.



(c) Changes in the sum-of-squared differences $\sum E_{mc}^2$



(d) Changes in the phase-based detection measure $\mathcal{R}_L(E_{pm})$

Figure 3.9: Comparison with other confidence measures under a shifting window. The left vertical line marks the starting position from which the motion field has two distinct motions, and the right vertical line marks the position when this disruption ends. The block size in use is $N = 64$

correlation peak, and the sum-of-squared differences under the shifting windows. On each graph, values of the corresponding measure are plotted at each position as the viewing window moves along.

With an ideal boundary detection, any disruption to the motion coherency ought to trigger a sharp transition in the value of the confidence measure. After all, a desirable output from the detection measure is binary, whether a boundary is detected or not. Such transitions are, however, not observed in the changes of the phase-correlation peak, nor in the motion-compensated sum-of-squared differences.

The performance of the proposed measure is illustrated in Figure 3.9-d. As seen in this graph, the values associated with the phase-based detection measure $\mathcal{R}_L$ are low in the regions of translated motion, because there is only one object whose motion is correctly estimated in these regions. As soon as the viewing window moves into a region containing the motion boundary, a sharp rise is observed in the value of $\mathcal{R}_L$, indicating its sensitivity to the presence of non-dominant motions. Likewise, a rapid decline is also observed under the reverse transition from a region containing multiple motion back to a region with single-motion on the right hand side of the figure.

In contrast with the phase-based detection measure, such sharp transitions are not observed on the other two graphs associated with the phase-correlation peak and the sum-of-squared differences. Figure 3.9-b, for example, shows the variation of the peak value in the phase correlation surface. The value of this peak is at its highest when the entire region inside the window is subject to a single translation motion, as on the left and right side of the graph. When the motion field is not singular, while there still is a peak associated with the dominant motion, other local maxima may also exist in the correlation surface due to other non-dominant motion. Since the total energy under the correlation surface is always equal to unity, the existence of other considerable local maxima reduces the energy under the dominant peak. This can be observed in the gradual decrease of the peak value under the transition from a single to a multiple-motion region.

A similar behaviour is also seen in the sum-of-squared differences in Figure 3.9-c, although the direction of change in its value is reversed in this case. A correct estimation of the dominant motion minimises the residual difference, when it is the only motion associated with the region under the window. When other motions also coexist in the scene, areas under those motions give rise to the residual difference because they do not conform with the estimated motion. The sum-of-squared differences therefore increases when moving from a single-motion region to a region with a motion boundary as is seen in this case.

In observations of both phase correlation and sum-of-squared differences, the changes in their respective values are gradual. More specifically, they can be seen as following a ramp - upward and downward - in both cases. This strongly suggests a *linear* dependency of the value on the amount of outliers (or disruption) to a single-motion field. Since a binary interpretation of this result is often expected, these gradual changes are undesirable as they indicate a poor response under circumstances where the outliers are present in a lesser quantity.

It should also be noted that in formulating this detection measure, motion vectors of integer-accuracy are used. The method does not require any fractional interpolation at this classification stage, hence reducing the effects that interpolation errors might otherwise cause.

3.5 Analogy of a phase-matched difference image

It was shown experimentally from the last section that in a phase-matched difference caused by a single object motion, the low-pass energy is significantly reduced as a proportion of the total energy. In this section, a more detailed analysis into the formulation of the phase-matched difference aims to provide supporting arguments to this claim. Analytical models are developed to characterise the phase-matched difference based on the spectrum of the original image. For simplicity, 1-D signals are assumed in the analytical models; the results, however, can be readily extended

to represent 2-D data as in video.

The following two cases are considered: a video region with a single object motion, and a video region affected by a moving object boundary.

3.5.1 Phase-matched difference due to a single motion

Let i_k represent the intensity level of an image block in the current frame, and i_{k-1} its best-matched block in the reference frame. Assume that the residual shift between two blocks is Δ, we have:

$$i_{k-1}(x) = i_k(x + \Delta)$$

It is assumed that from an initial estimation using phase correlation, the motion has been identified up to a *nearest* integer motion. As a result, by registering the reference toward the current frame by this estimate, the residual motion is reduced to fractions of a pixel, or $|\Delta| \leq \frac{1}{2}$. The right-hand side of this equation can then be re-written using Taylor's series expansion as follows:

$$i_k(x + \Delta) = i_k(x) + \Delta.i'_k(x) + \frac{\Delta^2}{2!}i''_k(x) + \frac{\Delta^3}{3!}i'''_k(x) + higher\_order\_terms \quad (3.16)$$

Since $|\Delta| \leq \frac{1}{2}$, the higher-order terms beyond $\Delta.i'_k(x)$ on the right-hand side of the expansion can be omitted in the approximation, thus simplifying it to:

$$i_k(x + \Delta) \approx i_k(x) + \Delta.i'_k(x) \quad (3.17)$$

Let E_{mc} be the motion-compensated difference between the current block and its best-match in the reference frame. From the above approximation, this difference can be simplified to the gradient of the original block multiplied by the residual motion:

$$\begin{aligned} E_{mc} &= i_k(x) - i_{k-1}(x) \\ &= i_k(x) - i_k(x + \Delta) \\ &= -\Delta.i'_k(x) \end{aligned} \quad (3.18)$$

Let $I_k(w)$ denote the Fourier transform of $i_k(x)$, then the transform of $i'_k(x)$ is $j.w.I_k(w)$. The Fourier transform of the residual image can be expressed as:

$$\begin{aligned}\mathcal{F}(E_{mc}) = \mathcal{F}(i_k(x) - i_{k-1}(x)) &= -\Delta.\mathcal{F}(i'_k(x)) \\ &= -j.\Delta.w.I_k(w) \end{aligned} \tag{3.19}$$

In the transform domain, the presence of the linear term $\Delta.w$ in the right-hand side of this equation is an attenuation factor on the frequency components, in which low-frequency components are suppressed more than high-frequency components [62]. In terms of energy in the difference image, this attenuation translates to a smaller proportion of the low-pass energy in the residual image.

On the other hand, the spectrum of a phase-matched difference image is affected in a different way. Following the approximation in (3.17), the spectrum of the reference block is related to the spectrum of the current block by the following relationship (note that the transform of $i(x)$ is written as $I(w)$ in these derivations):

$$\begin{aligned} I_{k-1}(w) = \mathcal{F}(i_{k-1}(x)) &= \mathcal{F}(i_k(x+\Delta)) \\ &= \mathcal{F}(i_k(x) + \Delta.i'_k(x)) \\ &= \mathcal{F}(i_k(x)) + \mathcal{F}(\Delta.i'_k(x)) \\ &= I_k(w) + \Delta.j.w.I_k(w) \\ &= (1 + \Delta.j.w)I_k(w) \end{aligned} \tag{3.20}$$

Following this equation, the magnitude components of the two transforms are related by:

$$\begin{aligned} |I_{k-1}(w)| &= |I_k(w)|.|1 + \Delta.j.w| \\ &= |I_k(w)|\sqrt{1 + (\Delta.w)^2} \end{aligned} \tag{3.21}$$

From the definition in section 2, the phase-matched difference can be described in terms of the reference and current images as follows:

$$\begin{aligned} E_{pm} &= i_k - \mathcal{F}^{-1}(|I_{k-1}|.e^{j\theta_k}) \\ &= \mathcal{F}^{-1}((|I_k(w)| - |I_{k-1}(w)|).e^{j\theta_k}) \end{aligned} \tag{3.22}$$

where $e^{j\theta_k}$ represents the phase components of I_k. The spectrum, or Fourier transform of this phase-matched difference is easily seen as:

$$\mathcal{F}(E_{pm}) = (|I_k(w)| - |I_{k-1}(w)|).e^{j\theta_k} \tag{3.23}$$

Replacing (3.21) into this latest equation, we have:

$$\begin{aligned}\mathcal{F}(E_{pm}) &= (|I_k(w)| - |I_k(w)|\sqrt{1+(\Delta.w)^2}).e^{j\theta_k} \\ &= (1-\sqrt{1+(\Delta.w)^2}).|I_k(w)|.e^{j\theta_k} \\ &= (1-\sqrt{1+(\Delta.w)^2}).I_k(w)\end{aligned} \tag{3.24}$$

This equation describes the spectrum of the phase-matched difference, which can be seen as the product of the spectrum of the original image and a non-linear term $(1-\sqrt{1+(\Delta.w)^2})$. In comparison, the transfer function associated with the motion-compensated difference is $(-j.\Delta.w)$. The amplitudes of these two transfer functions

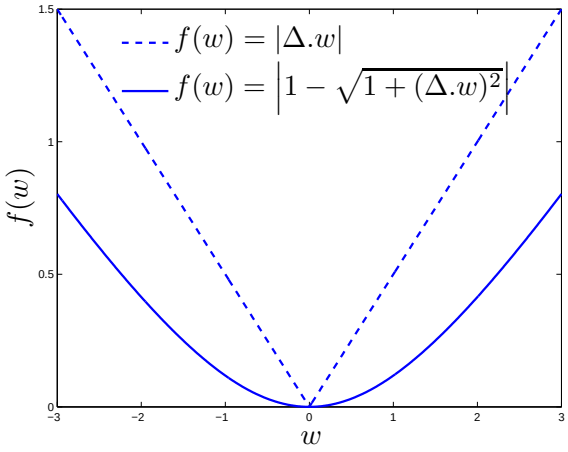


Figure 3.10: Transfer functions for the motion-compensated difference and the phase-matched difference

are plotted in Figure 3.10. From this plot, the following statements can be made about the characteristics of the phase matched difference image E_{pm}:

- Low-frequency components in the spectrum are attenuated more than the high-frequency components.
- The attenuation factor increases non-linearly as the frequency decreases. In particular, near-DC components are suppressed to a much larger degree.

The comparisons between the models for the phase-matched difference and the motion-compensated difference can also be summarised in a few points. First, as a common feature in both images, the low frequency components are attenuated more than the high frequency components, as their transfer functions show in Figure 3.10. Second, the attenuation associated with the phase-matched difference is consistently lower than the attenuation on the motion-compensated difference across the spectrum, which can also be seen from the graph and readily proven according to the inequality:

$$|1 - \sqrt{1 + (\Delta.w)^2}| \leq |\Delta.w| \tag{3.25}$$

Thirdly, the low-frequency attenuation in the phase-matched difference is much stronger than the low-frequency attenuation in the motion-compensated difference, but are approximately the same at high frequency:

$$\lim_{w \to \infty} (|1 - \sqrt{1 + (\Delta.w)^2}| - |\Delta.w|) = 0 \tag{3.26}$$

It appears that in a motion-compensated difference, the low frequencies are also suppressed more than the high frequencies. Naturally, these common characteristics of both types of difference image may raise the question of whether the motion-compensated difference also provides a comparable performance if used as the medium in the detection measure, instead of the phase-matched difference, especially when a detection measure is based on the distribution of energy in the difference image. In other words, does the phase-matched difference offer any improvements that make it a better choice than the motion-compensated residual difference?

To answer this question, the previous experiment with a shifting window is again repeated, this time however with the detection measure being formulated using the motion-compensated difference. The measure is denoted as $\mathcal{R}_L(E_{mc})$ to distinguish it from the proposed phase-based measure which is based on the phase-matched difference, $\mathcal{R}_L(E_{pm})$. Using the same setup as in subsection 3.4.4, the proportion of the low-pass energy in the motion-compensated difference is calculated while shifting

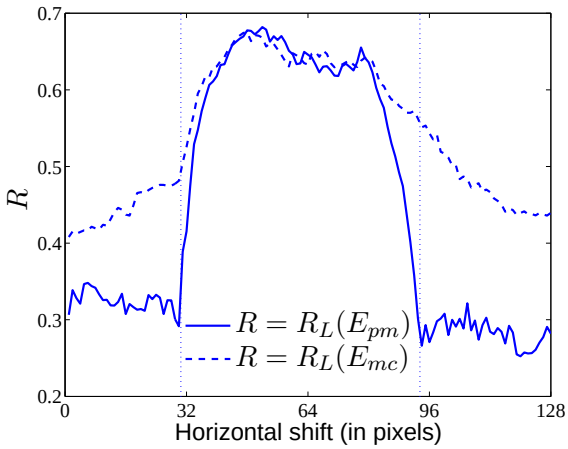


Figure 3.11: Comparison between the proportions of low-pass energy in the motion-compensated difference and the phase-matched difference. The block size in use is $N = 64$

the block across the moving object boundary. Figure 3.11 shows these two detection measures superimposed on the same graph.

The improvements of the measure using the phase-matched difference can be clearly seen from this figure. The more sensitive response associated with the phase-matched difference, as this example shows, turns out to be due primarily to a higher suppression of the low-frequency components. The regions of interest are outside the dotted vertical lines, where the motion fields are homogeneous and the detection measure is supposed to produce a smaller value, indicating a lower proportion of the energy in the low-frequency components. Because of the stronger attenuation at these low-frequency components, $\mathcal{R}_L(E_{pm})$ shows a significantly smaller value in these regions than $\mathcal{R}_L(E_{mc})$.

In the regions between the dotted lines, which contains the motion boundary, both detection measures indicates a higher proportion of low-pass energy in its difference image, however there is much more similarity between the two responses here than in the outside region. It is therefore logical to say that the sharp transitions associated with an occurrence of a moving object boundary are facilitated to a large degree by the stronger suppression at the lower frequencies of the image signal. Setting a

threshold for a binary classification of $\mathcal{R}_L(E_{mc})$ would be much more difficult than doing so with $\mathcal{R}_L(E_{pm})$.

3.5.2 Multiple motions in the frequency domain

The effect of multiple motions in a scene cannot be considered simply as only inducing a phase-shift in its Fourier transform, as in the case of a single translational motion. There have been some recent attempts to characterise the spectrum of images due to multiple motions and non-Fourier motions [67–69]. A common feature of such analysis is the usage of a Heaveside step function [62] to model the discontinuity in the motion field. While this function helps simplify the nature of the occlusion in a mathematical model, the complexity of subsequent expressions obtained in these studies makes them less intuitive for understanding the behaviour of images such as a phase-matched difference. For example, the multiple-motion analysis in [69] even truncated the low-frequency components as they were seen as distortions caused by the occlusion. While using a similar approach to modelling occlusion, this part of the analysis aims to derive a simpler explanation for the dominance of low-pass energy in the phase-matched difference when the region is affected by multiple motions.

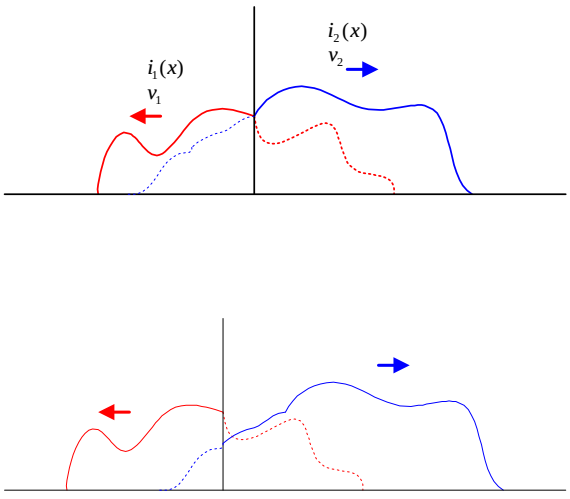


Figure 3.12: Occlusion model

Let $i_1(x)$ represent the occluding object with a corresponding velocity v_1, and $i_2(x)$

and v_2 represent the occluded object. With reference to Figure 3.12, the occlusion can then be formulated as:

$$i(x) = i_1(x - v_1.t).H(x - v_1.t) + i_2(x - v_2.t).(1 - H(x - v_1.t)) \tag{3.27}$$

In this representation, t indicates the time in the temporal dimension, and the Heaviside function $H(x)$ is implemented to model the occlusion at the object boundary. Since the phase-matched difference is created from two consecutive frames i_k and i_{k-1}, their relation can be simplified based on this model to facilitate further analysis. Let $t = 1$, $d_1 = v_1$ and $d_2 = v_2$. The relative expression that can be obtained for i_k and i_{k-1} is:

$$i_k(x) = i_1(x).H(x) + i_2(x).(1 - H(x)) \tag{3.28}$$

$$i_{k-1}(x) = i_1(x - d_1).H(x - d_1) + i_2(x - d_2).(1 - H(x - d_1)) \tag{3.29}$$

Let $f(x)$ be a generic function and $F(w)$ its Fourier transform. From [62], the area under $f(x)$ is equal to its Fourier transform at the origin, or:

$$\begin{aligned} \int_{-\infty}^{+\infty} f(x) &= \int_{-\infty}^{+\infty} f(x).e^{-j.x.w|_{w=0}} \\ &= F(0) \end{aligned} \tag{3.30}$$

and by reciprocal

$$\begin{aligned} \int_{-\infty}^{+\infty} F(w) &= \int_{-\infty}^{+\infty} F(w).e^{j.w.x|_{x=0}} \\ &= f(0) \end{aligned} \tag{3.31}$$

This analogy can be applied on the difference in the spectra of the current and reference images, $I_k(w)$ and $I_{k-1}(w)$, in the following relation:

$$\int_{-\infty}^{+\infty} I_{k-1}(w) - \int_{-\infty}^{+\infty} I_k(w) = (i_{k-1}(x) - i_k(x))|_{x=0} \tag{3.32}$$

In other words, the difference between the areas under two spectrums is equal to the change in the spatial domain at the location of the motion discontinuity (i.e. at $x = 0$). Because this discontinuity is associated with a boundary between two

different objects, significant changes in the pixel values in the occluded/uncovered region can be expected when the two objects are of different color and textures. Most importantly, the change is relatively *independent* of the sizes of the objects involved, because they only depend on the difference between pixel values across this object boundary.

In the frequency domain, most images are predominantly low-pass in nature. Therefore when the area under a spectrum changes, it is logical to expect that most of the change is carried by the low-pass components. Consider the difference between the magnitudes of two spectrums, $|I_{k-1}(w)| - |I_k(w)|$, which is used in creating the phase-matched difference. From (3.32), it is expected that the energy in this difference is also concentrated at low frequencies. It then follows readily that the phase-matched difference is dominated by lower-frequency components, as opposed to the case with no motion discontinuity, where the low-frequency components are strongly attenuated.

The following example further illustrates the assertion that the detection of multiple motion using the proposed measure is relatively independent of the size of a second object. From the sequence "Mobile and Calendar", using frame 2 and frame 0 as the current and reference frame, Figure 3.13-a shows a block chosen so that it contains only the translating background. The phase-matched difference is then calculated for this block. In Figure 3.13-b, by shifting the current block window vertically by 6 pixels downward, the window then also contains a very small portion of the train chimney, which moves with a slightly different motion. Although the size of the second object (i.e. the chimney) is rather insignificant as compared to the window, there is a high contrast at the occlusion boundary between the black chimney and the white background. As reflected in the value of the boundary detection measure, this small disturbance to a single motion field causes a significant change (more than double) in the proportion of low-pass energy in the resulting phase-matched difference image.

From subsection 3.4.4, the improved sensitivity of the proposed detection measure,

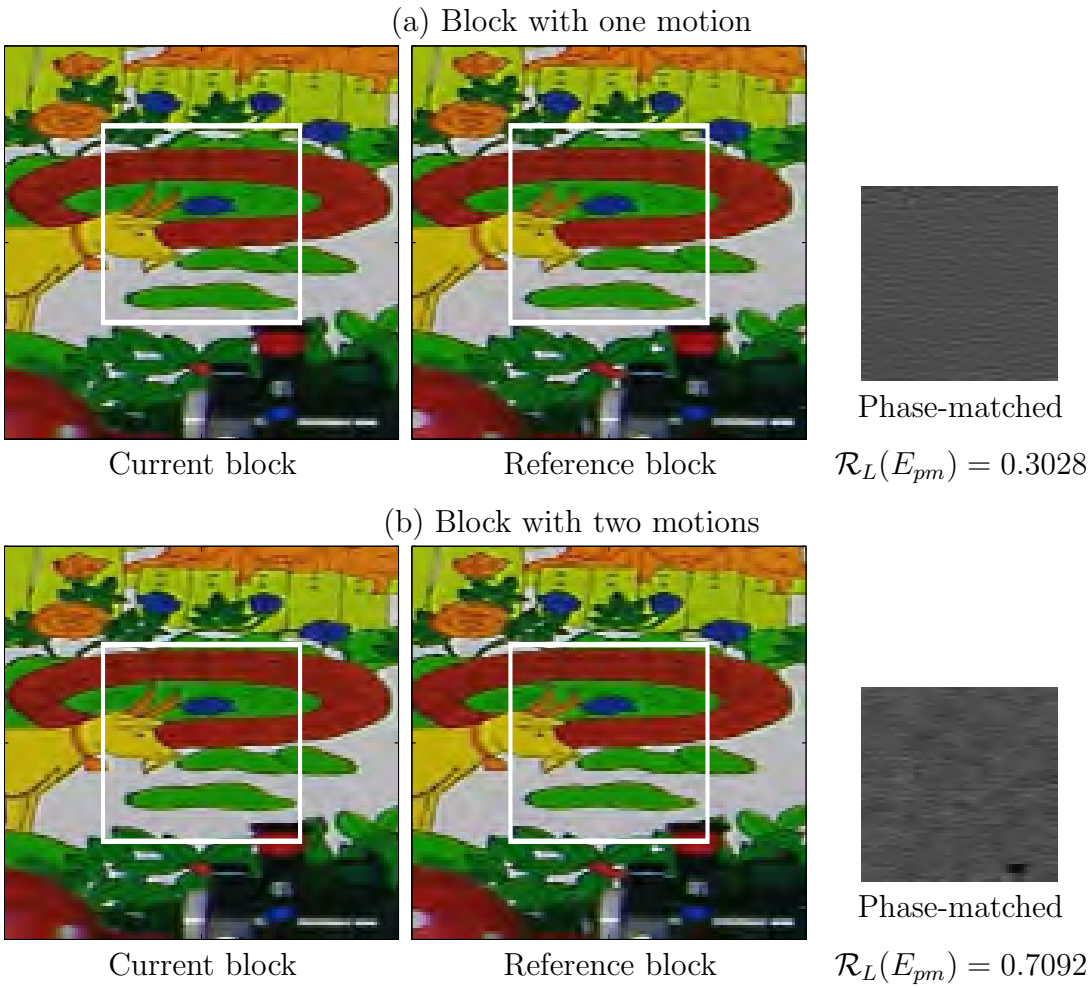


Figure 3.13: Effect of a second motion on a phase-matched difference image

based on the distribution of low-pass energy in a phase-matched difference, was shown experimentally. To sum up, the analysis in the current section gives further insights into the improved performance of the detection measure, which can be attributed to the following factors:

- When the selected block undergoes a single motion, the low-pass frequencies in the phase-matched difference are attenuated by a much higher factor than in the motion-compensated difference.
- When there are two or more motion patterns in the image, the occlusion at the motion discontinuity gives rise to the low-pass components in the phase-matched difference. This increment is strongly affected by the change in the pixel values at the occlusion boundary, and therefore less affected by the size

of the non-dominant object (outliers). As a result, the proposed detection measure shows markedly improved sensitivity even when the outliers appear insignificant if using other detection criteria.

3.6 Phase-matched difference for parametric motions

The term single-motion has been used to indicate a translational motion in this chapter. Jianbo Shi et. al. suggested in [70] that using a translational model produced more reliable and accurate results for tracking when the interframe displacement is small. An affine model, it was said, is more appropriate when interframe displacement is large, such as when comparing distant frames. In various mathematical modelling of multiple motions, component motions assuming a constant, translational trajectory were also adopted for purpose of simplicity [67–69, 71].

The proposed phase-based boundary detection measure aims to classify each image block as either single-motion or multiple-motion, based on the spatial information available in two adjacent frames. For this purpose, a single motion using a translational model suffices for the task. It is however beneficial to further explore if this measure can be adapted to a more general parametric motion. In particular, the implication of image transformation using an affine model is studied in this section.

A translation in the spatial domain can be interpreted as corresponding to a phase shift in the image spectrum in phase correlation, or approximately modelled using a Taylor expansion series as in section 3.5. The effect of an affine motion in the Fourier domain is considerably more complex. Rotation and zoom can be transfered back into a study of translation after a change of axis [72] in the spatial domain: for rotation, it requires changing from Cartesian to polar coordinates, and in the presence of zooming, a conversion of the axis to logarithmic scale is needed. These conversions are however only meaningful if some prior knowledge about the motion is given, i.e. knowing whether the object is rotating or being zoomed in. After such

conversions, motion estimation can be treated as a standard phase-correlation in its converted coordinate system.

For a generic affine motion, Bracewell et al. [73] showed that if an image is subject to an affine transformation in the spatial domain such as

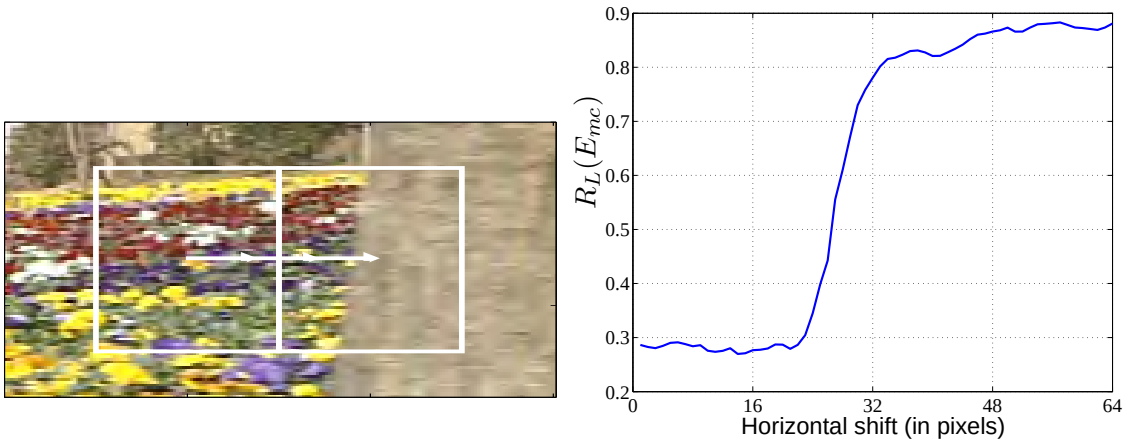
$$g(x) = f(a_{11}x + a_{12}y + a_{13}, a_{21}x + a_{22}y + a_{23}) \tag{3.33}$$

then its Fourier transform is obtained by the following formula:

$$\begin{aligned} G(u,v) = & \frac{1}{|a_{11}a_{22} - a_{12}a_{21}|} e^{\frac{j2\pi}{a_{11}a_{22}-a_{12}a_{21}}[(a_{22}a_{13}-a_{12}a_{23})u+(a_{11}a_{23}-a_{13}a_{21})v]} \\ & F\left(\frac{a_{22}u - a_{21}v}{a_{11}a_{22} - a_{12}a_{21}}, \frac{-a_{12}u + a_{11}v}{a_{11}a_{22} - a_{12}a_{21}}\right) \end{aligned} \tag{3.34}$$

From this expression, both the base function and the phase shift can be seen as undergoing a parametric transformation. To formulate the phase-based boundary detection measure, it is first necessary to calculate a phase-matched difference between the pair of image blocks. This requires the parameters of the affine model to identify the *matching* block. In other words, the two images need to be registered using the dominant motion before the difference can be calculated. Once the model parameters are obtained, a matching image is produced by warping the reference toward the current frame. From the two registered images, a phase-matched difference can then be created in a similar manner to that described in section 3.4.2.

The example in Figure 3.14 shows the response of the phase-based boundary detection measure to a motion discontinuity when the two moving objects are transformed under different affine motions. Two consecutive frames are taken from the sequence "Flower Garden", and in the smaller inset window 3.14-a the tree is seen as occluding the flower bed to its left as the camera is moving. Both motions are not translational, as the flower field is not in parallel but positioned at an angle to the camera plane, and the tree is not strictly a planar surface. It is assumed, however, that each motion can be approximated by an affine model over two consecutive frames. In this case, the affine motion parameters are estimated using the non-linear least-squares method [74].



(a) (b)

Figure 3.14: Performance of the phase-based detection measure in the presence of non-translational motion, (a) shifting of an observation window across two objects with parametric motion, and (b) response of the detection measure

The performance of the detection measure under this setting is shown in Figure 3.14-b. After registering the reference window toward the current window according to the estimated affine parameters, any residual motion between them can then be approximated as translational, and from which the analysis in section 3.5 would also become applicable. The number of DCT coefficients selected for the boundary detection measure is the same as used in the previous example of "Mobile and Calendar" in Figure 3.9-d. It can be seen from table 3.2 that the range of $[\overline{R_L^{low}}, \overline{R_L^{high}}]$ here is inclusive of the corresponding range in table 3.1, for $N_0/N \in [1/8, 6/8]$. In other words, the separation of single and multiple-motion regions is even stronger than in example 3.9-d. This is due to two factors: (a) a much larger difference between motions of the two objects, which also induces a larger occluded area in the multiple-motion region, and (b) the affine model can be seen as providing a more accurate approximation than a translational model with integer-pel accuracy in a single-motion area.

However, it should be pointed out that there is one impediment to an extension to the affine model, besides an increase in computational complexity. When there are multiple object motions in a scene with small differences between them, an estimation

N_0/N	1/8	2/8	3/8	4/8	5/8	6/8	7/8	8/8
$\overline{R_L^{low}}$	0.0024	0.0204	0.0512	0.0993	0.1801	0.2825	0.4575	0.7111
$\overline{R_L^{high}}$	0.1897	0.4325	0.5642	0.6464	0.7176	0.7956	0.8690	0.9291
$\overline{R_L^{high}} - \overline{R_L^{low}}$	0.1873	0.4120	0.5129	0.5471	0.5375	0.5131	0.4115	0.2180

Table 3.2: Mean values of the detection measure at the boundary and boundary-free regions, at different values of N_0, for "Flower Garden"

using a higher-order motion model may be subject to the effect of overfitting. For example, two distinctive translational motion may be ambiguously represented by one affine motion as illustrated previously in Figure 3.1. As a result, a multiple-motion region may be wrongly classified as single-motion. This can be viewed alternatively as an aperture-related problem. Motion parameters are best estimated within their region of support. A multiple-motion field, especially when the deviation between different motions is small, might be ambiguously supported (in a mean-squared error sense) by a single-motion hypothesis. The upfront effort to minimise the residual difference in motion estimation actually hinders the a posteriori detection of multiple motions. In earlier works such as [35], it is suggested that the problem can be avoided by taking frames further apart, hence increasing the difference between the dominant and non-dominant motions. To a certain extent, however, doing so would limit the generality of the segmentation method.

The approach adopted in this work is not to commit to an early classification of single-motion using affine models, to avoid the inherent ambiguity in the motion estimation. Instead, the boundary detection measure is limited to distinguishing regions of single translational motion from the rest of the image. Therefore regions transformed under a non-translational motion may also be initially classified as a boundary region. A process of motion estimation and motion-based region merging will subsequently be used to determine their association with moving objects in the video.

3.7 Implementations

One of the central issues in calculating the phase-matched difference is the window size to use. The primary objective of the boundary detection measure is to separate blocks containing a motion discontinuity, and for the purpose of localising such boundaries, the smaller the block size, the more useful the gathered information. Such blocks provide an *approximation* to the object boundaries, and a smaller block size provides a finer resolution. However, a smaller window is also more likely to be subject to the aperture problem, making the motion estimation result less reliable, as well as the boundary detection. These conflicting requirements are predictably often the case in a "chicken and egg" problem as with many other issues in computer vision, such as the inter-dependency between motion estimation and segmentation.

From a video compression point-of-view, an ambiguous motion estimation may not matter much, as long as the residual difference remains insignificant. For segmentation, mis-labelling of a motion discontinuity as a homogeneous region has a much larger impact. It is a well-known fact that undersegmentation - e.g. a lost object boundary - is difficult to correct in post-processing [41]. For that matter, a wrong single-motion classification of a small window is worse than a correct classification on an encompassing larger window, as the latter conveys useful information while the former is simply misleading.

If a window is thresholded as boundary-free, then the classification result can be passed on to any smaller windows it contains. To facilitate the localisation of a motion discontinuity, it is proposed that the detection measure is applied using overlapping windows across the video frame. Using a block-based framework, each block is assigned a *tag* of either single-motion (boundary-free), or multiple-motion for a boundary block. The block size used in the algorithm is 16-by-16. The classification is however obtained indirectly from larger, overlapping windows using the following steps:

1. Divide each image frame into blocks of 16-by-16
2. Mark all the blocks initially as *multiple-motion*

3. Select a 64-by-64 block on the top-left of the frame and apply the boundary detection criterion using equation (3.12)
4. If single-motion is detected, change the labels of all 16-by-16 sub-blocks to *single-motion*; otherwise leave their tags unchanged
5. Shift the 64-by-64 block to the right by 16 pixels, and iterate the detection/labelling process. When reaching the end of the row, the process is restarted after vertically shifting the first block on the previous scan by 16 pixels. The iteration is carried out until the window reaches the bottom right corner of the frame.
6. Restart from step 3 using a block size of 32-by-32, however this time omit those windows whose four 16-by-16 sub-blocks have already been classified as *single-motion*.

In other words, the algorithm can be described as a process of sweeping a window across the frame in a fixed grid, and the boundary detection criterion is assessed at each node. The order of this scanning is illustrated in Figure 3.15. The result of the classification process contains two groups. The first group containing those blocks which are labelled as single-motion, which are expected to be translated by a linear shift. The second group is made up of multiple-motion blocks, because they either contain a moving object boundary, or do not conform to a single translational motion. The first group of blocks is then further segregated into clusters of the same motion. A comparison of adjacent motion vectors is straightforward at this stage because the estimation results are of integer resolution. This group can be readily allocated into a number of core objects based on the estimated motions.

Recall that this classification serves as a prerequisite for object segmentation, which ultimately requires the integration of the two groups. In other words, the real boundary between a single-motion region and a multiple-motion region does not typically rest on the grid imposed by the block-based classification. A multiple-motion label is implying that a spatial segmentation is required on that block in the next stage of segmentation. Therefore an aggressive option would be to "grow" a single-motion region to the nearest spatial edge in its neighbouring multiple-motion region, while a

more conservative approach would be to "shrink" such regions. Considering the often irreversible effect of incorrectly labelling multiple motions as a single region, the conservative approach is adopted here. As a result, all the blocks at the perimeter of the single-motion clusters are included in the spatial segmentation. A 4-connectivity criterion is used to identify these perimeter blocks. Reiterating, oversegmentation is preferred to undersegmentation in this initial boundary detection and motion classification.

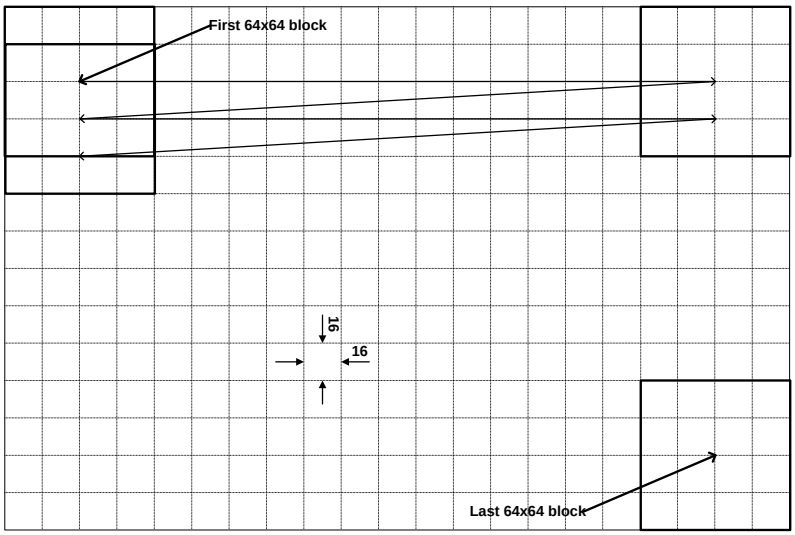


Figure 3.15: Scanning order to calculate the motion confidence measure at each 64-by-64 block. The process is then repeated at the block size of 32-by-32

3.8 Experiments

The experiments are carried out on a number of adjacent frames for three sequences: "Mobile and Calendar", "Flower Garden" and "Table Tennis". The objective is to single out the regions at each frame which can be certified as boundary-free areas. Preprocessing on the video data includes the followings:

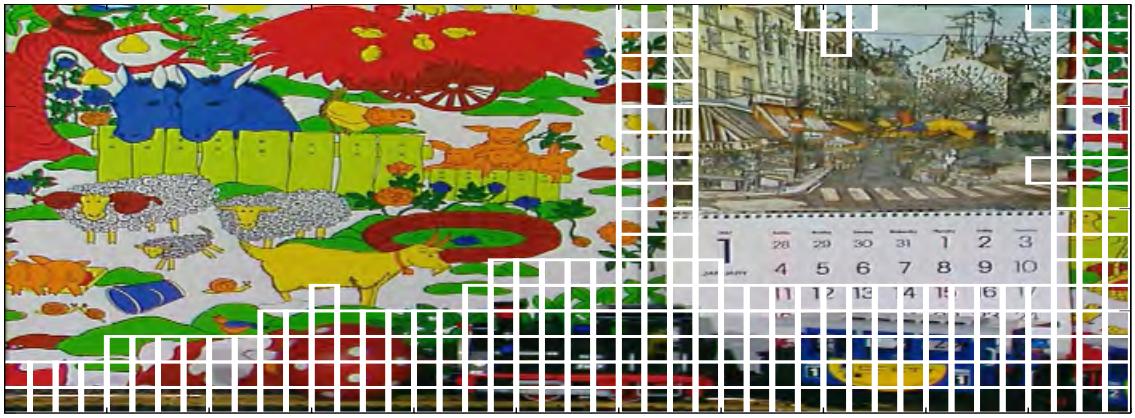
- Deinterlacing: Because these are interlaced sequences, every second horizontal row of pixels are removed from each frame, so that only the odd field remains to be used for calculation of the phase-matched difference.

- Removing the sync-pulse: The first and last two pixels of every horizontal row are also removed from each frame, as they are often affected by the synchronisation pulse. This interference, if not removed, usually leads to erroneous classification of the first and last blocks on each scanning line because they often represent a high level of noises.
- Cropping: All the video frames are cropped so that their (deinterlaced) vertical and horizontal dimensions are the next highest multiple of 16.

The samples selected for the experiment represent a number of different motions, which are due to both object motions and camera movements. On each frame, the boundary is plotted around each block (size 16x16) where the motion singularity is not yet confirmed under the detection measure.

Figure 3.16 shows the classification result of the boundary detection on three frames from the sequence "Mobile and Calendar". While all the objects in the sequence are moving, either under their own movements or due to the camera panning, the motion of the wallpaper and the calendar can be closely approximated as translations. The classification results show that most parts of these two objects are labelled as single-motion, whereas the areas surrounding their boundary are marked as multiple-motion. For the other two objects, the rolling ball and the train, their motions are not well-approximated by a translation as both of them move along the curvature of the track, as well there are parts of the background which are visible through the train windows. The areas containing these objects are therefore also initially classified as multiple-motion. Note that in Figure 3.16-a, some blocks at the top of the calendar are also marked as multiple motion, which is due to the fact that a very small part of the wallpaper is also visible at these locations, and even this small disruption to the dominant motion is detected.

Figure 3.17 shows the results from the sequence "Table Tennis". In these scenes the background is relatively stationary, while the moving objects are the pingpong ball and the player's hand. In this example, the single-motion areas detected are the background part of the frames, while the areas containing the moving objects



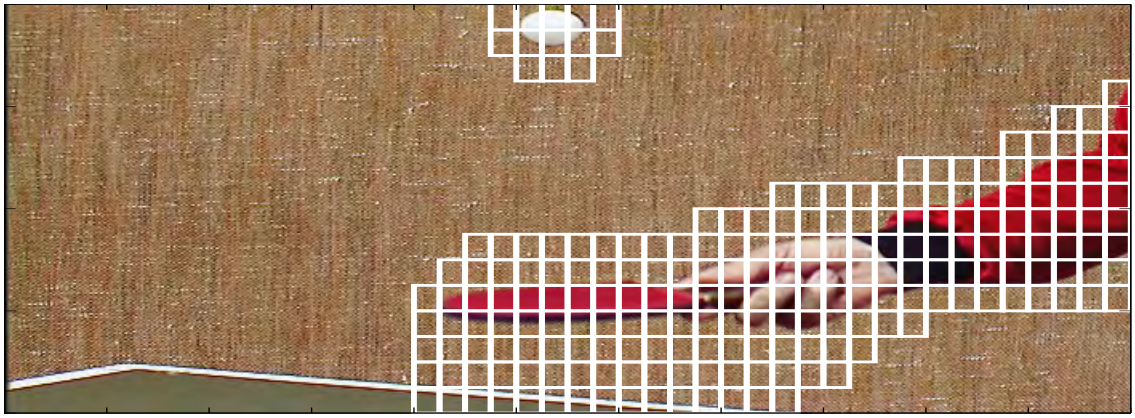
(a) Frame 1

(b) Frame 11

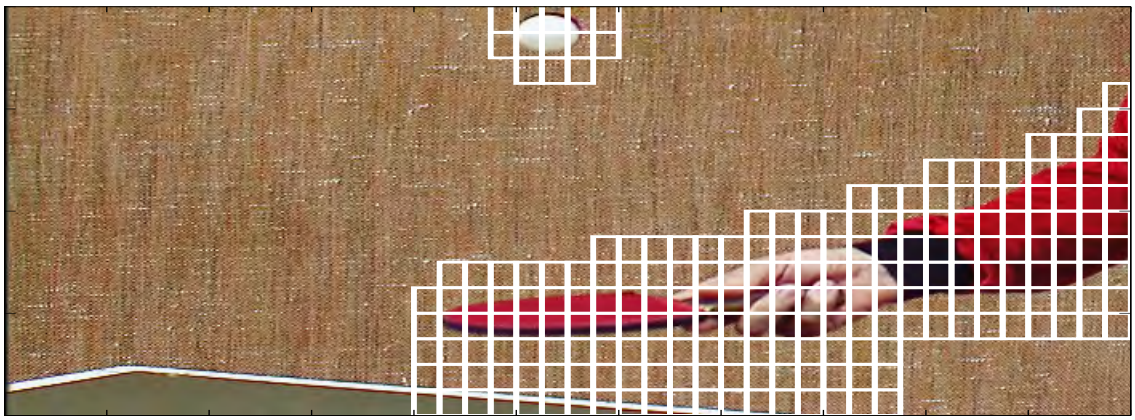
(c) Frame 30

Figure 3.16: Classification results on the sequence "Mobile and Calendar"

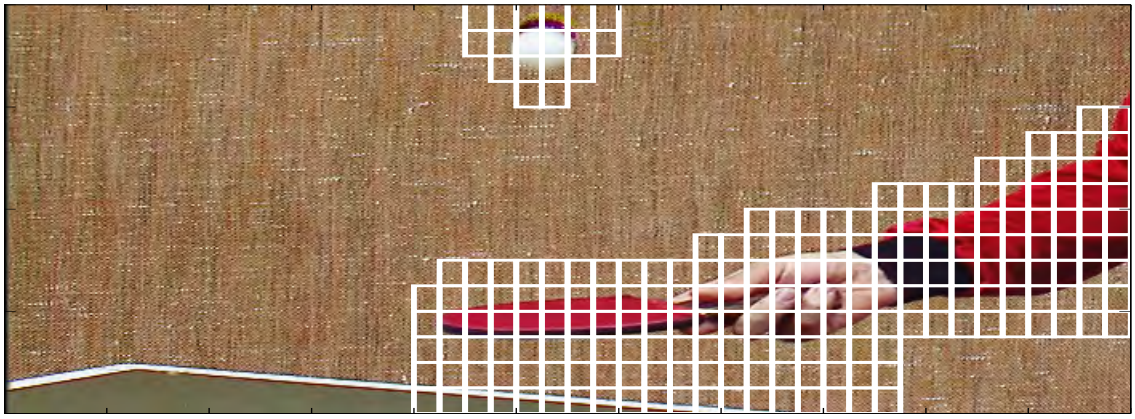
are classified as multiple-motion. Note that the motion of the player's hand is not strictly translational, and there is also little variation in illumination in its upper region, therefore affecting the accuracy of the initial block-based motion estimation there. For this reason, the entire area containing the moving hand is marked as



(a) Frame 1



(b) Frame 2

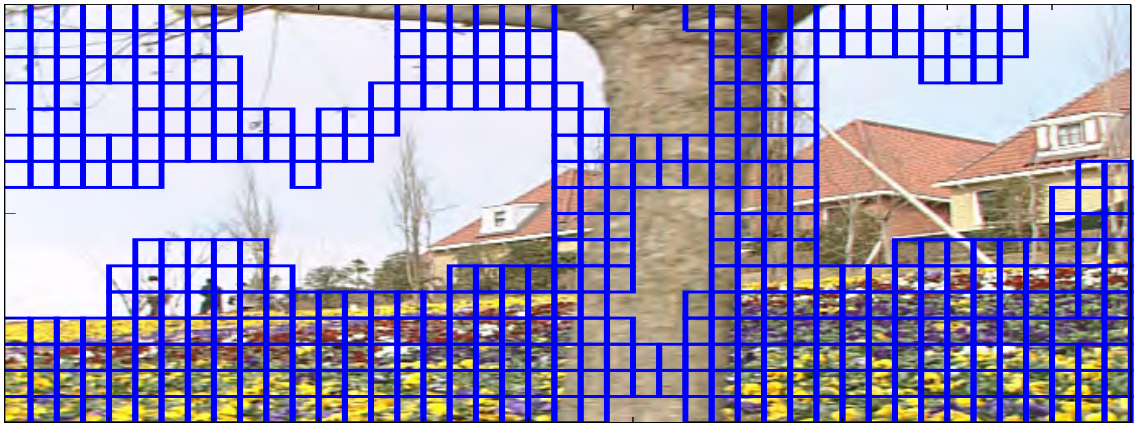


(c) Frame 3

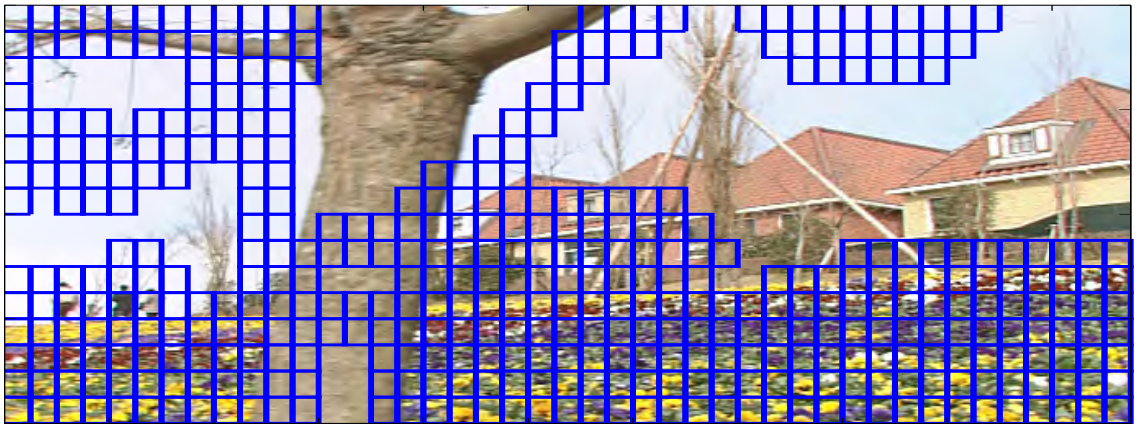
Figure 3.17: Classification results on the sequence "Table Tennis"

multiple-motion.

