NATIONAL UNIVERSITY OF SINGAPORE

Video Segmentation: Temporally-constrained Graph-based Optimization

by

Liu Siying

A thesis submitted in partial fulfillment for the
degree of Masters of Engineering

in the
Faculty of Engineering
Department of Electrical and Computer Engineering

January 2010

NATIONAL UNIVERSITY OF SINGAPORE

Abstract

Faculty of Engineering

Department of Electrical and Computer Engineering

Master of Engineering

by Liu Siying

Video segmentation not only spatially performs intra-frame pixel grouping but also temporally exploits the inter-frame coherence and variations of the grouping. Traditional approaches simply regard pixel motion as another prior in the MRF-MAP framework. Since pixel pre-grouping is inefficiently performed on every frame, the strong correlation between inter-frame groupings is largely underutilized. In this work, spatio-temporal grouping is accomplished by propagating and validating the preceding graph that encodes pixel labels for the previous frame, followed by spatial subgraph aggregation subject to the validated labeling information. Graph propagation is achieved by a global motion estimation which relates two frames temporally, thus transforming the segmentation of the current frame into a highly constrained graph partitioning problem. All propagated pixel labels are carefully validated by similarity measures. Trustworthy labels are preserved and erroneous ones removed. The unlabeled pixels are merged to their labeled neighbors by pair-

wise subgraph merging. Experimental results show that the proposed approach is highly efficient for the spatio-temporal segmentation. It makes good use of temporal correlation and produces encouraging results.

Acknowledgements

This thesis would not have been successfully completed without the kind assistance and help of the following individuals.

First and foremost, I would like to thank my supervisors Associate Professor Ong Sim Heng and Dr. Yan Chye Hwang for their unwavering guidance and support through the course of this research. I am grateful for their continual encouragement and advice that have made this project possible.

I would like to express my heart-felt gratitude to Dr. Guo Dong, a senior research staff from DSO National Laboratories, for his generous sharing of knowledge and continual guidance on the subject of video segmentation.

I would also like to thank Mr. Francis Hoon, the Laboratory Technologist of the Vision and Image Processing Laboratory, for his technical support and assistance.

Last but not least, I would like to extend my gratitude to my fellow lab mates for their help and enlightenment.

Contents

Abstract **i**

Acknowledgements **iii**

List of Figures **viii**

List of Tables **xii**

1 Introduction **1**

1.1 The Video Segmentation Problem 1

1.2 Contributions 3

1.3 Organization of the Thesis 4

2 Background and Previous Work **5**

2.1 Image Segmentation: Spatial Grouping 5

2.1.1 The MRF-MAP Framework 5

2.1.1.1 Energy Minimization 7

2.1.2 Segmentation by Clustering 8

2.1.3 Graph-based Segmentation 10

2.2 Video Segmentation: Spatio-temporal Grouping 13

2.3 Previous Video Segmentation Approaches 16

2.4 Segmentation with Spatial Priority 17

2.5 Trajectory Grouping 18

2.5.1 Grouping by Motion Similarity 19

2.5.2 Grouping by Model Fitting 20

2.6 Joint Spatial and Temporal Segmentation 21

2.7 Summary of the Previous Approaches 23

3 Proposed Method **25**

3.1 Efficient Fusion of Spatial and Temporal Information 25

3.2 System Overview 26

3.3 Notation 27

3.4 Graph Propagation 29

3.4.1 Scale Invariant Feature Detection 30

3.5 Validation 30

3.6 Independent Motions 32

3.6.1 Regional Changes 33

3.7 Aggregation 38

3.7.1 Edge Information 39

3.7.2 Color 41

3.7.3 Shape 42

3.7.4 Cost 42

3.7.5 Complexity Analysis of Subgraph Aggregation 44

3.7.6 Algorithm 45

3.8 Connections to Transductive Learning 46

4 Experimental Results and Discussion **47**

4.1 Experiment Settings 48

4.1.1 First Frame Initialization 49

4.2 Segmentation Evaluation Methodology 51

4.3 Standalone Segmentation Quality Evaluation 53

4.3.1 Spatial Uniformity 53

4.3.2 Independent Motion 58

4.3.3 Newly Appeared Objects 58

4.3.4 Benefit of Temporal Propagation 60

4.4 Relative Segmentation Quality Evaluation 62

4.4.1 Overall Segmentation Evaluation 62

4.4.2 Comparison against State-of-the-art Video Segmentation . . 72

5 Future Work and Conclusions **77**

A Mathematical Models **80**

A.1 Markov Random Field (MRF) 80

A.1.1 MRF for Image Segmentation 81

A.2 Max-flow/Min-cut Algorithm 82

A.2.1 Ford−Fulkerson Algorithm 83

Bibliography **84**

List of Figures

2.1 This diagram shows a distribution of data points in the feature space. Mean Shift vector points towards the denser region in the feature space and converges at the mode of the data set through density gradient estimation. 9

2.2 (a) A graph $\mathbf{G}$ with 2 terminals S and T. (b) A cut on $\mathbf{G}$. Edge costs are reflected by thickness. 13

2.3 Structural flow of grouping along spatial and temporal axes. 15

2.4 Structure of grouping approaches with spatial priority. 17

2.5 Taxonomy of trajectory grouping approaches. 19

2.6 Taxonomy of joint spatial and temporal grouping approaches. . . . 21

3.1 Spatio-temporal grouping by the propagation, validation and aggregation of a preceding graph. 26

3.2 A strong temporal correlation implies similar grouping in most corresponding regions between two frames. (a) Grouping results in the previous image frame. (b) Pixel labels in the previous frame are propagated and validated in the current frame. About 94.25 % of labels are reusable in segmenting the current frame. 31