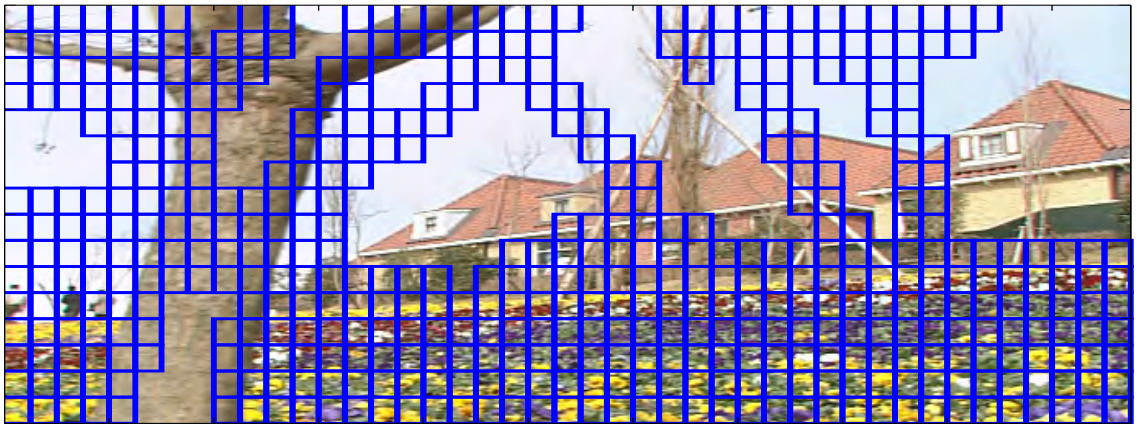
In the third sequence "Flower Garden", most of the object motions are generated by a moving camera over different object geometries, with an exception of the people walking at the left hand side of the flower garden. Few of these motions are well-



(a) Frame 1



(b) Frame 20



(c) Frame 30

Figure 3.18: Classification results on the sequence “Flower Garden”

approximated by a translation, for example on the flower garden, areas closer to the camera are seen as moving at a faster rate than an area located further away. In the result shown in Figure 3.18, parts of the these video frames which come out

consistently under a single-motion class is the front of the houses, since they are positioned the furthest from the camera and their motion can be approximated as translational, although disruptions are still occasionally observed due to the presence of foreground objects.

As it was seen in the examples, the detection measure is shown to be capable of distinguishing regions which move under a single-motion *and* are free of moving object boundaries. The constraint is that the single-motion must be translational, a condition which is necessary to make the initial motion estimation unambiguous. In the first example, it is shown that the algorithm can accurately identify the regions surrounding the boundary of two translating objects, even though the difference between their motions is small. The second example shows that it can separate a large part of the background from a foreground moving object. The third example shows that in a scene with different object motions, the algorithm can identify the object regions which follow the single translational motion assumption.

3.9 Summary

This chapter has proposed an approach to identify and distinguish areas in a video which do not contain a moving object boundary and follow a unique translational motion. The strength of this phase-based method is the improved sensitivity in detection of non-dominant object motion under the estimation window, which has been demonstrated both experimentally and analytically. The constraint on the method is the prerequisite that an unambiguous estimation is obtained for the object motion, and for this reason the initial dominant object motion is assumed to be translational. In summary, the key findings of this chapter in relation to the phase-matched difference image are as follows:

- When there is no moving object boundary inside a block, and the estimated motion is within the sub-pixel neighbourhood of the true motion, the phase-matching operation results in a difference image not only with less energy

than a conventional motion-compensated difference, but also characterised by a strong attenuation imposed on its low-frequency components.

- When there is a moving object boundary inside a block, or the estimated motion does not adequately address the underlying motion, the rise in energy of a phase-matched difference image is accompanied by an even more dramatic rise in the proportion of energy in its low-frequency components, even when the presence of non-dominant motions are relatively small. This sharp change in the proportion of low-pass energy is further emphasised by the stronger suppression of the low-frequency components in those blocks without a moving object boundary.

- The proportion of low-pass energy in the phase-matched difference image can be used as a detection criterion to discriminate boundary blocks from those without a moving object boundary. This discrimination was also shown as being more sensitive to outliers than measures such as sum-of-squared differences on a motion-compensated difference image.

- The phase-matched difference image can be modelled analytically to substantiate the above findings

- The detection measure was shown as extendable to parametric motions in general, provided that an initial estimation of the dominant motion is unambiguous.

When viewed under the context of a larger object segmentation scheme, the output from this single-motion/multiple-motion classification represents a *coarse* segmentation, as practically each single-motion region would belong wholly to one object, whereas a region is labelled as multiple-motion because it either straddles a moving object boundary, or the motion has not been accurately estimated. In other words, the single-motion regions can now be excluded from the *search space* for the object boundaries. The task of locating the object boundary within the remainder of

the frame is the focus for the next chapter, which will be accomplished by using a region-based motion estimation and motion-based region merging strategy.

Chapter 4

Spatial segmentation and motion-based region clustering

4.1 Introduction

While a block-based approach may be appropriate for motion estimation of a region positioned *inside* an object, the boundaries of natural objects rarely coincide with such a grid. Each block straddling a motion boundary between two moving objects is subject to a non-homogeneous motion field, and therefore applying a block-based estimation would produce a misleading result. Because the shape of a moving object is defined by its boundary with other objects, segmentation requires that the locations of such discontinuities in the motion field to be accurately detected and recovered. This chapter proposes a segmentation approach which is carried out in the following four stages:

1. Identification of areas within a frame where multiple object motions exist
2. Decomposition of the multiple-motion area into smaller, single-motion components
3. Estimation of motions for the component segments
4. Use of the motion information to merge these segments and the rest of the single-motion blocks

The first step of this approach is accomplished by the classification of frame blocks as either single-motion and multiple-motion, using the boundary detection method developed in the previous chapter. In this chapter, solutions to the subsequent three segmentation stages are proposed. In particular, the focus is on the spatial-based segmentation within the multiple-motion areas, and the motion-based region merging strategy.

4.2 Spatial segmentation

In a scene containing multiple moving objects, the motion of each individual object sets it apart from others and the background. Therefore, if the motion associated with every pixel is identified and estimated correctly, a segmentation approach can be formulated from just the motion information. It is quite obvious that for such a method to function effectively, the accuracy of the motion must be well maintained during the estimation process. Unfortunately, the inter-dependency between segmentation and motion estimation usually makes it difficult to obtain accurate estimation without prior knowledge of object boundaries. Because object boundaries are generally considered to be a subset of all the spatial edges in a frame, the spatial content of an image provides a valuable source of information to help locate such boundaries.

The effects of interdependency between object segmentation and motion estimation is most pronounced in the vicinity of the object boundary. In a block-based approach, if a selected rectangular block contains regions of different moving objects, a single motion estimate would not produce satisfactory results. Instead, it is necessary to first divide the block into regions corresponding to individual objects. This division is to ensure that each split region is associated with one single motion prior to estimation. Because a segment selected based on the spatial coherency is much less likely to span across a motion discontinuity, a spatial segmentation scheme is implemented for this division.

In previous spatio-temporal video segmentation techniques, spatial segmentation has often been performed as a first step, followed by motion estimation and motion-based region merging [42, 47]. While region-based motion estimation is more computationally intensive than a block-based approach, the most important advantage of the former results from the fact that a well-formed spatial segment is much less likely to contain parts of a different object, or outliers. The coherency of pixel values is seen as a *de facto* support for the single-object assumption within each segment, therefore improving the reliability of motion estimation. Because each segment belongs to one object, it is also more likely that the estimated motions can be used to identify independently-moving objects in a scene.

However, results of a spatial segmentation process are also highly dependent on the content of the scene. Since both the number of objects and their texture are unknown quantities, there is little that can be assumed about the initial segments. In fact, the majority of spatial segmentation would result in either over-segmentation or under-segmentation, affected mainly by settings of the parameters which drive the segmentation process. The former refers to an object being divided into many segments, due to a variation in the color and texture within the object itself, or changes in lighting condition. Under-segmentation, on the other hand, is the formation of spatial segment over an object boundary, which in most parts is caused by a lack of contrast at such boundaries. While the two problems have different implications on object segmentation results, it is generally agreed that over-segmentation can be corrected using motion information, whereas such an option is usually not available for the problem of under-segmentation [41].

It often occurs that the spatial segmentation of a single frame produces a much larger number of segments than the number of moving objects. Even when an object is of a uniform color, effects of noises, changing illumination and reflection may still prevent it from being spatially segmented as one single region. The most useful information that links these segments to the same object is their motion, e.g. if the segments take up the same motion trajectory. In this context, the effect of

over-segmentation can be considered *repairable*, pending a proper motion estimation. On the contrary, due to inhomogeneity of the motion field, motion estimation on an under-segmented region usually produces an ambiguous result. Wrong motion information usually leads to a loss of the object boundary, an effect which is difficult to correct by post-processing.

As happens with many other inter-dependency issues in image processing, the preference for over-segmentation comes with a certain trade-off. While it relies on using motion to re-group neighbouring regions, over-segmentation tends to produce more regions of smaller sizes, making motion estimation less reliable. Secondly, an increased number of regions and candidate motions may introduce instability into the classification process [35].

Effects of these problems can be addressed with the single/multiple motion classification scheme developed in the previous chapter. These trade-offs are most relevant in the context of an object boundary. Undersegmentation is most damaging if it occurs over the boundary of two moving objects, but much less so if it happens *inside* one object. On the other hand, while oversegmentation is a safer option around the object boundary, it is also highly redundant if it takes place inside an object. A more efficient spatial segmentation mechanism can be formulated if it is limited to the region which contains the moving object boundary, instead of the entire frame.

The remainder of this section is organised as follows. Subsection 4.2.1 looks at the relationship between the selective spatial segmentation scheme and boundary detection measure. Subsection 4.2.2 describes the quadtree algorithm to carry out the segmentation, based on the framework of region growing in [17]. Lastly, in subsection 4.2.3, an extension of the quadtree algorithm to a self-expanding segmentation scheme is proposed, based on the local motion information and result from the spatial segmentation process.

4.2.1 Selective segmentation based on boundary detection

A selection of areas for spatial segmentation results directly from classifications under the boundary detection measure. The previous detection scheme indicates which frame areas are more likely to contain a boundary between moving objects.

There is a conceptual resemblance of this segmentation approach to the concept of non-linear diffusion in image processing. When a conventional filtering operation, for example using a mean or Gaussian filter, is applied uniformly on a still image, the overall smoothing effect is observed at all the image features. Within a region of relative constant intensity, such operations help reduce the effect of noise. When applied *across* two regions of different colors, however, it results in a blurring effect, or a reduction of the contrast at the boundary. While smoothing inside a region is acceptable and generally considered as an enhancement to the image quality, smoothing across the region boundary is rather undesirable. A non-linear filter, such as the one proposed by Perona and Malik in [33], is aimed at controlling the effect of diffusion depending on the amount of local image features.

For object segmentation, in a similar way, spatial segmentation should not be necessary in regions which are free of moving object boundaries. Previously, classifications from the boundary detection measure are based on the uniqueness of local motions. In an area where the presence of one motion is not unique, spatial segmentation helps to divide it into individual object components based on the local color property. This partition is helpful for a subsequent motion-based region grouping process. On the other hand, if a block contains no moving object boundary, then its motion can be further refined using the existing block-based framework. The advantages are seen in conjunction with the effectiveness of the single/multiple motion classification:

- The uniqueness of the motion on a region with significant spatial texturedness makes it an ideal target for a reliable motion estimation. Alternatively, further unintended spatial segmentation of such a region may unnecessarily complicate the motion-based region grouping process.
- A reduced number of both object segments and candidate motions from a more

selective spatial segmentation helps simplify the region grouping process, as well as improving its stability.

In the selection of blocks which are included in the spatial segmentation, there is an extension to those which are a neighbour to an existing multiple-motion block. The main reason for this inclusion is to improve the accuracy of motion estimation on segments in the neighbourhood of an object's boundary. When the spatial segmentation is confined to a particular block, the segments are then partially dictated by the block boundary. As the accuracy of motion estimation for a segment depends on the correct boundary formation, it may also be affected by the existence of an unintended block boundary. Consideration given to the immediate neighbouring blocks is meant to reduce such effects and improve the quality of estimated motion.

4.2.2 Quadtree segmentation

As mentioned earlier, the purpose of spatial segmentation is to identify the candidate regions for a subsequent motion-based classification. This process is not expected to produce a final object segmentation. An important issue, however, is that it is performed conservatively so that undersegmentation is avoided. While there are a variety of spatial segmentation techniques, an algorithm which deals with image intensity and color is preferred as it directly maintains the spatial integrity of a region. In this section, based on the framework of region growing in [17], a quadtree-based approach is implemented to segment the selected areas into individual spatial regions of uniform colors.

Let S denote an area to be segmented. Following the region-oriented framework in [17], segmentation is considered to be a partition of S into individual regions $S_1, S_2, ..., S_n$, with n being the number of regions, subject to the following conditions:

1. $\bigcup_{i=1}^{n} S_i = S$
2. $S_i \bigcap S_j = \emptyset, \forall i \neq j$
3. S_i is a connected region, $i = \overline{1, n}$

4. $P(S_i) = TRUE, \forall i = \overline{1,n}$
5. $P(S_i \cup S_j) = FALSE, \forall i \neq j$

where for the last two conditions, $P(.)$ is a logical predicate over the input argument. For example, condition (4) can be used to test if pixels in region S_i are spatially homogeneous, by setting:

$$P(S_i) = \begin{cases} TRUE, & if\,((max(S_i) - min(S_i)) < T_s) \\ FALSE, & otherwise \end{cases} \tag{4.1}$$

and condition (5) is used to assert the separability between different segments.

The first condition requires that the combined set of individual regions is equivalent to the area selected for segmentation, while the second condition states that any two regions should be exclusive. These conditions are often self-evident in the implementation of most segmentation methods. The third condition requires that for any two pixels in a region, there exist a connected path from one pixel to the other. Figure 4.1 illustrates the concepts of four-connectedness and eight-connectedness for a pair of pixels. Throughout this chapter, four-connectedness is assumed when addressing neighbouring segments and/or pixels.

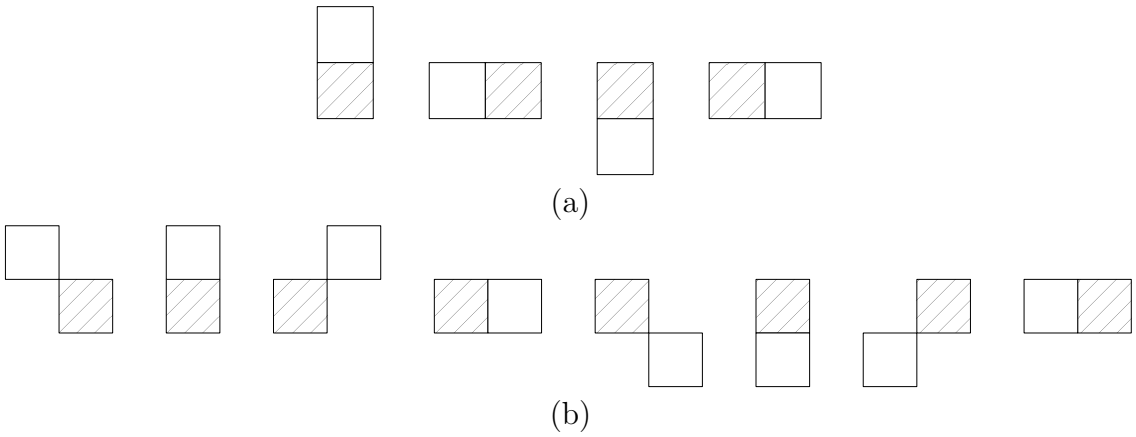


Figure 4.1: Pairs of pixels with (a) four-connectedness, and (b) eight-connectedness

The fourth condition is used to enforce the homogeneity criteria within each spatial region. This criterion is formulated using a constraint on the standard deviation amongst the group of pixels from each region as follows.

Let r be a square block of N-by-N pixels, and the intensity at pixel location (x, y) be r_{xy}, where $1 \leq x \leq N$ and $1 \leq y \leq N$. The mean value of the region is

$$\overline{r} = \frac{\sum_{x=1}^{N}\sum_{y=1}^{N} r_{xy}}{N^2} \tag{4.2}$$

and the variance, or second central moment, within this group of pixels is:

$$\sigma^2(r) = \frac{1}{N^2 - 1}\sum_{x=1}^{N}\sum_{y=1}^{N}(r_{xy} - \overline{r})^2 \tag{4.3}$$

where $\sigma(r)$ is the standard deviation. The value of $\sigma(r)$ is an indicator of the variation of pixels within the block. When $\sigma(r) = 0$, the block is made up of identical pixels; the larger the value of $\sigma(r)$, the less likely it is that pixels within the block are of a similar value. For a segment S_i, this constraint can be imposed by setting a threshold on the standard deviation $\sigma(S_i)$, as follows:

$$P(S_i) = \begin{cases} TRUE, & \text{if } (\sigma(S_i) < T_\sigma) \\ FALSE, & \text{otherwise} \end{cases} \tag{4.4}$$

As the classification of multiple/single motion is made at the block size of 16-by-16 pixels, the quadtree is performed using this block size as the top node. There are 5 levels in the quadtree from the top node to the pixel level. This step is referred to as splitting, as each block is split into individual nodes to satisfy $P(S_i) = TRUE$ in equation (4.4).

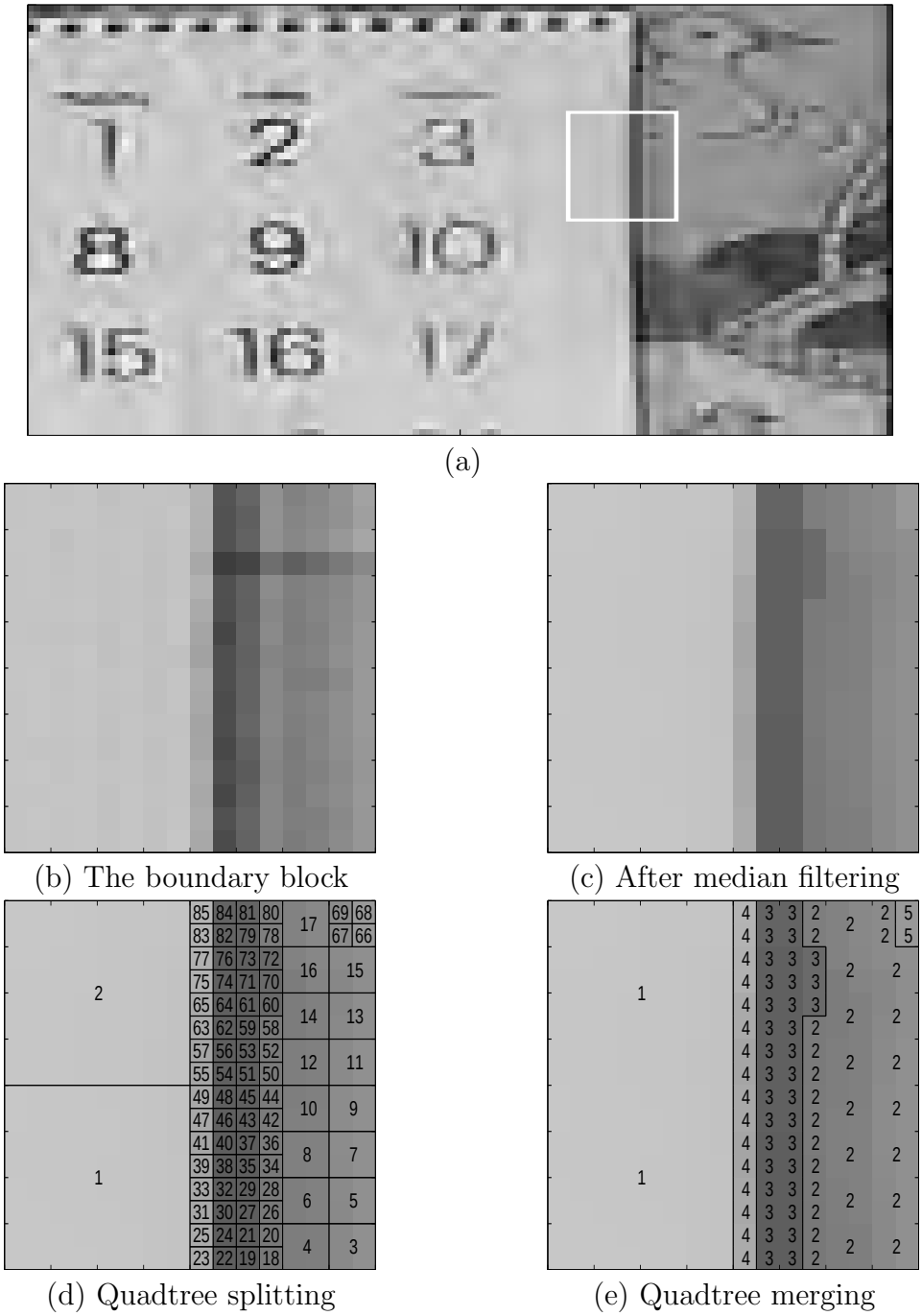
While the above homogeneity criterion is tested at each node of the quadtree, a spatial segment may also spread across a number of neighbouring nodes. By itself, the quadtree splitting process usually results in over-segmentation. To reduce this effect, a merging step is performed between adjacent nodes of the quadtree, by a verification of the fifth condition. The similarity measure between any two adjacent segments is realized by a comparison of their mean values as follows:

$$P_2(S_i \cup S_j) = \begin{cases} TRUE, & \text{if } (|\overline{S_i} - \overline{S_j}| < T_{mean}) \\ FALSE, & \text{otherwise} \end{cases} \tag{4.5}$$

The following example demonstrates how the above segmentation process functions at the block level. Figure 4.2-a shows a 16-by-16 block which straddles the boundary between the calendar and the wallpaper from the sequence "Mobile and Calendar". An enlarged view of this block is displayed in Figure 4.2-b. Note that the region near the boundary is affected by lighting conditions, as there is a shadow cast on the background by the calendar object. Figure 4.2-c shows the same block, after the frame is pre-filtered by a median operator. Using a 3x3 median filter, each value in the image is replaced by the median value of the 3x3 block centered on this pixel. The objective of the prefilter is to reduce the effect of noise and isolated features on an image before segmentation.

Figure 4.2-d shows the output from the splitting phase, where each numbered square represents an end-node in the quadtree. The sizes and locations of those nodes depend on the local texture within the block. For example, where there is a large patch of pixels of a relatively constant value, such as in nodes 1 and 2, the node may remain as one larger block; it may also produce a single-pixel node at the lowest level, as in nodes numbered 18 and higher, because a larger encompassing node has a large standard deviation. From this figure, it also appears that a number of adjacent nodes seem to be of very similar gray level. Because an arbitrarily-shaped segment can always be decomposed into components which are nodes of a quadtree, the subsequent merging stage allows an object to form freely across the branches of a quadtree by grouping neighbouring nodes with similar values. Figure 4.2-e shows the result of this merging process.

With the merging step, it also occurs that the process should be allowed to take place across the block boundary, because many spatial segments are not confined to a single block. Under a similar procedure, consideration is then given to adjacent segments which belong to different blocks in an inter-block merging stage. If two neighbouring regions satisfy condition (4.5), they are combined to form a new region. The boundaries between different regions are therefore no longer dictated by the structure of the quadtree or the shared edges of adjacent blocks. A spatial region



(a)

(b) The boundary block

(c) After median filtering

(d) Quadtree splitting

(e) Quadtree merging

Figure 4.2: Different stages of the quadtree segmentation

which is segmented correctly may also improve the accuracy of its estimated motion.

The above quadtree split-and-merge segmentation approach can be summarised in the following steps:

- For each square block S, calculate its $P(S)$. If $P(S) = TRUE$, label the

block as one segment (no splitting). If $P(S) = FALSE$, split the block into four square quadrants of identical size and label each quadrant as a separate segment (splitting).

- At each splitting, re-assess the condition P on each split quadrant. Further split each quadrant in a similar manner if the condition is $FALSE$.
- Stop splitting when for any segments, either $P = TRUE$ or the segment size is one (i.e. a single pixel).
- Within each block, look for adjacent segments where the difference in their mean values is less than a pre-determined threshold T_{mean}. Merge and re-label the combined segment, and re-calculate its mean value.
- Continue the intra-block merging, until no further merging is possible (i.e. the difference in the mean values of any two adjacent segments is more than or equal to T_{mean}).
- Repeat the above merging procedure, allowing new segments to form across the block boundary.

4.2.2.1 Inclusion of color in segmentation

The criteria for splitting and merging have been formulated based on the gray values, or the luminance, of the frame. From an YUV color model, the luminance is understood to induce a higher sensitivity in human perception than the chrominance [17]. There are, however, circumstances where using the values from only one color dimension creates the potential for undersegmentation. For example, when the difference between two regions is reflected mainly in the chrominances, a luminance-only segmentation will fail to detect the boundary between them. Inclusion of color in the segmentation process has therefore been considered in a number of studies [39, 40, 43, 56].

The main objective of using colors is to reduce the effects of undersegmentation. In the split-and-merge strategy proposed earlier, the causes of undersegmentation can be traced back to two following sources:

- Under splitting: there are nodes of the quadtree which sit across a spatial edge, or the boundary of two objects. This is due to a lack of contrast of luminance levels within the node, preventing it from being further split.
- Over merging: segments which belong to different objects are grouped together during the merging step. Again, this is mostly due to the difference in the luminance between these objects not being large enough to distinguish them.

Both of these problems can be addressed directly by incorporating other color dimensions into the test criteria in (4.4) and (4.5). Specifically, spatial homogeneity within each region should be maintained across all color dimensions, instead of just the luminance. Since both the splitting and merging operations are driven by this requirement, they can be readily extended to include the other dimensions of the color space in the following manner:

- Splitting: This process is carried out until condition (4.4) is satisfied under all three color dimensions, i.e.

$$P(S_i) = \begin{cases} TRUE, & \text{if } max\left(\sigma(S_i(y)), \sigma(S_i(u)), \sigma(S_i(v))\right) < T_\sigma \\ FALSE, & \text{otherwise} \end{cases} \tag{4.6}$$

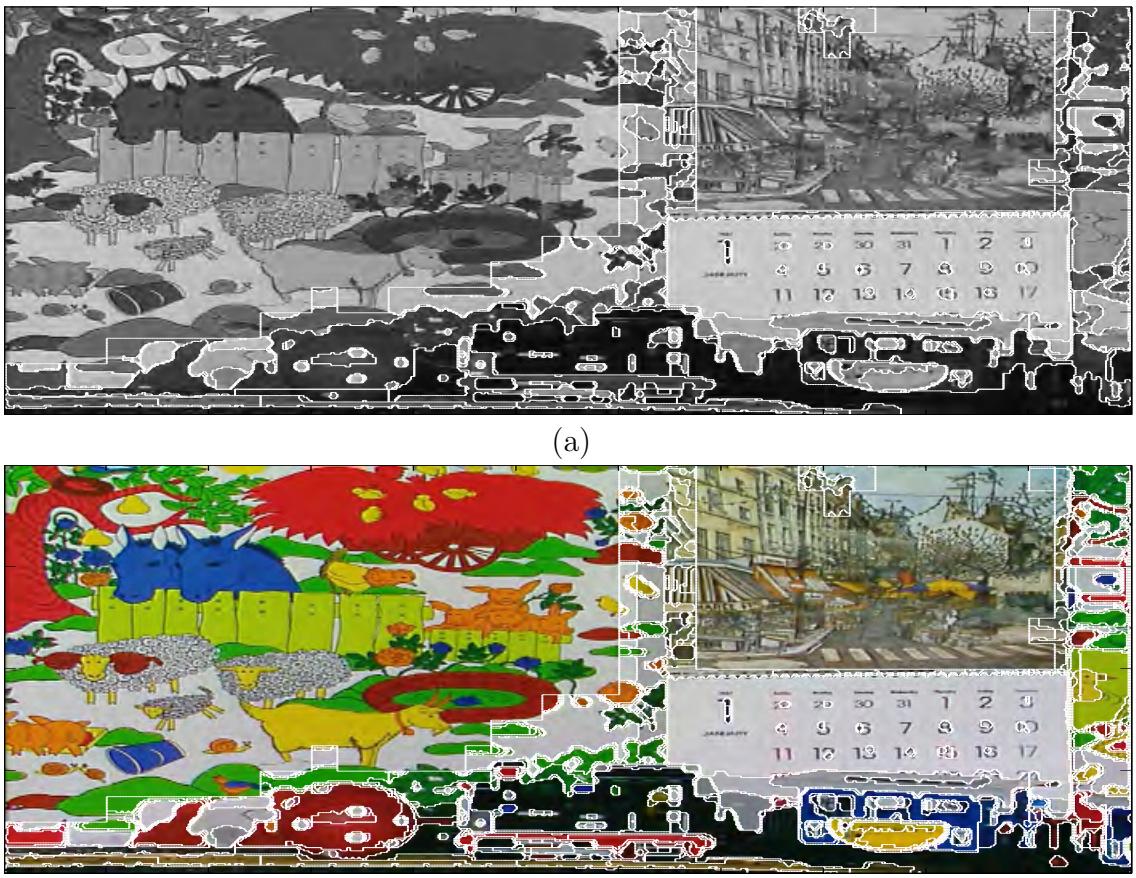
- Merging: Two regions are merged only if the similarity measure (4.5) is satisfied concurrently at all three color dimensions, i.e.

$$P_2(S_i \cup S_j) = \begin{cases} TRUE, & \text{if } max \begin{pmatrix} |\overline{S_i(y)} - \overline{S_j(y)}|, \\ |\overline{S_i(u)} - \overline{S_j(u)}|, \\ |\overline{S_i(v)} - \overline{S_j(v)}| \end{pmatrix} < T_{mean} \\ FALSE, & \text{otherwise} \end{cases} \tag{4.7}$$

In the above expressions, $S(y)$, $S(u)$ and $S(v)$ indicate the samples taken respectively at the luminance and two chrominances in the YUV color space of segment S. The potential for undersegmentation is significantly reduced by ensuring that the measurement criteria is satisfied at every color dimension.

The example in Figure 4.3 shows the advantage of using color in segmentation, as compared to a luminance-only approach. In the first case, it is clearly seen that

part of the ball is wrongly segmented into the background, due to a lack of contrast between the two regions if only the luminance component is used. When the other two chrominance components are taken into consideration, as demonstrated in the second result, these regions are successfully separated from each other. It is easily seen that in the colored frame, the ball and the background are of rather distinctive colors. On the other hand, the gray-scale version, which represents the luminance levels, shows little difference in the shades of gray between two regions. The results are obtained by using the same set of thresholds in both cases.



(a)

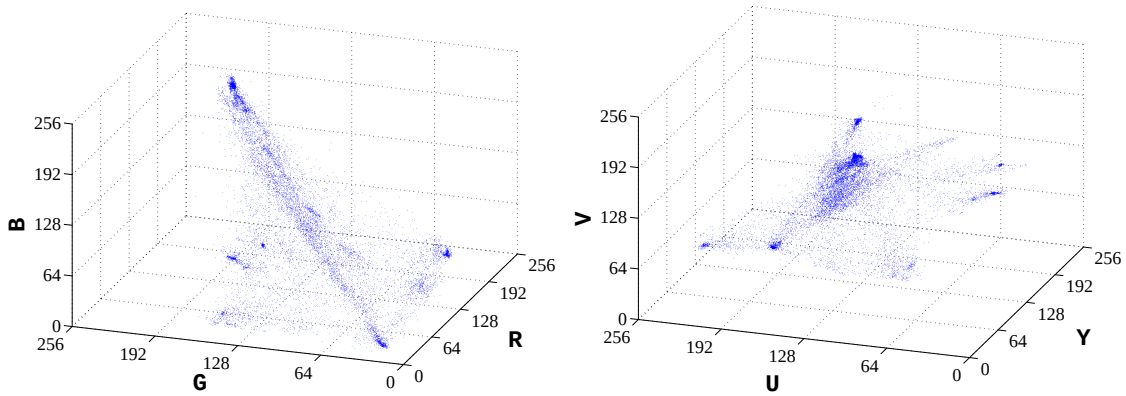
(b)

Figure 4.3: Results of the spatial segmentation, (a) using only the luminance component, and (b) using luminance and two chrominances

The parameter $P_2(S_i \cup S_j)$ in (4.7) expresses the similarity of two regions via the difference between their mean values, which can be considered as a *distance* measure. In a number of color-based segmentation approaches in the recent literature, such

distance measures have been proposed using a linear combinations of the color components [43, 53]. While doing this may simplify the calculation, it also introduces an averaging effect on the measure, where a more distinguishable color component may have to compensate for those which are less so. The condition in (4.7) requires that two regions must show a similarity in *all* color dimensions in order to satisfy the merging criterion.

The YUV color model is used because of the lower correlation between its color components. Figure 4.4 shows a 3-D representation of data using RGB and YUV color models respectively, from the color video frame in Figure 4.3-b. It can be seen from the RGB plot in Figure 4.4-a that the pixels tends to congregate along the diagonal axis, which translates into a higher correlation between the color components using the RGB model, while the YUV plot in Figure 4.4-b does not have this tendency.



(a) (b)

Figure 4.4: Distribution of image data, in (a) RBG color space, and (b) YUV color space

The correlation between different components in each color space can also be expressed quantitatively via the correlation coefficient [75]. The correlation coefficient between two sets of data X_1 and X_2 is calculated as follows:

$$\rho(X_1, X_2) = \frac{E(X_1, X_2) - E(X_1)E(X_2)}{\sigma^2(X_1)\sigma^2(X_2)} \tag{4.8}$$

where E indicates the expected value, and σ^2 is the variance. A value of ρ close to zero indicates little or no correlation between two sets of data, while a larger

absolute value of ρ is associated with a higher level of correlation. Table 4.1 contains the values for the coefficients calculated between pairs of color components under RBG and YUV color models for one color video frame. It shows that the absolute values of the coefficients obtained on the RGB color model are significantly higher than the values associated with the YUV color model. A high correlation often reduces the effectiveness of using color for segmentation, as there would be a high level of redundancy in the responses of the color dimensions. On the other hand, the lower correlation amongst components of the YUV color model would enhance the detection by switching the emphasis to the most distinguishable component.

RGB		*YUV*	
$\rho(R,G)$	0.6216	$\rho(Y,U)$	-0.0416
$\rho(G,B)$	0.6832	$\rho(U,V)$	-0.2354
$\rho(B,R)$	0.5320	$\rho(V,Y)$	-0.1264

Table 4.1: Correlation coefficients from RGB and YUV color space

4.2.3 Self-expanding quadtree on featureless regions

The above quadtree segmentation operated within an area which had been decided on by the boundary detection measure. Because the detection measure is formulated using a block-based approach, it is also subject to the aperture problem, in a similar way to a block-based motion estimation. The spatial segmentation therefore should not be treated simply as a passive process, but it should also be made capable of detecting, and making correction to such errors if they occur. The self-expanding quadtree in this section is designed to provide such capabilities to the previous quadtree segmentation scheme.

In block-based motion estimation, most algorithms are focused on finding a motion vector to match the current region to a region in the reference frame. Under some circumstances, however, there might exist two or more matching blocks within the same neighbourhood, leading to an ambiguous estimation. The result of a block-

based boundary detection scheme may be affected in a similar way, as illustrated in Figure 4.5. This figure shows an example where a multiple-motion block may be ambiguously classified as single-motion. Suppose that three objects A, B and C in the scene are moving in parallel to the camera plane. Assume that C is the background object and occluded by two foreground objects A and B, both of which are moving. Furthermore, both objects A and B are textured, while the surface of object C is of a relatively constant color and illumination. Under this setup, those blocks lining the shared edges between objects A and C, or between object B and C, should be classified as boundary blocks, as they contain a part of the background object C, and part of either object A or B. From a viewer's perspective, this is quite obvious because objects A and B are closer to the viewer than object C, and the motion boundary coincides with a depth discontinuity as well.

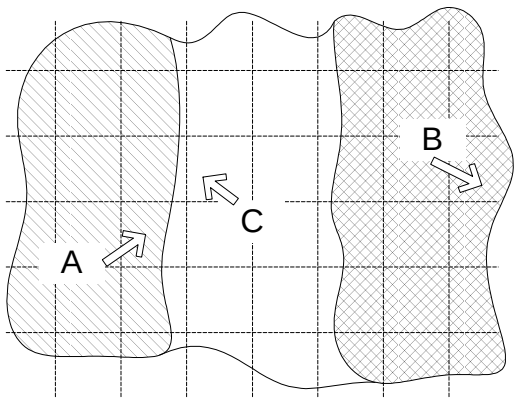


Figure 4.5: Ambiguity in classifications of blocks along object boundaries

From a two-dimensional representation of the above scene, the differentiation in depths between different objects is not obvious. In this example, due to the lack of texture in object C, a block positioned across the boundary between objects A and C can still be matched in its entirety using only motion information from the foreground object A. While parts of the block belonging to the background object are a temporal-mismatch with respect to its true motion, the lack of texture means that it still is a good match spatially. Without being aware of the depth difference, it is difficult to detect the coexistence of multiple motions within this window. The consequence is a mis-classification of this boundary block as single-motion, which

may lead to a loss of the object boundary in a subsequent segmentation because the algorithm fails to identify the motion discontinuity at such locations.

The potential confusion between single/multiple motion classification is attributed largely to a lack of local features over one object in the scene. It is well known that the reliability of motion estimation is highly dependent on the amount of local features. For example, from the Lucas-Kanade motion estimation method [76], the optical flow vector (Δ_x, Δ_y) at an image point in a neighbourhood region R is expressed as:

$$\begin{bmatrix} \Delta_x \\ \Delta_y \end{bmatrix} = \left(-\Sigma_R \begin{bmatrix} I_x.I_t \\ I_y.I_t \end{bmatrix} \right) \left(\Sigma_R \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right)^{-1} \tag{4.9}$$

where I_x and I_y are the spatial gradients along the horizontal and vertical axis, I_t is the temporal gradient (i.e. the interframe difference). The stability of this solution is dependent on the non-singularity of the second matrix on the right-hand side of this equation. On a region with little or no spatial features, the values of I_x and I_y are close to zero, therefore making this matrix singular. In [52], the reliability factor associated with a locally estimated motion vector is expressed as:

$$R = \frac{\lambda_{min}}{\lambda_{max}} \tag{4.10}$$

where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of this matrix. It also shows that the matrix becomes increasingly ill-conditioned as $(I_x^2 + I_y^2) \rightarrow 0$.

Within the context of motion segmentation, the consequence of mis-classifying a multiple-motion region as a single-motion region is severe and should be corrected as early as possible. When the non-dominant object is non-textured, the difficulty is how to realize that such matching is actually *misleading*, as a good match in this case does not equate to a single-motion. It is usually not possible to detect this on the basis of a local estimation, so in order to re-evaluate the validity of an initial single-motion assumption, it is necessary to rely on observations of the motion field on the neighbouring blocks. The details of this process is explained in the remainder of this section.

It should be recalled that the purpose for creating the spatial segments is because their motion can be estimated with higher reliability than the block-based estimation. While it would be useful to keep the area under spatial segmentation as small as possible, the process should also be capable of expanding a local segmentation into its neighbouring region where such expansions are beneficial for creating an unambiguous segment boundary. In conjunction with the multiple/single-motion classification, it is proposed that the segmentation area may grow adaptively according to the local spatio-temporal content, under two expansion criteria as follows:

1. Inclusion of single-motion blocks, which are a neighbour to another single-motion block with a different motion vector.
2. If a block is included in the spatial segmentation, and remains as one single segment after the quadtree segmentation, then segmentation is extended to all of its immediate neighbouring blocks.

The objective of the first expansion is to increase the resilience of the segmentation method to cases where the local classifications are ambiguous. When two adjacent blocks are both labelled as single-motion and at the same time have different motion vectors, it rarely means that the object boundary rests on the shared border of the blocks, but more likely because one block is subject to multiple object motions, but has not been classified as such due to a lack of local features on the objects. Rather than using a block-based motion estimation for such regions, it is a more prudent strategy to include these block into the spatial segmentation to enable a region-based estimation approach at such location.

The second expansion criterion is based on the deduction that a block only remains as a single segment after the quadtree segmentation because it has little or no texture. Such a block is often part of a larger region, so it is logical that the neighbouring blocks should be included to allow the spatial region to form without obstruction. For example in Figure 4.6, it would be incomplete for a spatial segmentation process to include the center block and not its neighbours.

As both expansions are driven by the result of the segmentation process itself, this

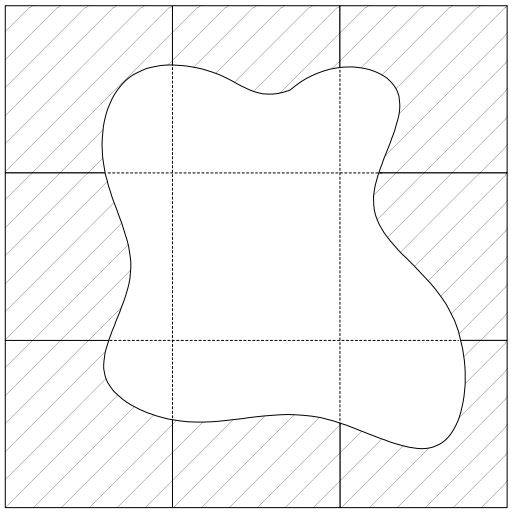


Figure 4.6: Expanding the segmentation area from the center block to the region boundary

segmentation approach would hereto be referred to as a *self-expanding quadtree*. A revisit to the example in Figure 4.7 may help to demonstrate the usefulness of this strategy. Consider a group of four successive blocks from *1* to *4*, where initially the moving object boundary may not be detected on blocks *1* and *4* due to the non-texturedness of the outliers. Furthermore, assume that each block is assigned a motion vector $mv_i, i = 1..4$. However, if objects A and B are moving differently, or $mv_1 \neq mv_4$, then it may be deduced that there must exist at least a value of $mv_i, 1 \leq i \leq 3$ so that $mv_i \neq mv_{i+1}$. In other words, in this group of four blocks, there are two adjacent blocks with different motion vectors. The spatial segmentation is then expanded to include these two blocks, by the first expansion criterion. In addition, if an included block is inside the non-textured object C, the second expansion criterion would eventually allow the entire region under C to be segmented spatially.

The self-expanding quadtree algorithm is summarised in diagram 4.8. In the previous color segmentation in Figure 4.3-b, for example, the algorithm expanded the segmentation area into the lower part of the calendar, which is a region with relatively little texture

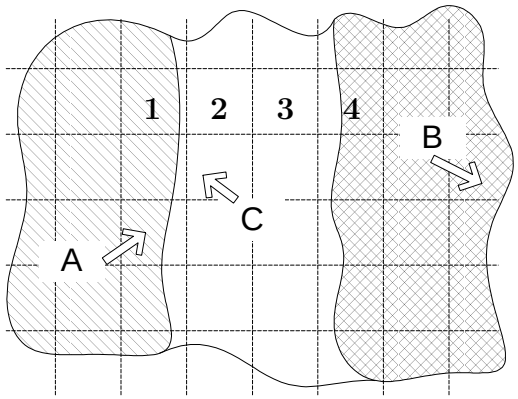


Figure 4.7: Application of the self-expanding quadtree over regions of ambiguous motion

4.3 Motion-based region merging

Having a video frame partitioned into groups of single motion blocks and a set of spatial segments, the next task is to identify the object that each of them belong to, and form the object masks accordingly. For most spatio-temporal algorithms, this involves two steps:

- Estimating motion parameters for every segment
- Grouping of segments into individual objects, based on a similarity measure.

The design of a similarity measure between regions is often considered as a critical factor in a segmentation approach [40], along with the region-merging strategy. In fact, designs of such measures may play a key role in differentiating one method from another.

For a video scene with background which is stationary and a moving foreground, the task can be reduced to assigning any region with a non-zero motion to the foreground object [49]. In sequences involving a non-stationary background, for example due to camera movement, a similar approach can also be devised after registrations have eliminated the background motion, assuming that this motion has been properly derived from a global estimation [53]. A region can be assigned either a foreground or background label, depending on whether its motion deviates from or conforms

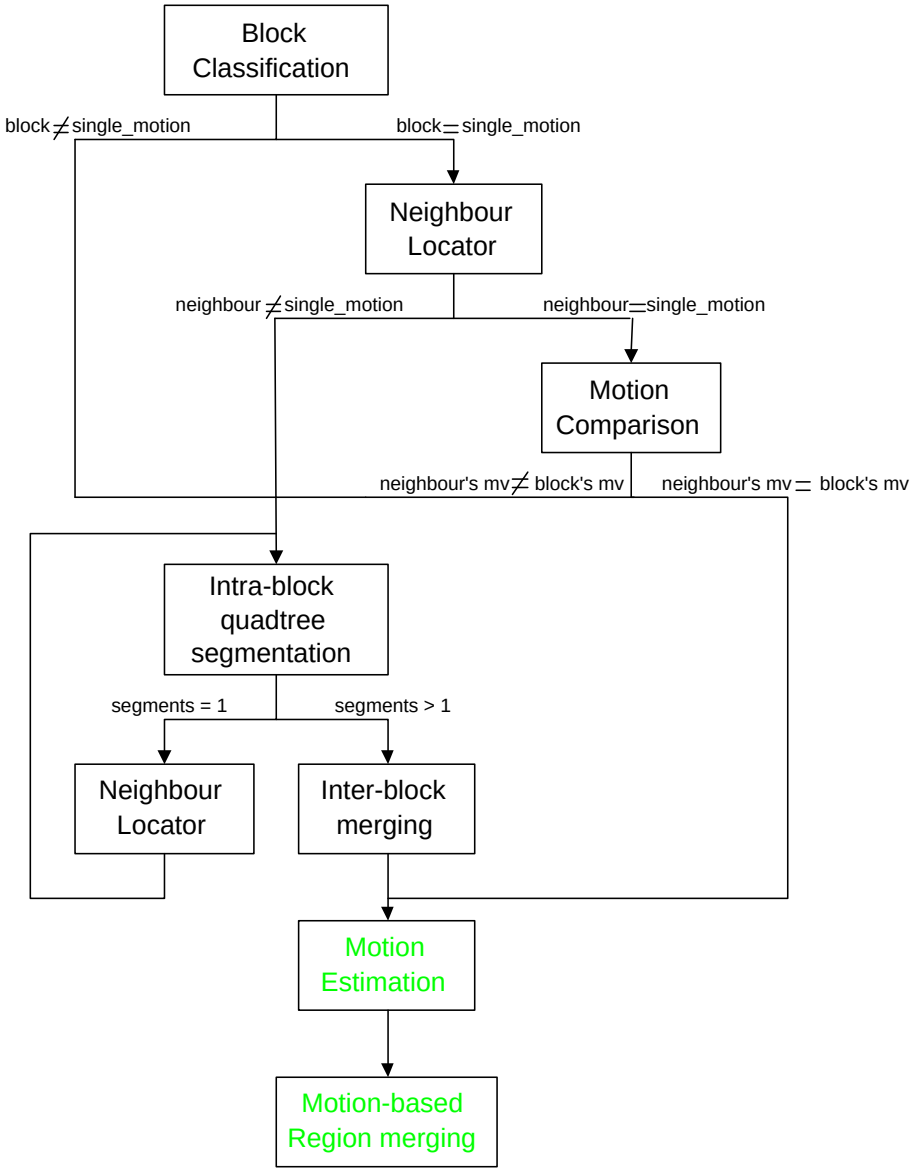


Figure 4.8: Self-expanding quadtree spatial segmentation

with the global motion, respectively. In such cases, it is usually unnecessary to provide an exact motion model for the foreground objects. However, there are a number of disadvantages with this approach. First, the very existence of foreground objects would ultimately affect the estimation of the global motion. The significance of such effects depends on the relative sizes and the difference in movements between the two groups. Second, with the decision to assign segments to an object based entirely on the deviation of its motion from the background, it is rather difficult to

distinguish between foreground objects which overlap. It has also been recognised as a common weakness in such classification strategies [26]. This work, on the other hand, bases the formation of each object mask on the motion similarity amongst individual neighbouring segments.

4.3.1 Geometry of motions

In the previous chapter, formulation of the single-motion classification is based on a translational motion model. For the purpose of motion estimation in tracking, a translational model is said to be more reliable and accurate over small inter-frame displacements [70]. The validity of this model over the scope of a video frame is also assumed in some other works [77]. Furthermore, a piece-wise linear object motion can usually be approximated by a series of successive translations.

A translational model however presents a number of limitations, especially when a measure of similarity between different object motions is desired. Segments at opposite ends of a rotating object, for example, may have motion vectors of opposite directions even though they belong to the same object. The panning effect also produces a larger translation for some regions of the same object if they are closer to the camera. In designing a similarity measure between regions, these issues are difficult to deal with if only translations are considered.

4.3.1.1 The affine model

Let P be a point in a 3-D Cartesian systems, with its coordinates being (x, y, z). When an object moves through a three-dimensional space, its rigid motion can be decomposed into 3 successive rotation about the axes X, Y and Z, followed by one translation. Let the translation shift be (u, v, w) and the three rotation angles be

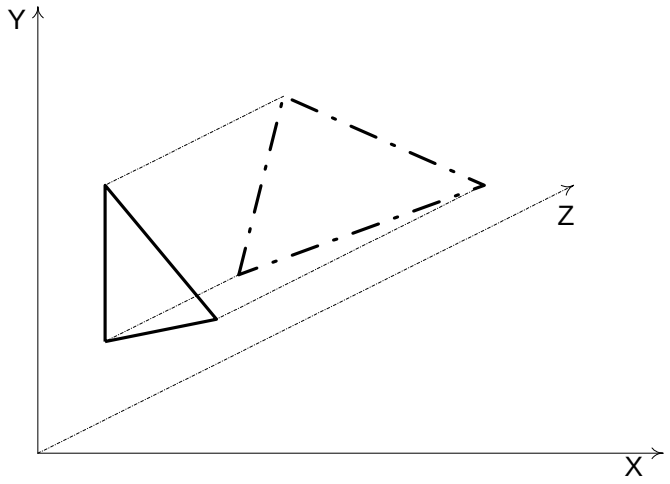


Figure 4.9: Projection onto the image plane XY under an affine camera model

α, β and γ. The rotation component of the transformation can be written as [78]:

$$\begin{aligned}\mathcal{R} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \end{aligned} \tag{4.11}$$

Let P', with coordinates (x', y', z'), be the image of P under this transformation. The mapping of the coordinates is expressed as:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4.12}$$

In the remainder of this chapter, an affine model is used for estimation and comparison of the estimated motions. Under this model, an orthogonal projection of all points on an object surface onto the image plane XY is assumed, as illustrated in Figure 4.9. The relation in scene geometries in a two dimensional representation

can be deduced from equation (4.12) as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} r_{31}z \\ r_{32}z \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix} \tag{4.13}$$

In addition, assume that P belongs to the outer surface of an object, and this surface can be approximated by a planar constraint as follows:

$$z = ax + by + c \tag{4.14}$$

Equation (4.13) is then equivalent to:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} (r_{11} + r_{31}a) & (r_{12} + r_{31}b) \\ (r_{21} + r_{32}a) & (r_{22} + r_{32}b) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} (u + r_{31}c) \\ (v + r_{32}c) \end{bmatrix} \tag{4.15}$$

Therefore for two adjacent frames k and $k-1$, an affine motion estimation involves finding a six-parameter model which approximates the transformation between image coordinates as:

$$\begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \tag{4.16}$$

While translations are often only adequate for object trajectories which are in parallel to the image plane, an affine model allows for a greater range of linear motions [57]. However, it is also important to note that a 2-D estimated motion not only depends on the object movement in 3-D, but also the characteristics of the object surface. When the outer surface of an object lies on a plane, as described in equation (4.14), the integrity of the model is well maintained for a rigid motion. If this condition is not met, for example when the object surface corresponds to a polynomial with quadratic terms, the accuracy of the estimation using an affine model will be affected as such nonlinear terms are unaccounted for in equation (4.15). The extent of such omissions depends on the characteristic of the object outer surface itself, and the parameters r_{31} and r_{32} of the rotational component, noting that

$$r_{31} = -cos(\alpha)sin(\beta)cos(\theta) + sin(\alpha)sin(\theta) \tag{4.17}$$

$$r_{32} = cos(\alpha)sin(\beta)sin(\theta) + sin(\alpha)cos(\theta) \tag{4.18}$$

Motion estimation is carried out in the remainder of this chapter based on the assumption that the interframe displacement is small so that an affine approximation is adequate for the estimation. In the next chapter, where the integrity of object masks are to be maintained over more distant frames, then a model with quadratic terms is selected for more flexible estimation.

4.3.2 Region-based motion estimation

The motion estimation is formulated as an optimisation problem, which seeks to minimise the residual difference on a region under an affine transformation. The residual difference over a region $\mathcal{R}$ in frame I_k, with respect to the reference frame I_{k-1}, is defined as:

$$\mathcal{E} = \sum_{(x,y)\in\mathcal{R}} \left(I_k(x,y) - I_{k-1}(a_{11}x + a_{12}y + a_{13}, a_{12}x + a_{22}y + a_{23})\right)^2 \tag{4.19}$$

4.3.2.1 Initialisation of affine parameters

A translational motion can be re-written as a special case of the affine model:

$$\begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix}$$

For each region, the optimisation is initialised from its estimated 2-parameter translational motion. When the region is a single-motion block, the initialisation is the same as the motion vector associated with the block. On the other hand, a spatially-segmented region in the multiple-motion zone is usually associated with more than one candidate motion vector. The initialisation is selected amongst these candidates by using a three-frame matching approach, as described in the following procedure.

Let R be a spatially-segmented region on frame I_k, and V be the set of candidate motion vectors passed onto R from estimations in its larger encompassing blocks. Using only I_k and its reference frame I_{k-1}, a best-matched vector from the candidate

set is usually selected as:

$$(v_x, v_y) = arg \min_{(v_x,v_y)\in\mathcal{V}} \left(\sum_{(x,y)\in\mathcal{R}} |I_k(x,y) - I_{k-1}(x - v_x, y - v_y)| \right) \tag{4.20}$$

The underlying assumption for this selection criterion is that a good match can be located for the given region at the reference frame. This assumption would be violated if the region is occluded at the reference frame. In fact, because the spatial segmentation is targeted at the areas with moving object boundaries, there is a higher probability that each region is bordering on a motion discontinuity, and therefore much more likely be subject to the occlusion effects. Under occlusions, the residual difference may still remain significant even when the true motion vector is used, because an occluded region would remain unmatched. In other words, a high residual difference does not always mean a wrong match. For this reason, it becomes more difficult to distinguish between correct and incorrect estimations if selections strictly adhere to the above criterion.

The ambiguity in estimation can be reduced if the next video frame is also taken into account. Consider the following example of two overlapping objects in three frames, illustrated in Figure 4.10. The part belonging to object A at frame k is occluded in reference frame $k-1$ by object B. This region therefore can only be partially matched in a backward estimation. However, if the velocity of the object A remains relatively unchanged, then a good match for this portion of the occluded object can be located in the next frame $k+1$, as seen in Figure 4.10.

This is often referred to as the uncovered background problem, but it certainly does not prevent a good match from being found. It is noted that a similar scheme has also been implemented in [47] for improved motion estimation. In this work, however, a stronger association between segments and motion boundaries in the multiple-motion areas provides for a more effective implementation, as occlusions only become a problem at such boundaries.

The criterion for a best-match motion vector is therefore extended over three frame

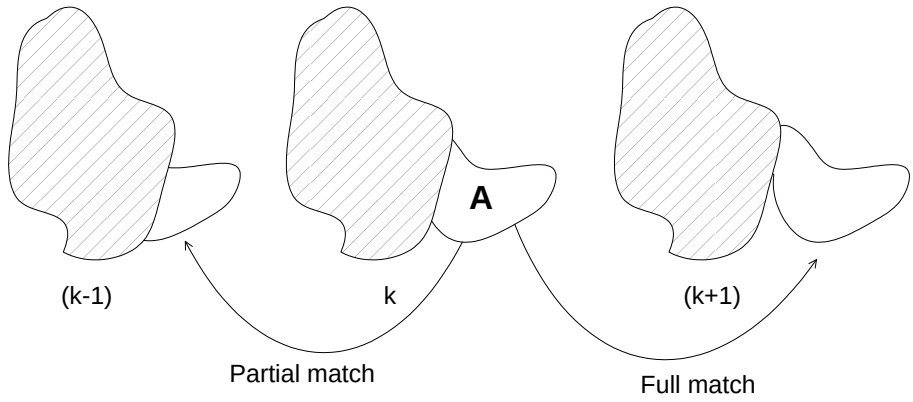


Figure 4.10: Selection of candidate motion using a three-frame approach

as follows:

$$(v_x, v_y) = arg \min_{(v_x,v_y)\in\mathcal{V}} \left(min \left\{ \begin{array}{c} \sum_{(x,y)\in\mathcal{R}} |I_k(x,y) - I_{k-1}(x - v_x, y - v_y)| \\ \sum_{(x,y)\in\mathcal{R}} |I_k(x,y) - I_{k+1}(x + v_x, y + v_y)| \end{array} \right\} \right) \quad (4.21)$$

After a best-match motion vector is identified for each region, it is passed on as the initialisation to the affine estimation in the next section.

4.3.2.2 Estimation of parameters

The Gauss-Newton optimisation method [74] is implemented for the motion estimation. From starting values $(1, 0, v_x, 0, 1, v_y)$, the optimisation aims to derive the motion parameters $(a_{11}, a_{12}, a_{13}, a_{12}, a_{22}, a_{23})$ by minimising the motion-compensated difference:

$$\begin{aligned} &\mathcal{F}(a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}) \\ &\quad = \frac{\sum_{(x,y)\in\mathcal{R}} (I_k(x,y) - I_{k-1}(a_{11}x + a_{12}y + a_{13}, a_{12}x + a_{22}y + a_{23}))^2}{number\ of\ pixels\ in\ \mathcal{R}} \end{aligned} \quad (4.22)$$

where $\mathcal{R}$ denotes the region under estimation.

The accuracy of the estimation depends on whether the minimum located on the error surface corresponds to the true motion, rather than a local minimum. This convexity can usually be assumed with a good initialisation, which places the first estimation within the proximity of the true motion [79]. Note that the sum-of-squared

differences, rather than the sum-of-absolute differences, is used in the above cost function. Because a spatially-segmented region usually presents little variation in its intensities, the motion-compensated difference associated with a pixel well inside the region is small. The difference between a correct match and an incorrect match is therefore most significant at pixels close to region boundary. In comparison to the absolute difference, the squared difference places a higher emphasis on an accurate matching of those boundary pixels. This is because any mismatch is reflected and amplified in its squared value rather than just the absolute difference.

In equation (4.22), a bilinear interpolation scheme is used to approximate the value of a pixel at position $(a_{11}x + a_{12}y + a_{13}, a_{12}x + a_{22}y + a_{23})$. With reference to Figure 4.11, the value of a pixel at an interpolated location is calculated as:

$$i = w.l.i_{11} + (1 - w).l.i_{12} + w.(1 - l).i_{21} + (1 - w).(1 - l).i_{22} \tag{4.23}$$

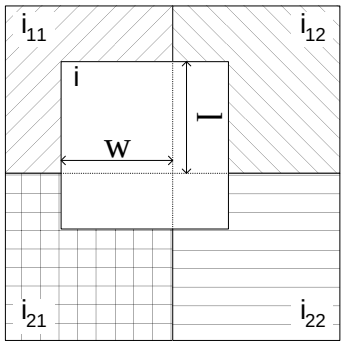


Figure 4.11: Bilinear interpolation

Occasionally, there may be pixels whose interpolated position in the reference frame falls outside the frame limits. Such pixels are not included in the sum of differences, or in the pixel count of the denominator of equation (4.22).

4.3.3 Region clustering using motions

Due to oversegmentation, an object is usually broken up into incomplete regions during an early stage of classification. Clustering involves finding regions which

are subject to a common grouping criterion. Such criteria are often expressed via a measure of similarity, which aims to provide a quantitative comparison of different regions. As is the case with the human vision system, this comparison is a multi-dimensional process and can be carried out in both the spatial and temporal domains. The difference between two regions can be expressed in terms of colors, motions and textures. A measure derived only from the color/intensity component alone often forms part of a spatial segmentation scheme [18]. When an affine motion model is used, the difference can be calculated directly in the affine parameter space [35]. Other measures based on multiple features are found in works such as [40, 54, 56, 79], where the differences calculated at individual feature spaces are often combined to produce a single quantity showing the relative distance of two regions.

Before proceeding further, recall that the segmentation strategy in this work comprises of three major stages which are carried out sequentially in the following order:

1. Classification of each image block as either single- or multiple-motion
2. Spatial segmentation in the multiple-motion areas
3. Grouping of spatial segments and single-motion blocks into objects

The second stage, spatial segmentation, included a built-in mechanism which allows expansions of the area under segmentation, as described by the self-expanding quadtree. This can also be seen alternatively as a *feedback* to the first stage, which decides the initial limits for such spatial segmentations inside a video frame. The third stage is composed essentially of motion estimation and a grouping strategy. The grouping strategy, which is now under consideration, is responsible for labelling each spatial region and single-motion block into a corresponding object. The difficulty, of course, is that neither the objects nor their defining characteristics (such as motion) are known explicitly at the beginning of this process.

Every segment can logically be considered as a candidate object. However, the number of these candidates could well exceed the number of actual objects, and therefore region merging is required to consolidate them into a smaller number of

objects. Regardless of whether a top-down or bottom-up approach is adopted for this merging, the primary function of a similarity measure is to decide if the difference between two regions is small enough so that they can be merged into a single region. The most important feature spaces where such differences are observed include their motion and color.

The following merging strategy relies on motion as its principle cue. While color can also be included for merging of neighbouring regions at this stage, it appears that such a reliance on color, even partially, would either be redundant or run counter to the earlier spatial segmentation from the second stage of the algorithm. It would also create the opportunity for inconsistencies along the decision tree, and may ultimately lead to instability of the method. In this work, a combination of two motion-based measures are used to express the degree of similarity between regions as follows:

- An absolute distance measure, $d(R_1, R_2)$, which quantifies the difference between any two regions in a single vector. The smaller the vector, the more similar two regions are, and vice versa.
- A relative measure of fitness, which aims to classify if a region's motion model is *fit* to represent the motion of its neighbour's. This results from a comparison of the residual differences between the motion models involved.

The details of both measures are provided below. In constructing the measures as well as formulating an overall region merging strategy, the following assumptions are made:

- A larger region is deemed to have more accurate motion parameters than a smaller region
- Merging of two regions is considered as a transfer of ownership of the smaller region to the larger region, and not the other way around. The combined region is therefore *owned* by the motion model associated with the larger region, until the model is updated.

4.3.3.1 The absolute distance measure

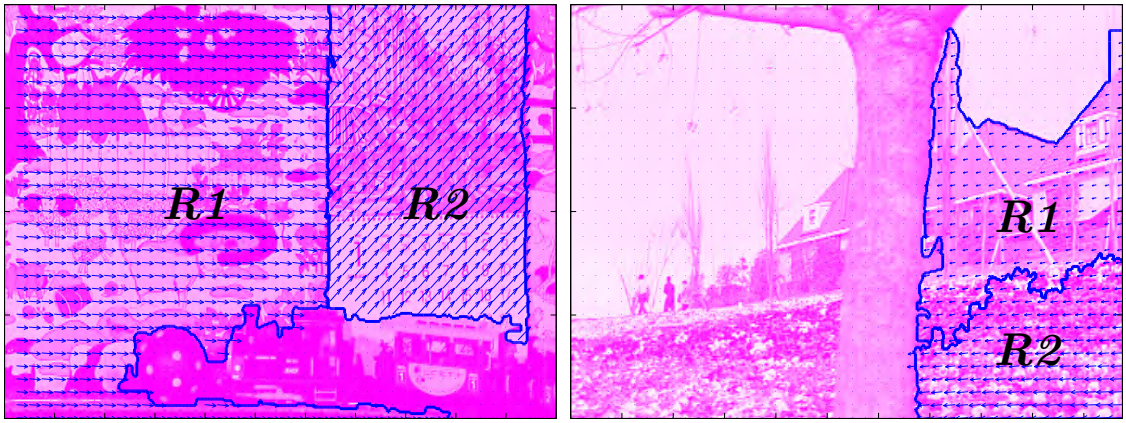
The basis for the distance measure used in this proposal is derived from the velocity-based similarity measure, or VSM, as in [38]. However, a modification is introduced to enable merging of regions with preference for larger regions, as stated in the previous assumptions.

Let R denote a region with affine motion parameters $A = (a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23})$, $p \in R$ is a pixel located inside region R, and $\overrightarrow{v}(p, A)$ represent a motion vector reconstructed at pixel p using the affine model A. If R_1 and R_2 are two separate regions, which are transformed by affine models A_1 and A_2 respectively, and $size(R_1) \geq size(R_2)$, then the distance between them is defined as follows:

$$d(R_1, R_2) = \frac{\sum_{p \in R_2} ||\overrightarrow{v}(p, A_2) - \overrightarrow{v}(p, A_1)||}{number\ of\ pixels\ in\ R_2} \tag{4.24}$$

In the given region R_2, this quantity expresses the difference between two motion fields created by the motion models A_1 and A_2. As the comparison is taken at every pixel, its value relates directly to the discrepancies in the motion fields, as opposed to measures formed in parameter space which may not correspond directly to a meaningful physical attribute of motion [79]. Note the skewness of the measure as it is averaged over the smaller region R_2, rather than the combined region $(R_1 \cup R_2)$ as in [38]. When calculated based on the smaller region R_2, both $l2$-norm and $L2$-norm measures appear to provide a better distinction than they are calculated based on the combined region, as shown in the following examples.

In Figure 4.12-a, taken from the sequence "Mobile and Calendar", R_1 corresponds to the wallpaper, and R_2 corresponds to the smaller calendar object. In Figure 4.12-b, taken from the sequence "Flower Garden", R_1 contains the front houses, and R_2 is the smaller patch corresponding to the flower garden. In both cases, R_1 and R_2 represent separate objects, although the difference between their motions is small. However, table 4.2 shows that the distinction obtained using the averaged difference on the smaller object R_2 is relatively stronger than the distinction represented by the averaged difference taken across the combined region.



(a) "Mobile and Calendar" (b) "Flower Garden"

Figure 4.12: Neighbouring regions and their motion fields

	$(R_1 \cup R_2)$	R_2
$l2-$norm	0.6714	0.7014
$L2-$norm	0.4517	0.4922

(a) "Mobile and Calendar"

	$(R_1 \cup R_2)$	R_2
$l2-$norm	1.4167	1.5801
$L2-$norm	2.7907	2.9981

(b) "Flower Garden"

Table 4.2: The distance measure between two different-moving regions, calculated from the combined region, and from the smaller region

The absolute distance measure then indicates the deviation of the estimated motion field in the smaller region from the estimated motion model owned by the larger region. Based on this deviation, a decision on whether to merge the small region into the large region is made according to a simple rule:

$$if\ d(R_1, R_2) \leq T_d$$
$$then\ (merge\ R_2\ into\ R_1)$$

As the previous experiment from Chapter 3 shows, two slightly different motions may be ambiguously supported by a single motion model, a condition which will result in undersegmentation. To avoid undersegmentation, T_d needs to be set conservatively so that objects with slightly different motions are not incorrectly merged during this stage. In a sequence with very fast motion, this setting may create oversegmentation. In this work, slow inter-frame object motion is assumed as a

necessary condition for an accurate recovery of object boundaries through segmentation. Effects of oversegmentation are subsequently dealt with through a temporal stabilisation process as described later in Chapter 5.

Figure 4.13 shows the results of region merging using the distance measure, with the threshold values for T_d ranging from 1/4 pixel to 1 pixel. From the two sequences in this figure, it is apparent that undersegmentation across moving object boundaries started to appear at values of $T_d > \frac{1}{2}$. In Figures 4.13-d and 4.13-e, the calendar object is undersegmented into the same object as the wallpaper. Likewise, in Figures 4.13-j and 4.13-k, the houses on the left side of the tree are segmented into the same object with the flower bed.

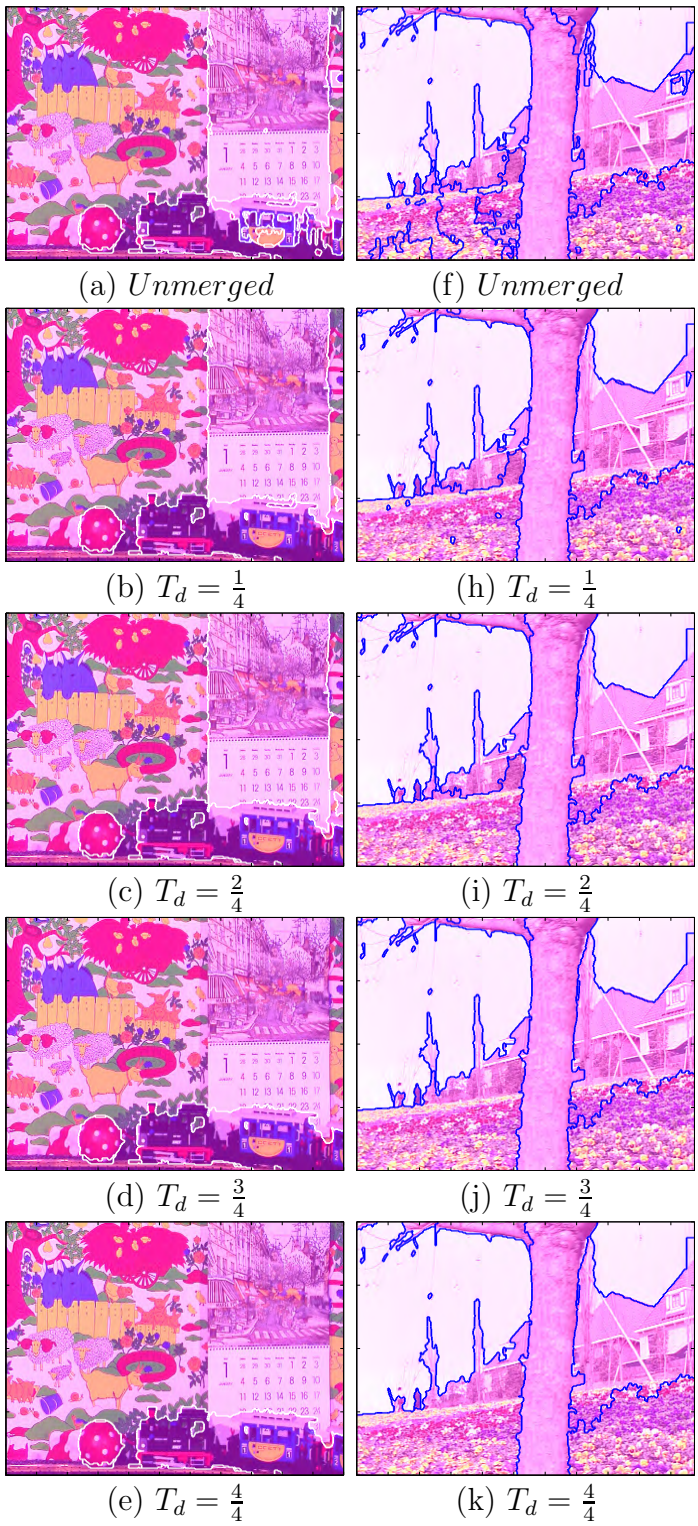
For the rest of this work, the value of T_d is set at $\frac{1}{2}$ pixel. In other words, an average difference of less than half a pixel in the motion fields allows merging two neighbouring regions. While this value may appear as rather conservative, it ensures the coherency of the motion model as the region grows, and prevents over-merging at object boundaries.

4.3.3.2 Comparing fitness of motion models over a region

The second criterion used in region-merging is derived from a relative comparison of the residual differences yielded on the same region by different models. Let E be the residual difference when a region R is transformed from frame I_k to frame I_{k-1} using its estimated motion A

$$E(R, A, I_k, I_{k-1}) = \frac{\sum_{(x,y)\in\mathcal{R}}(I_k(x,y) - I_{k-1}(a_{11}x + a_{12}y + a_{13}, a_{12}x + a_{22}y + a_{23}))^2}{number\ of\ pixels\ in\ \mathcal{R}} \tag{4.25}$$

In the simplest case, a model with the smaller difference can be chosen as a more suitable representation. Given two regions R_1 and R_2, a direct comparison can be formulated by calculating the remaining residual difference on the region R_2 using the motion model of region R_1, then comparing it with the original difference from



(a) *Unmerged* (f) *Unmerged*

(b) $T_d = \frac{1}{4}$ (h) $T_d = \frac{1}{4}$

(c) $T_d = \frac{2}{4}$ (i) $T_d = \frac{2}{4}$

(d) $T_d = \frac{3}{4}$ (j) $T_d = \frac{3}{4}$

(e) $T_d = \frac{4}{4}$ (k) $T_d = \frac{4}{4}$

Figure 4.13: Region merging using different values of threshold on the distance measure

estimation on R_2. In pseudo code form, the comparison may be stated as follows:

$$if\ (E(R_2, A_1, I_k, I_{k-1}) - E(R_2, A_2, I_k, I_{k-1})) \leq 0$$
$$then\ (merge\ R_2\ into\ R_1)$$

While this measure is simple, its effectiveness for region-merging is quite limited. Because the motion estimation procedure is optimised towards minimisation of the residual difference at each region, a locally estimated model A_2 usually produces a smaller difference on region R_2 than an adopted model taken from a neighbouring region, meaning that the above inequality would not usually be satisfied. The problem with such a direct motion-compensated difference is the intrinsic entanglement between a cost function designed for motion estimation and the comparison itself. In other words, it would be ideal if the comparison process is independent of the cost function being used in the estimation. For this purpose, a three-frame approach is again proposed for merging of regions under affine transformations.

Assume that the affine motion for each object remains relatively unchanged across three frames I_{k-1}, I_k and I_{k+1}. The motion on a region between frames I_k and I_{k+1} therefore can be approximated as the *inverse* of the estimated motion model between frames I_k and I_{k-1}. While the cost function is optimised for backward estimations, it does not necessarily produce a minimised cost function in a forward estimation. In fact, an over-ambitious optimisation scheme usually results in over-fitting of parameters in one estimation direction, hence degrading the accuracy of its matching in the inverse direction. In either case, the inclusion of a forward cost function can help to verify the similarity between two motion models.

The affine relation between a current and reference frame

$$\begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \tag{4.26}$$

can also be written in reversal as:

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \left(\begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} - \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \right) \tag{4.27}$$

Let $A^{-1} = (a'_{11}, a'_{12}, a'_{13}, a'_{21}, a'_{22}, a'_{23})$ be defined as the set of parameters corresponding to this inverse model. A derivation of these parameters can be obtained readily from the existing model A:

$$\begin{bmatrix} a'_{11} & a'_{12} \\ a'_{21} & a'_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \tag{4.28}$$

$$\begin{bmatrix} a'_{13} \\ a'_{23} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \tag{4.29}$$

It is proposed that the following comparisons are included in the testing whether a region R_2 is better represented by its neighbour motion model A_1 than its locally estimated model A_2:

1. Comparison of residual differences on R_2 due to A_1 and A_2

$$E(R_2, A_1, I_k, I_{k-1}) - E(R_2, A_2, I_k, I_{k-1}) \leq 0 \tag{4.30}$$

2. Comparison of residual differences on R_2 due to A_1 and the inverse model A_2^{-1}

$$E(R_2, A_1, I_k, I_{k-1}) - E(R_2, A_2^{-1}, I_k, I_{k+1}) \leq 0 \tag{4.31}$$

3. Comparison of residual differences on R_2 due to two inverse models A_1^{-1} and A_2^{-1}

$$E(R_2, A_1^{-1}, I_k, I_{k+1}) - E(R_2, A_2^{-1}, I_k, I_{k+1}) \leq 0 \tag{4.32}$$

Note that the motion-compensated difference under an inverse motion is calculated between a current and next frame. The last two comparisons are made to indirectly assess the performance of the inverse model A_2^{-1}. A larger difference associated with the inverse model implies that the estimated model is only optimised in an error term but is likely overfitting the region in one direction. The more accurate motion from its larger neighbour, if they both are under the same object motion, can be used to detect and correct such effects, if one of the above inequalities is satisfied.

4.4 Implementation

Details of the segmentation method are presented in the flowchart shown in Figure 4.14. It should be read in conjunction with the previous diagram in Figure 4.8, which documented details relevant to the spatial segmentation process.

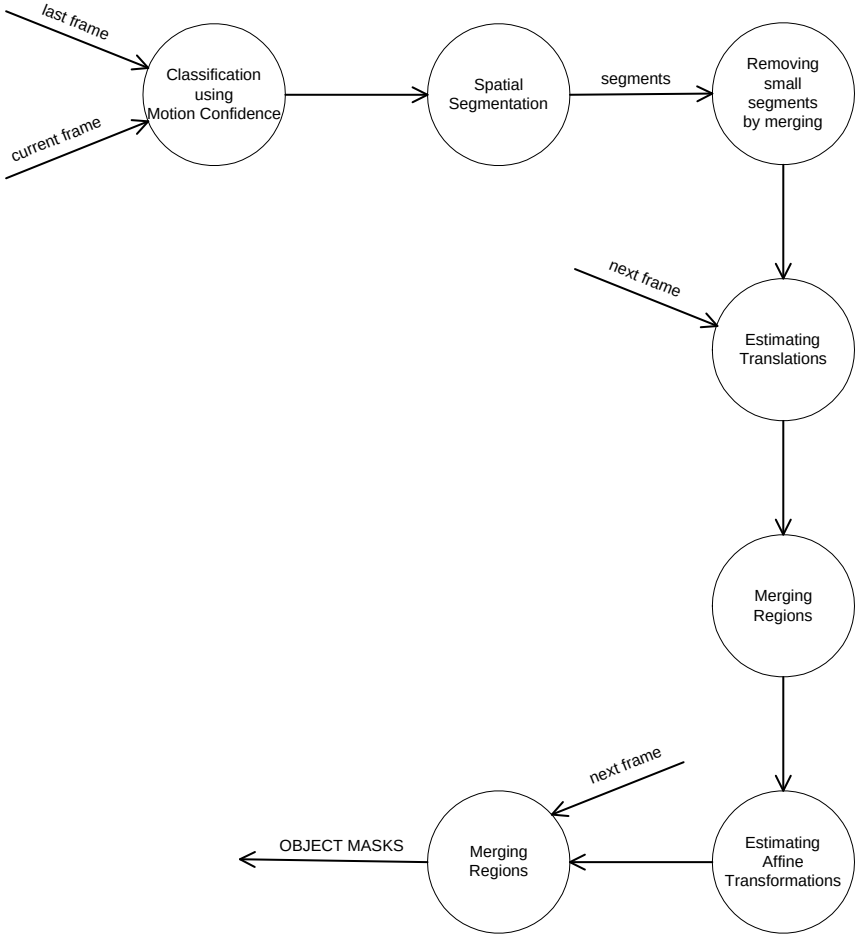


Figure 4.14: Two stages of motion-based region merging

The implementation of the algorithm, written using *MATLAB*, comprised the following modules:

1. Classification of each (16-by-16 pixel) block as either single-motion or multiple-motion, via the confidence measure
2. Selection of blocks for the spatial segmentation:
   - multiple-motion blocks

- single-motion blocks which are adjacent to a multiple-motion
- adjacent single-motion blocks with different motion vectors

3. Performing a quadtree segmentation at each block selected in step 2, by a split-and-merge approach
   - Check if the block *remains in one single region* after segmentation
   - If condition = *true* then expand segmentation to its eight neighbours. At each of these block, repeat the procedure if above condition is true.
4. Merging regions across neighbouring blocks.
5. Removal of small, isolated regions
   - Label a region with pixel count ≤ 16 (equal in size to a 4-by-4 block) as a *small* region.
   - Merge such regions with a neighbouring *non-small* region which shares most of its boundary.
6. Estimation of the translational motion for each region
   - Assign candidate motion vectors to a region, using motions from the blocks which overlap this region.
   - Select a best-match motion vector from these candidates using a three-frame approach in equation (4.21)
7. Merging of regions using their translational motion.
8. Estimation of affine motions for each region
   - Initialise the parameters using the translational motion
   - Optimise the motion-compensated difference according to equation (4.22)
9. Region-merging based on affine motions
   - Sort regions according to their sizes
   - For each region and its larger neighbour, check for satisfactory merging criteria, and form a new region accordingly.

- If more than one neighbour satisfies the criteria, choose the one which produces the smallest motion-compensated difference on the region to be merged
- Update the affine model for the merged region, using the existing model as initialisation
- Iterate the process until no further merging can be committed.

4.5 Results and comments

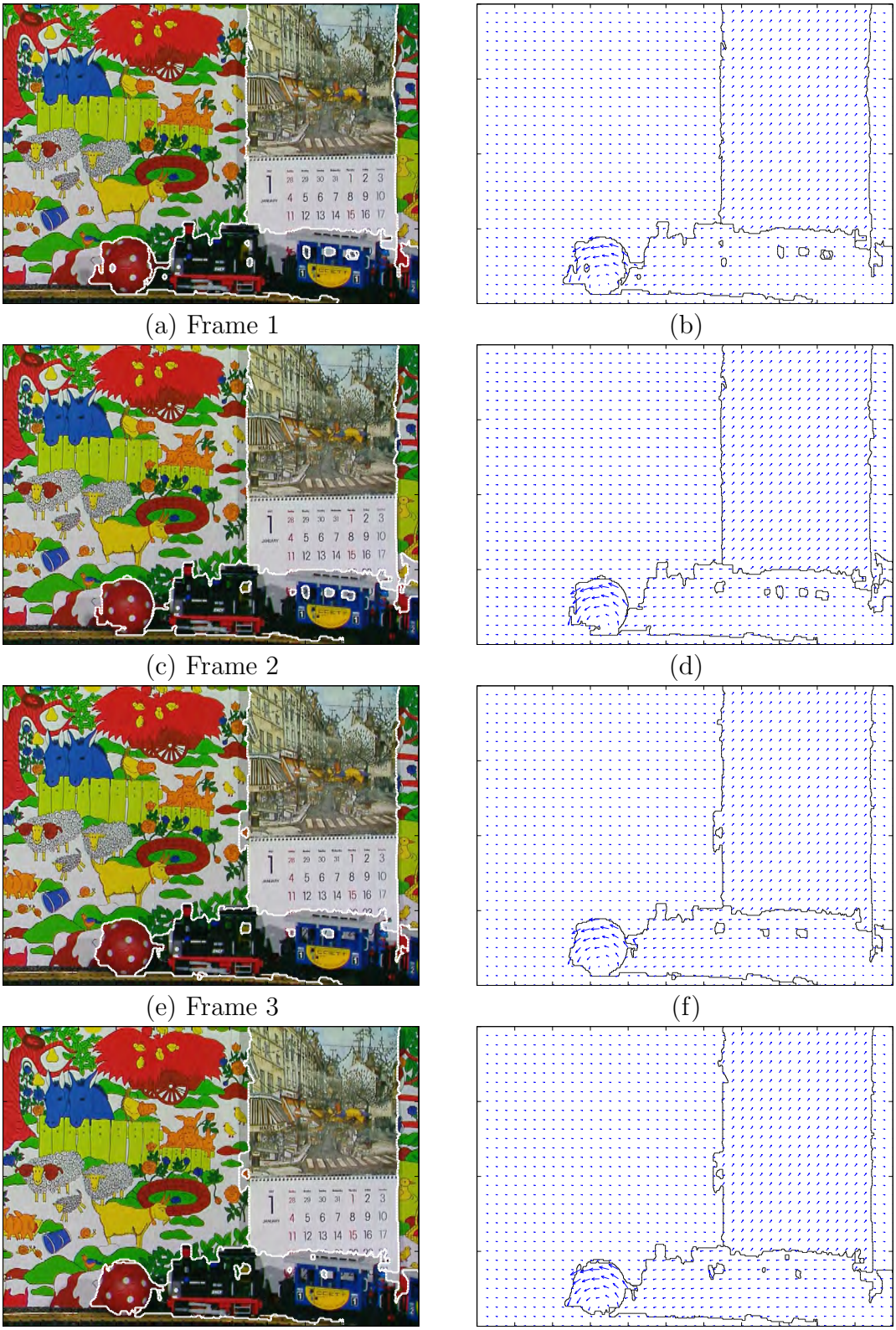
This section displays some segmentation results from the proposed algorithm, obtained at individual frames of the sequences "Mobile and Calendar" and "Flower Garden". The segmentation at each frame requires the input from three frames, including the current frame, its immediate previous and next frame in the sequence. The movements of most objects in these videos can be described as 3-D rigid parametric models, although not all motions follow a strictly affine transformation. The same set of threshold parameters are used in all experiments.

The first test sequence "Mobile and Calendar", the results from which are shown in Figures 4.15 and 4.16, has four main objects with distinctively different motion characters. The wallpaper is subject to the left-panning movement of the camera, while the calendar is moving up and down vertically in addition to this panning. The ball appears to share the same trajectory and very similar horizontal velocity with the toy train, however it is rolling with a circular motion, while the toy train motion is governed by the curvature of the track. As displayed in the results, all of these objects are identified in the segmentation masks. The calendar object is detected from the wallpaper, even though the difference between their motion is relatively small, i.e sub-pixel across adjacent frames. Similarly, the rolling ball and the train are also separated from each other, a distinction which is retrieved mainly based on the characteristics of their motions.

Note that at some frames, the part of the wallpaper positioned between the ball and the toy train is confused as being part of the train. This is because the texture of

this region is hidden under the heavy shadow of both foreground objects, making it difficult to estimate its motion accurately. When the lighting improves in subsequent frames, the region is successfully recovered and aligned with the wallpaper. It should also be noted that there are a small number of over-segmented regions, most notably around object boundaries. To deal with such effects in a consistent manner, a multi-frame approach is proposed in the next chapter.

Results for the second test sequence "Flower Garden" are shown in Figures 4.17 and 4.18. In this sequence, the actual motion on each object involved is largely due to a moving camera, apart from the walking people at the far left. Segmentation of objects is therefore mainly facilitated by the variation in the geometry of the scene. The present objects are part of different surfaces and distances to the camera, which affect their motions with regard to the camera position. In this particular example, the flower bed and the houses are seen as lying on two intersecting planes, hence their motion can be approximated using affine transformation. The tree, on the other hand, does not lie on such a planar surface, but it is assumed that the interframe displacements are small enough that an affine model would suffice for estimation. As seen from the results, the algorithm is successful at separating these objects based on their motion patterns. In addition, the people at the left of the flower garden have been segmented in the majority of frames, even though the distance from the camera means that their relative movements with regard to the surroundings are very small.



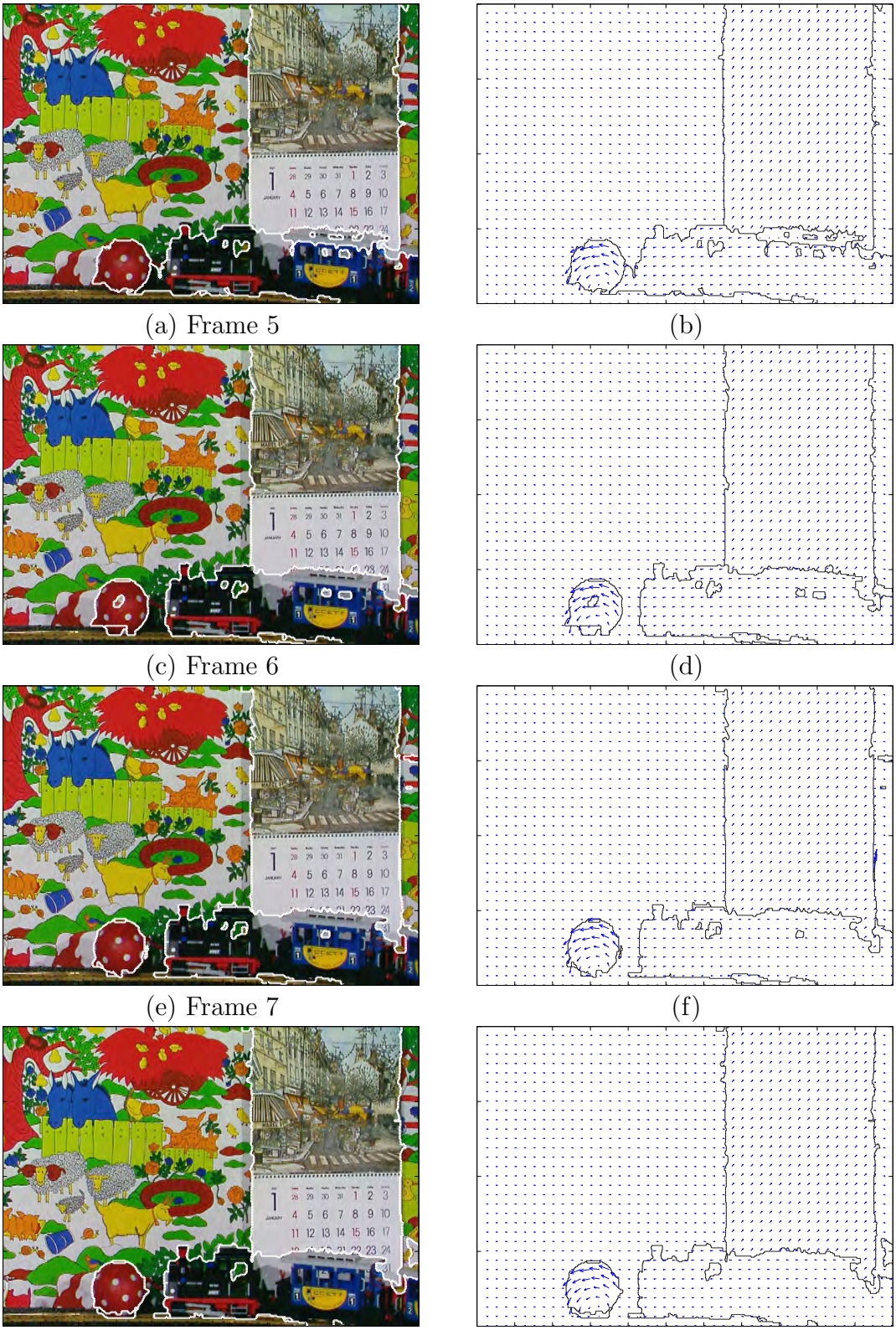
(a) Frame 1 (b)

(c) Frame 2 (d)

(e) Frame 3 (f)

(g) Frame 4 (h)

Figure 4.15: Segmentation results of "Mobile and Calendar", frames 1 to 4



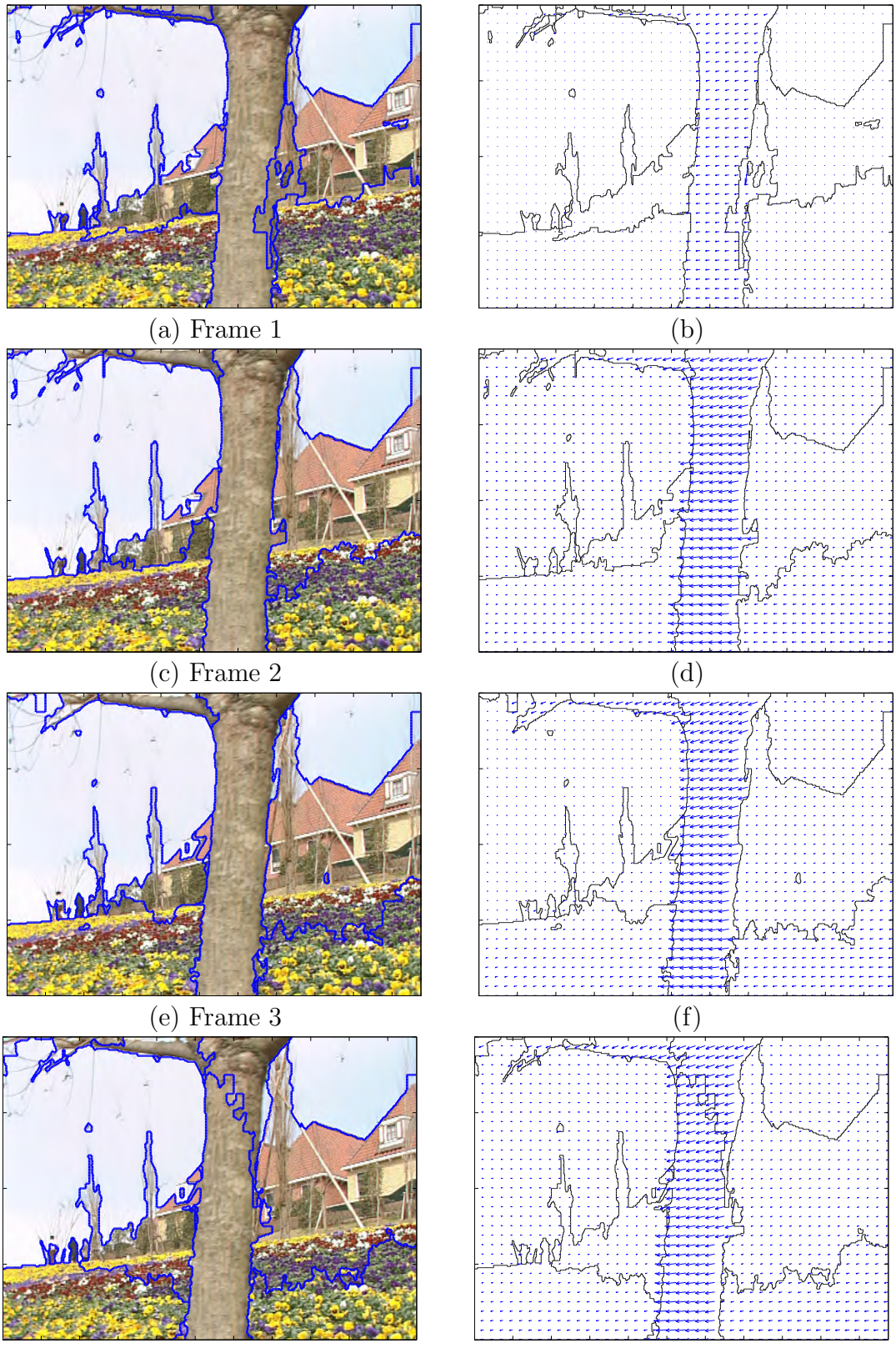
(a) Frame 5 (b)

(c) Frame 6 (d)

(e) Frame 7 (f)

(g) Frame 8 (h)

Figure 4.16: Segmentation results of “Mobile and Calendar”, frames 5 to 8



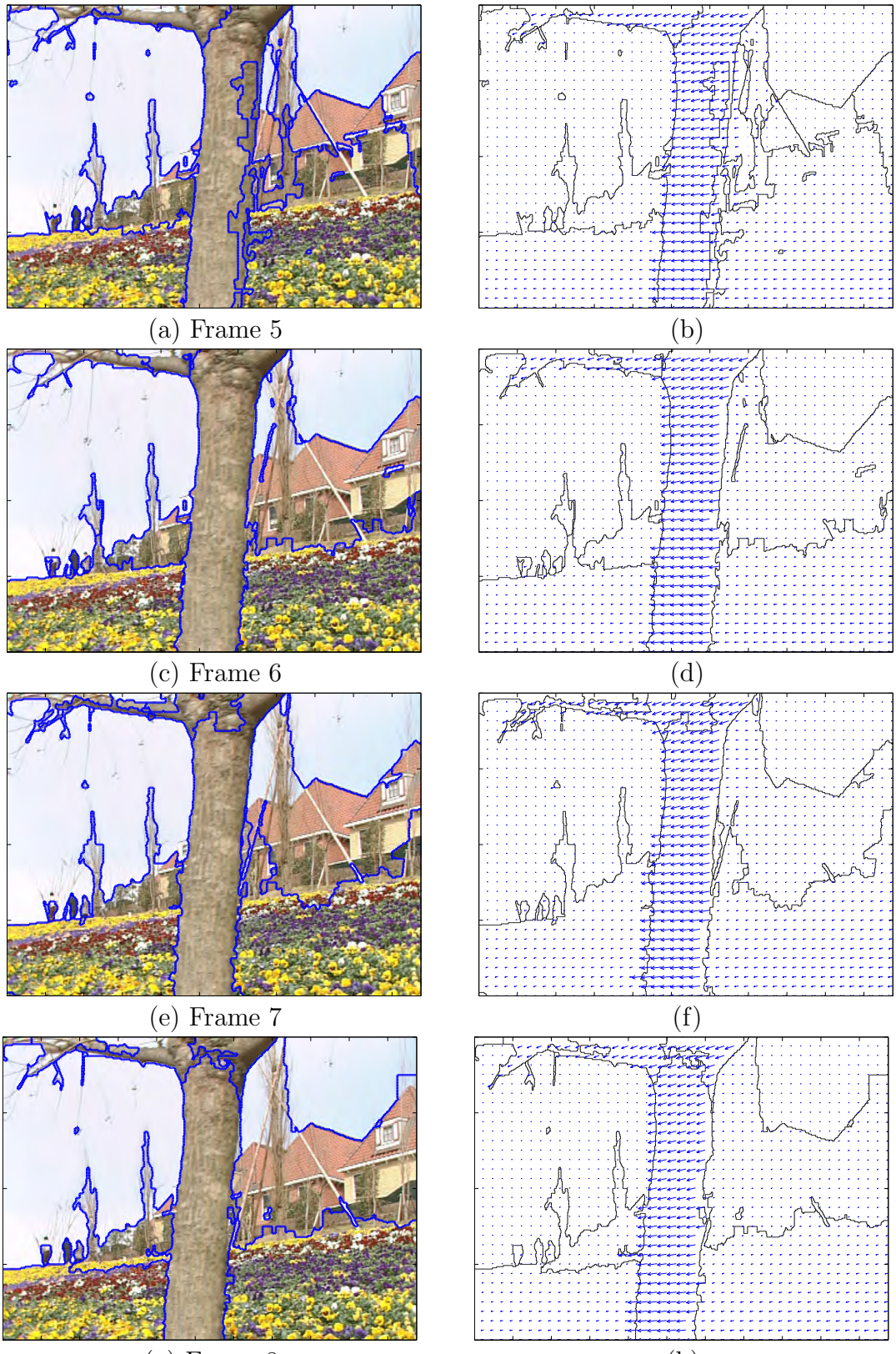
(a) Frame 1 (b)

(c) Frame 2 (d)

(e) Frame 3 (f)

(g) Frame 4 (h)

Figure 4.17: Segmentation results of "Flower Garden", frames 1 to 4



(a) Frame 5 (b)

(c) Frame 6 (d)

(e) Frame 7 (f)

(g) Frame 8 (h)

Figure 4.18: Segmentation results of "Flower Garden", frames 5 to 8

4.5.1 Comparison with a spatio-temporal segmentation approach

While there exist a number of different segmentation techniques in the literature, this proposed method offers two strengths that can be seen as addressing some critical issues in a video segmentation problem:

- A selective spatial segmentation approach, resulting from a classification of motion types using the motion confidence measure. Perceptually, the burden of spatial segmentation and subsequent region-based motion estimations is confined to the areas containing motion discontinuities, which are also where such spatial supports are most needed. The method effectively avoids the need for a *blanket* spatial segmentation of an entire frame into candidate regions, which would otherwise introduce unnecessary ambiguity into the classification process.
- A region clustering process which allows each spatial region to locate and adapt its best-fit motion model. The adaptation of both forward and backward differences into the region merging strategy improves the resilience of the method in the presence of regions with overfitting motion parameters.

For comparison purposes, an implementation of a recent spatio-temporal segmentation algorithm in [43] is presented. The method started with a set of initial spatial regions obtained via a process of non-linear spatial filtering and watershed segmentation at each frame. Motion estimation is then performed on individual regions, followed by an object labelling process carried out based on the direction of motion vectors. Figure 4.19 shows the results from the initial spatial watershed segmentation on a frame of the sequence "Mobile and Calendar".

An initialisation using a global spatial segmentation can be effective in a scene where a clear distinction exists between foreground and background in both spatial and temporal domains, such as for a frame in the sequence "Table Tennis" in Figure 4.20-d. However, in a complex scene with multiple large moving objects, relying

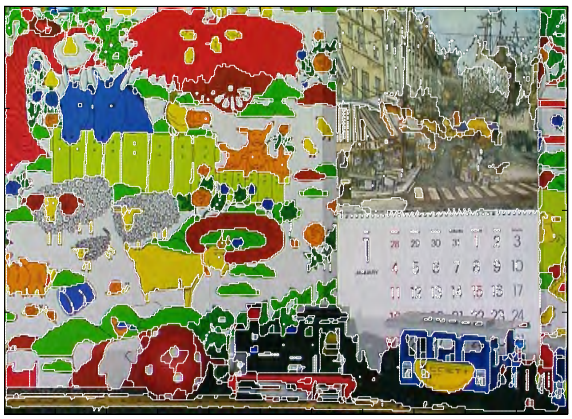
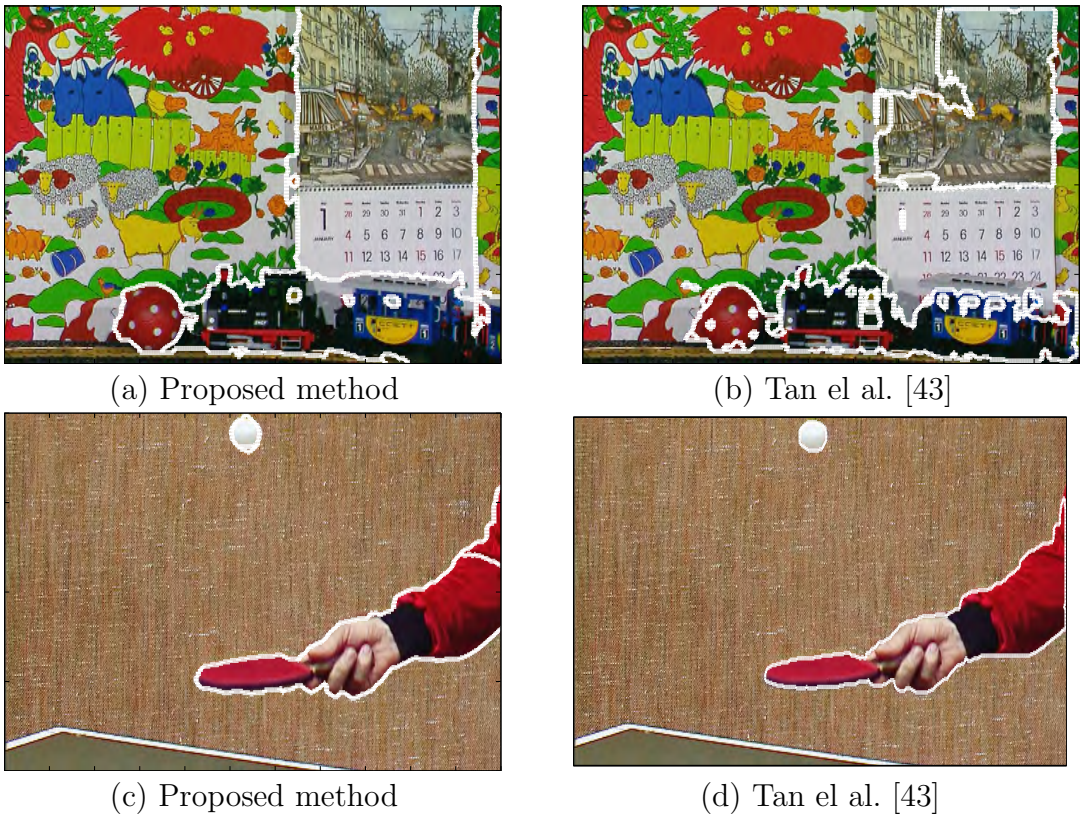


Figure 4.19: Spatial segmentation result based on non-linear filtering and watershed segmentation using Tan et al.'s algorithm [43] on frame 3 from "Mobile and Calendar"

on an initial spatial segmentation may produce irrecoverable defects on the final object masks. In Figure 4.20-b, the segmentation result from the previous scene of "Mobile and Calendar" clearly shows that the defects from a global spatial segmentation from Figure 4.19 has a follow-on effect on the final result. For example, the lower part of the calendar is incorrectly grouped with the wallpaper object and the top of the train carriage. Due partially to the effects of prefiltering prior to spatial segmentation, spatial regions are inadvertently formed over the boundaries of these objects. Because then the single-motion assumption is violated, relying on compromised motion estimation eventually leads to defects in the segmentation masks.

4.5.2 Remarks on the accuracy of the masks

As the segmentation is performed independently from frame to frame, a number of effects on the segmentation masks are observed. Many of these effects are due to the limited temporal dimension of the method, where each segmentation result only depends on three local frames. As we will see in the next chapter, they can be corrected when more frames are taken into account. Some of them, as observed on the segmentation results, are:



(a) Proposed method (b) Tan el al. [43]

(c) Proposed method (d) Tan el al. [43]

Figure 4.20: Comparison between the proposed segmentation method and the spatio-temporal approach of [43]

- Small sections of an object boundary may be incorrectly identified in some frames. This is usually due to a lack of contrast in the color space across a motion discontinuity, leading to the creation of a single spatial segment over the boundary. This is, however, expected to be momentary and can be corrected over time as the involved objects move past each other.
- Presence of over-segmented regions, due often to either inaccurate motion estimation of the region, or inadequacy in the representation of the motion model. In either case, it prevents a region from being united with the valid object that it should otherwise belong. A relatively ad-hoc solution to this problem would involve elimination of stand-alone regions less than a certain size limit, because over-segmented regions also tends to be smaller than others. Such a solution, however, may also mistakenly remove valid but small objects. Since

the appearance of an over-segmented region is usually sporadic, both spatially and temporally, a more sustainable approach would involve evaluations of the object masks over a longer sequence of frames, where non-persistent objects can be effectively filtered out and removed.

- Inconsistency of the object masks between frames: Due to the memory-less nature of the algorithm, where segmentation at each frame is performed independently of the masks obtained from the previous frames. Again, the solution to this problem would involve increasing the temporal resolution of the method in both forward and backward directions.

While a local solution may be devised to fix some of these problems, their effectiveness tends to be dependent on the spatial content of the scene. It appears that a more efficient and dynamic solution should involve inputs from multiple frames, through a temporal process which allows good aspects of each mask to be retained, while removing incorrect features at the same time.

4.6 Summary

The work in this chapter aimed to produce the masks for moving objects, based on a combination of spatial segmentation and motion-based region clustering. The process is initiated by first partitioning the area into spatially-coherent segments using a color quadtree approach, followed by a motion-based region clustering stage. The flexibility in formulation of this chapter is reflected by two expansions: a move beyond the block-based framework to a region-based algorithm, and upgrading of motion representations from a translational model in the boundary detection stage to an affine model for the purposes of both estimation and region merging. Two features which are seen as key contributions to the results of this chapter are:

- A self-expanding quadtree segmentation scheme: The spatial segmentation is only useful if the boundary of each region is not preconditioned by the block boundary. If the color segmentation encounters a “flat” block en route,

indicated by having only a single node in the quadtree at that block, then it automatically extended the segmentation to neighbouring blocks until a block with authentic texture was found. In other words, while this segmentation was spatially confined by design, it was not texture-blind.

- Region merging using a three-frame approach: Because motion estimation was carried out in only one direction, i.e. between the current and previous frames, region merging using these two frames tended to be affected by the bias of the objective function used by the estimation. For example, it is difficult to dismiss an overfitting motion on a small region as a misfit, because after all it is still optimised for the best motion-compensated difference on that region. However, inclusion of an extra picture (in this case the next frame) alleviated this prejudice in region merging. In the previous example, assuming a constant velocity over the short duration of three frames, a reversal of the overfitting backward motion is likely to yield a poor performance in the forward direction between the current and next frame, whereas an accurate motion should maintain its integrity in both directions. A three-frame approach therefore allowed a more objective consideration for fitness of a motion over neighbouring regions.

The segmentation results presented at the end of this chapter compared favourably with another technique based on an indiscriminate initial spatial segmentation. While there still exist a small number of oversegmented regions in the results, it is argued that a sustainable solution to such effects should be formulated in the context of long-term, rather than individualistic, segmentations. This will be considered within the framework of temporal stabilisation for the object masks, the topic which will be addressed in Chapter 5.

Chapter 5

Object mask stabilisation using temporal integration

5.1 Introduction

To establish the correspondences for a moving object in video segmentation, the inputs of two or more frames are usually required. The recovery of object boundaries from a scene is the result of the application of a number of constraints to the input video data. In the last chapter, these constraints were realized in a regularisation process where the estimated motions are gradually fitted into a number of affine models, which in turn characterise the movement of individual objects.

The requirement for a temporally-stable segmentation mask arises when the result is needed on a longer video sequence, rather than on some isolated and individual frames. Since the dynamics of a scene change with object movements, the quality of a local segmentation may vary as well. Based on the segmentation results obtained previously, this chapter addresses the question of how to produce a temporally-consistent segmentation in a long sequence of images.

5.2 Temporal consistency

One of the main purposes of motion estimation in a video compression technique is to exploit the temporal redundancy which exists between consecutive frames. Except at a scene cut, it is expected that the shape mask of a moving object only changes slightly across adjacent frames. However, in obtaining a segmentation mask based solely on a particular frame and its immediate neighbours, the requirement for a gradual and smooth transition of the mask from one frame to the next is neither explicitly stated nor enforced. While such requirements would certainly be redundant if the segmentation result is always perfect, in practice it is almost impossible to eliminate all segmentation errors, especially in local processing where only a few video frames are used. When the temporal smoothness of a shape mask is taken into account as an additional constraint, it not only addresses the above requirement but also works as a feedback to correct inconsistent segmentation errors. Apparently, the application of this constraint can only be implemented under multiple-frame processing.

If two-frame segmentation is viewed as a static process, a multiple-frame algorithm is usually seen as part of an *active vision* system [2], where "the introduction of an active perceiver facilitates the application of previously acquired information to relevant ensuing contexts". A range of problems present in a static segmentation can be solved reliably if a multiple-frame approach is adopted. Some examples are:

- Lack of contrast at object boundaries: A problem occurring when an object moves past another of a similar color. In this instance there is little or no variation in illumination in a spatial region over a motion discontinuity, making it impossible to recover the correct boundary using only local motion information. On the other hand, there are other pairs of frames where this distinction can be identified more readily. Under a multi-frame approach, the decision on segmentation can also be inferred from distant frames and so an ambiguous boundary may be corrected.
- Separation of objects with similar velocity: Moving objects rarely maintain a

constant velocity throughout a sequence. When a moving object slows down to a stop, it may be seen as belonging to a stationary background if viewed in a two-frame context. The same can also be said about a stationary object before it starts moving. For robust performance, an algorithm should have the ability to detect an object even when its motion is momentarily similar to its neighbours'. This can only be achieved if multiple frames are taken into account.

In [26], Meier and Ngan proposed a segmentation method based on matching edge pixels of an object model. The segmentation task at each frame is considered as a *model update* process, which uses a Hausdorff distance measure to determine a subset of the edge pixels representing the best match to an existing model in a previous frame. The process is designed to maintain the consistency of the model with regard to locations of the edge pixels. The follow-on effect of this updating, however, is a dependence on accuracy of the previous mask to produce a satisfactory result for the current frame. There is no feedback from other frames to an initial model to ensure its accuracy, because this operation is performed sequentially using motion and spatial data from frame to frame.

In a more explicit way, the constraint for temporal consistency is factored in as a contributing term in a statistically-based, maximum likelihood estimation approach in [48]. The authors formulated the segmentation under a Bayesian framework, with the probability density function being calculated from a 7-parameter feature vector. These parameters can be divided into three groups: two coordinates (x, y) for each pixel, three color components (Y, U, V), and two coordinates for the flow vector (u, v). The likelihood of any classification labelling is treated as a sum of three independent PDFs calculated from the image data. The temporal consistency is then stipulated by imposing a model on the spatial probability of each object, therefore seeking a solution on the next frame "with least change in location of segments". However, the initial set of segmentation masks is also produced from the optical flow data of two frames, making subsequent results subject to the accuracy of the initial segmentation.

In both of the aforementioned examples, the temporal process is primarily concerned with maintaining a degree of consistency for the masks, while the accuracy of each mask is dealt with at the level of two-frame segmentation. On the other hand, it is possible to simultaneously improve both the accuracy and temporal stabilisation of the shape masks in the same process. A more accurate mask naturally leads to a temporally-stable segmentation, and while a temporally-stable segmentation may not strictly correspond to an accurate shape mask at every frame, it was reported that a stable mask provided a better perceptual quality when viewing a video [80].

The post-processing, as described in this chapter, is proposed as a batch scheme which incorporates the previous segmentation results from all frames, and then inferring a decision on each individual segmentation mask. By design, it is therefore a top-down approach and not evolutionary. The diagram in Figure 5.1 illustrates the procedure for incorporating the information from k continuous sets of segmentation masks into a temporally-stable representation over this same sequence of images. Processing segmentation masks in a batch mode offers a number of advantages:

- Removing the need to perform a direct segmentation between frames that are far apart: When a segmentation is performed using two distant frames, it becomes more difficult to estimate motion accurately as an object moves further away from its original position. In addition, the area under occlusion also grows due to a large displacement, therefore affecting the segmentation quality.
- Improving stability of the shape masks universally: because the feedback is sought collectively, rather than accumulatively, from all other frames, the accuracy and stability of each mask is maintained regardless of the frame location within the sequence.

5.2.1 Mask revalidation by referencing

When considering the stability of segmentation masks over a video segment, two separate issues arise: First, the consistency of an object *presence* from frame to frame,

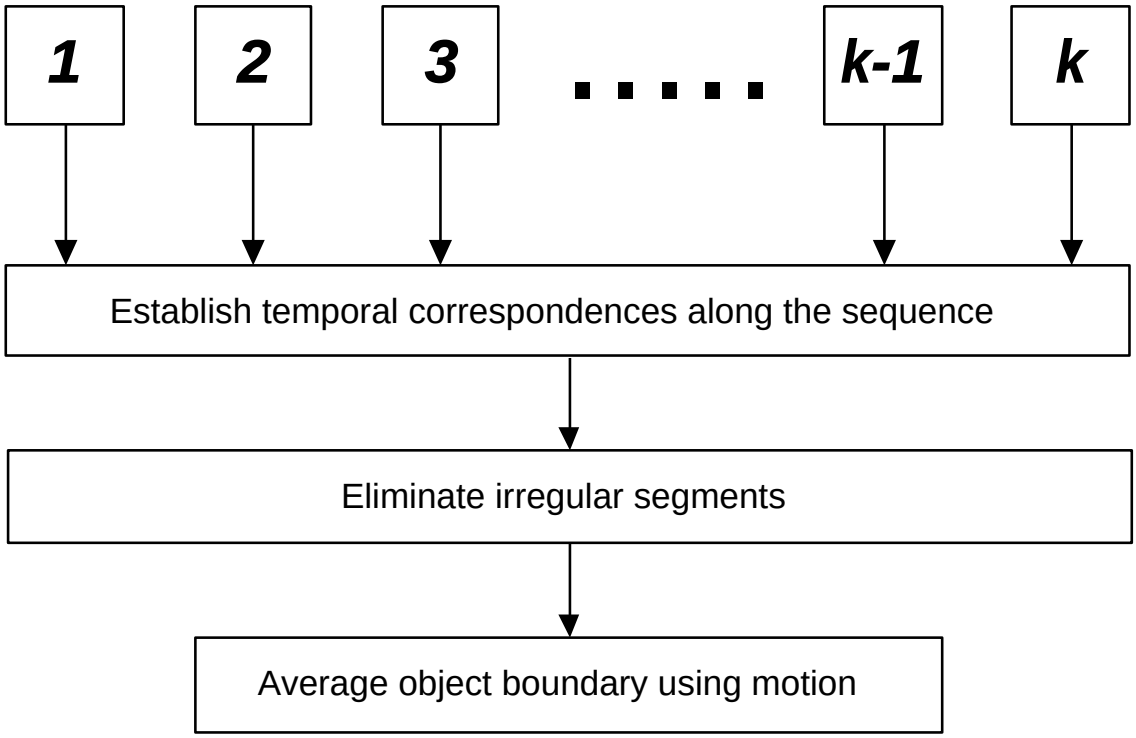


Figure 5.1: Temporal stabilisation via multiple-frame processing

and second, the consistency that the object's shape mask maintains throughout the video. The following section addresses the first issue.

From a set of segmentation masks, an important criterion that needs to be established prior to post-processing is the validity of each mask as the correspondence to an independent object, and not a defect caused by either oversegmentation or undersegmentation. In the majority of object segmentation techniques, the presence of a moving object is often decided at the inter-frame segmentation. In [51], the number of objects was preset for a region clustering stage. Similarly, a fixed number of initial regions were also used in [39]. In [42], regions under a certain size were seen as the consequence of oversegmentation, and removed according to a size limit criterion. In [40], different thresholds were imposed on different sequences to produce the segmentation.

Since an object does not usually appear or disappear abruptly in a video sequence,

the same behaviour also ought to be expected of their masks. In reality, errors in the segmentation process usually result in inconsistencies in the shape masks, especially when segmentation on each frame is carried out independently of others. A common case is the effect of oversegmentation, which may occasionally break up one object into multiple segments of smaller sizes in some frames. Furthermore, inconsistencies due to undersegmentation are also a concern. As the velocity and trajectory of an object changes, it may temporarily assume the same motion as one of its neighbours', making it difficult to rely on motion to separate two objects. It is apparent that a local segmentation (i.e. using only a small number of adjacent frames) only reflects the short-term dynamics of the local frames [81], and any decision taken at this level may not necessarily generalise to the rest of the sequence. On the other hand, temporal processing usually involves the concept of tracking, by using the segmentation result from a current frame as the initialisation, or a reference, for the next frame segmentation [26]. Sequence-based object stabilisation can be performed on a given foreground object [80], preconditioned by a consistent presence of this object throughout the image sequence. Multiple-frame processing is also carried out by treating the video data as 3D spatio-temporal volumes [82, 83], but such approaches are seen as more applicable to sequences where an initial spatial segmentation does not result in an excessive number of regions.

In the proposed mask referencing scheme, no assumption is made on the validity of each object mask at the beginning. Rather, the reasoning for temporally-consistent masks is made based on the following observations:

- When a mask corresponds to a valid object, its presence should be recognised in most of frames. On the other hand, the occurrence of an over-segmented region is usually sporadic. In the example illustrated in Figure 5.2, it is highly likely that region number 4 in each mask is the result of an oversegmentation, while the other three segments actually correspond to valid objects, since their positions are spatially consistent in all frames.
- In most cases, undersegmentation due to a lack of differentiation amongst local

object motions is a temporary problem. Undersegmentation may be "unrepairable" in principle with a two-frame segmentation approach [41], but can be corrected with multiple frames. If an object is segmented and found to be consistent at other frame locations, then its position in an undersegmentation may be recovered by inferences from those frames. Although this strategy may not work if the similarity between motion fields persists, but then again in such circumstances there would be little practical incentive to separate two objects of a persistently-similar movement.

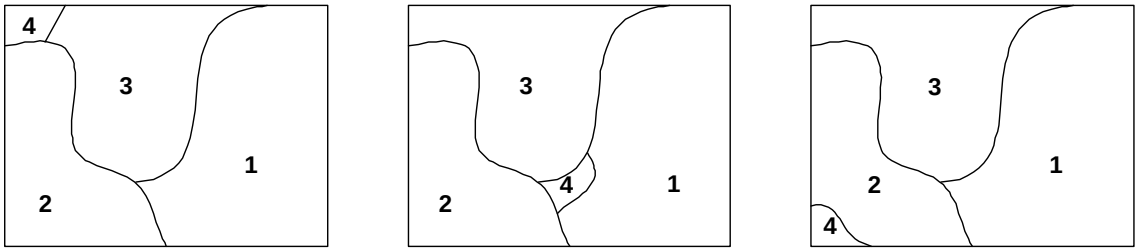


Figure 5.2: Sporadic occurrence of an oversegmented region

The objective of the referencing scheme is to filter out inconsistent segments and eliminate their existence as independent objects. It also aims to recognise those objects which are consistent and then registers their presence throughout the sequence. From the previous chapter, even though the segmentation masks have been produced on individual frames, the label assigned to each mask is localised at the current frame and the global correspondence amongst them is yet to be established. In other words, it is unknown whether two masks at two different frames correspond to the same object. The referencing scheme is also designed to register this information, the details of which will now be described.

Consider a video sequence with k frames, indexed as $1...k$. Let n be an arbitrary frame number, subject to $2 \leq n \leq k$, and the respective object label fields at frames $n-1$ and n be L_{n-1} and L_n. Each label field comprises individual object masks, for example $O_{n,i} = \{(x,y)|L_n(x,y) = i\}$, where $O_{n,i} \in L_n$ is the mask for the i^{th} object at frame n. A shape mask $O_{n,i} \in L(n)$ is accompanied by its estimated affine motion parameters, which describe the object's motion from frame n toward frame

$n-1$. The motion parameters allow the binary mask to be projected into frame $n-1$ as $O_{n-1}^{n,i}$. When this projection is superimposed on the label field L_{n-1}, it is not expected that the projected mask lines up perfectly with only one mask in this field. At frame $n-1$, a mask $O_{n-1,j} \in L_{n-1}$ is defined as a *reference* for the original $O_{n,i} \in L(n)$ if its overlapped area with $O_{n-1}^{n,i}$ is the largest of all the overlaps of this projection with other masks at this frame. Alternatively, it may be stated that $O_{n-1,j}$ is *referred to* by $O_{n,i}$. A similar procedure can also be applied to find the reference for $O_{n-1,j}$ at frame n, by using the forward estimation parameters to project the mask from frame $n-1$ to frame n. It should be noted that at this stage referencing is not commutative, i.e. "$O_{n,i}$ referencing $O_{n-1,j}$" does not necessary mean "$O_{n-1,j}$ referencing $O_{n,i}$" and vice versa.

Having defined the referencing scheme for two adjacent frames, the procedure can be extended to cover objects between any two frames in the sequence. For two arbitrary frame numbers m and n, $1 \leq m < n \leq k$, the backward and forward references are establish as follows:

Backward reference from frame n to frame m

- For each object mask $O_{n,i} \in L_n$, find its reference $O_{n-1,j} \in L_{n-1}$.
- Redefine the projection as the area it overlaps with the reference, i.e. $O_{n-1}^{n,i} = O_{n-1}^{n,i} \cap O_{n-1,j}$
- Using this projection and motion parameters associated with $O_{n-1,j}$, locate the next reference $O_{n-2,l} \in L_{n-2}$ at frame $n-2$
- Repeat these steps until reaching frame m

Forward reference from frame m to frame n

Because the motion parameters from the segmentation process are derived from a backward estimation, the procedure to establish the forward reference involves first obtaining parameters for reverse motion.

- For each object mask $O_{m,i} \in L_m$, create the forward projection $O_{m+1}^{m,i}$ using the reversed version of its affine model, and then locate its reference mask $O_{m+1,j} \in L_{m+1}$.

- Re-create the projection $O_{m+1}^{m,i}$ using the reversed motion of the reference $O_{m+1,j}$. This is because the latter model is a more accurate estimation of the forward motion, while the former is an estimation based on the assumption that the velocity of the object is constant.
- Redefine the forward projection as the area it overlaps with the reference, i.e. $O_{m+1}^{m,i} = O_{m+1}^{m,i} \cap O_{m+1,j}$
- Using this projection, repeat the above procedure to locate the reference object at frame $m + 2$
- Repeat these steps until reaching frame n

For a given shape mask at one frame location, it is a straightforward procedure to identify its references at any other frame along the sequence. However, the fact that such a mask has references elsewhere in the sequence does not necessarily strengthen its credibility as a valid, stand-alone object. A much more reliable indicator is the number of times it is *referred to* (i.e. being a reference) by masks from other frames. In this process, a mask cannot self-nominate to be a reference. Because locating a reference is a process originated from another frame, a strong group support is required for any mask which is held up as a consistent reference. This group support is likely to hold true for a properly segmented object, as its masks provide the references for one another along the sequence. In comparison, an isolated, over-segmented region in the segmentation would rarely be consistently referred to by others, although it may still have references at other frames.

An example of the frame referencing approach is illustrated in the following figures and tables. For all the examples in this chapter, only the top fields of each interlaced video are used to carry out segmentation. In this example the segmentation result is considered within a series of 30 frames. From an initial segmentation result in Figure 5.3, there exists 15 object labels in the first frame of the sequence "Mobile and Calendar", some of which are the results of oversegmentation. At this stage, the labels are assigned to objects according to their sizes, i.e. 1 corresponds to the largest object (the wallpaper) and 15 to the smallest object. Table 5.1 shows how the

objects in this first frame make references to the objects at other frames. The first column are the labels assigned to objects in frame one, and the subsequent columns show the corresponding labels for its references at other frames in the sequence.

A second table, Table 5.2, on the other hand, shows how the objects in this first frame are *referred to* by the objects in other frames. Similarly, the first column of the table indicates the labels assigned to objects in this first frame, while subsequent columns show the corresponding labels for objects at other frames which make a reference to the objects in the first frame. It is observed that some objects are referred to more often than others, and some are not referred to at all. In this table, 0 indicates a lack of refereed objects at the destination frame. When an object is not referred to by objects at other frames, then its existence as a stand-alone segment is more likely questionable. Those objects are therefore easily identified and removed by merger with one of it neighbours, with the results tabulated in Table 5.3. This process reduces the number of objects from fifteen to seven, with the updated mask then displayed in Figure 5.4.

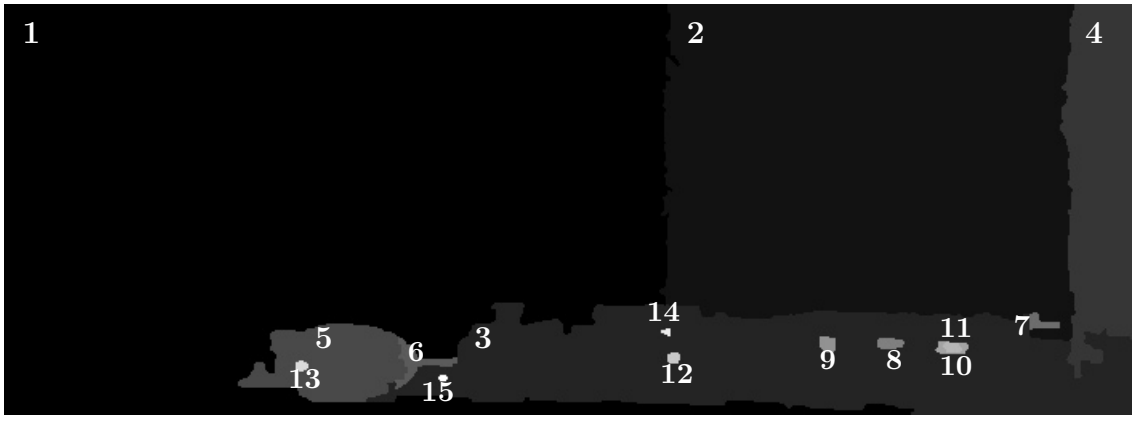


Figure 5.3: Initial segmentation mask of the first frame of "Mobile and Calendar" (15 segments)

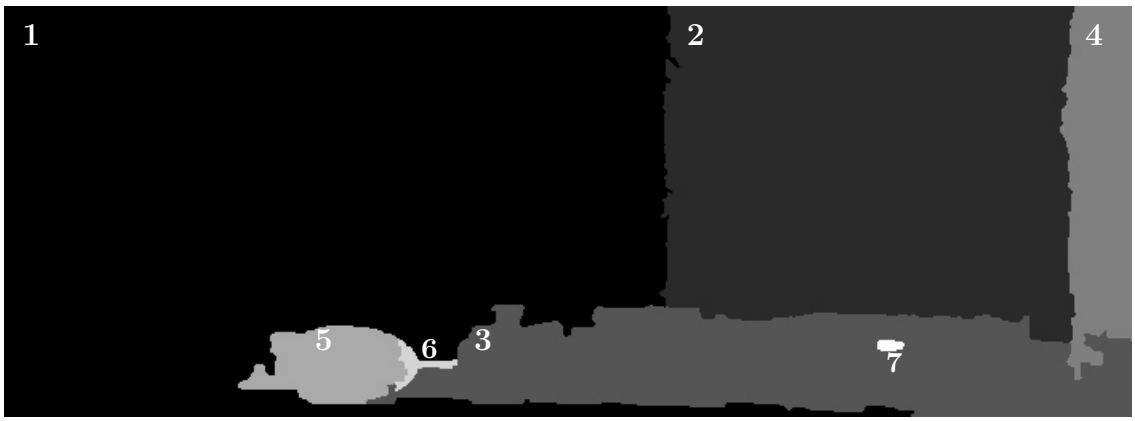


Figure 5.4: Segmentation mask of the first frame of "Mobile and Calendar" *after* the first frame referencing is performed (7 segments)

frame number																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	2	2	2	0	0	0	0	0
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4
6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4
7	7	3	8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
8	11	8	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
9	10	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	8	7	3
10	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	11	3	10	10	10	9	14	3	3	7
11	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	11	3	10	10	10	9	14	3	3	7
12	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
13	5	5	5	5	5	5	5	6	5	6	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4
14	9	7	7	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
15	3	3	3	3	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Table 5.1: References *from* object labels in the first frame *to* other frames

frame number																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	0	17	13	0	0	0	0	0
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4
6	0	0	0	0	0	16	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	6	16	14	20	0	14	13
7	7	0	8	19	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	11	8	12	13	12	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	9	0
9	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.2: References *to* object labels in the first frame *from* other frames

frame number																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	2	2	2	0	0	0	0	0
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4
6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4
7	6	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Table 5.3: References *from* object labels in the first frame *to* other frames, after the first reference checking stage

frame number																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	2	2	2	0	0	0	0	0
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4

Table 5.4: References *from* object labels in the first frame *to* other frames, after the second reference checking stage (5 segments)

An additional criterion being used to recover missing segments due to undersegmentation is to recognise the objects which are consistent and subsequently ensure their presence in the segmentation masks. As was mentioned earlier, referencing is not necessarily a commutative relationship. One of the reason is undersegmentation, where two independent objects may refer to the same under-segmented mask in another frame. The solution is to reconstruct the correct object boundaries within the under-segmented region. Through this process, when all inconsistencies are removed, the remaining objects must be kept consistent. The difficulty of course is not to lose any valid object in the process.

A segment is considered to be consistent if it is not only referred to, but such references must also be commutative over a number of frames, i.e. $O_{m,i}$ is the reference of $O_{n,j}$ while $O_{n,j}$ is also the reference of $O_{m,i}$. Furthermore, it is required that these frames are consecutive. The second requirement would then filter out most segments whose estimated motions are sporadic, as the one-to-one mapping of references is not sustainable under incorrect motions.

To resume the previous example, Table 5.4 extends the result of the consolidation process after enforcing this requirement of commutative referencing on the segmentation masks. As seen in this table and Figure 5.5, the number of objects after merging has been reduced to five. In this particular example, the five segmented objects are: 1 - the wallpaper, 2 - the calendar, 3 - the train, 4 - part of the wallpaper at the right-hand side of the frame, and 5 - the ball. Because of the panning of the camera, the partial view of the wallpaper on the right-hand side is diminishing toward the end of the sequence. This is reflected in Table 5.4 as from frame 26, there is no correspondence for this object as it disappears from the camera view.

The result from this referencing scheme can also be used to reassign labels to video objects in a consistent manner. In the initial stage, each object mask is assigned a number according to their size at that particular frame. As the size of an object may change from frame to frame, so does the label assigned to its mask. An example of which is seen in Table 5.4, where the change in the size of an object triggers a

swap of labels between object four and five from frame 18 onwards. However, after the references have been established amongst a group of object masks, the object labels can be re-assigned so that each label is associated with one specific object throughout the duration of the video.

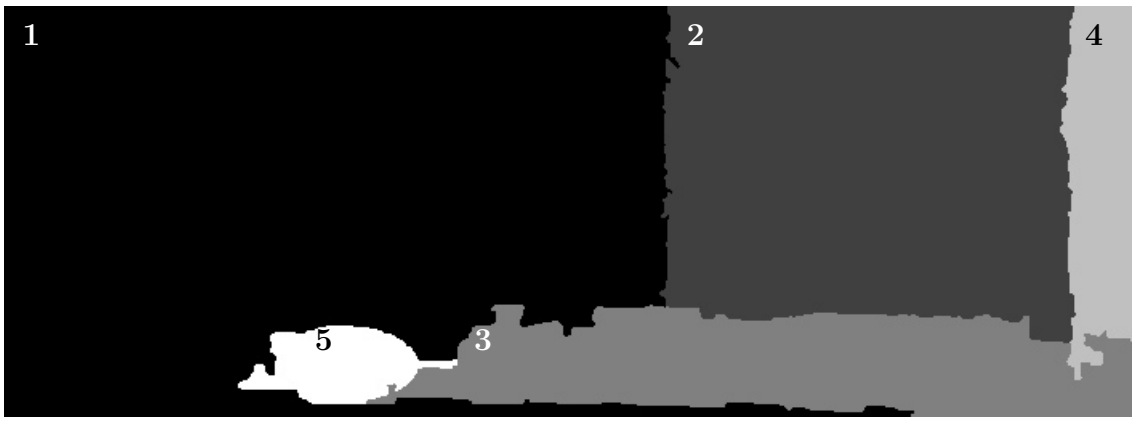


Figure 5.5: Segmentation mask of the first frame of "Mobile and Calendar" *after* the second frame referencing is performed

A merging target for a given segment is identified according to the following procedure:

- For each segment to be eliminated, identify all its immediate neighbours
- Locate all the references from this segment at other frames
- In addition, locate all the references from each neighbour.
- For each neighbour, count the number of references which are the same as the references from the segment to be merged
- Merge the segment with the neighbour with the highest count

In addition, the procedure to correct the effect of undersegmentations at individual frames is carried out in conjunction with smoothing of the object boundaries, with details being elaborated in section 5.4.3.

Finally, we should comment on the sustainability of the mask referencing scheme. As far as the end result concerns, the above procedure resembles a "clean-up" operation

which is usually performed as a post-processing step in a segmentation algorithm. Many other approaches opt to apply a more heuristic criterion to identify and remove invalid segments, such as requiring all segments falling below a certain size be merged with a larger neighbour. While such strategies may be quicker to execute, they are often inadequate to reflect the long-term dynamics of a scene, which can only be observed through a larger temporal aperture. In addition, a limit such as on the object size is often chosen as a criterion of convenience, as it confers little objectivity on the validity of an object in either spatial or temporal domain. In the mask referencing scheme, on the contrary, the validity of a segment is assessed in direct conjunction with the consistency of the estimated motion. Validation of each segment as an independent object is obtained through the strength of cross-referencing that exists between the segment itself and others in the sequence. While it is almost impossible to ensure a local segmentation as error-free, it would be reasonable to argue that as long as the majority of results are correct, a valid object mask will be retained based on its consistent presence. On the other hand, an invalid object mask is often characterised by a weak correlation to others and can be rejected accordingly. The referencing scheme may be seen alternatively as a top-down approach to directly maintain the consistency of each object's presence during the video scene.

5.3 Consistency of object boundaries

Given that the masks corresponding to each object have been identified at every frame, the next task is to refine and remove the temporal inconsistencies in their shape. These inconsistencies often manifest as defects in the segmentation masks, which are usually caused by insufficient information from the local motion, or a lack of spatial contrast across an object boundary. Both reasons can be attributed to the limited temporal aperture, i.e. the small number of frames involved in the initial segmentation.

Figure 5.6 shows an example of such a defect, where a red ball is rolling across a background image which also has patches of similar color. Even though the spatial segmentation was performed using both luminance and chrominances, it is always difficult to recover the true boundary between two moving regions when their colors are very similar. The solution to this problem is to look beyond the current frame into the rest of the sequence. By comparing an existing mask with its correspondences at other frames, it soon becomes obvious that the outlier regions are irregular and inconsistent, therefore making the case for their rejection from the shape mask.

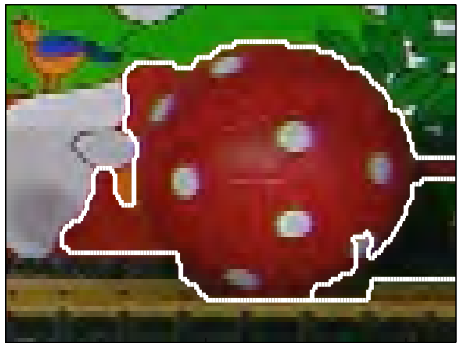


Figure 5.6: Defects in a segmentation mask caused by a lack of spatial contrast across the object boundary

The correction may be described as an averaging process, where the masks from other frames are first *registered* toward a current frame. Assuming that the motion has been estimated accurately, all the registered masks should overlap at the same object position. The consistency of a pixel in the mask may then be evaluated by the frequency of its appearance in the registered masks. It should also be noted that an initial object mask may be subject to both oversegmentation, i.e. only having partial correspondence to the object, and undersegmentation, i.e. containing pixels from neighbouring objects. The corrections therefore involve not only rejection of inconsistent pixels, but also inclusion of pixels which may not be present inside the initial mask.

While the process is straightforward in principle, implementation in this form is only suitable for a foreground object. The following issues need to be addressed for

a scene containing multiple moving objects:

- Effect of occlusion and uncover: When an object is subject to occlusion, its mask at one frame may not correspond in its entirety to the mask in another frame. Averaging on the registered masks, each of which are partially occluded, would eventually produce a shape mask which is smaller than the object itself. Likewise, if parts of an object become uncovered during its movement, the above procedure would lead to an average shape which is larger than the object.
- Estimation of object motions between distant frames: In order to accurately project the object mask from another frame into the current frame, reliable motion estimation is required. A mask projected according to a wrong motion would compromise the temporal stability of the segmented object and its neighbours. On the other hand, as the displacement of an object between two frames grows, it becomes more difficult to estimate this motion accurately. While an affine model is sufficient for estimation of small inter-frame motions, it may also become inadequate when required to cope with larger displacements.

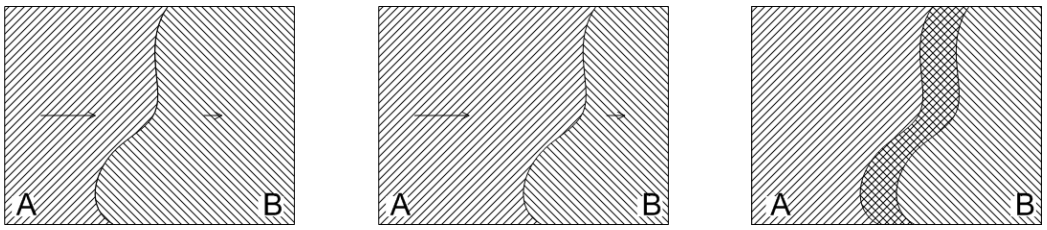
The solution to these issues are provided in the next part of this chapter. To properly deal with occlusion and uncover problems, the depth ordering is first established amongst objects. As these effects are directional (i.e. occlusion in a forward play becomes uncovered area in a backward play), they are then taken into account in setting a threshold in the set of registered masks for those pixels which are subject to either effect. In addition, the motion model used for registrations of the object mask is also extended from a 6-parameter affine model to a 12-parameter representation.

5.3.1 Depth ordering

The depth ordering between adjacent moving objects is established using the concept of the *ownership* of a boundary region, as described in [41, 84]. When the shape masks of two moving objects overlap in a two-dimensional video frame, the boundary

separating them is dictated by the occluding i.e. the one closer to the camera. An effort to smooth the boundary of the shape masks is therefore essentially linked to the foreground object at each boundary location.

Suppose that the object masks are already obtained at both frames m and $m+k$ ($k > 0$), then a simple procedure can be implemented to detect which object is in front of the other. The examples in Figures 5.7-a and 5.7-b show the masks for two adjacent objects A and B at frames m and $m+k$ respectively. Using the estimated motions for objects A and B, from frame $m+k$, both objects can be registered toward frame m, and the result of this registration is shown in Figure 5.7-(c). The double-crossed area indicates the overlapping area between the two registered masks. Since this area is associated with the occluding region, it is under the *ownership* of the foreground object. By simply considering whether the texture under this overlapped area belongs to object A or B at frame $m+k$, the occluding object can be determined. In this example, A is the foreground object as its mask contains the overlapping areas.



(a) Frame m (b) Frame $m+k$ (c) Registered objects at frame $m+k$

Figure 5.7: Ownership of an occluded region

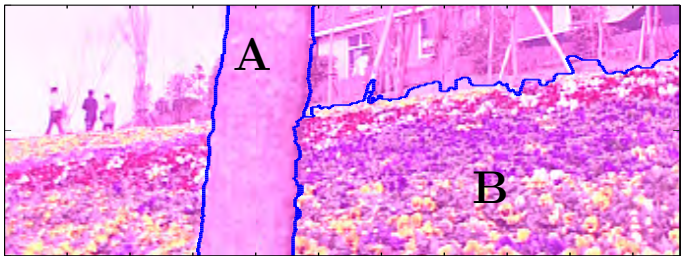
For the registered versions of the adjacent masks to overlap using this approach, it is assumed that occlusion takes place along the direction of motion from frame m to frame $m+k$. In case of an uncovered background, there will be an unfilled gap in the frame between two registered masks instead of an overlapping zone. It is useful to observe that uncover in one temporal direction is occlusion in the opposite direction. For any two spatially-adjacent objects A and B, both of which are present in frames 1 to N, the following scheme is used to decided which object is closer to

the viewer:

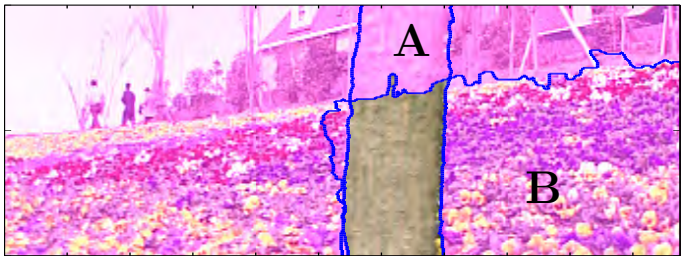
- Register all the masks for both object A and B toward the first frame 1
- Register all the masks for both object A and B toward the last frame N
- At both frames, mean-threshold each set of registered masks to produce an averaged mask for each object.
- Select the frame location where the two averaged masks overlap. From the motion parameters for both objects A and B, identify the model which produces the smaller residual difference on the overlapped region. The object associated with this motion model is then considered as foreground to the other.

The example in Figure 5.8 gives an illustration of this process for two objects, over a series of 30 frames in the sequence "Flower Garden". In this example, under the camera movement, the tree moves to the left at a faster speed than the flower bed and uncovers parts of the flower bed along the sequence. This is also reflected by the clear separation between the averaged masks in the last frame in Figure 5.8-c. On the other hand, if the series of frames is watched in the reversed order, than the tree may be seen as occluding the right hand side of the flower bed. If the object masks are all registered toward the first frame, then two averaged masks will overlap, as shown in the non-pink area of Figure 5.8-b.

Depth ordering is established by testing for fitness of the motion parameters of both objects over this overlapped region. The sum-of-squared difference using the motion parameters of object A (tree) is 127, and the corresponding result using the motion parameters of object B (flower bed) is 480. Consequently, the motion parameters of object A are seen as providing a better fitting model for the overlapped region than the parameter for object B. The overlapped region therefore can be considered as being under the *ownership* of object A, which is equivalent to saying that object A is closer to the camera than object B.

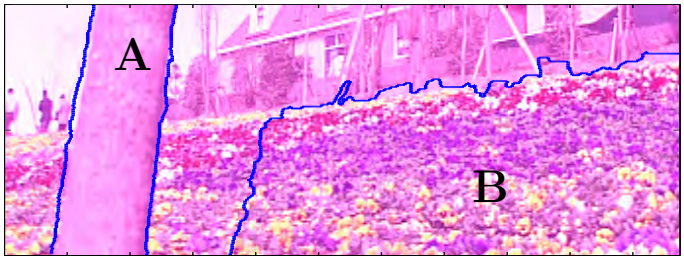


(a) Frame 15



(b) Frame 1

$SSE_A(A \cap B) = 127$; $SSE_B(A \cap B) = 480$



(c) Frame 30

Figure 5.8: (a) - The object masks at the frame 15; (b) and (c) - The averaged results for the registered masks at different frames. Along the direction of occlusion, the averaged masks are overlapping in (b)

5.3.2 Occlusion and uncover regions

For a foreground object, the registered mask is not affected by occlusion or uncover problems, and ideally it would occupy all the pixel locations in the original object. In practice, most individual segmentations contain some types of errors, and the accuracy of a registered mask is also affected by the accuracy of the estimated motion for the object under this mask. However, if the local (three-frame) segmentation approach has been successful in assigning each pixel location to its correct object mask in the *majority* of video frames, then correction to an individual mask can be obtained by mean-thresholding on the set of registered masks. The consistency

criterion can be established for a pixel location if it is present in the majority, i.e. more than half the number, of the registered masks. Other pixel locations, which are absent in the the majority of these masks, should then be rejected as inconsistent pixels.

If a scene is composed of objects which are strictly categorised as either foreground or background, then a smoothing process performed on the foreground masks will naturally leave the background mask as smooth as well. In such cases, only a single threshold value is required to determine the consistency of a pixel inside a foreground object mask. However, in a generic case with multiple moving objects, an object can be a foreground object for one object, but at the same time be partially occluded by another. As a consequence, a pixel may be absent from a registered mask due to occlusion. During the process to determine the consistency of a mask pixel, an absence due to occlusion should not be counted against the validity of the mask at that pixel location. In other words, the threshold used for the consistency check should be adapted at each pixel location, to reflect the number of times that pixel is occluded by other masks in the sequence.

A consistent pixel is re-defined as one which is present in the majority of registered masks, excluding those masks where it is not present due to occlusion by a neighbouring object. Considering an object mask at one temporal location during a sequence of N frames; if a pixel p in the mask is subject to occlusion in $N_O(p)$ other frames in this sequence, then the consistency-check for this pixel will be set at $\frac{N-N_O(p)}{2}$, instead of $\frac{N}{2}$.

To enable an adaptive setting of the above threshold, the value $N_O(p)$ needs to be calculated for every pixel. If the depth ordering between adjacent objects is known, and the object motion is accurately estimated along the sequence, the number of occlusions for each pixel may be calculated as follows:

- For each object O in a current frame, identify all objects which are adjacent and *foreground* to this object.
- From other frames in the sequence, register each of the foreground masks

toward the current frame.

- Identify pixels on the current frame which are overlapped by both the mask for object O, and the registered masks from other foreground objects. For each pixel p in this region, the number of times it is positioned inside the registered mask of a foreground object also represents the number of times it is occluded, or $N_O(p)$

At each frame, it would become possible to build a per-pixel map, which indicates the number of times a pixel is subject to occlusion at other frames in the sequence. Based on this information, the consistency check can then be carried out without the assumption that the object must always be in a foreground region.

An important issue rising from the use of multiple frames to contribute to the segmentation of a local frame is how to maintain the accuracy of the result, given that the object displacement tends to grow larger as the distance between frames increases. Because the validity of a local object mask is also verified by the consistency of other motion-compensated masks in the sequence, it is vital that the motion parameters relating a distant object mask to the current frame is correct, so that it does not compromise the result of the consistency check. In particular, the consistency check has relied on the availability of the motion parameters between any pair of frames, which involves a relatively large number of calculations if all the estimation are performed directly. When an affine motion model is assumed between adjacent frames, it is possible to calculate the motion parameters between any two frames by model multiplication. However, the estimation quality is expected to degrade when the affine model becomes inadequate to represent the object motion. In the next section, the accuracy and appropriateness of the affine model is examined under the presence of larger object motions, and a strategy is adopted to reduce the number of estimations while still maintaining the ability to register an object mask between any two frames in the sequence.

5.4 Object motion

With individual segmentation masks produced from three frames (previous, current and next), an affine motion model has been assumed for the estimation process. The primary advantages of an affine model are:

- It provides a means to model non-translational object motions. Some earlier work, for example [85], argued that an affine motion model provides the best trade-off between accuracy, computation and conciseness of representation.
- Small movements of an object can be approximated by affine motions. In fact, the reliability of a task such as tracking depends on objects having a small interframe displacement [70].
- Affine motions are multiplicable and reversible, allowing bidirectional estimation and accumulation of results

Amongst these advantages, the consideration for the trade-off between factors such as computational requirement and accuracy would indeed be relevant when the result is also affected by the choice of an estimation window, i.e. without any prior knowledge of segmentation. When an estimation window is located at the wrong location, such as across a motion boundary, the accuracy of a more complicated model may be compromised as it attempts to overfit the parameters to the actual discontinuity in the motion field. However when the shape of an object is already known and the motion is estimated on the basis of its mask rather than an arbitrary window, then it becomes feasible to consider the suitability of a higher order motion model for the objects. As illustrated by examples later in this section, an affine model may not cope as well when a large rigid motion is involved, especially if the motion trajectory or the object curvature is on a non-planar surface.

As mentioned earlier in the previous chapter, the integrity of an affine model is subject to two assumptions; an affine camera projection model and a planar constraint on the surface of moving objects. The first condition depends on a linear projection of the object motion trajectory onto the camera, which can usually be assumed when

the distance between the camera and an object is much more significant than the object size, or the magnitudes of the object and/or camera movements are small. The second condition requires that the surface of an object coincides with a plane (first-order surface). While this condition is harder to satisfy, its effect depends on the magnitude of the motion and therefore can also often be ignored when the interframe object motions is small. For two frames positioned far apart in a sequence, however, it becomes increasingly likely that either or both assumptions may not be met, which in turn affects the estimation quality.

5.4.1 Motion model: quadratic vs. affine

Within the context of rigid 3-D motions, a perspective projection motion model allows more flexibility in accounting for object motions in a scene than an affine model. While the model is still limited in the sense that it is not meant for non-rigid motions, the most significant advantage is its ability to deal with motion of both planar and non-planar parametric surfaces, which is also very common in video objects.

The following analysis is based on the work by Longuet-Higgins and Prazdny [86], which provides a brief insight into the estimation under a projective model. The first conclusion from their work, given a static scene and a moving observer whose view is modelled as being perspective, argued that the scene motion perceived on the retina can be decomposed into a translational and a rotational component. Interestingly, the rotational component is *independent* of the scene geometry. For a point at an arbitrary location (X, Y, Z) on the surface of an object, the rotational component (u_r, v_r) of its instantaneous velocity is modelled as follows:

$$u_r = -B + Cy + Axy - Bx^2 \tag{5.1}$$

$$v_r = \quad A - Cx + Ay^2 - Bxy \tag{5.2}$$

where (A, B, C) represents the angular rotations of the rigid motion in the Cartesian coordinate system, and $(x, y) = (\frac{X}{Z}, \frac{Y}{Z})$. The translational component (u_t, v_t), on

the other hand, is also found to be *independent* of the angles of rotation:

$$u_t = \frac{(-U + xW)}{Z} \tag{5.3}$$

$$v_t = \frac{(-V + yW)}{Z} \tag{5.4}$$

where (U, V, W) are the translations of the rigid motion. The total velocity is then equal to the vector sum of the translational and rotational components:

$$u = \frac{(-U + xW)}{Z} - B + Cy + Axy - Bx^2 \tag{5.5}$$

$$v = \frac{(-V + yW)}{Z} + A - Cx + Ay^2 - Bxy \tag{5.6}$$

Furthermore, assume that the surface of the object can be described by a parametric means as in the following constraint:

$$Z = d + \alpha X + \beta Y + \sum_{n=2}^{\infty} O_n(X, Y) \tag{5.7}$$

where the last term on the right-hand side indicates the non-linear components of the surface. By setting $(u_0, v_0, w_0) = (U, V, W)/R$ and $z = (Z - d)/d$, the velocity equation can then be simplified to:

$$u = (-u_0 + xw_0)(1 - z) - B + Cy + Axy - Bx^2 \tag{5.8}$$

$$v = (-v_0 + yw_0)(1 - z) + A - Cx + Ay^2 - Bxy \tag{5.9}$$

The presence of the quadratic and other non-linear terms is contributed to by both the curvature of the object surface and rotation as seen in these functions. The number of unknowns in the right-hand side of equations (5.8) and (5.9) is apparently dependent on the complexity of the object surface in (5.7), with the presumption that the surface can be properly modelled by such a geometric means. Besides the unknown object geometry, solving for all parameters is further complicated by the inseparability of the translational terms (u_0, v_0, w_0) and coefficients of the polynomials O_n.

To provide more stable boundaries for a set of two-dimensional masks, however, does not necessarily require a full recovery of all these parameters. The objective of

the registration process is limited to relating the *projections* of moving objects, not their complete geometric structure. To this end, the following generic 12-parameter model is used to represent the motion of an object between two frames k and $k-n$.

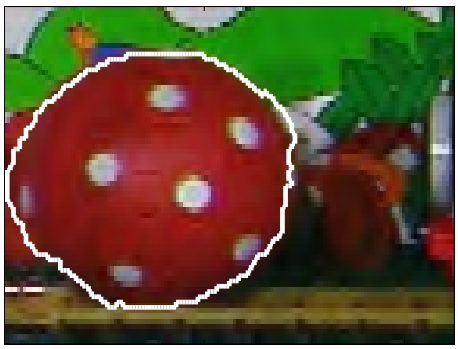
$$\begin{bmatrix} x_{k-n} \\ y_{k-n} \end{bmatrix} = \begin{bmatrix} a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} x_k^2 \\ x_k y_k \\ y_k^2 \end{bmatrix} + \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \quad (5.10)$$

Figures 5.9, 5.10 and 5.11 show some examples of how a 12-parameter model can offer a better fitting motion for objects with a non-planar surface. In those figures, (a) shows the object and its mask at a current frame; (b) shows a distant frame in the sequence, usually less than 10 frames away; (c) and (e) show the reconstructed versions of the object inside the mask at the current frame using a 12-parameter and 6-parameter motion model respectively, based on the data from the other frame; (d) and (f) show the result of registering the binary mask of the object on the current frame toward the other frame, also using the 12-parameter and 6-parameter motion model respectively.

In all three cases, the estimation using a 12-parameter motion model results in a significant reduction in the sum-of-squared differences compared to the affine model. This reduction is also attributed in part to the initialisation of the parameters using the 6-parameter model. A more important improvement from this parametric model, however, is the accuracy that it provides when registering an object mask towards a destination frame. As depicted in figures (d) and (f) respectively, registering a mask using the 12-parameter model produces a more accurately-aligned outline of an object than a 6-parameter model, because the former allows for the non-linear components to be reflected in the estimation.

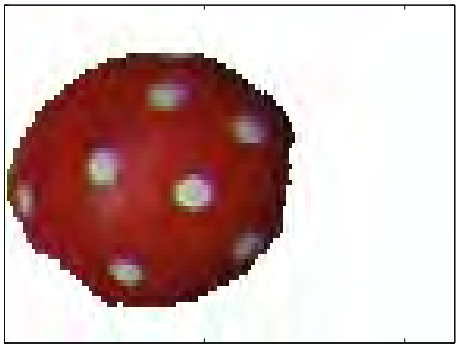
5.4.2 Accumulation of motions

As demonstrated in the previous section, a 12-parameter motion model may improve the quality of a registered object mask. However, the manipulation of such a model is subject to a number of inflexibilities which are not present in a linear model such as



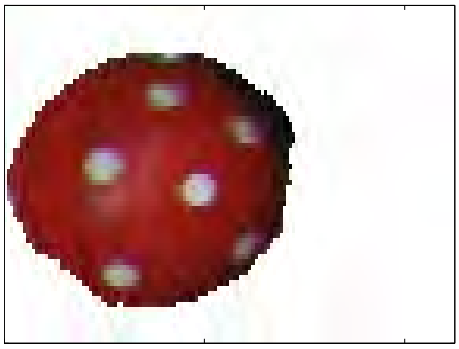
(a) Frame 15

(b) Frame 5



(c) $SSD_{12param} = 281.4$

(d) Registered mask with 12-parameter



(e) $SSD_{6param} = 458.8$

(f) Registered mask with 6-parameter

Figure 5.9: Estimating the motion of the "Ball" object. (a) - Object with its segmentation outline at reference frame 15; (b) - Object at frame 5; (c) and (e) - Object being warped toward the reference frame using the 6-parameter and 12-parameter parametric motion models, respectively, together with the residual mean-squared errors; (d) and (f) - The outline of the segmentation mask after being warped from frame 15 toward frame 5

(a) Frame 15

(b) Frame 5

(c)$SSD_{12param} = 139.7$

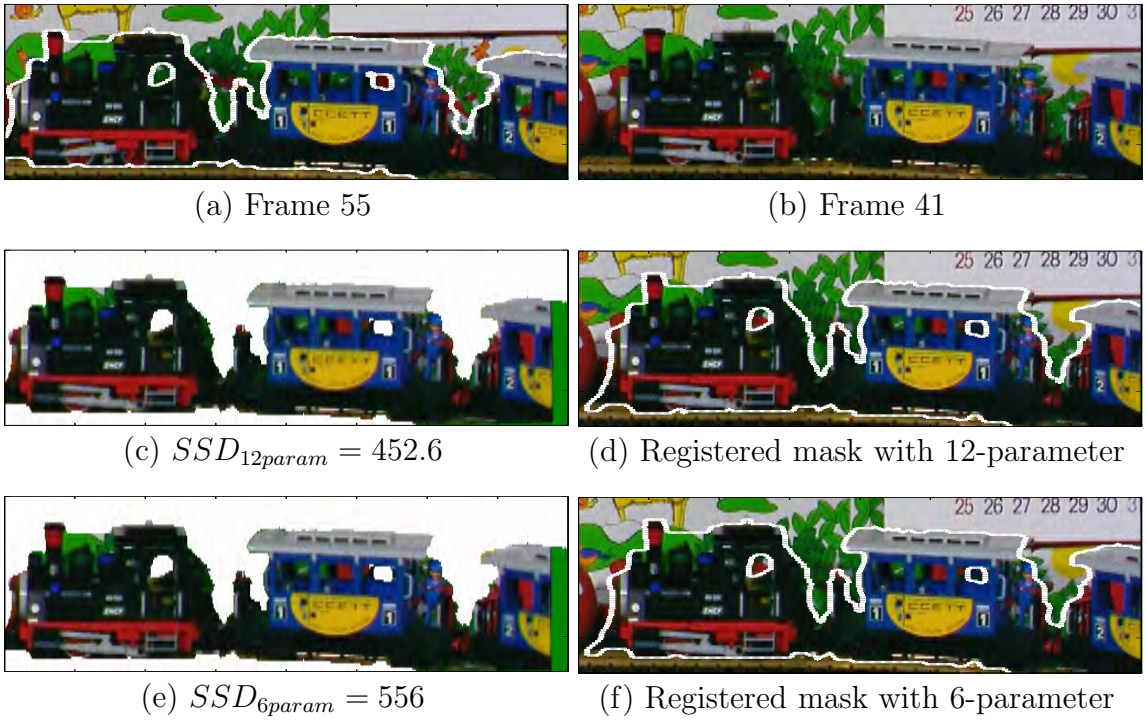
(d) Registered mask with 12-parameter

(e) $SSD_{6param} = 215$

(f) Registered mask with 6-parameter

Figure 5.10: Estimating the motion of the "Tree" object between frame 15 and frame 5. Displacements are approximately 10pixels/frame horizontally and > 1 pixel/frame vertically

affine or translational. More specifically, a 12-parameter model is not fully reversible to produce an estimation in the opposite direction, nor is it possible to accumulate consecutive motions without changing the order of the model itself. While all rigid motions are apparently reversible, and any group of them can be concatenated into



(a) Frame 55 (b) Frame 41

(c) $SSD_{12param} = 452.6$ (d) Registered mask with 12-parameter

(e) $SSD_{6param} = 556$ (f) Registered mask with 6-parameter

Figure 5.11: Estimating the motion of the "Train" object between frame 55 and frame 41

another rigid motion, the inseparability between two components, namely the scene geometry and the motion parameters, makes it difficult to perform such task without a recovery of the individual components.

The boundary smoothing process requires an ability to register an object mask at each frame to any other frame in the sequence. As the motion model is not accumulative across adjacent frames, it would mean a large number of calculation are required if the motion is to be estimated directly between any pairs of frames. For example, a sequence with 30 frames would require:

$$C_{30}^{2} = \frac{30 \times 29}{2} = 435 \; estimations$$

for each object in either forward or backward direction, and twice as much (870) for estimations in both directions (due to the irreversibility of the model).

To reduce the number of calculations, the middle frame in a sequence is selected as a reference, and the motions are estimated for each object between this reference

and any other frame in the series. A two-step procedure is then followed to register the binary mask of an object from one frame to another in the sequence:

- Create an intermediate mask at the reference frame, by registering the mask from the first frame toward this reference frame
- Register the intermediate mask again toward the second frame

By using a reference point, the calculation required for each object is reduced to

$$29 * 2 = 58 \; estimations$$

for both forward and backward registration of its mask. This is equivalent to $1/15^{th}$ the number of calculations that would have been required for direct estimations between every pair of frames. In addition, if the motion parameters need to be re-estimated due to an updated shape, only an estimation between the current and reference frames is required, and not between the current and every other frame.

(a) Frame 30 (a) Frame 15 (ref) (a) Frame 1

Figure 5.12: Registering a mask through a reference frame. (a) Object mask at the original frame; (b) first registration toward the reference frame; (c) second registration toward the destination frame

Figure 5.12 shows the registration of an object mask over a 30-frame distance, with the middle frame being used as the reference point. The registered masks can be

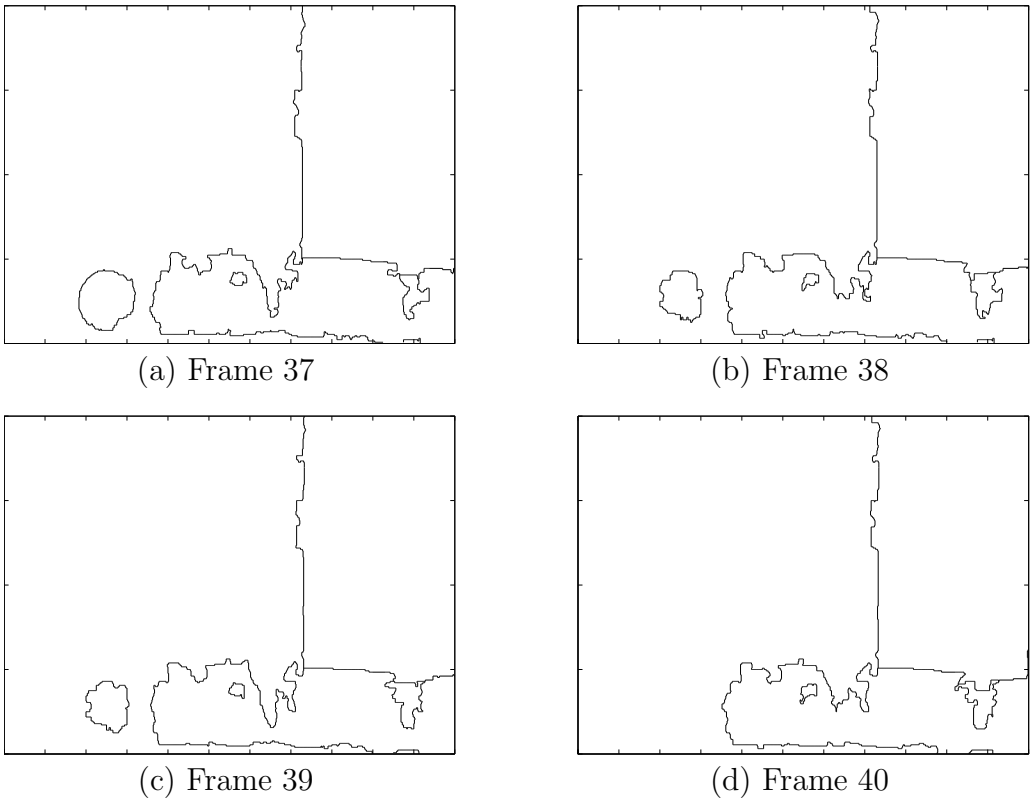
seen as matching very well to the object outline at both the reference and destination frame.

5.4.3 Recovery from undersegmentation

In this algorithm, since each object is separated from its neighbours based on the motion, segmentation becomes more difficult when adjacent objects assume the same velocity in a 2-D representation. Under this circumstance, the local motion estimation may still be correct, but by itself is not sufficient to justify the difference amongst objects.

In practice, independent objects are more likely to move with different velocities. It may still occur that due to variations in the movement of each object, the difference in motion fields between two objects may diminish momentarily. An example to consider is a car coming to a stop at a traffic light, only to continue to move after the light turns green. If observations are made continuously before and after the changing lights, then it would become obvious that the vehicle is moving and the stop is only temporary. However, if observations of the car are made only during the red light, then it would be seen as stationary as other fixed features on the road, hence the local motion information on this part of the video would not provide enough information to obtain a motion segmentation mask for the car.

Figure 5.13 shows an example where the ball object is slowing down to a stop, which results in it being grouped together with the wallpaper object in the last frame as both of them are subject only to the camera motion. It is seen from this example that when the difference between motions becomes too small, the quality of a segmentation using local motion tends to degrade for the objects involved. As compared to a spatial undersegmentation due to a lack of spatial contrast across an object boundary, the above degradation can be considered as due to motion undersegmentation, and have to be corrected to maintain a consistent temporal quality of the mask.



(a) Frame 37 (b) Frame 38

(c) Frame 39 (d) Frame 40

Figure 5.13: Degradation in quality of a local segmentation due to the convergence of local object motions

As described earlier in 5.2.1, the presence of each object in a sequence is validated by a referencing process. The results from this process are used to rationalise the likelihood of undersegmentation in a sequence. When an object is subject to undersegmentation at one frame due to the similarity of local motions, there is a lack of one-to-one correspondences to and from this frame for the unaffected object mask from other frames. Upon such an event occurring, the missing shape mask can be reconstructed from the detected masks in the sequence by using the same averaging procedure as was used in the temporal stabilisation.

5.5 Objective measures of temporal stability

Two measures are adopted to demonstrate the improvements of the temporal stabilisation on the object masks. The first measure, introduced by Arkin et al. [87], provides a metric for comparison of two arbitrary polygonal shapes, using a distance measure on the turn function (also known as the cumulative angular function). The second measure was introduced by Erdem et al. [88] and is based on the result of a chi-squared test on the color histograms of an object between two frames.

5.5.1 Turn function and the shape-similarity measure

The turn function [87] provides a representation for a shape. From a reference point chosen on its boundary and following the edges counterclockwise, the value of the function at a point along the boundary is defined as the angle between the tangent to the object at that point and the axis x. The overall length of the boundary is normalised to unity prior to calculation so that all objects are represented within the interval $[0, 1]$ along the x axis.

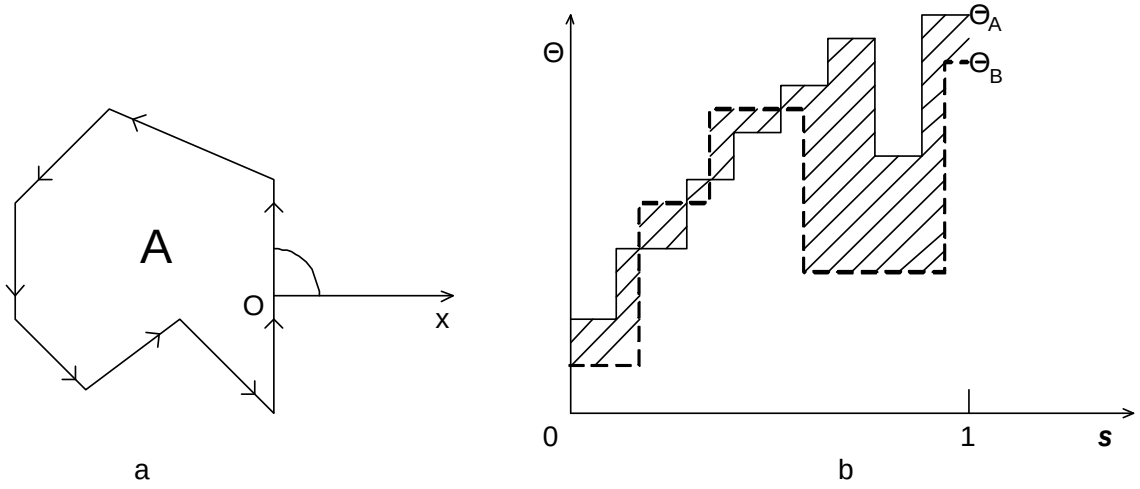


Figure 5.14: (a) A polygon A and its turn function $\Theta(A)$. (b) The shaded area is equivalent to the distance measure between two turn functions $\Theta(A)$ and $\Theta(B)$

For a polygon A as in Figure 5.14, the turn function Θ_A is a series of single-value

steps which are the angles between polygon edges and the reference axis, with the rises and falls coinciding with its vertices. A rise corresponds to a left turn on the boundary, while a fall is due to a right turn. A shift t of the origin O along the boundary corresponds to a horizontal shift in the turn function, and a rotation θ of the shape causes the function to shift vertically. To compare two shapes A and B, a distance measure is formulated as a minimisation problem on the difference between respective turn functions with the variable t and θ:

$$d_p(A,B) = \underset{t\in[0,1]}{min}\left(\underset{\theta\in[0,2\pi]}{min}\left(\int_0^1 (\Theta_A(s+t) - \Theta_B(s) + \theta)^p ds\right)\right) \tag{5.11}$$

For the case of $p = 2$ and a fixed t, it was shown that $d_2(A, B)$ is convex and θ can be solved for analytically [87].

To compare two segmentation masks, each mask is first converted into a polygon by treating each edge pixel as a vertex, using 8-connectedness amongst these pixels. The distance measure between two shapes is then calculated as (5.11), with the origin selected as the top left point in each mask.

5.5.2 Chi-squared test on color histograms

The chi-squared test is a statistical goodness-of-fit test designed to verify if two sets of binned data come from the same distribution [89]. Given that the colors of an object do not usually change abruptly, it follows that their distributions should be approximately the same if the segmentation is performed properly. It was proposed that this test can be used to assess the quality of a set of segmentation masks [88]. Specifically, the χ^2 distance measure between two color histograms H_1 and H_2, each with n_1 and n_2 samples respectively, and arranged in B bins, was expressed as:

$$\chi^2(H_1,H_2) = \sum_{i=1}^{B} \frac{\left(\sqrt{\frac{n_2}{n_1}}H_1(i) - \sqrt{\frac{n_1}{n_2}}H_2(i)\right)^2}{H_1(i)+H_2(i)} \tag{5.12}$$

The results from the two tests, performed before and after the temporal stabilisation, are elaborated in the following section.

5.6 Results

5.6.1 Segmentation masks after stabilisation

Temporal stabilisation is carried out in groups of frames, each consisting of 30 consecutive frames. The results presented here demonstrate the smoothing effects that the stabilisation brings to the segmentation.

Figures 5.15, 5.17 and 5.19 are the unprocessed masks which are used as inputs for the algorithm. These masks are taken directly from the end of chapter 4 for two sequences "Mobile and Calendar" and "Flower Garden", and they contain a numbers of undesirable defect. For example spatial undersegmentation is affecting the shape of the ball (Figure 5.15, frames $14-19$), and the tree (Figure 5.19, frames $21-22$). Effects of motion undersegmentation can be seen in Figure 5.19, when the calendar stops moving (frames $49-50$), and the ball is undetected for the same reason (frames $40-46$). There are also effects of oversegmentation observed throughout the masks, as motion estimations on some smaller segments are not accurate.

The respective results after performing stabilisation are presented in Figures 5.16, 5.18 and 5.20. Besides the removal of oversegmentations, they show significant improvements both in terms of the (semantic) accuracy of the masks, as well as their consistency when progressing from one frame to the next. In Figure 5.16, the previous defects on the ball object have all been eliminated, and the tree object in Figure 5.20 has also been dramatically improved.

The result is not completely free of all errors, but most remaining errors are peculiar to a scene. In Figure 5.16, the algorithm does not reject a strip between the ball and the train as undersegmentation, because the area is dark in color and also subject to the heavy shadow of the ball, hence there are few features to set it apart. When the area becomes better lit, as in Figure 5.18, then its features provide for better estimation and segmentation results. Another example is a little bump on the left side of the calendar in Figure 5.16, which is due to a low contrast level in this area

persisting for almost all frames (while the right side is almost error-free). When the spatial contrast improves then such effects are also removed - as seen in Figure 5.18.

Since the stabilisation algorithm bases its decision to keep or remove a segment on motion consistency and not the segment size, it does not reject objects of small sizes as long as the motion can be detected consistently. As seen in the first sequence, the small window on the train is reproduced in the segmentation masks. On the second sequence, the mask associated with the people walking near the flower field are also detected and retained.

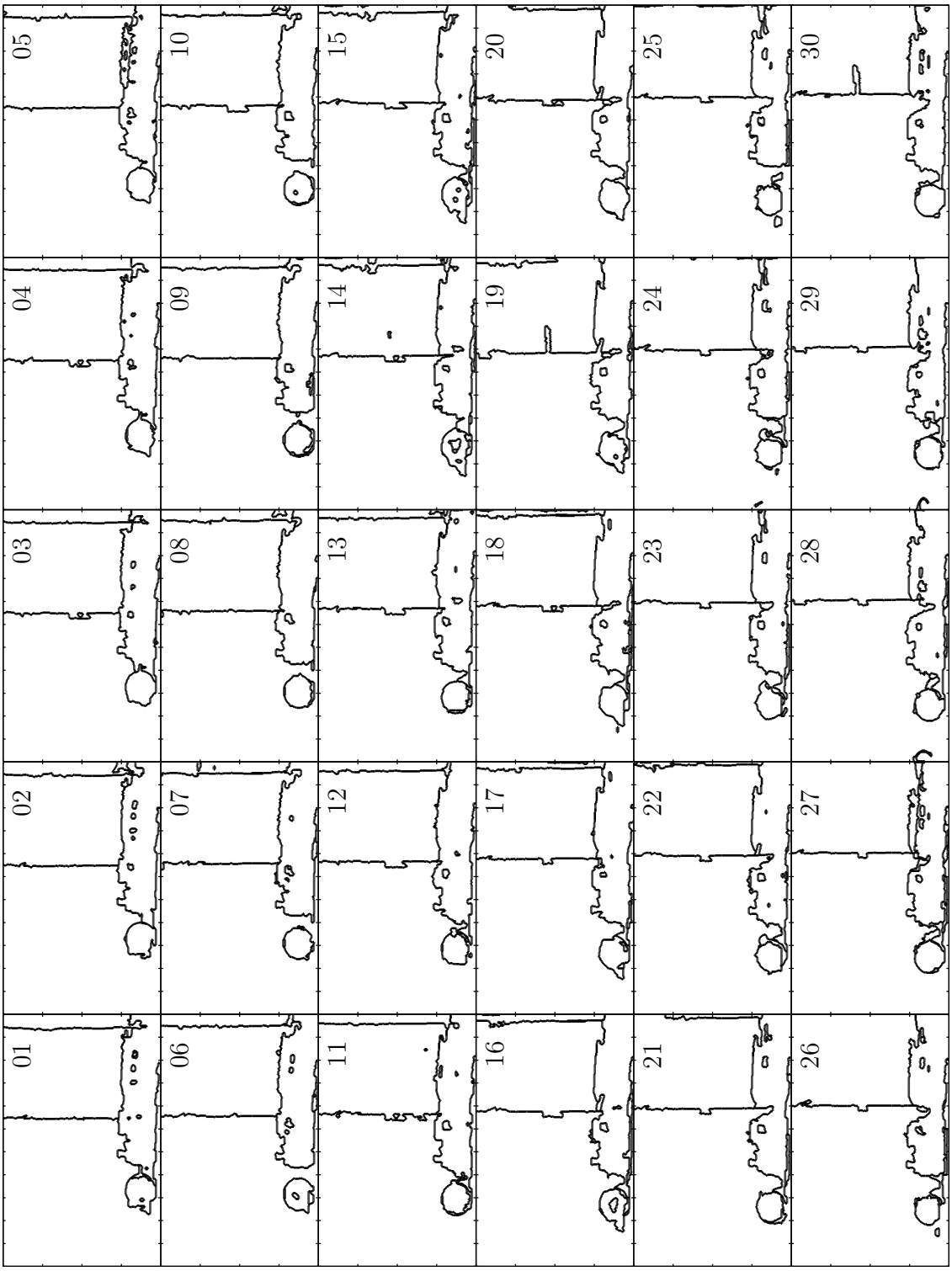


Figure 5.15: Before stabilisation: “Mobile and Calendar”, frames 01-30

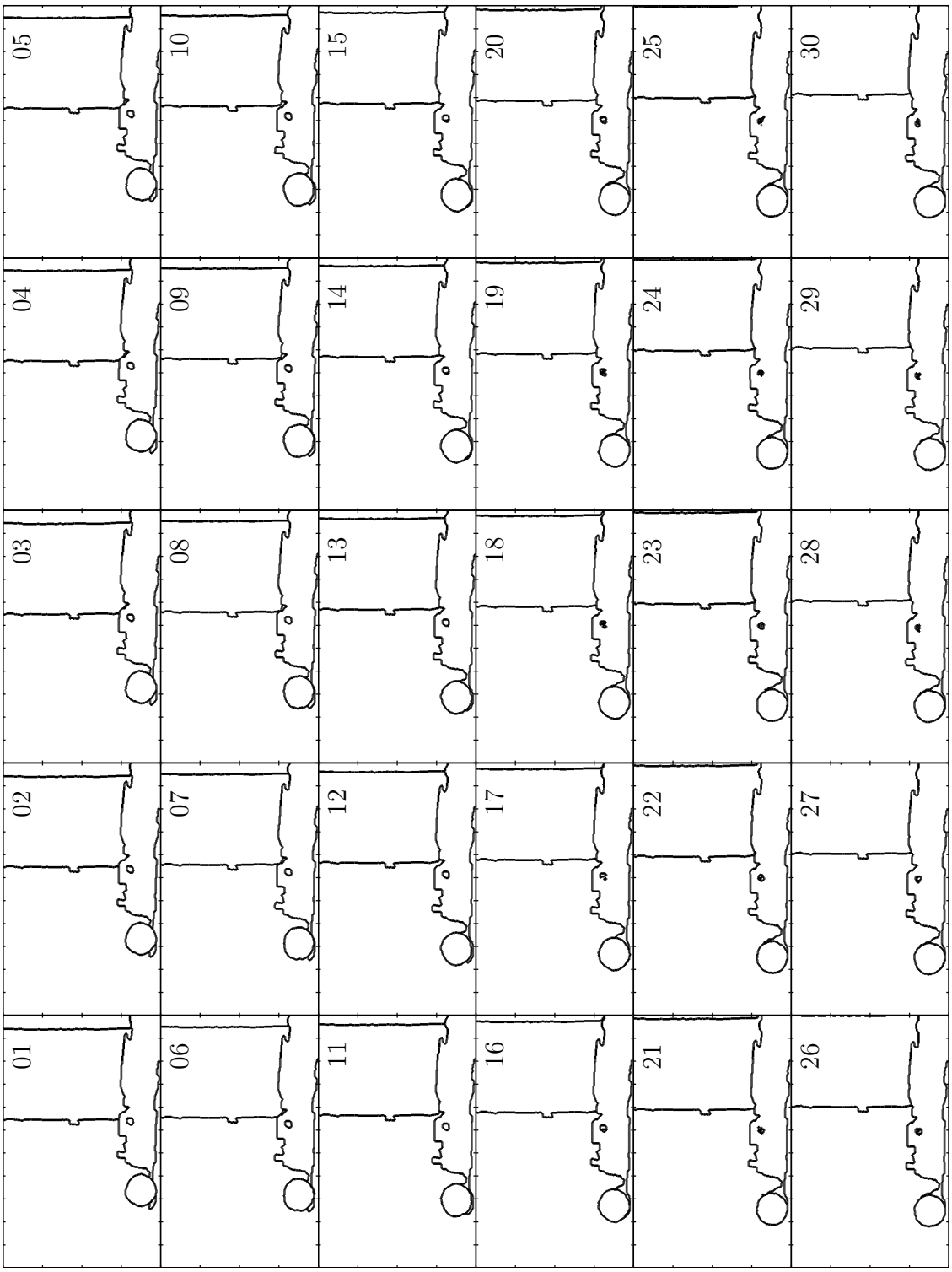


Figure 5.16: After stabilisation: "Mobile and Calendar", frames 01-30

Figure 5.17: Before stabilisation: "Mobile and Calendar", frames 21-50

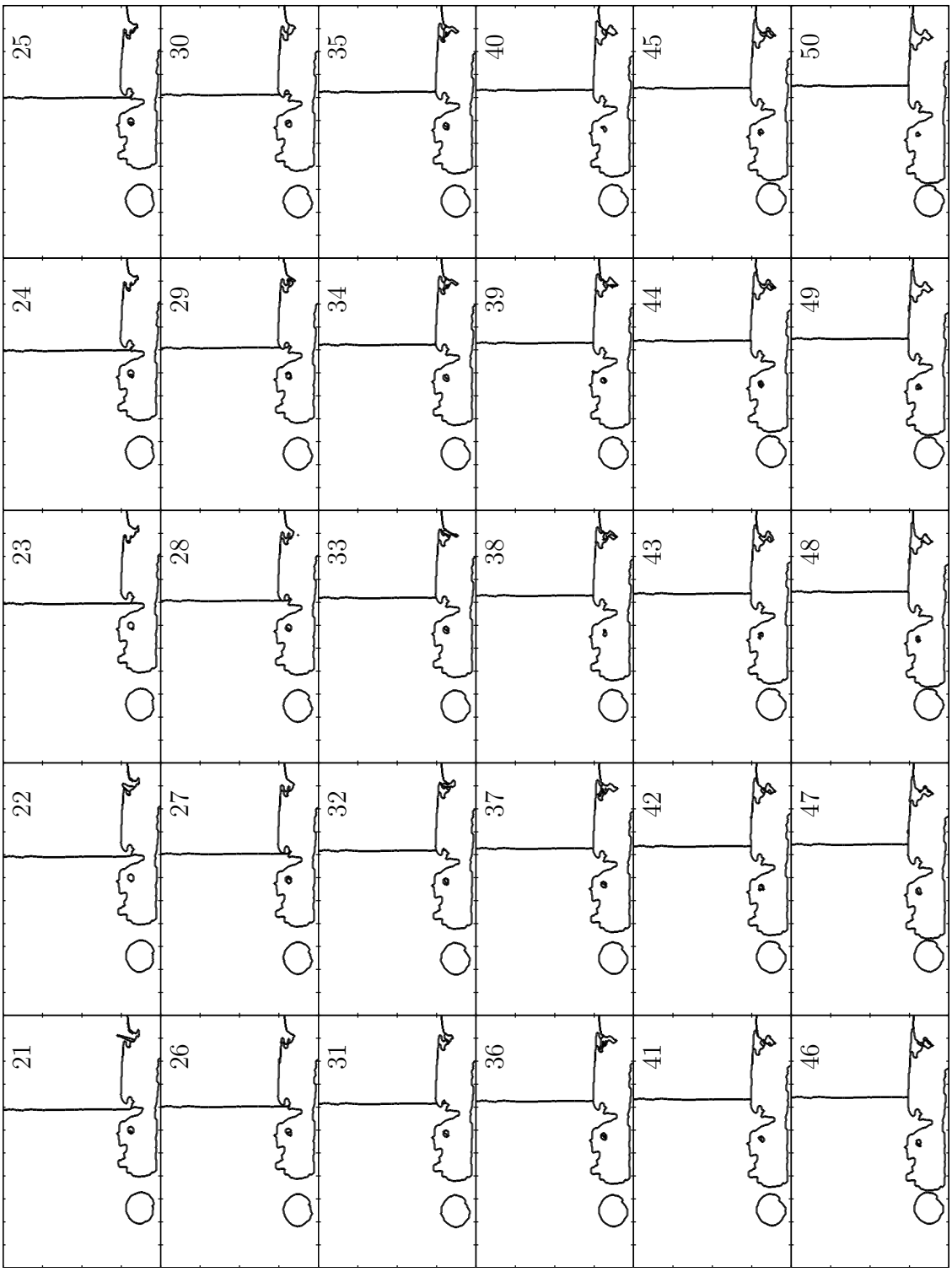


Figure 5.18: After stabilisation: "Mobile and Calendar", frames 21-50

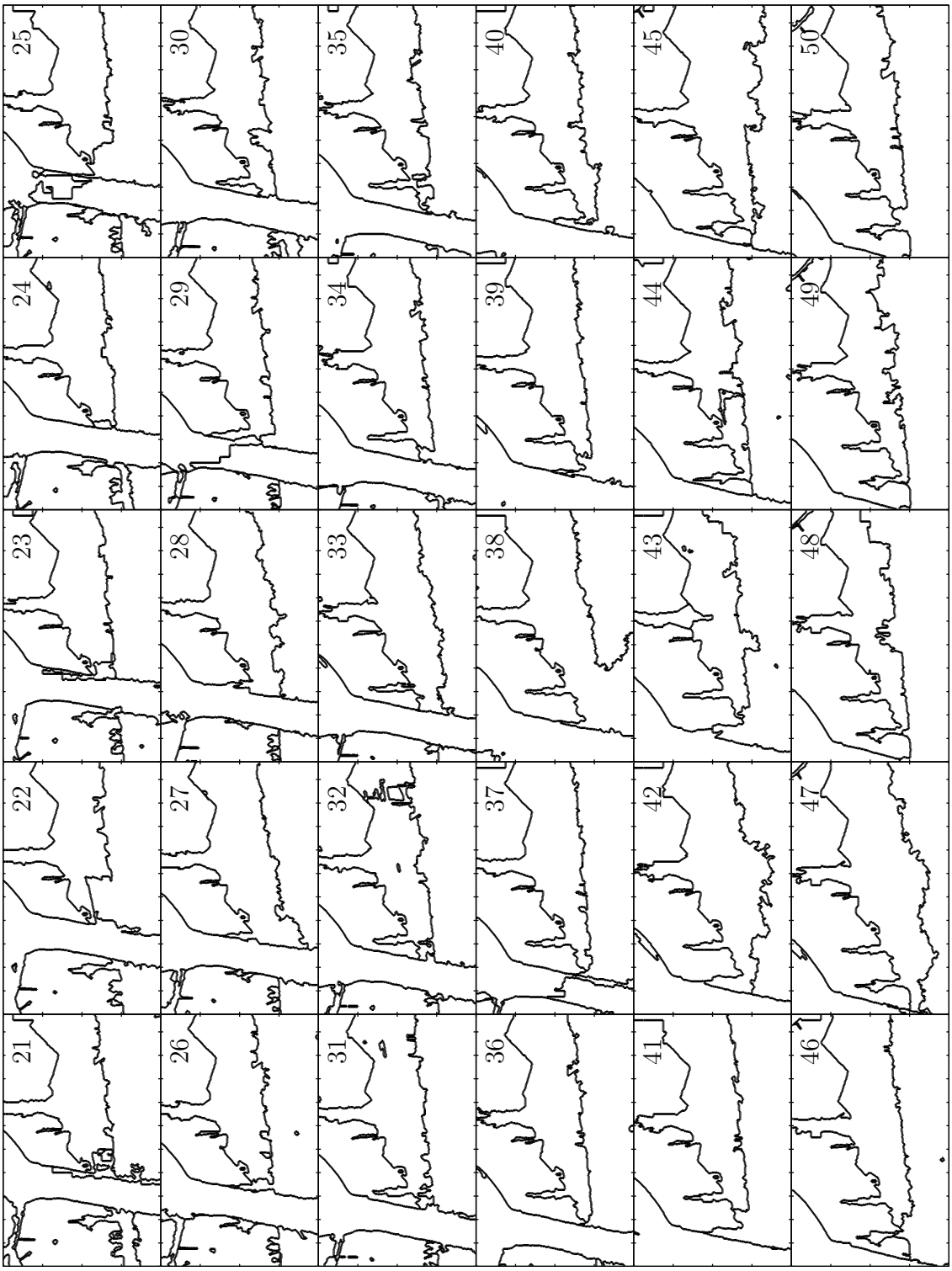


Figure 5.19: Before stabilisation: "Flower Garden", frames 21-50

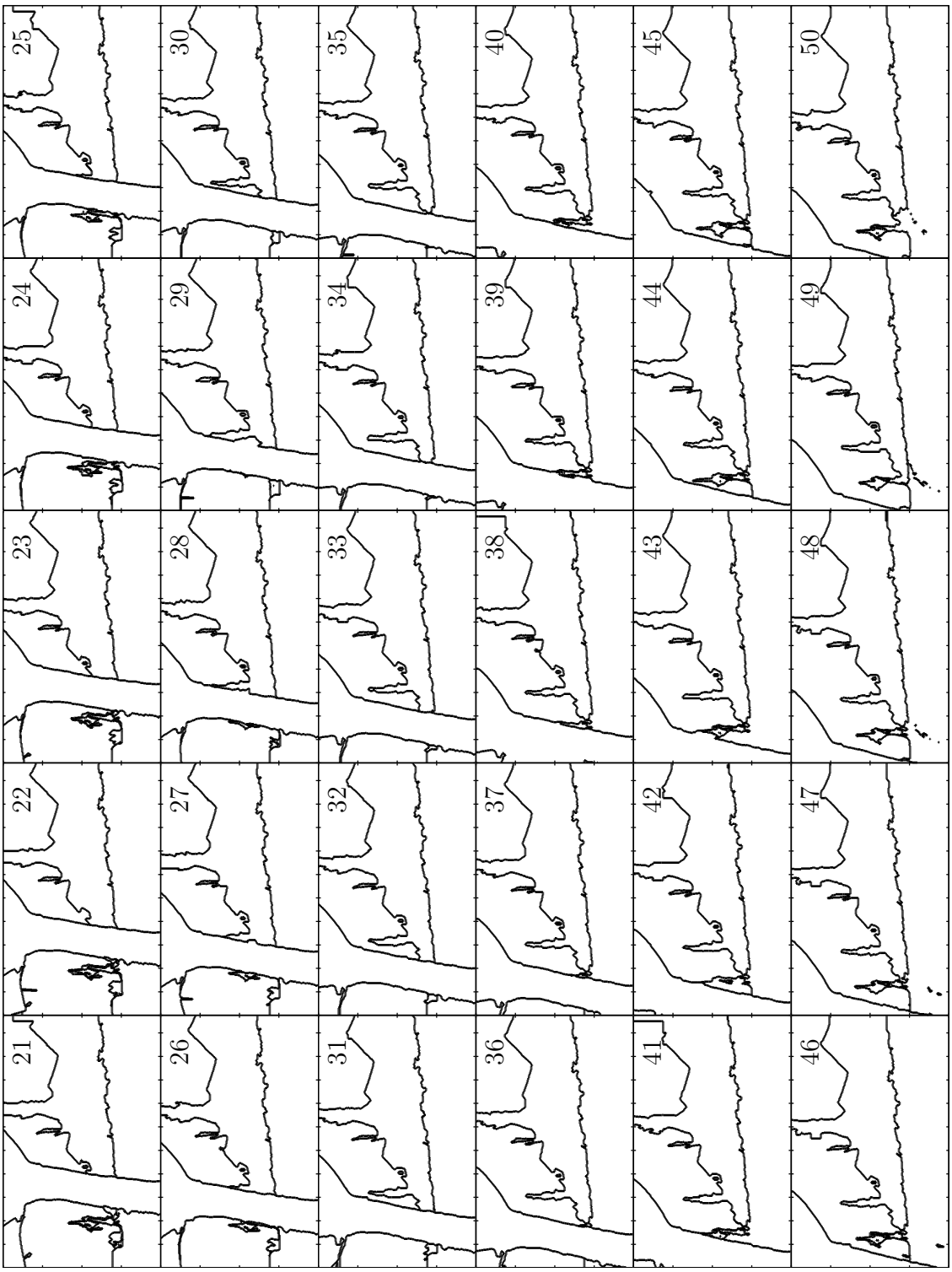


Figure 5.20: After stabilisation: “Flower Garden”, frames 21-50

5.6.2 Statistical measures on improvements of the shape mask

To quantify the performance of the temporal stabilisation as compared with the unprocessed masks, the two measures of similarity are calculated on the shape and the color histograms of the objects. For each test, the supplied inputs are the binary masks for an object and their color components at two adjacent frames [80]. A smaller value for the shape measure indicates that there is a higher level of similarity between the two shape masks, and a smaller chi-square value means a higher probability that the two segments carry similar color components as is expected of a good segmentation.

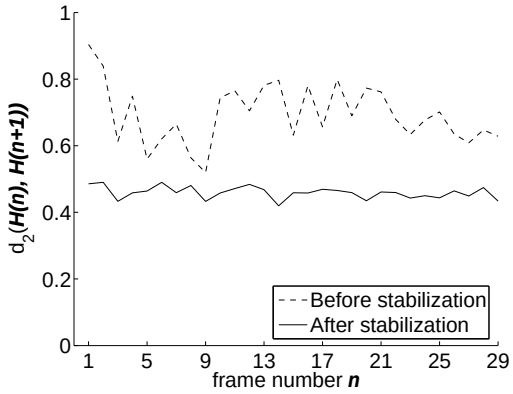
Table 5.5 lists the mean results from the test as performed on three objects. The results of the ball and the train are calculated between frames $01 - 30$, while the tree between frames $21 - 50$. It can be seen that the mean values *after* temporal stabilisation are significantly smaller than those obtained beforehand.

	Shape		χ^2	
	Before	After	Before	After
Ball	0.6941	0.4593	199	103
Train	0.8841	0.4586	221	152
Tree	0.8822	0.6200	512	203

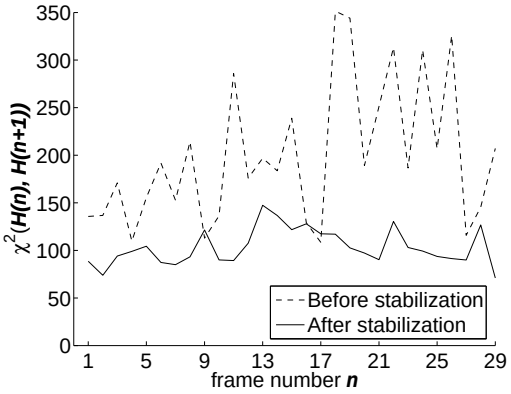
Table 5.5: Mean results of the shape-similarity measure and the chi-squared test on pairs of adjacent segmentation masks

A further insight into these values is shown in Figure 5.21. The graphs show the calculations at every pair of adjacent frames. It can be seen that the values associated with the stabilised masks are lower than those of the initial mask at most points, meaning that the consistency of a mask transition between two adjacent frames has improved at most frames and not just in the mean-value sense. An exception for the shape measure occurs toward the last few frames of "Flower Garden" where the tree moves out of the picture, hence leaving an edge effect on the shape measure.

Ball

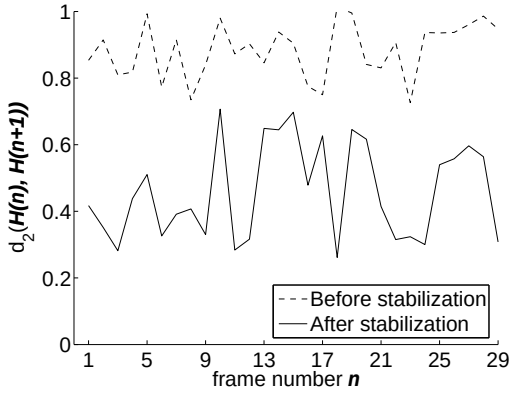


(a) Shape-similarity

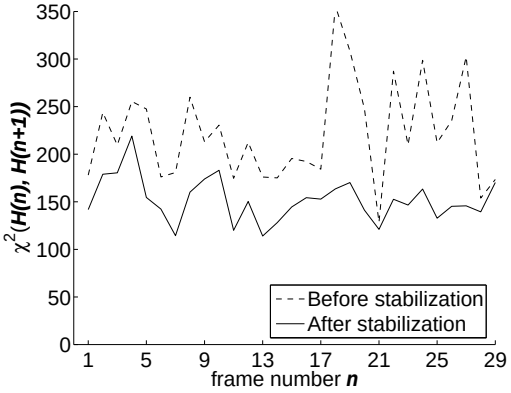


(b) Color χ^2-test

Train

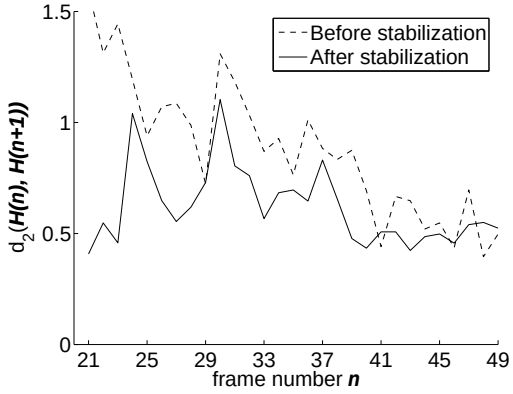


(c) Shape-similarity

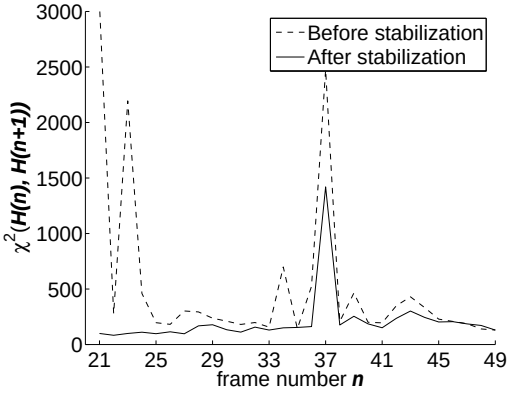


(d) Color χ^2-test

Tree



(e) Shape-similarity



(f) Color χ^2-test

Figure 5.21: Comparison of segmentation before and after the temporal stabilisation. The measure of shape similarity is on the left column, and the chi-square test on the color histograms is on the right column

5.6.3 Comparison to a manual segmentation

In this section, the segmentation results from before and after the temporal stabilisation are compared against a sample set of manually-segmented masks. The manual segmentations are derived by hand using the Polygonal Lasso Tool in Adobe Photoshop.

For the "Ball" object, three sets of segmentation masks are shown in Figures 5.22, 5.23 and 5.24. The first set, in Figure 5.22, contains the manually-segmented object masks. The second set, in Figure 5.23, is obtained directly from the three-frame segmentation algorithm and prior to the stabilisation. The third set, in Figure 5.24, shows the results after these masks have been stabilised. Note that the unprocessed shape mask for this object is not available in the initial segmentation between frames 40 and 46, as its local movement is too small to be detected (as seen previously in Figure 5.17). For the purpose of comparison, in the unprocessed segmentation in Figure 5.23, the shape mask at frame 37 is reused on frame 40, and the shape mask at frame 49 is reused on frame 45.

In the same order, the corresponding masks for the "Train" object are shown in Figures 5.25 to 5.27, and for the "Tree" object in Figures 5.28 to 5.30. In Table 5.6, the mean values of the shape measures and the chi-squared tests are shown for the three objects when comparing the automatic segmentation results to the manually-segmented masks, both before and after temporal stabilisation. The closer resemblance of the stabilised masks to the manually-segmented masks are clearly seen in all three cases, demonstrated by the smaller values associated with both the shape measure and the chi-squared test obtained on the color histogram.

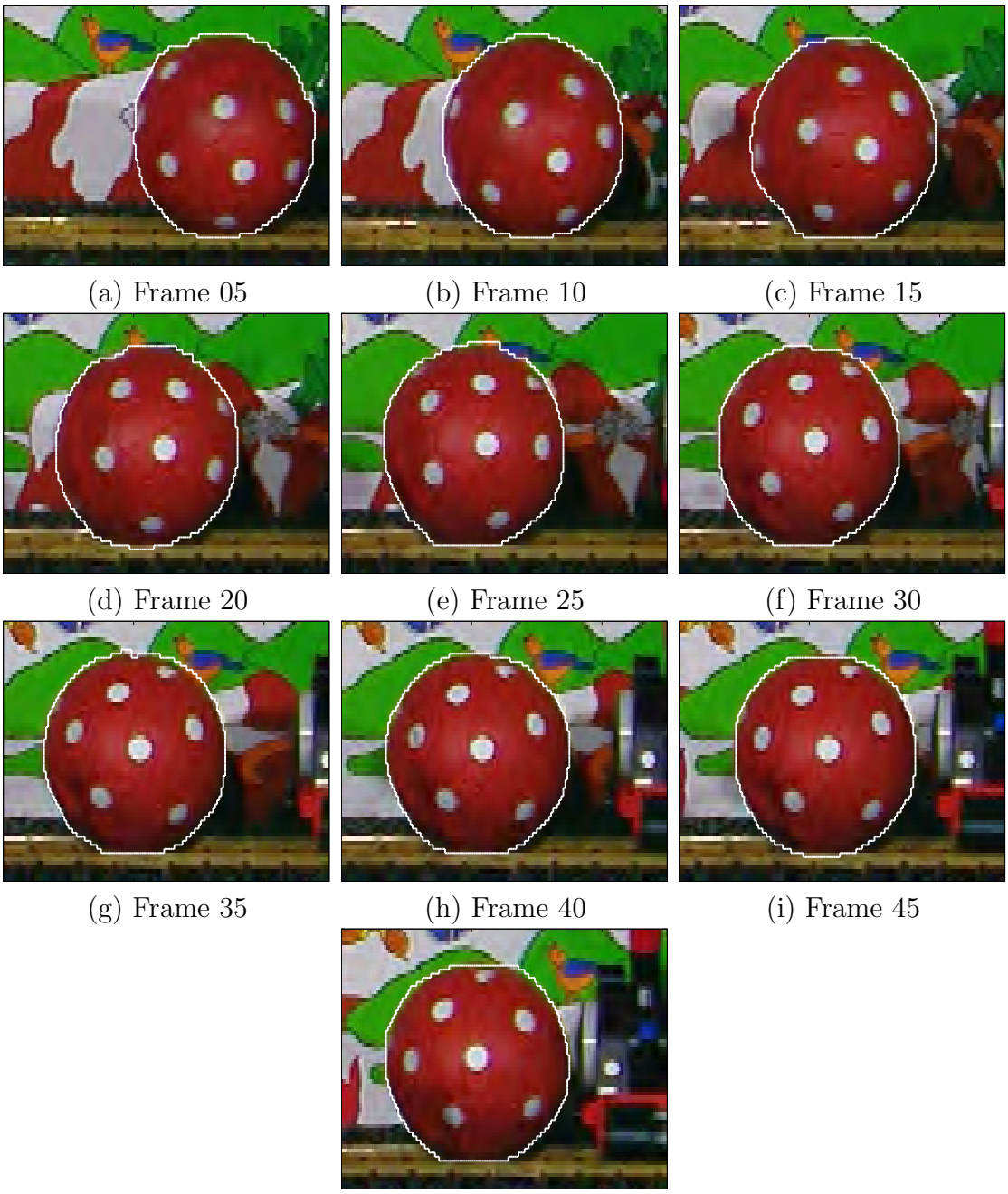
The graphs in Figure 5.31 show the frame-by-frame result of the comparisons for three objects. For the shape measure, the improvement of the stabilised masks is seen most clearly for the "Ball" and "Tree" objects. As compared to these two objects, the "Train" object shows relatively more discrepancies in the shape measure between the hand segmentation and the automatic segmentation, for both the unprocessed and stabilised results. This is attributed to a number of undetected

concave gaps that exist between the carriages of the train. The exact object boundaries in these areas are more difficult to detect, since the background is subject to both occlusion and uncover as the train moves, and poor lighting due to the shadow of the carriages. However, as the results in Table 5.6 show, the segmentation masks after the temporal stabilisation have also improved as compared to the unprocessed masks.

The accuracy of the stabilised masks, with reference to the object boundary, can be inspected in Figures 5.24, 5.27 and 5.30. A visual observation of these images show that the resulting masks match up very well to the actual objects after the motion-based stabilisation process.

	Shape		χ^2	
	Before	After	Before	After
Ball	0.5735	0.4615	189	135
Train	1.2410	1.2157	645	642
Tree	0.9818	0.7786	1134	852

Table 5.6: Mean results of the shape-similarity measure and the chi-squares test between the manually-segmented masks and the automatic segmentation, before and after temporal stabilisation



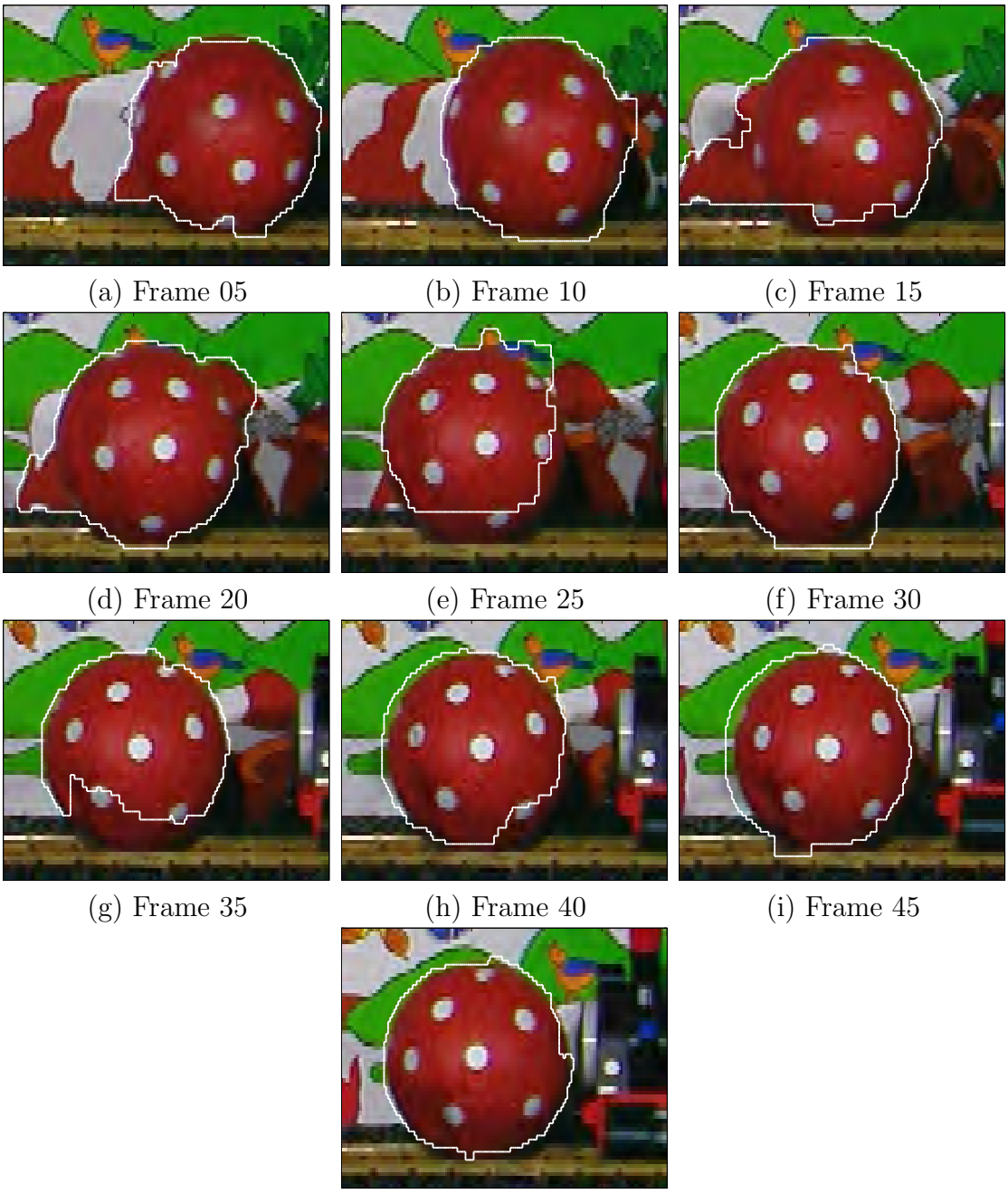
(a) Frame 05 (b) Frame 10 (c) Frame 15

(d) Frame 20 (e) Frame 25 (f) Frame 30

(g) Frame 35 (h) Frame 40 (i) Frame 45

(j) Frame 50

Figure 5.22: Manual segmentation for “Ball” object



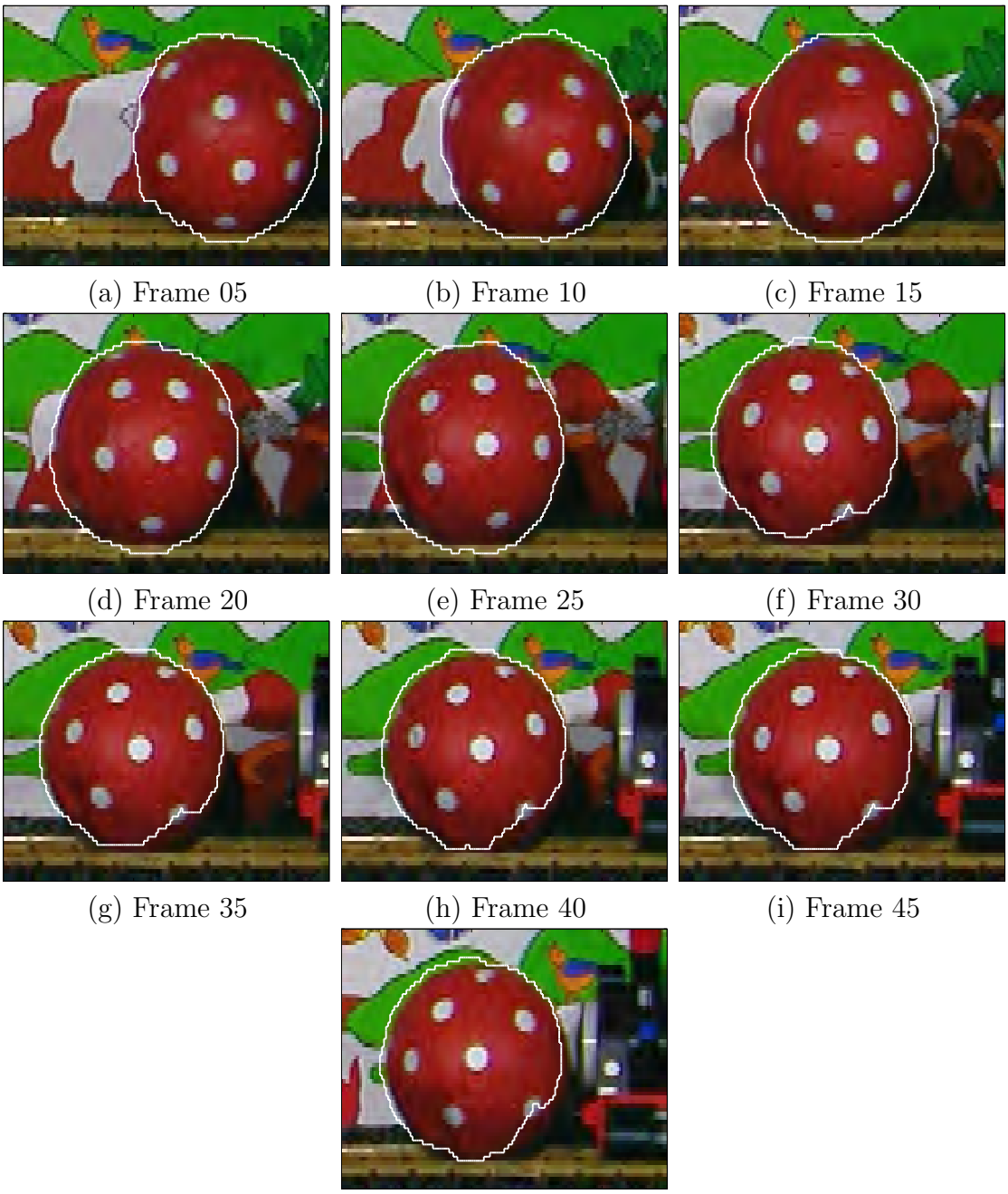
(a) Frame 05 (b) Frame 10 (c) Frame 15

(d) Frame 20 (e) Frame 25 (f) Frame 30

(g) Frame 35 (h) Frame 40 (i) Frame 45

(j) Frame 50

Figure 5.23: Initial (unprocessed) segmentation for “Ball” object



(a) Frame 05 (b) Frame 10 (c) Frame 15

(d) Frame 20 (e) Frame 25 (f) Frame 30

(g) Frame 35 (h) Frame 40 (i) Frame 45

(j) Frame 50

Figure 5.24: Segmentation after intra-group stabilisation for “Ball” object

(a) Frame 05 (b) Frame 10

(c) Frame 15 (d) Frame 20

(e) Frame 25 (f) Frame 30

(g) Frame 35 (h) Frame 40

(i) Frame 45 (j) Frame 50

Figure 5.25: Manual segmentation for “Train” object

(a) Frame 05 (b) Frame 10

(c) Frame 15 (d) Frame 20

(e) Frame 25 (f) Frame 30

(g) Frame 35 (h) Frame 40

(i) Frame 45 (j) Frame 50

Figure 5.26: Initial (unprocessed) segmentation for “Train” object

(a) Frame 05 (b) Frame 10

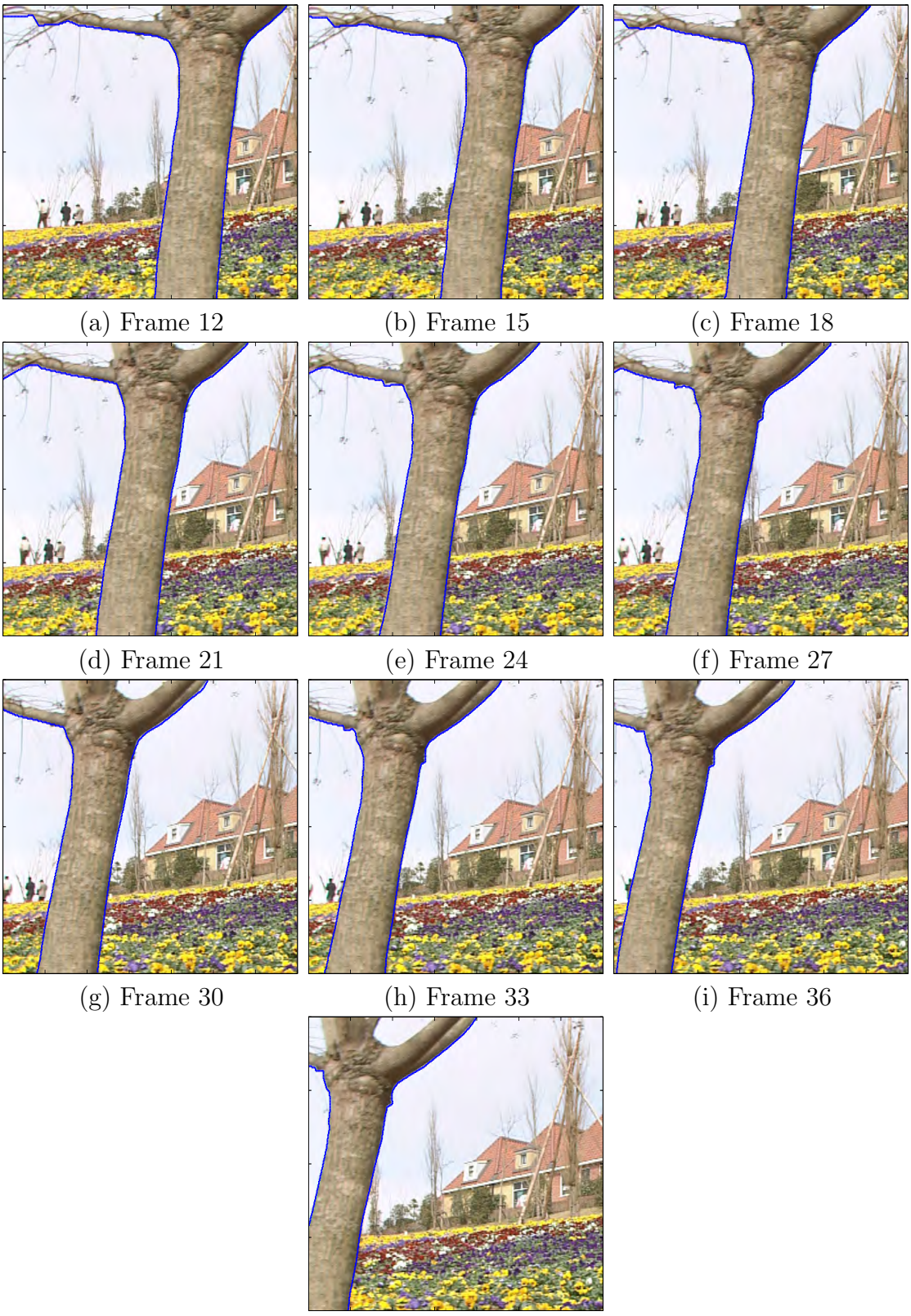
(c) Frame 15 (d) Frame 20

(e) Frame 25 (f) Frame 30

(g) Frame 35 (h) Frame 40

(i) Frame 45 (j) Frame 50

Figure 5.27: Segmentation after intra-group stabilisation for “Train” object



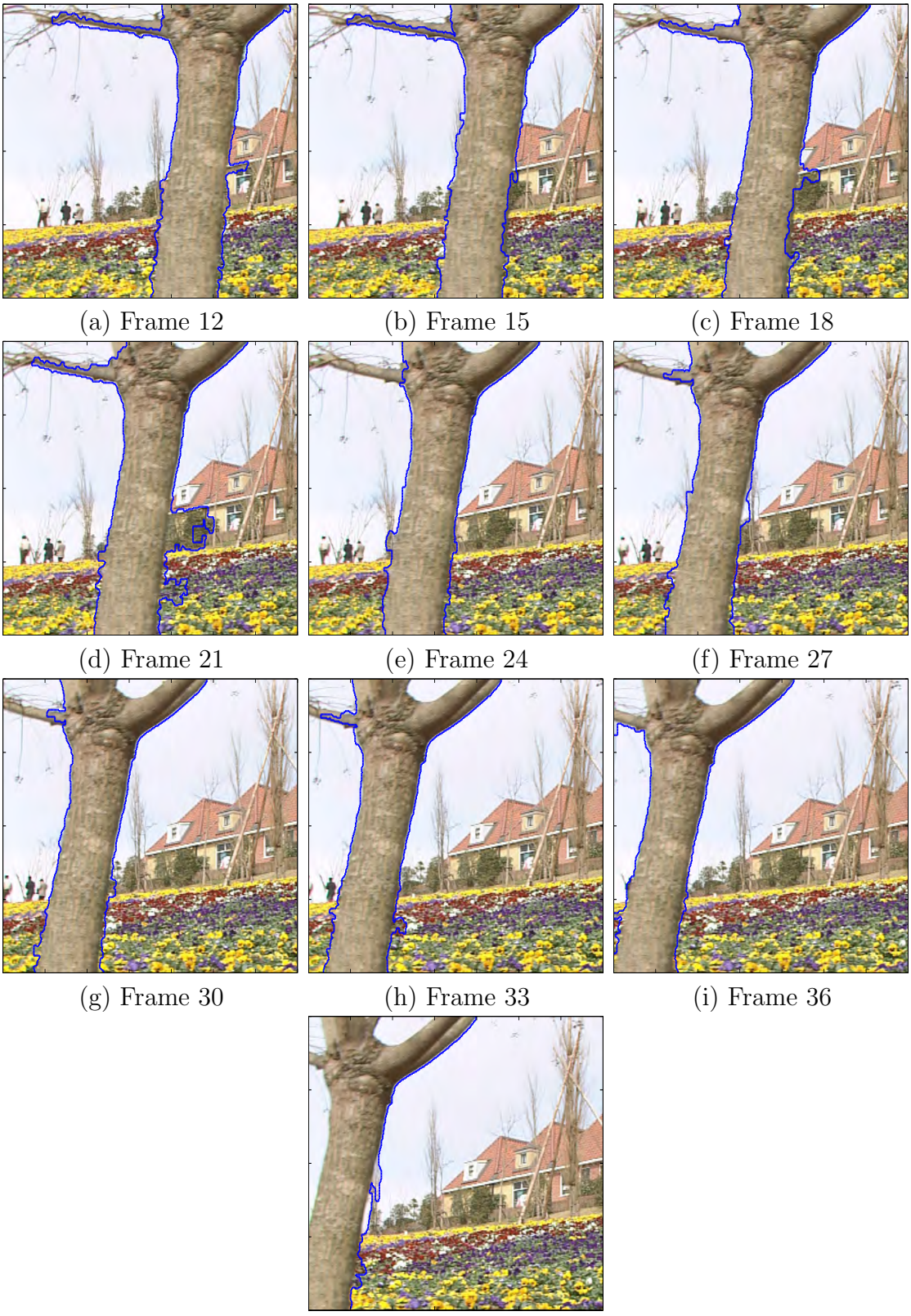
(a) Frame 12 (b) Frame 15 (c) Frame 18

(d) Frame 21 (e) Frame 24 (f) Frame 27

(g) Frame 30 (h) Frame 33 (i) Frame 36

(j) Frame 39

Figure 5.28: Manual segmentation for “Tree” object



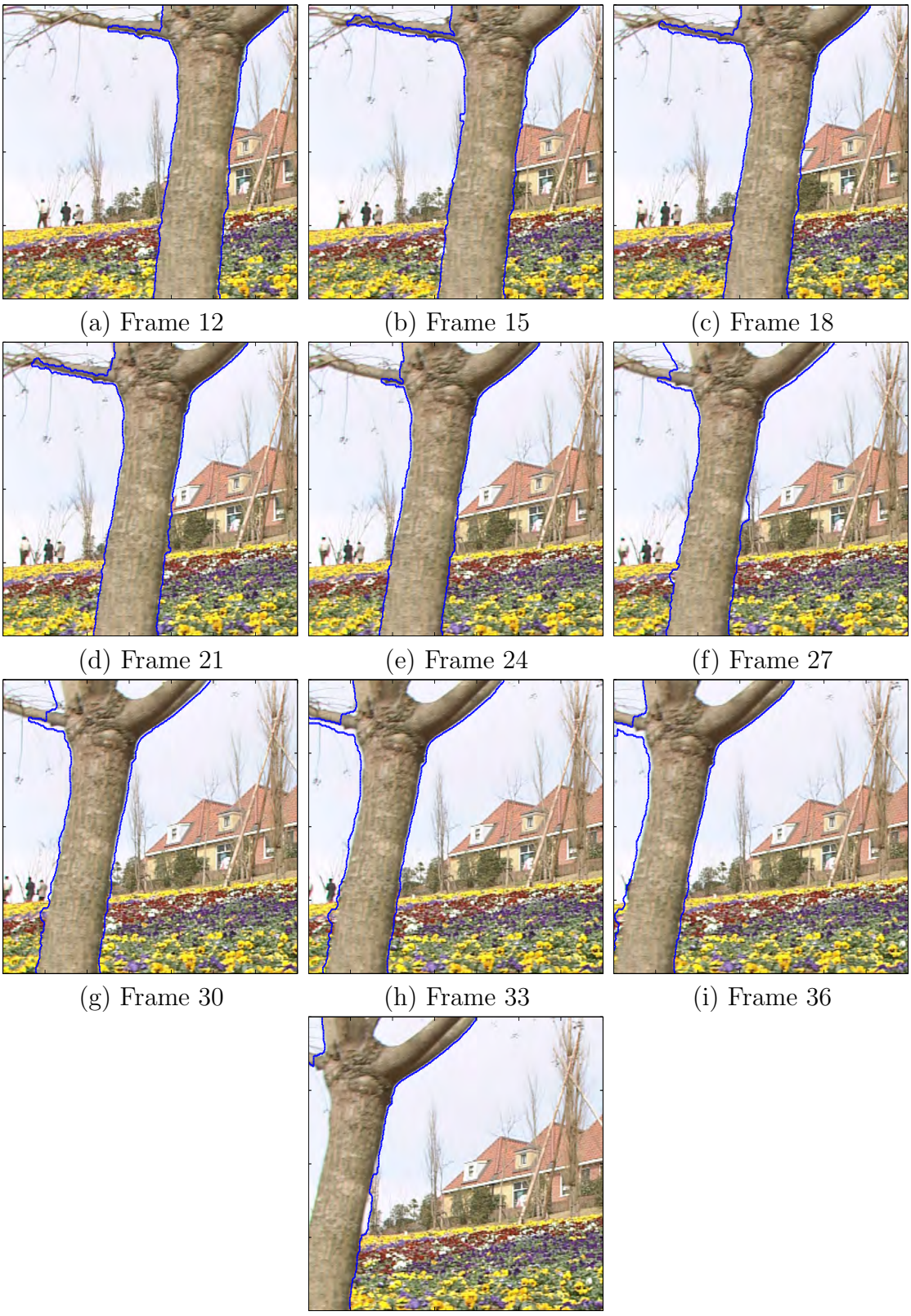
(a) Frame 12 (b) Frame 15 (c) Frame 18

(d) Frame 21 (e) Frame 24 (f) Frame 27

(g) Frame 30 (h) Frame 33 (i) Frame 36

(j) Frame 39

Figure 5.29: Initial (unprocessed) segmentation for “Tree” object



(a) Frame 12 (b) Frame 15 (c) Frame 18

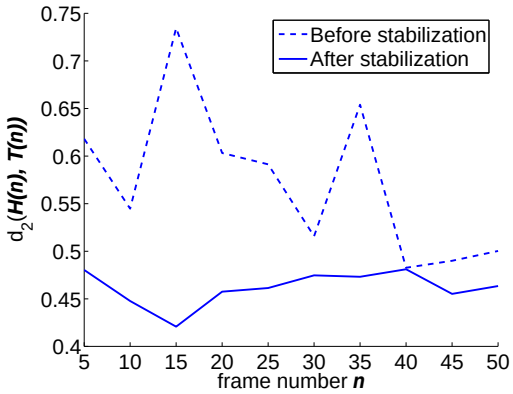
(d) Frame 21 (e) Frame 24 (f) Frame 27

(g) Frame 30 (h) Frame 33 (i) Frame 36

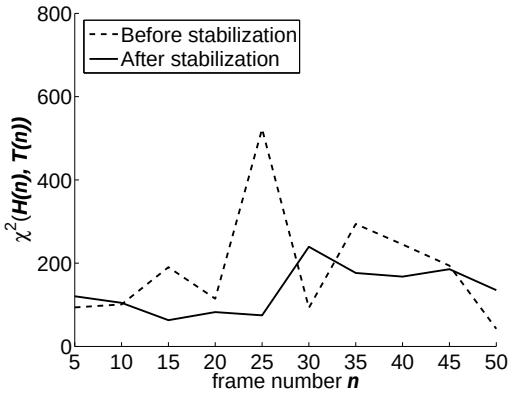
(j) Frame 39

Figure 5.30: Segmentation after intra-group stabilisation for “Tree” object

Ball

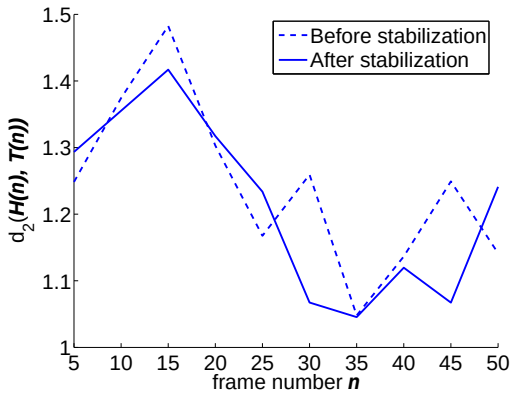


(a) Shape-similarity

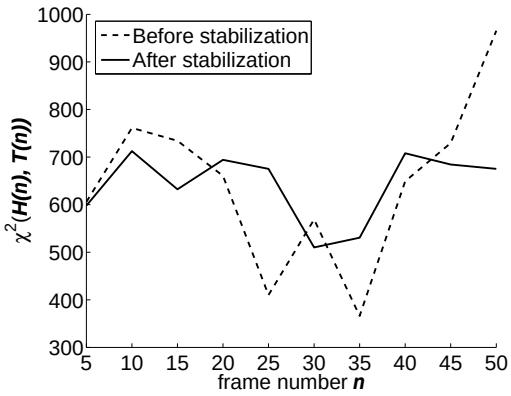


(b) Color χ^2-test

Train

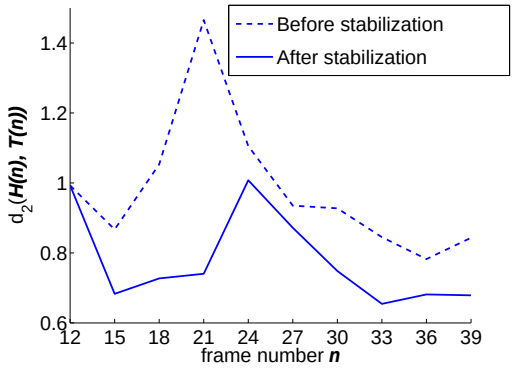


(c) Shape-similarity

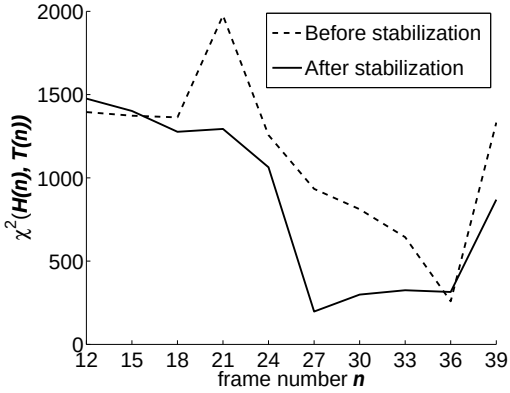


(d) Color χ^2-test

Tree



(e) Shape-similarity



(f) Color χ^2-test

Figure 5.31: Comparison to the manually-segmented masks, before and after stabilisation

5.7 Segmentation on extended data sets

For many applications which rely on the results of video segmentation as their input, it is often required that the supplied data is sustainable for the extended duration of a video sequence. Therefore, in order to make the results usable in practice, it is necessary to consider the feasibility of the segmentation approach in longer video sequences.

It has been demonstrated so far in this chapter that one of the central issues in extending object segmentation beyond individual frames is the ability to maintain the temporal consistency of the object masks, as well as to make corrections to segmentation errors, which may be caused by a temporary lack of motion information and/or color contrast at a present frame. As the segmentation results obtained in the previous section show, multiple-frame processing can lead to significant improvements in the accuracy of the shape masks, as long as the motion estimation is of good quality. Semantically, once the individual masks become more accurate, the temporal consistency will invariably improve as well.

In the integration of more pictures into a segmentation process, it becomes more difficult to find the correct correspondence when object relations are sought between frames too far apart. Post-processing using incorrect motions would eventually compromise the accuracy and consistency of the shape masks. In a batch-processing environment, the law of diminishing returns also applies when the number of frames grows large.

In extending the algorithm to cover longer sets of data, it is desirable to retain the advantages offered by batch-processing, such as the temporal stability already achieved in each group of frames. The method however needs to address the constraint of having a smooth transition for the shape masks from one group of frames to the next group when they are all concatenated together in the order of the sequence.

The proposed solution is illustrated in Figure 5.32. In principle, the segmentation masks are still processed in groups of frames. Consecutive groups are however

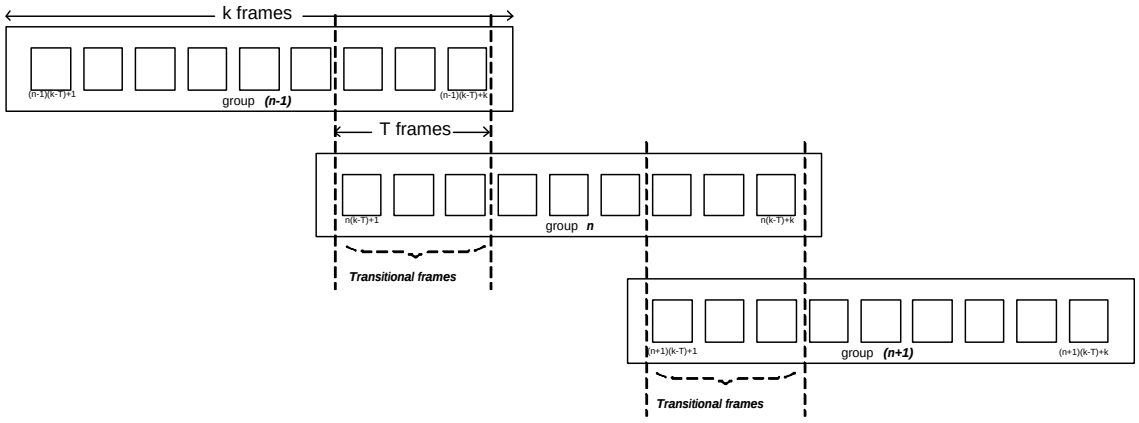


Figure 5.32: Transition for object masks across overlapping groups of segmentations

overlapping, so that the last frames in the first group coincide with the first frames in the second group. The objective of this overlap is to provide a temporal space where the masks from two groups can be gradually adapted to each other, which would otherwise be abrupt without any overlapping. In the algorithm described hereto, the overlapped interval serves as a transitional period where the shape masks from the first frame group are gradually *morphed* into the masks on the second frame group.

Let $1...T$ represent the frame number of an overlapping transition between two sets of video clips, and $M_1(n)$ and $M_2(n)$ be the binary masks of an object taken respectively from group 1 and group 2, at frame location n, $1 \leq n \leq T$. The two masks are then combined into a mask $M(n)$ as follows:

$$M(n) = (T - n + 1)M_1(n) + nM_2(n) \tag{5.13}$$

The combined mask $M(n)$ is not binary, as each of its pixels p may assume one of

four values: [1]

$$\begin{array}{lllllll} p & \overline{\subset} & (M_1 \cup M_2) & \Leftrightarrow & p & = & 0 & (outside) \\ p & \subset & (M_1 \cap M_2) & \Leftrightarrow & p & = & T+1 & (shared\ mask) \\ p & \subset & (M_1 \cap \overline{M_2}) & \Leftrightarrow & p & = & T-n+1 & (Type1) \\ p & \subset & (\overline{M_1} \cap M_2) & \Leftrightarrow & p & = & n & (Type2) \end{array} \tag{5.14}$$

The last two items correspond to pixels which exist in one mask but not the other, which are causing instabilities when moving across frame groups. Removing all such pixels produces stable results *within* the overlapping section, but at the same time creates instabilities when moving in and out of this section. The solution is therefore not to remove those discrepancies altogether, but to reduce their appearances in a temporally-consistent way within, and at both entry and exit points of the transition.

To facilitate the explanations, let the two types of discrepancies be called $Type1$ and $Type2$, where $Type1$ consists of pixels belonging to M_1 but not M_2, and $Type2$ is the opposite. At the beginning of the transition, pixels of $Type2$ are more likely to cause a discrepancy in the mask than $Type1$, since they do not belong to the segmentation results from the group 1 which precedes the transition. Likewise, pixels of $Type2$ are less visible as discrepancy towards the end of the transition, as they also belong to the segmentation masks in group 2 which follow the transition. The algorithm therefore aims to reject more of $Type1$ pixels towards the end and more of $Type2$ pixels towards the beginning of a transition.

From a mask $M(n)$, the pixels of $Type1$ and $Type2$ are first arranged into a number of spatial clusters, and each cluster only contains one pixel type. In addition, let a shared mask be the set of pixels which belong to both M_1 and M_2. Removal of pixel from any cluster of either type is carried out with reference to the shared mask and illustrated in Figure 5.33 as follows:

- Set the number of pixels to reject, depending on the pixel type of the cluster
  - $Type1$: Reject $\frac{n}{T+1}C$ pixels, i.e. proportional to the frame distance from

[1] T is chosen as an even number in implementations to make the distinction between the last two conditions unambiguous, i.e. $T + n - 1 \neq n\ \forall n$

the start of the transition. C is the total number of pixels inside the cluster.

- $Type2$: Reject $\frac{T-n+1}{T+1}C$ pixels, i.e. proportional to the frame distance from the end of the transition

- For each cluster, grow the shared mask from its original position until it contains up to the number of pixels to be retained in that cluster, and reject all other pixels in the cluster from the mask.

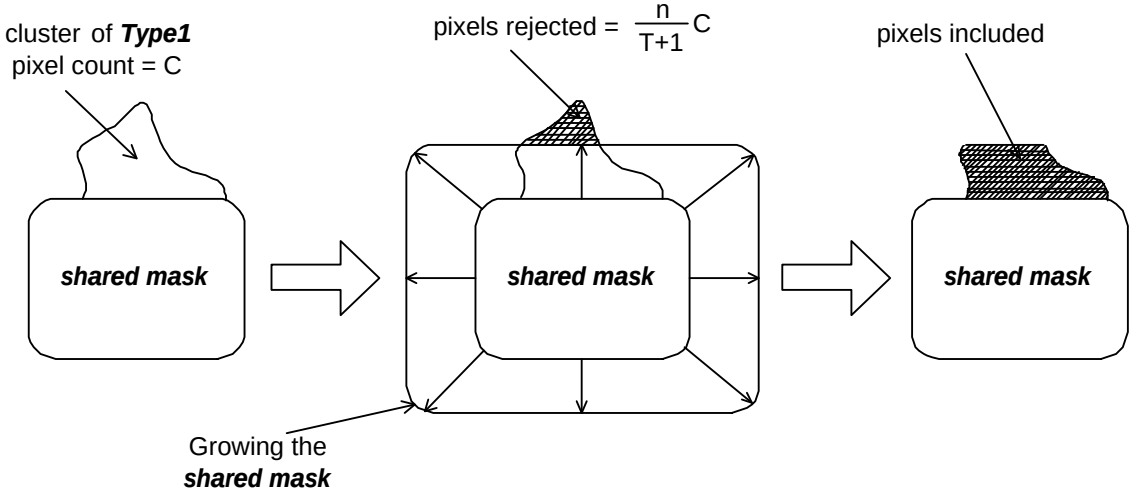


Figure 5.33: Selection of pixels for an object mask in transition

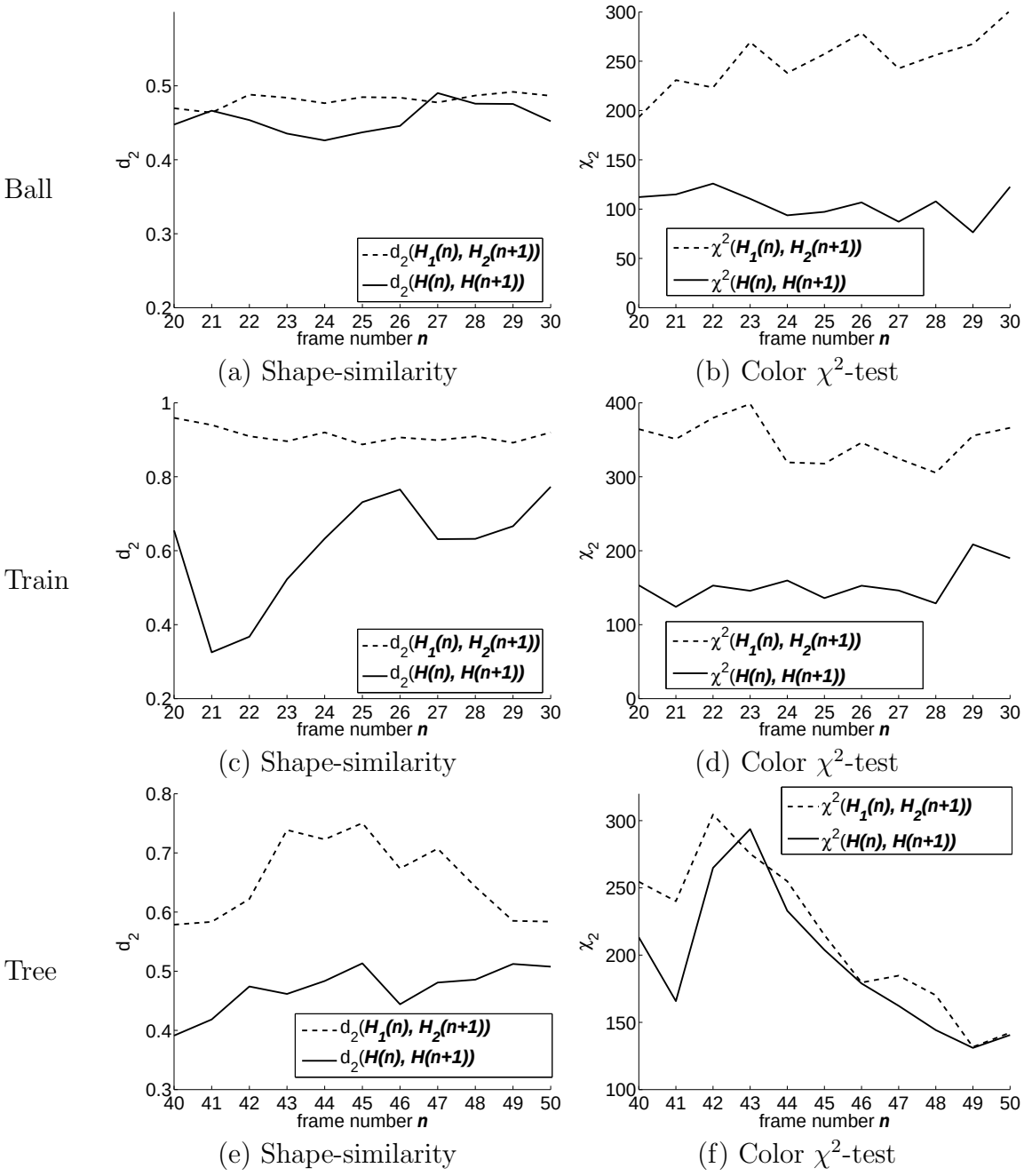
In the selection of pixels to keep or reject, the use of the shared mask as the core is designed so that the spatial integrity of the final shape mask is maintained. The temporal stability across a transition interval also depends on the number of frames involved. A longer overlap allows a more gradual morphing of the differences between the masks, while a short interval would force it to be resolved more quickly.

It should be restated that the main objective of this group-morphing algorithm is to facilitate a smooth transition across different sets of segmentation results. The algorithm therefore does not have an explicit target of improved segmentation accuracy. It assumes that the object masks produced by the motion-based stabilisation process in the previous section are already of an acceptable quality and accuracy, but also recognises that there may still exist some discrepancies when the masks are

produced using different sets of neighbouring video frames. Objectively, the performance of the group-morphing strategy is assessed by comparing the smoothness of the frame-to-frame transition of the object mask after the operation, against a direct adaptation of the shape mask from one frame group to the next. The measures of the similarity in object shapes and color histograms are again used to demonstrate these improvements.

The graphs in Figure 5.34 show the results of these comparisons for three objects. In the results prior to the group morphing operation (i.e. the dotted graphs), the measures are calculated on the masks between pair of adjacent frames inside the group transition. In each pair, the first mask is taken from the segmentation performed on the first group of frames, and the second mask is taken from the segmentation performed on the next group of frames. These dotted graphs show the inconsistency of the frame-to-frame transition that would have been present in the segmentation masks, if they are directly assembled without consideration for the inter-group transition. On the other hand, the results obtained after completion of the inter-group morphing operation shows considerable improvements for both the shape-similarity and color measures between adjacent frames, as shown in Figures 5.34-a to 5.34-f. Note that the improvements seen in the first and last graphs, 5.34-a and 5.34-f, is not as strong as in 5.34-b to 5.34-e. In Figure 5.34-a, the shape of the ball is relatively constant in both groups, hence there is less gain through the inter-group morphing. In Figure 5.34-f, the "Tree" object is disappearing from the camera view as the sequence progresses, which also affects the frame-to-frame comparison of its mask.

To finalise this section, the segmentation results, after joining two groups of segmentations masks, are displayed in Figures 5.35 and 5.36. Each set of masks is obtained by concatenating two adjacent groups of 30 frames, which overlap by 10 frames (i.e. for the sequence "Mobile and Calendar" Figure 5.35, group1 = frames 1-30 and group2 = frames 21-50; for the sequence "Flower Garden" in Figure 5.36, group1 = frames 21-50 and group2 = frames 41-70). As observed from the results,



(a) Shape-similarity (b) Color χ^2-test

(c) Shape-similarity (d) Color χ^2-test

(e) Shape-similarity (f) Color χ^2-test

Figure 5.34: Comparison of the frame-to-frame inter-group transition. The dotted graphs show the results without the the group-morphing operation. The solid graphs show the result after the group-morphing operation is completed for the transitional frames.

the transformation of the shape masks across adjacent frames within each transition is as inconspicuous as it is elsewhere in the sequence.

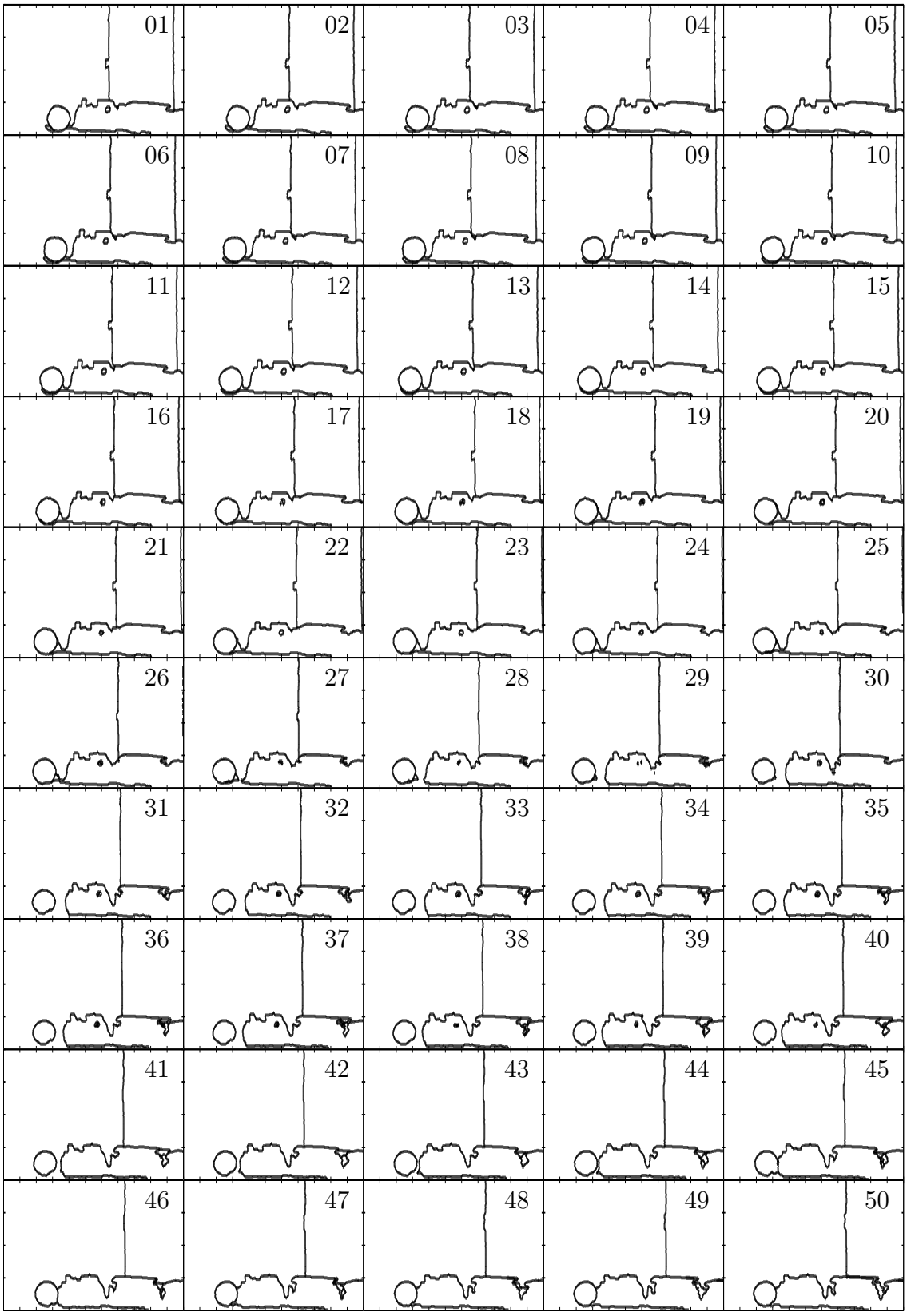


Figure 5.35: "Mobile and Calendar": Frames 01-50, created by concatenating two separate sets of masks, 01-30 and 21-50

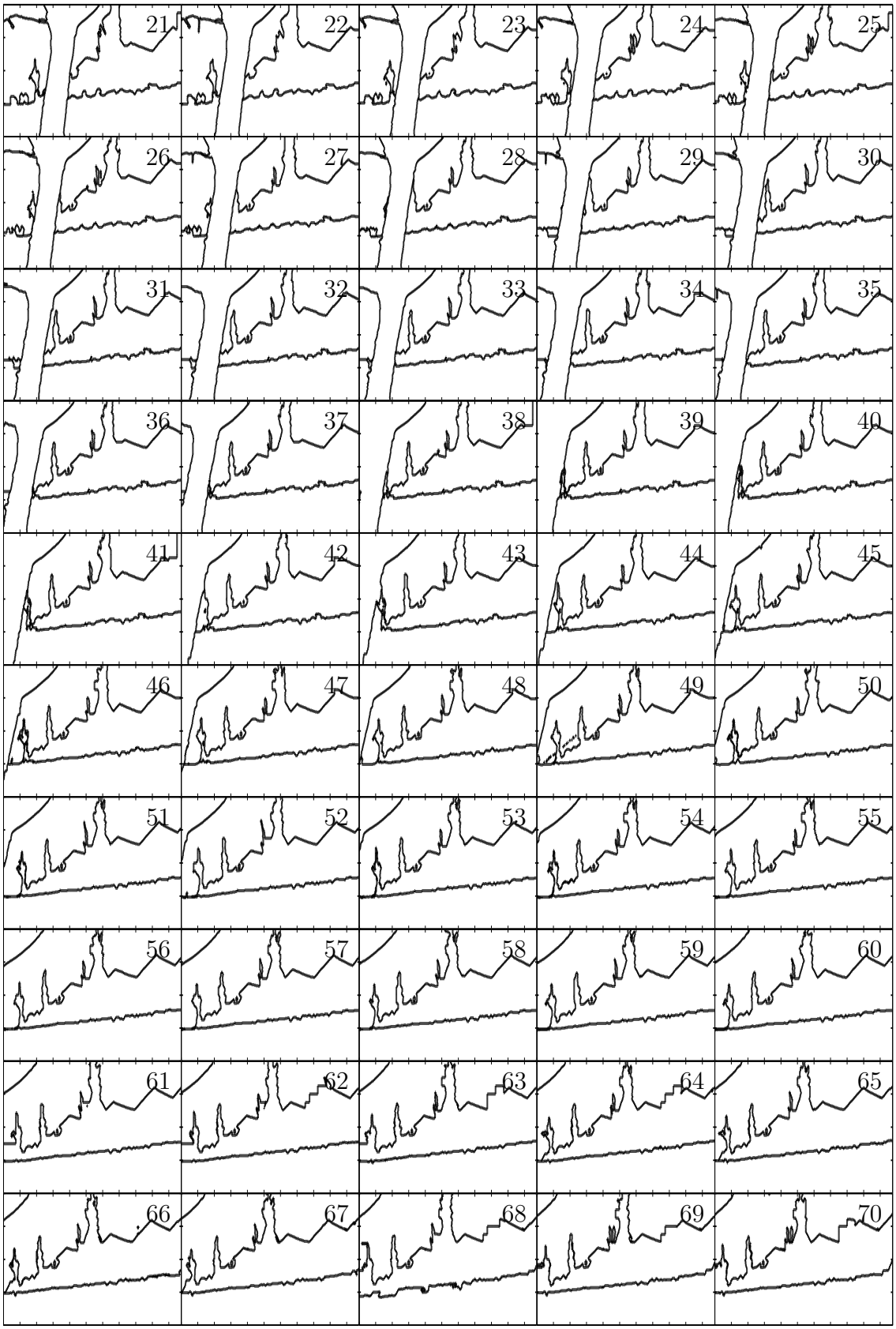


Figure 5.36: "Flower Garden": Frames 21-70, created by concatenating two separate sets of masks, 21-50 and 41-70

5.8 Summary

This chapter has proposed a stabilisation approach for segmentation masks of multiple objects in a sequence of frames. The stability objective was twofold, (a) stability in the presence of objects throughout the sequence, and (b) stability of the object boundaries. By enforcing the former criterion, oversegmentation and undersegmentation errors were removed because such instances are often represented as isolated and irregular segments. The second criterion was achieved via temporal averaging of the object masks, after the stability of object presence had been established. This chapter also saw the adoption of a 12-parameter projective motion model, to replace the affine model used in Chapter 4, in order to achieve more accurate mask-registration results when respective frames are positioned further from each other. The contributions of this chapter include:

- Use of a temporal "referencing" framework to validate the presence of individual masks: Support for a mask at one frame was evaluated by whether it had been *referred to* by masks at other frames in the sequence. The irregularities of an over-segmented region, together with its often incorrect motion, translated to a low (or zero) number of referees. An under-segmented region, on the other hand, was detected by a presence of double references from another frame.

- Adaptive thresholding for consistency checking on occluded parts of an object mask: Boundary smoothing was performed by a consistency check amongst all the masks registered towards a current frame. With a fixed number of frames in use and stabilisation performed concurrently on multiple objects, an object may be foreground to one but background to another. The consistency check-level for a mask pixel was reduced according to the number of times it is occluded at other frames in the sequence.

- A group morphing operation to concatenate sets of segmentation results from fixed-length, overlapping groups of frames. Its objective is to make the overall segmentation algorithm expandable, without an excessive reliance on a single

reference frame over a longer sequence.

At the end of this chapter, objective measures were employed to show that the final object masks were more temporally consistent than the initial set of individual segmentation, an effect which can also be confirmed from a visual inspection of the masks. In addition, these improvements were also demonstrated through comparisons made between the automatic segmentations and the manually-segmented object masks on a number of frames. The improvement achieved through the group-morphing algorithm has also been shown.

In proceeding with this approach, two main assumptions were made: first, (a) individual segmentation masks are already correct in the *majority* of frames, so that their overall quality will be improved through the group processing, and second, (b) all object transformations are approximately rigid in three-dimensional space, so that the consistency criteria can be enforced by a parametric means. The improvements shown in the result sections are the evidence that the first assumption has indeed been validated. As per the second assumption, while a quadratic model has been used to accommodate a wider range of rigid motion, it is also expected that the approach may not function as well for objects whose transformation falls outside the model capacity.

In comparison, the majority of segmentation techniques which exist in the literature are concerned mainly with a frame-to-frame processing of the object masks. A sequence-based method, on the other hand, may assume that a foreground object can be tracked for the entire video, and focus its processing on such objects. As it has been shown in the current and previous chapters, errors in a local segmentation are difficult to avoid, and sometimes cannot be detected based on the spatio-temporal content of just a few frames. The strength of the proposed method is its ability to use the information from a large number of frames, to concurrently reflect the dynamics of the scene onto multiple objects at each individual frame. This process does not only remove a number of defects that would have been difficult to correct in a local segmentation, but also brings stability to the resulting object masks in

each frame group in a top-down and regulated manner. In addition, the method also offers a flexible mechanism to enable a gradual transition between adjacent groups of segmentation masks. This extension makes it practical to extend segmentation result to longer sequences, without compromising the existing group-based stability of the object masks.

Lastly, as object segmentation is still a field which requires much further research, many strengths and weaknesses of a technique do not usually manifest until the segmentation is actually required for processing of a video sequence. While the work so far in this thesis has been concerned with the stability and accuracy of the segmentation, these issues have an important implication on the efficiency of other techniques such as video coding and compression. Together with accurate modelling and estimation of the object motion, these issues are central to an ability to obtain a concise, object-based representation for a video sequence. For example, transmission of subsequent frames in a video scene can be reduced to an accurate association of the masks and motion parameters to the video object, rather than the actual spatial textures in these frames.

To further demonstrate the accuracy and sustainability of the proposed segmentation approach, the next chapter presents an experimental video compression framework, which aims to encode and reconstruct each video frame from a collection of individual object sprites. The temporal consistency of an object, and the ability to properly model its motion along the sequence, are the contributing factors to the efficiency of such a video representation.

Chapter 6

An application for object-based video compression

6.1 Introduction

As in any evolving field of research, it would be premature to conclude that contemporary efforts have completely addressed the problems of unsupervised motion estimation and object segmentation, or image understanding at large. On the other hand, continuous improvements being made in object segmentation often directly benefit many other applications and related fields of research. A good segmentation provides a solid starting point for an object tracking system. A structure-from-motion study may deduce the object geometry based on long-term observation of an individual object and its associated motions, whereas useful semantic information may also be inferred from the binary shape mask of an object and its movements.

Using segmentation results after stabilisation from Chapter 5, this chapter looks at an experimental framework for object-based video compression which is shown to deliver much improved coding efficiency.

6.2 Representing objects in video

At the physical level, each video clip may simply be viewed as a sequence of images being displayed at a given frame rate. Semantically, however, a video is more likely perceived as a composition of objects, each with its own audio and visual characteristics. The concept of representing videos as a composition of such audio-visual objects, or AVOs, was embraced within the framework of the MPEG-4 standard [90]. An object-based coding scheme allows the manipulation of video data at the physical object level at both the encoder and decoder, a feature seen as essential for many future content-based multimedia applications and interactive services [91].

For compression purposes, the implication of an object-based approach is quite straightforward. Besides the intraframe spatial redundancy, the coding efficiency of most video codecs is largely dependent on how well they can exploit the temporal redundancy via accurate motion estimation. In this respect, an object-based scheme has even more leverage as motions are associated with actual objects, therefore avoiding the larger interframe differences which would otherwise take place due to indiscriminate treatments of image blocks across object boundaries. If object motions are accurately modelled and estimated, it may well result in a smaller data overhead for motion vectors as only one set of motion parameters are required for each object from the segmentation masks.

Segmentation, or generation of the video object plane at each frame location, is considered a non-normative part of the standard and can often be performed beforehand and independently of both the encoding and decoding processes. Nevertheless, reliability of segmentation results is essential to maintain the integrity of an object-based video coding scheme.

6.3 Sprite coding of objects

An attractive feature of object-based video compressions is sprite coding [92], which has the potential for significant savings in the transmission bandwidth. Based on the assumption that an object is omnipresent between scene cuts, in the absence of spatial undersampling, and knowing the trajectory and velocity along the sequence, it becomes feasible to reconstruct the presence of an object at a current frame using its reference image from any of the other frames, excluding areas of the object may have been occluded at the reference frame. For coding purposes, a sprite for each object is a reference image, from which a complete representation of the object can be rendered at any instance of time in the sequence. While this manipulation is often seen as being more applicable to a background object being transformed by a global motion, the same treatment can also be given to other objects in a multiple-object environment, as long as their sprite images are also properly created. Subject to these preconditions, a video sequence can be reconstructed at the receiver from three sets of data: (a) Sprite images (one for each object), (b) Segmentation masks at all frames, and (c) transformation parameters relating each object to their reference sprite image. As a consequence, the actual texture details of the pictures may only need to be transmitted once for each object, for the duration of the video. As compared with an approach where intra-coded images (I-frames) are required at more frequent frame intervals, a sprite-based scheme promises a substantial reduction in transmission bit rate through a more efficient and global reuse of coded texture. In addition, the flexibility and compact representation of an object-based scheme holds the potential for further improvement in applications such as frame rate conversion, where a temporal interpolation can be realized through the object trajectories rather than directly in picture fields or textures.

An object sprite is created with reference to a designated frame in the sequence. Figure 6.1 illustrates such a scenario. The shape masks and textures corresponding to the object from other frames are registered towards the reference, using the motion parameters estimated based on the texture inside the object mask. All the

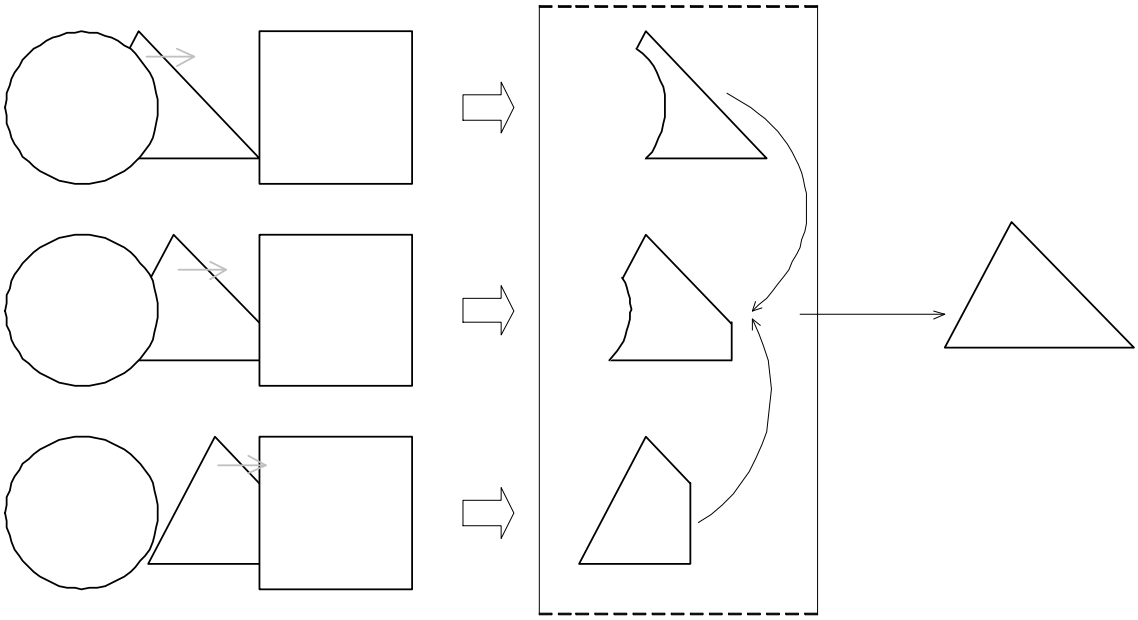


Figure 6.1: Sprite generation

registered images are then combined to produce the sprite, shown on the right hand side of this figure, by using a temporal sampling operation at each pixel location such as a median or mean filter. In generating a sprite image, occluded areas of an object in the reference frame can be recovered from other images in the sequence, hence contributing to a more complete reconstruction of the object image in the sprite. In this illustration, even though the triangle object is not fully visible in any individual frame, the generated sprite contains its entire content, therefore enabling reconstruction of the object at any of the contributing frames.

Figure 6.2 shows a generated sprite for the wallpaper on the "Mobile and Calendar" sequence of the first 30 frames, using frame number 15 as the reference. The white box identifies the location of this reference frame within the sprite. In this example, the sprite is of a larger dimension than the frame size as the result of the panning movement of the camera. In addition, the sprite also contains visible areas which are usually occluded by other objects, such as the background behind the ball and the train.

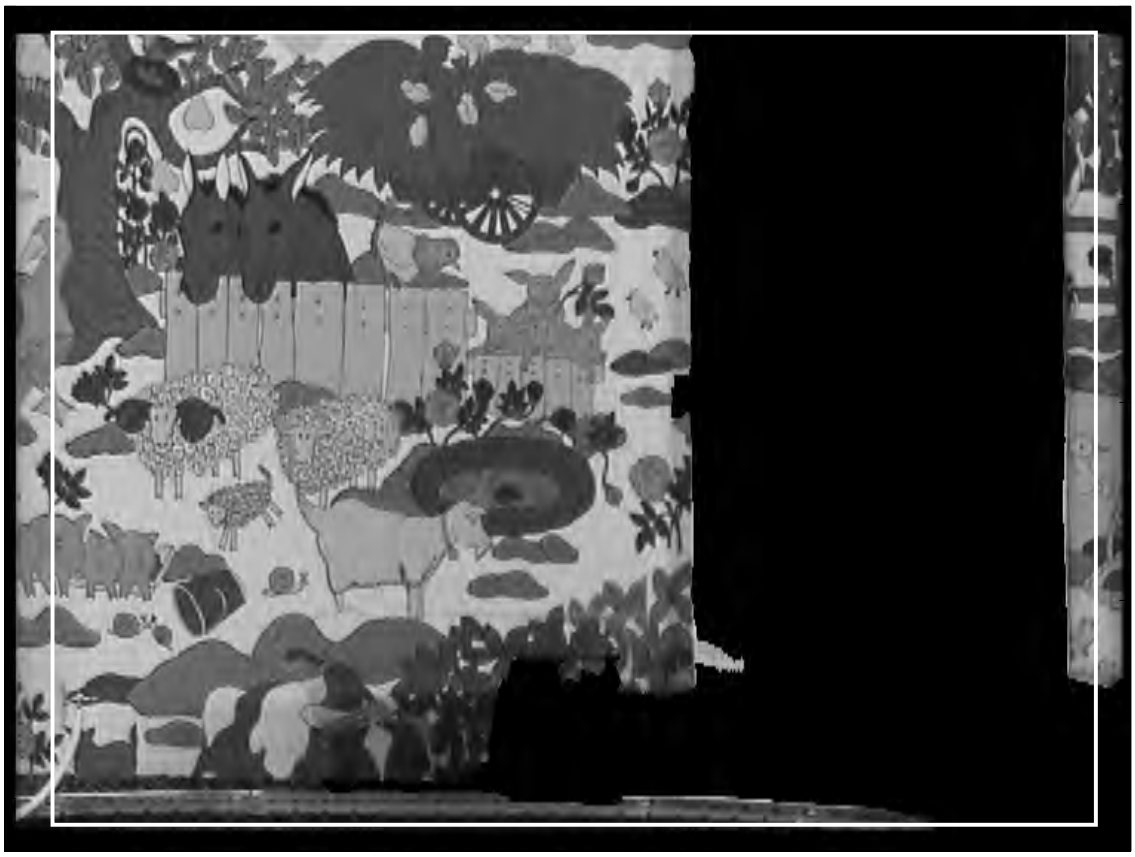


Figure 6.2: Sprite of the wallpaper object, "Mobile and Calendar". The box marks the position of the reference frame.

6.4 Sprite representation at super-resolution

A sprite can be considered as a form of temporal compression, representing an attempt to build a panoramic view which encompasses every individual shot of an object as it is portrayed throughout a sequence. Creating a sprite is often a lossy process, partly because of temporal sampling in construction of the sprite image, and therefore the quality of individual images is usually affected when reconstructed at the decoder. Assuming that motion between the object and its sprite has been estimated accurately, spatial undersampling is one of the remaining factors and has the greatest effect on the reconstructed images.

According to sampling theories, a perfect reconstruction from samples can be attained for a continuous signal if it is sampled at or above the Nyquist sampling rate,

which is twice the highest frequency in the original signal [93]. For a video image, these ideal sampling conditions would allow a picture to be reproduced at any arbitrary shift, including sub-pixel, from the reference in a sprite with little reduction in quality. In practice, however, such conditions are rarely achieved in a digital image system, due to the bandwidth limit of the optical path in a camera, and resolution of the image sensor [94]. This gives rise to the problem of undersampling, also known as aliasing, in which high-frequency components masquerade lower frequency components. In the presence of aliasing, if two pictures are taken of the same scene but related by a non-integer shift, then one cannot be completely reconstructed from the other and vice versa. Likewise, if fractional motions are involved when registering images toward the reference frame, some information would be lost upon those registered images being combined into the sprite, resulting in poorer quality for objects reconstructed from this sprite. An approach to alleviate the effects of aliasing is to consider using a super-resolution version of the sprite image [95].

The super-resolution approach takes advantage of the sub-pixel shifts which exist between different images of the same scene [96] to construct a view at a higher resolution. A super-resolution image can be constructed under a frequency-based or a spatial-based approach. Advantages of each method have been considered in a previous study [97], where a spatial-based approach is perceived as more versatile as it accommodates a wider range of object motions, beyond a translational model which is usually a prerequisite for a frequency-domain approach. In particular, in conjunction with the results from previous chapters which rely on parametric motions for segmentation and long-term stabilisation of the object shape masks, it would be feasible to create a super-resolution sprite for each individual video object based on a parametric approximation of its motion. Super-resolution sprites are then seen as a means to achieve an improvement in the quality of transmitted pictures.

Figure 6.3 shows a simplified view of a video compression system which utilises object-based representations and super-resolution sprites for coding purposes. Under this approach, the super-resolution sprites are intra-coded at the encoder, together

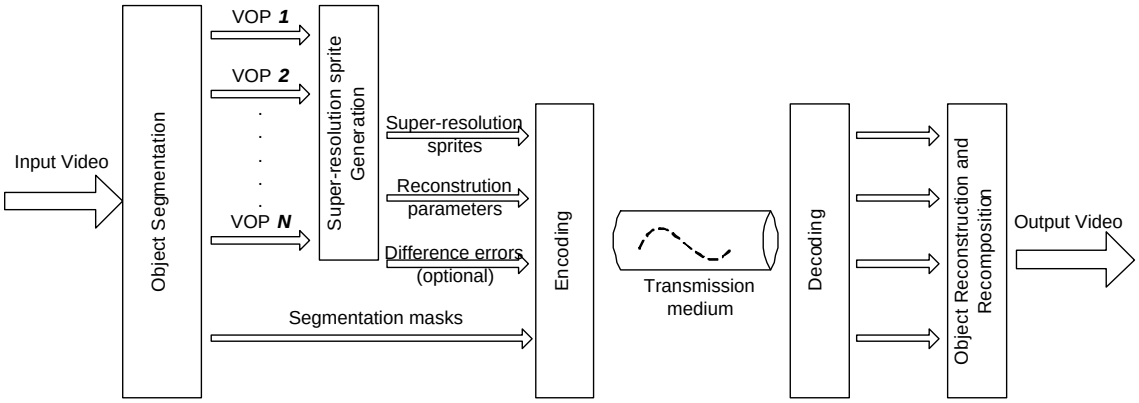


Figure 6.3: Object-based coding with super-resolution sprites

with the parameters required to render the objects inside segmentation masks from the sprites. The differences between each recomposed frame and its original are also encoded as an optional component, as discrepancies may occasionally be present in the reconstructed objects due to interpolation errors in sampling to and from the super-resolution sprites, incorrect motion parameters when the object has moved too far from the reference frame, or irregularities from the segmentation masks.

A super-resolution sprite is larger than a standard sprite, and in this coding scheme is specified at double the resolution of the original video. The entire set of object sprites therefore require considerably more storage and transmission bandwidth than a single intracoded video frame. The justification is the fact that this information only needs to be updated once for each scene. With persistent segmentation masks, there are potential savings in data rate, and improvements in the consistency of the reconstructed video.

6.5 Experiments and results

According to the coding framework in Figure 6.3, the object-based encoding process is performed as follows:

- Object sprites are generated at double the resolution of the original video using the technique described in [95].

- Sprite images are encoded using JPEG-2000 Region-of-Interest [98].
- Segmentation masks are encoded as binary shape coding, using the technique defined in MPEG-4 [90].
- Motion relating each object to its reference sprite are approximated as an eight-parameter projective model, and parameters are encoded as 20-bit floating-point numbers.
- Motion-compensated prediction difference images are encoded using H.264 [99].

At the decoder, textures inside the masks for each object are first reconstructed at the high resolution of the object sprites. They are then recomposed into individual pictures also at the high resolution, before being downsampled to the same resolution of the original video.

The first experiment is based on 100 frames from the sequence "Mobile and Calendar", which is presented in color (YUV 4:2:0) at the frame resolution of 704x512 and using interlaced sampling. Figure 6.9-a shows the original odd field of frame 45. The four objects present are the wallpaper, calendar, train and ball, all of which are initially encoded as double-resolution sprites. The difference signals are also encoded for this sequence.

Figure 6.4-a shows the quantitative performances between the object-based coding using super-resolution sprites and block-based H.264 Advance Video Codec (AVC) using 5 reference frames. Only the first frame of the H.264 video is intracoded (I-frame), and the subsequent frames are coded as consecutive groups of BBP frames. In this example, the improvement in peak signal-to-noise ratio (PSNR) of the proposed object-based approach using super-resolution sprites over H.264 is between 1 to 2 dB.

Figure 6.5-a shows an example of a reconstructed frame at the decoder using the object-based method at the data rate of $417kb/s$, as compared against a reconstruction from a H.264 coded video at a higher data rate in Figure 6.5-b. An enlarged view for the texture inside two inset windows from each reconstructed frame is also

shown in Figure 6.6. From these enlarged images, it can be seen that the reconstructed frame from the super-resolution sprites is able to retain many of the finer details from the original video frame. On the other hand, much of these details have been lost in the H.264 reconstructed frame, such as the grain of the red tree at the top left corner (see inset window (a)).

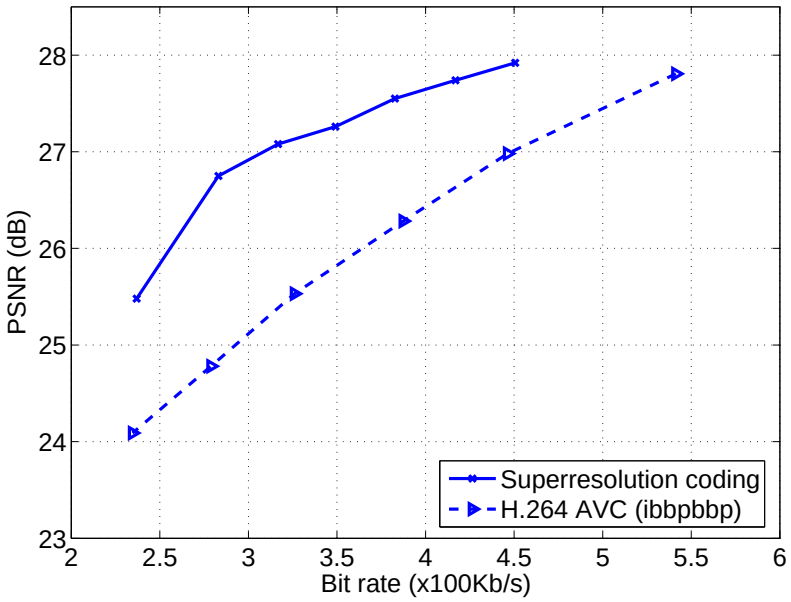
Texture (kb/s)	*Difference (kb/s)*	*Total Bitrate * (kb/s)*	*PSNR (dB)*
84.5	63.7	236.9	25.48
134.0	60.4	283.1	26.75
168.0	60.1	316.8	27.08
201.6	58.9	349.2	27.26
235.6	58.4	382.7	27.55
270.1	58.1	417.0	27.74
304.1	57.7	450.6	27.92

* Includes shapes coded at 46.45 kb/s and motion parameters at 42.26 kb/s

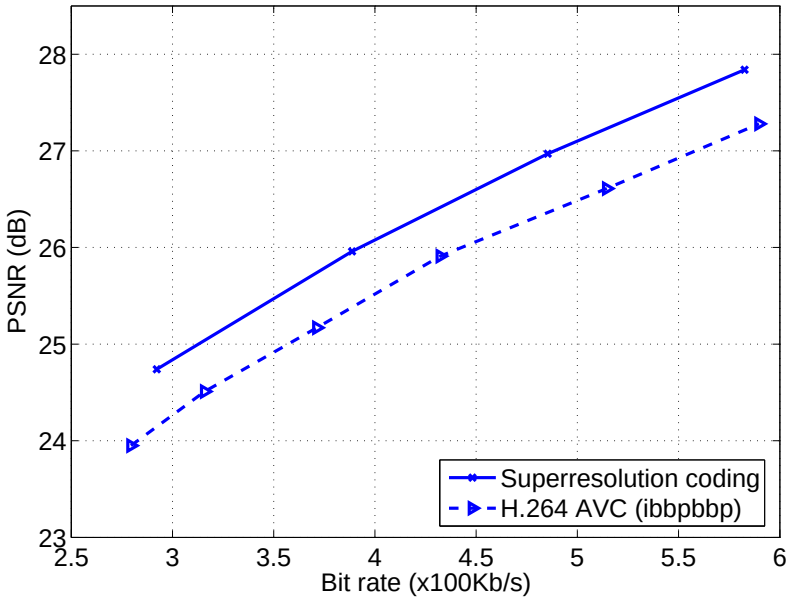
Table 6.1: Bandwidth allocations in "Mobile and Calendar"

Table 6.1 shows the detailed bandwidth allocations given to each of the components in this coding scheme, at different data rates and signal-to-noise ratios. The variation in the total bandwidth is induced primarily by the level of quantisation applied to the super-resolution sprites at the encoder. As shape masks and motion parameters are crucial for reconstructions at the decoder, the bandwidth dedicated to these two components remains unchanged in the experiment.

The second experiment is carried out on the first 40 frames of the sequence "Flower Garden" at the resolution of 688x512, also in YUV 4:2:0 format and interlaced sampling. While the object masks exist for a longer series of frames, it is more difficult to obtain accurate motion to generate the sprites, because of the large displacements of objects from the reference frame. For the purpose of sprite generation, the sky and the house in this example are considered as one single object, due to a lack of texture on the sky. The difference signals are not encoded in this example, as the



(a) "Mobile and Calendar"



(b) "Flower Garden"

Figure 6.4: Rate-distortion performance of the object-based super-resolution codec against H.264

shorter video duration reduces the likelihood that a reconstructed video frame is affected by undersampling, and the object masks are also well-maintained over all frames.

As observed from the rate distortion performance in Figure 6.4-b, the object-based coding approach offers an improvement in PSNR of approximately 0.6dB over the H.264 video at a similar data rate. A closer inspection of the reconstructed images using both methods in Figure 6.7, as well as the enlarged textures from the inset windows in Figure 6.8, also shows that image details obtained by the object-based reconstruction remain considerably sharper than those taken from the H.264 reconstructed frame. Note that in its current form, the object-based method does not implement any extra post-processing on the boundary regions between arbitrarily-shaped objects on the reconstructed frame. Techniques such as feathering, or color blending within these regions may further improve the subjective quality of the reconstructed video.

(a) Object-based codec with super-resolution sprites at 417 kb/s.

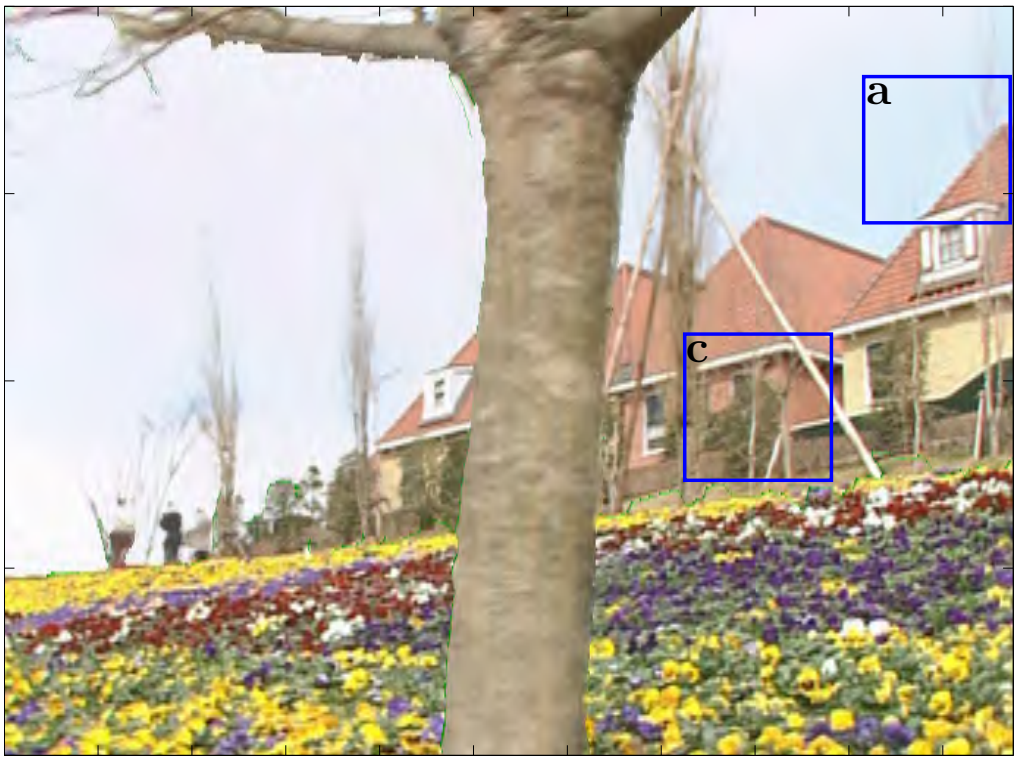
(b) H.264 AVC at 446 kb/s

Figure 6.5: Reconstructed "Mobile and Calendar" at frame 45. Zoomed-in details inside the inset windows are also shown Figure 6.6

(a) (b)

(c) (d)

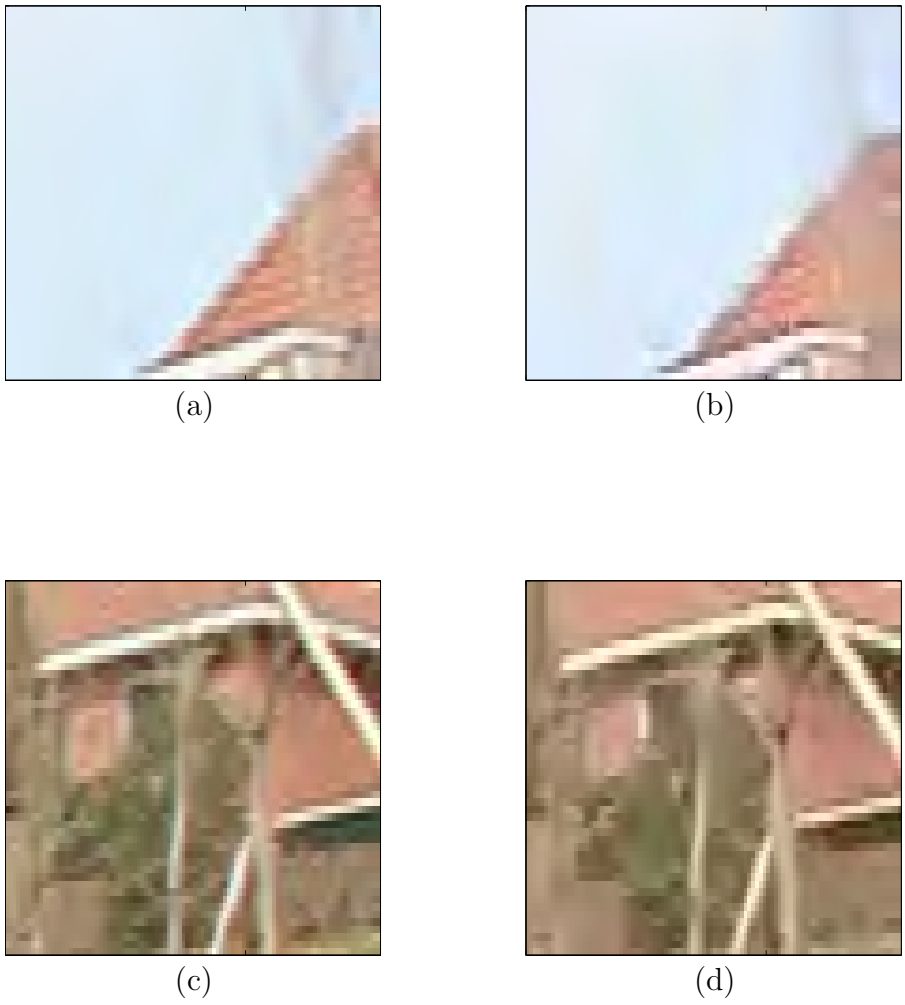
Figure 6.6: "Mobile and Calendar": (a) and (c) - Details extracted from an object-based reconstructed frame; (b) and (d) - Details extracted from a H.264 reconstructed frame



(a) Object-based codec with super-resolution sprites at 583 kb/s

(b) H.264 AVC at 589 kb/s

Figure 6.7: Reconstructed "Flower Garden" at frame 5. Zoomed-in details inside the inset windows are also shown in Figure 6.8



(a) (b)

(c) (d)

Figure 6.8: "Flower Garden": (a) and (c) - Details extracted from an object-based reconstructed frame; (b) and (d) - Details extracted from a H.264 reconstructed frame

(a) "Mobile and Calendar" at frame 45

(b) "Flower Garden" at frame 5

Figure 6.9: The original video frames

6.6 Summary

This chapter has presented an object-based video compression framework which combines the use of segmentation masks and super-resolution sprites. The examples have shown that there are considerable improvements to be obtained by using the proposed method, both in terms of PSNR and visual quality of the reconstructed video frames. In addition, this coding approach has the potential to offer a range of object-based functionalities at both the encoder and decoder.

On the other hand, there also exist a number of deficiencies in this coding approach. At its current design, the method is not suitable for real time applications, as both procedures to refine the segmentation masks and to generate the object sprites require the availability of many future frames in a sequence. In addition, while intensive computation is involved in both segmentation and sprite generation, there are redundancies which have not been exploited in the process. For example, motion estimation has been performed independently at both stages, whereas a reuse of such information may lead to more efficient processing.

Another factor which affects the performance of this object-based coding method is the frame-length for which each object sprite can accommodate. Due to the initial bandwidth required to transmit the super-resolution object sprites, having fewer frames between two scene cuts would increase the average data rate, whereas a scene with more frames would lead to further reduction of the overall bandwidth requirement. Depending on the particular requirements of an application, one may contemplate alternating between various coding modes at different sections of a video sequence, so as to achieve an optimum performance.

Chapter 7

Conclusions

7.1 Summary of results

This thesis has put forward a systematic framework for the extraction of moving objects from a video sequence. The framework is based on an understanding that in order to gain a good segmentation mask, correlations at both spatial and temporal domains need to be exploited in a complementary manner.

A conventional spatio-temporal segmentation approach is usually based on the notion of using a complete spatial segmentation as the basic building blocks, to be clustered into objects by the temporal similarity amongst their corresponding motions. Although still being a valid and logical proposition, it overlooks the fact that regardless of the spatial content, a video is often made up of a small number of objects, implying that temporal integrity may already exists in most parts of a picture. An initial spatial segmentation in such parts would at best become redundant in the segmentation result, and at worst the increased number of segments it produces would affect the complexity and stability of the object classification. While a region clustering stage is almost always present in one form or another, it would be beneficial to confine such processes to the areas of greatest importance, which are in the vicinity of moving object boundaries, rather than performing it ubiquitously. In

addition, such an approach also reduces the effects that errors from an initial spatial segmentation may have on regions outside the targeted area.

To identify the areas where segmentation should be approached by clustering, a new scheme for the detection of moving object boundaries was proposed in Chapter 3. Under a block-based approach and using motion information obtained from phase-correlation, it aimed to locate and classify the image blocks which are more likely to straddle a boundary between different moving objects, or blocks whose estimated motion is unlikely to be correct or unique. The detection criterion built on the characteristics of a phase-matched difference image, which was created by taking the difference only between the Fourier magnitudes of two motion-compensated blocks, and matching of the phase components by reusing the phase of one block on the other. The key findings of this chapter were:

- When there is no moving object boundary inside a block, and the estimated motion is within the sub-pixel neighbourhood of the true motion, the phase-matching operation results in a difference image not only with less energy than a conventional motion-compensated difference, but also characterised by a strong attenuation imposed on its low-frequency components.

- When there is a moving object boundary inside a block, or the estimated motion does not adequately address the underlying motion, the rise in energy of a phase-matched difference image is accompanied by an even more dramatic rise in the proportion of energy in its low-frequency components, even when the presence of non-dominant motions are relatively small. This sharp change in the proportion of low-pass energy is further emphasised by the stronger suppression of the low-frequency components in those blocks without a moving object boundary.

- The proportion of low-pass energy in the phase-matched difference image can be used as a detection criterion to discriminate boundary blocks from those without a moving object boundary. This discrimination was also shown as be-

ing more sensitive to outliers than measures such as sum-of-squared differences on a motion-compensated difference image.

- The phase-matched difference image can be modelled analytically to substantiate the above findings

- The detection measure was shown as extendable to parametric motions in general, provided that an initial estimation of the dominant motion is unambiguous.

Results of the block-based boundary detection scheme represent a coarse segmentation, which then allows the spatio-temporal efforts to be focused more on the image regions which are either boundary blocks, or blocks having an unreliable motion. Subsequent work in Chapter 4 aimed to resolve the object boundaries to a pixel-accurate level, via a spatio-temporal segmentation on the selected area. This process is initiated by first partitioning the area into spatially-coherent segments using a color quadtree approach, followed by a motion-based region clustering stage. The flexibility in formulation of this chapter is reflected by two expansions: a move beyond the block-based framework to a region-based algorithm, and upgrading of motion representations to an affine model for both estimation and region merging. Two features which are seen as key contributions to the results of this chapter are:

- A self-expanding quadtree: The spatial segmentation is only useful if the boundary of each region is not preconditioned by the block boundary. If the color segmentation encounters a "flat" block en route, indicated by having only a single node in the quadtree at that block, then it automatically extended the segmentation to neighbouring blocks until a block with authentic texture was found. In other words, while this segmentation was spatially confined by design, it was not texture-blind.

- Region merging using a three-frame approach: Because motion estimation was carried out in only one direction, i.e. between the current and previous frames, region merging using these two frames tended to be affected by the bias

of the objective function used by the estimation. For example, it is difficult to dismiss an overfitting motion on a small region as a misfit, because after all it is still optimised for the best motion-compensated difference on that region. However, inclusion of an extra picture, in this case the next frame, alleviated this prejudice in region merging. In the previous example, assuming a constant velocity over the short duration of three frames, a reversal of the overfitting backward motion is likely to yield a poor performance in the forward direction between current and next frame, whereas an accurate motion should maintain its integrity in both directions. A three-frame approach therefore allowed a more objective consideration for fitness of a motion over neighbouring regions.

The segmentation masks produced at the end of Chapter 4 were the results of an integration between the earlier boundary detection and the above spatio-temporal segmentation, and compared favourably with another technique based on an indiscriminate spatial segmentation. The proposed algorithm, however, elected not to impose elaborate, but often adhoc, criteria for "clean-up" of some occasional oversegmentations, by arguing that a more sustainable approach is to consider such problems in the context of long-term, rather than individualistic, segmentations.

This argument was jointly supported by the framework of Chapter 5, which aimed to reach a temporally-stable object representation through the collection of individual segmentations. The position taken in this chapter was that if the masks are not temporally stable, it is because they are not yet accurate, therefore rectifying such defects would improve the temporal stability. The stability objective was twofold, (a) stability of the presence of objects throughout the sequence, and (b) stability of the object boundaries. By enforcing the former criterion, oversegmentation and undersegmentation errors were removed because such instances are often represented as isolated and irregular segments. The second criterion was achieved via temporal averaging of the object masks, after the stability of object presence had been established. This chapter also saw the adoption of a 12-parameter projective motion model, to replace the affine model used in Chapter 4, in order to achieve more accu-

rate mask-registration results when respective frames are positioned far apart. Key features of this stabilisation process were:

- Use of a temporal "referencing" framework to validate the presence of individual masks: Support for a mask at one frame was evaluated by whether it had been *referred to* by masks at other frame in the sequence. The irregularities of an over-segmented region, together with its often incorrect motion, translated to a low (or zero) number of referees. An under-segmented region, on the other hand, was detected by a presence of double references from another frame.

- Adaptive thresholding for consistency checking on occluded parts of an object mask: Boundary smoothing was performed by a consistency check amongst all the masks registered towards a current frame. With a fixed number of frames in use and stabilisation performed concurrently on multiple objects, an object may be foreground to one but background to another. The consistency check-level for a mask pixel was reduced according to the number of times it is occluded at other frames in the sequence.

- Concatenation of stabilised segmentation results between overlapping, fixed-length sequences of frames to make the algorithm expandable, without an excessive reliance on a single reference frame over a longer sequence.

At the end of this chapter, objective measures were employed to show that the final object masks were more temporally consistent than the initial set of individual segmentation, and this conclusion was also derived from a visual inspection of the masks.

The last chapter presented, as a proof-of-concept experiment, an integrated object-based video compression system. With the segmentation results inherited from Chapter 5, along with the assumption of parametric transformations for object motions, the encoding process represented each temporal object as one static sprite, together with the corresponding set of binary masks and reconstruction parameters. The sprites were generated at super-resolution to reduce effects of spatial under-

sampling and to improve the quality of reconstructed objects. The coding efficiency surpassed the contemporary H.264 coding in terms of PSNR performance, and was also shown as offering reconstructed images at a higher visual quality.

7.2 Considerations for further research

Work in this thesis has been built in successive stages, where the results from each stage have a propagating effect further on. The accuracy of the output at each stage, along with the stability of the overall algorithm, is affected by the performance of its preceding stage. While there are a number of fail-safe mechanisms built into the implementation to detect and prevent propagation of errors, activation of such processing is costly and unhelpful for performance efficiency. Each step in the algorithm is therefore treated conservatively to ensure that they do not cause irreversible defects, or errors that may frequently require a backtrack in subsequent processing to correct. In hindsight, there exist a number of venues, both within and beyond the current proposed framework, where further investigations may lead to an improvement, or open up possibilities for future research. Some of these considerations are elaborated in this section.

The phase-base detection of moving object boundaries in Chapter 3 deals well with translational motions, and was also shown as being adaptable to an affine model as long as the dominant motion parameters are estimated accurately. In the actual implementation, the use of affine representations started in Chapter 4, for motion estimation and region clustering. The reservation from using this model at the boundary detection stage stems from the specific concern that in a two-dimensional windowed representation, a discontinuity between two *slightly different* translational motions may also be ambiguously modelled as an affine motion, an interpretation which might compromise the boundary detection scheme. Further investigations may be carried out to assert integrity of local motion interpretation, for example by considering the coherency of the estimated motion in overlapping windows, where an

ambiguous model may produce less coherent results. A successful adaption of affine motion in the detection stage is expected to reduce the processing load currently required at the regions grouping stage.

The second consideration is given to the assumption of 3-D rigid motions for object movements, which was used as the basis for many decisions in regions merging and object boundary smoothing. An extension may involve a hybrid algorithm, which can accommodate both rigid and non-rigid transformations. The strength of a rigid-motion classification is the ability to separate adjacent objects of distinctive motions, whereas a strength of a foreground extraction method is its capacity to identify contiguous foreground regions without having to explicitly specify the region's motion. For example, upon detection of a boundary region, a hybrid approach may choose to regularise the dominant motion by a rigid model, while relaxing this constraint on non-dominant motion, so as to achieve a form of localised foreground/background classification which can then be extended to a flexible global object classification. Such processes should also consider efficient coding for non-rigid foreground objects.

With reference to the sprite-based representation of an object, a number of issues may be explored, such as:

- Using the sprite image as feedback to improve the segmentation mask at individual frames. By comparing texture within the mask to the corresponding texture in a sprite, wrongly-segmented regions may be corrected.

- Relations between coding efficiency and the number of object-frames accommodated in each sprite image.

- Effects of latency at the receiver, as an upfront transmission of all the object sprites are required before decoding can commence.

- The benefit of sending the difference signal to update a sprite-based reconstructed object may also be considered subjectively. While the updating is helpful to reduce errors as well as improving the signal-to-noise ratio in sprite-base coding of a long object sequence, it may introduce unnecessary flickering

effects on some objects due to the quantisation of the difference signal. On the other hand, this difference signal can also be sent adaptively with consideration for local image structure and texture level.

7.3 Concluding remarks

At the center of most video object segmentation techniques is a method to realize the object abstractions through a physical means of spatial and temporal correlations. It is often accompanied by a number of additional constraints as an artificial interpretation of the true object semantics, in an attempt to establish orders within an otherwise uncorrelated stream of data. This work has relied on a phase-based boundary detection scheme to devise a complementary use of spatial and temporal information, where the usually more arduous spatial support is resorted to when temporal coherency is unverified. The additional constraints have been realized on the assumption of parametric object motions, progressing from translation and affine models for adjacent-frame segmentation, to projective model for stabilisation of videos at length. The author would like to think of the work presented here as a small but significant contribution towards the contemporary efforts to achieve a compact and precise representation of a video sequence as a composition of semantically-meaningful objects, the results of which would directly benefit many high level and practical applications.

Bibliography

[1] D. Regan, *Spatial Vision*, vol. 10 of *Visual and Visual Dysfunction*, chapter 1, "A Brief Review of Some of the Stimuli and Analysis Methods Used in Spatiotemporal Vision Research", pp. 1–42, Macmillan Press, 1991.

[2] M.J. Tarr and M.J. Black, "A computation and evolutionary perspective on the role of representation in vision," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 65–73, July 1994.

[3] Y. Aloimonos, "What have I learned," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 74–85, July 1994.

[4] R. Jain, "Expansive vision," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 86–88, July 1994.

[5] C.M. Brown, "Toward general vision," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 89–91, July 1994.

[6] S. Edelman, "Representation without reconstruction," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 92–94, July 1994.

[7] J.K. Tostsos, "There is no one way to look at vision," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 95–97, July 1994.

[8] M.A. Fischler, "The modeling and representation of visual information," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 98–99, July 1994.

[9] J.K. Aggarwal and W.N. Martin, "The role of R & R in vision: is it a matter of definition?," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 100–102, July 1994.

[10] H.I. Christensen and C.B. Madsen, "Purportive reconstruction: a reply," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 103–108, July 1994.

[11] G. Sandini and E. Grosso, "Why purportive vision," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 109–112, July 1994.

[12] M.J. Tarr and M.J. Black, "Reconstruction and purpose," *CVGIP: Image understanding*, vol. 60, no. 1, pp. 113–118, July 1994.

[13] V.S. Ramachandran, *The Emerging Mind*, Reith Lectures. BBC Radio, http://www.bbc.co.uk/radio4/reith2003/lectures.shtml, April 2003, Lecture 2: "Synapses and the Self".

[14] B.K.P. Horn and B.G. Schunck, "Determining optical flow, A.I. memo no. 572," Tech. Rep., MIT, April 1980.

[15] R.A. Wilson and F.C. Keil, Eds., *The MIT encyclopedia of the cognitive sciences*, The MIT Press, 1999.

[16] K. Koffka, *Principles of Gestalt Psychology*, Harcourt Brace., 1935.

[17] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1993.

[18] N. Otsu, "A threshold selection method from grey-level histograms," *IEEE Transansactions on Systems, Man, and Cybernetics*, , no. 1, pp. 62–66, 1979.

[19] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for content based image retrieval," in *IEEE International Conference on Image Processing*, 2002.

[20] N. Herodotou, K.N. Plataniotis, and A.N. Venetsanopoulos, "A color segmentation scheme for object-based video coding," in *IEEE Symposium on Advances in Digital Filtering and Signal Processing*, 1998, pp. 25–30.

[21] S. Vitabile, G. Pollaccia, G. Pilato, and E. Sorbello, "Road signs recognition using a dynamic pixel aggregation technique in the HSV color space," in *International Conference on Image Analysis and Processing*, 2001, pp. 572–577.

[22] S.Y. Wan and W.E. Higgins, "Symmetric region growing," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1007–1015, September 2003.

[23] L. Vincent and P. Soille, "Watershed in digital spaces: an efficient algorithm based on immersion simulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–589, June 1991.

[24] P.V.C. Hough, "Methods and Means for Recognizing Complex Patterns," U.S. Patent 3,069,654, 1962.

[25] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[26] T. Meier and K.N. Ngan, "Video segmentation for content based coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1190–1203, December 1999.

[27] M.S. Landy and N. Graham, *The visual neurosciences*, pp. 1106–1118, MIT Press, 2004.

[28] B.S. Manjunath and W.Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 18, pp. 837–842, August 1996.

[29] W.Y. Ma and B.S. Manjunath, "EdgeFlow: a technique for boundary detection and image segmentation," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1375–1388, August 2000.

[30] L. Shafarenko, M. Petrou, and J. Kittler, "Automatic watershed segmentation of randomly textured color images," *IEEE transaction on image processing*, vol. 6, no. 11, pp. 1530–1544, November 1997.

[31] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, August 2000.

[32] J. Weickert, *Anisotropic diffusion in image processing*, Teubner, 1998.

[33] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, Nov 1990.

[34] L. Vincent, "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 176–201, April 1993.

[35] J.Y.A. Wang and E.H. Adelson, "Representing moving images with layers," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 625–638, September 1994.

[36] G.D. Boshukov, G. Bozdagi, Y. Altunbasak, and A.M. Tekalp, "Motion segmentation by multistage affine classification," *IEEE Transactions on Image Processing*, vol. 6, no. 11, pp. 1591–1594, November 1997.

[37] H. Zheng and S.D. Bolstein, "Motion-based object segmentation and estimation using the MDL principle," *IEEE transaction on image processing*, vol. 4, no. 9, pp. 1223–1235, September 1995.

[38] T. Zaharia and F. Preteux, "Parametric motion models for video content description within the MPEG-7 framework," in *Proceedings International Symposium Communications*, December 2002, pp. 415–422.

[39] Y. Altunbasak, P.E. Eren, and A.M. Tekalp, "Region-based parametric motion segmentation using color information," *Graphical Models and Image Processing*, vol. 60, no. 1, pp. 13–23, January 1998.

[40] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatialtemporal segmentation based on region merging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 9, pp. 897–915, September 1998.

[41] D. Tweed and A. Calway, "Motion segmentation based on integrated region layering and motion assignment," in *Proceedings of Fourth Asian Conference on Computer Vision*, January 2000, pp. 1002–1007.

[42] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 539–546, September 1998.

[43] C.K. Tan and M. Ghanbari, "Using non-linear diffusion and motion information for video segmentation," in *International Conference on Image Processing*. IEEE, 2002, vol. 2, pp. 769–772.

[44] J.G. Choi, S.W. Lee, and S.D. Kim, "Spatio-temporal video segmentation using a joint similarity measure," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 279–286, April 1997.

[45] D.W. Murray and B.F. Buxton, "Scene segmentation from visual motion using global optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 2, pp. 220–228, March 1987.

[46] M.M. Chang, A.M. Tekalp, and M.I. Sezan, "Simultaneous motion estimation and segmentation," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1326–1333, September 1997.

[47] I. Patras, E.A. Hendriks, and R.L. Lagendijk, "Video segmentation by MAP labeling of watershed segments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 326–332, March 2001.

[48] S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," in *Computer Vision and Pattern Recognition*, 2001, vol. 2, pp. 746–751.

[49] S.Y. Chien, S.Y. Ma, and L.G. Chen, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 577–585, July 2002.

[50] A. Neri, S. Colonnese, G. Russo, and P. Talone, "Automatic moving object and background separation," *Signal Processing*, vol. 66, pp. 219–232, 1998.

[51] F. Dufaux, F. Moscheni, and A. Lippman, "Spatio-temporal segmentation based on motion and static segmentation," in *IEEE International Conference on Image Processing*, 1995, pp. 306–309.

[52] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motions," *IJCV*, January 1994.

[53] Y. Tsaig and A. Averbuch, "Automatic segmentation of moving objects in video sequences: a region labeling approach," *IEEE Transactions On Circuits And Systems For Video Technology*, vol. 12, no. 7, pp. 597–612, July 2002.

[54] R. Castagno, T. Ebrahimi, and M. Kunt, "Video segmentation based on multiple features for interactive multimedia application," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 562–571, September 1998.

[55] D.D. Giusto, F. Massidda, and C. Perra, "A fast algorithm for video segmentation and object tracking," in *International Conference on Digital Signal Processing*, 2002, pp. 697– 700.

[56] E. Chalom and Jr. V.M. Bove, "Segmentation of frames in a video sequence using motion and other attributes," in *SPIE Digital Video Compression: Algorithms and Technologies, 2419*, 1995, pp. 230–241.

[57] A.M. Tekalp, *Digital Video Processing*, Prentice Hall, 1995.

[58] T. Yoshida, H. Katoh, and Y. Sakai, "Block matching motion estimation using block integration based on reliability metric," in *IEEE International Conference on Image Processing*, 1997, vol. 2, pp. 152–155.

[59] A. Lundmark, H. Li, and R. Forchheimer, "Motion vector certainty reduces bit rate in backward motion estimation video coding," in *Proceedings of SPIE Visual Communications and Image Processing*, June 2000, vol. 4067, pp. 95–104.

[60] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *International Journal of Conputer Vision*, pp. 283–310, Feb 1989.

[61] C.D. Kuglin and D.C. Hines, "The phase correlation image alignment method," in *IEEE Conference on Cybernetics and Society*, 1975, pp. 163–165.

[62] R.N. Bracewell, *The Fourier Transform and Its Applications*, chapter 6, pp. 119–125, McGraw-Hill, 2000.

[63] A.V. Oppenheim and J.S. Lim, "The importance of phase in signals," in *Proceeding of the IEEE*, May 1981, vol. 69, pp. 529–541.

[64] R. Reininger and J. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Transaction on Communucations*, vol. COM-31, pp. 835–839, June 1983.

[65] S. R. Smoot and L. A. Rowe, "Study of DCT coefficient distributions," in *Proc. SPIE*, 1996, pp. 403–411.

[66] E.Y. Lam and J.W. Goodman, "A Mathematical Analysis of the DCT Coefficient Distributions for Images," *IEEE transaction on image processing*, vol. 9, no. 10, pp. 1661–1666, Oct 2000.

[67] S.S. Beauchemin and J.L. Barron, "On the Fourier properties of discontinuous motion," *Journal of Methematical Imaging and Vision*, vol. 13, pp. 1–19, 2000.

[68] S.S. Beauchemin and J.L. Barron, "The frequency structure of one-dimensional occluding image signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 200–206, February 2000.

[69] W. Yu, G. Sommer, and K. Daniilidis, "Multiple motion analysis: In spatial or in spectral domain?," *Conputer Vision and Image Understanding*, , no. 90, pp. 129–152, 2003.

[70] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 1994.

[71] D.J. Fleet and K. Langley, "Computational analysis of non-Fourier motion," Tech. Rep. RPL-TR-9309, Queen's University, Robotics and Perception Laboratory, Department of Computing and Information Science, April 1994.

[72] B.S. Reddy and B.N. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *IEEE Transactions on Image Processing*, vol. 5, no. 8, pp. 1266–1271, August 1996.

[73] R.N. Bracewell, K.-Y. Chang, A. K. Jha, and Y.-H. Wang, "Affine theorem for two-dimensional fourier transform," *Electronics Letters*, vol. 29, no. 3, pp. 304, February 1993.

[74] E.K.P. Chong and S.H. Zak, *An Introduction to Optimization*, Wiley, 2001.

[75] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, Addison-Wesley, 1989.

[76] B.D. Lucas and T. Kanade, "An iterative registration technique with an application to stereo vision," in *Proceedings DARPA Image Understanding Workshop*, 1981, pp. 121–130.

[77] L. Wiskott, "Segmentation from motion: combining Gabor- and Mallat wavelets to overcome the aperture and correspondence problem," *Pattern Recognition*, vol. 32, no. 10, pp. 1751–1766, 1999.

[78] L. Rade and B. Westergren, *Mathematics handbook for science and engineering*, Basel,Birkhauser, 1995.

[79] Y. Deng and B.S. Manjunath, "Netra-V: toward an object-based video representation," *IEEE Transactions On Circuits And Systems For Video Technology*, vol. 8, no. 5, pp. 616–627, September 1998.

[80] C.E. Erdem, F. Ernst, A. Redert, and E. Hendriks, "Temporal stabilization of video object segmentation for 3D-TV applications," in *International Conference on Image Processing*, October 2004, pp. 357–360.

[81] J. Konrad and M. Ristivojevic, "Video segmentation and occlusion detection over multiple frames," in *IS&T/SPIE Symposium on Electronic Imaging, Image and Video Communications*, January 2003.

[82] M. A. El Saban and B. S. Manjunath, "Video region segmentation by spatio-temporal watersheds," in *IEEE International Conference on Image Processing*, September 2003.

[83] F.M. Porikli and Y. Wang, "An unsupervised multi-resolution object extraction algorithm using video-cube," in *IEEE Int. Conf. Image Processing*, 2001, pp. 359–362.

[84] T. Darell and D. Fleet, "Second-order method for occlusion relationships in motion layers," Tech. Rep. 314, MIT Media Laboratory.

[85] S. Kruger, *Motion analysis and estimation using multiresolution affine models*, Ph.D. thesis, The University of Bristol, July 1998.

[86] H.C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 208, no. 1773, pp. 385–397, July 1980.

[87] A.M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, and J.S.B. Mitchell, "An efficiently computable metric for comparing polygonal shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 209–216, March 1991.

[88] C.E. Erdem, B. Sankur, and A.M. Tekalp, "Performance measures for video object segmentation and tracking," *IEEE Transactions On Image Processing*, vol. 13, no. 7, pp. 937–951, July 2004.

[89] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.

[90] ISO/IEC 14496-2, "Informationation Technology - Coding of Audio-Visual Objects - Part 2: Visual," in *International Standard*. International Organization for Standardization, December 2001.

[91] T. Sikora, "MPEG digital video-coding standards," *IEEE Signal Processing Magazine*, pp. 82–100, September 1997.

[92] K.N. Ngan, C.W. Yap, and K.T. Tan, *Video Coding for Wireless Communication Systems*, chapter 2, pp. 123–128, Marcel Dekker, 2001.

[93] A. Feuer and G.C. Goodwin, *Sampling in Digital Signal Processing and Control*, Birkhauser, 1996.

[94] G. de Haan, *Digital Signal Processing Handbook*, chapter 9, "Video Scanning Format Conversion and Motion Estimation", Boca Raton: CRC Press LLC, 1999.

[95] G. Ye, M. Pickering, M. Frater, and J. Arnold, "Super-resolution static sprite generation and its application to object-based video coding," in preparation, 2004.

[96] S.C. Park, M.K. Park, and M.G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, pp. 21–36, May 2003.

[97] S. Borman and R. Stevenson, "Spatial resolution enhancement of low-resolution image sequences: a comprehensive review with directions for future research," Tech. Rep., Laboratory for Image and Signal Analysis, University of Notre Dame, July 1998.

[98] D.S. Taubman and M.W. Marcellin, *JPEG2000 - Image Compression Fundamentals, Standards and Practice*, chapter 10, pp. 442–448, Kluwer Academic Publishers, 2002.

[99] ISO/IEC 14496-10, "Information Technology - Coding of audio-visual objects - Part 10: Advanced Video Coding," in *International Standard*. International Organization for Standardization, November 2003.