3.3 A pair of invalidated subgraphs due to whole region displacement. (a) The circle $\mathbf{g_1^-}$ in $\mathbf{I}^-$ and the pre-propagated location of its wrong prediction $\mathbf{g_2^-}$. (b) The predicted location of the circle is now at $\mathbf{g_1^p}$, while the correct location should be at $\mathbf{g_2^p}$. 33

3.4 Two invalidated subregions due to partial region displacement. (a) A rectangle $\mathbf{g_1^-}$ (orange) and the pre-propagated region of its wrong prediction in frame $\mathbf{I}^-$, annotated as $\mathbf{g_2^-}$ (green). (b) The actual location of the rectangle shifts to $\mathbf{g_2^p}$ and it partially overlaps with the predicted region $\mathbf{g_1^p}$. The non-overlapping subregions A and B are invalidated while the overlapping subregion C is validated. . . . 35

3.5 An invalidated subgraph due to a disappearing object. (a) A circle $\mathbf{g_1^-}$ in frame $\mathbf{I}^-$. (b) The circle disappears in $\mathbf{I}$, causing $\mathbf{g_1^p}$ to be invalidated. 35

3.6 An invalidated subgraph due to a newly appearing object. (a) $\mathbf{g_1^-}$ denotes the pre-propagation of a newly appeared circle. (b) A new circle appears in frame $\mathbf{I}$, causing $\mathbf{g_x}$ to be invalidated. Note that the subscript 'x' indicates that $\mathbf{g_x}$ is not a result of temporal propagation and it is yet to be grouped and labelled, whereas its pre-propagated subgraph $\mathbf{g_1^-}$ is labelled. 36

3.7 Three invalidated subregions due to region splitting. (a) A rectangle $\mathbf{g_1^-}$ in frame $\mathbf{I}^-$ and the pre-propagation of its separated parts denoted by $\mathbf{g_2^-}$, $\mathbf{g_3^-}$ and $\mathbf{g_{1B}^-}$ (the shaded regions). (b) The rectangle splits into two regions in $\mathbf{I}$, causing $\mathbf{g_2^p}$, $\mathbf{g_3^p}$ and $\mathbf{g_{1B}^p}$ to be invalidated. Only portions that still overlap with the split regions (solid yellow regions), $\mathbf{g_{1A}^p}$ and $\mathbf{g_{1C}^p}$, are validated. 37

3.8 A pair of invalidated labels of a single region due to region merging. a) Two rectangles $\mathbf{g_1^-}$ and $\mathbf{g_2^-}$ in frame $\mathbf{I}^-$ and the pre-propagation of the centre part of their merged version is denoted by $\mathbf{g_3^-}$. (b) The two rectangles merge into one region in $\mathbf{I}$, causing $\mathbf{g_{1A}^p}$, $\mathbf{g_{2B}^p}$ and $\mathbf{g_3^p}$ to be invalidated. Only portions that still overlap with the merged regions (solid yellow regions), $\mathbf{g_{1B}^p}$ and $\mathbf{g_{2A}^p}$, are validated. 38

3.9 If subgraphs $\mathbf{g}^i$ and $\mathbf{g}^j$ are to be merged to form $\mathbf{g}^k$, the strength of the boundary of these two subgraphs is the mean of all edge weights in $\mathbf{e}_k^B$ (denoted by black dotted lines). The strength of the joint between subgraphs $\mathbf{g}^i$ and $\mathbf{g}^k$ is computed as the mean of all edge weights in $\mathbf{e}^j$ (denoted by green dotted lines). 40

4.1 (a) and (b) Spatial uniformity (SI) of frames 1−30 of the "Table Tennis" Sequence and frames 10−35 of the "Coast Guard" Sequence respectively. The horizontal line marks the *SI* value of the initialized segmentation. The majority of the segmentation results have *SI* values close to that of the initialized segmentation. 56

4.2 (a) and (b) Spatial uniformity (SI) of frames 50−80 of the "Dog" Sequence and frames 1−20 of the "Coast Guard" Sequence respectively. The horizontal line marks the *SI* value of the initialized segmentation. The majority of the segmentation results have *SI* values close to that of the initialized segmentation. 57

4.3 (a)−(d) Segmentation results of frames 2, 5, 9 and 12 in the "Table Tennis" sequence by the proposed algorithm. The pingpong ball and human hand are segmented as independent moving objects. Note that pingpong ball is correctly associated despite no temporal overlapping after propagation. 59

4.4 (a)−(c) Segmentation results for frame 35, 37 and 39 of the "Table Tennis" sequence. The poster on the wall is successfully detected and segmented as a newly appeared object. 61

4.5 (a) Manually segmented ground truth of frame 1 of the "Table Tennis" sequence; (b) Manually segmented ground-truth of frame 10 of the "Coast Guard" sequence. 63

4.6 Overall segmentation quality: (a) Overall quality for frames 1−30 of the "Table Tennis" sequence. (b) Overall quality for frames 10−35 of the "Coast Guard" sequence. 65

4.7 Selected segmentation results for frames 1−30 in the "Table Tennis" sequence. (a),(c),(e),(g),(i) and (k) Frames 1,3,7,13,25 and 30. (b)Initialized segmentation for frame 1. (d),(f),(h),(j) and (l) Corresponding segmentation results. 67

4.8 Selected segmentation results for frames 10−35 in the "Coast Guard" sequence: (a),(c),(e),(g),(i) and (k) Frames 10,13,19,22,27 and 33. (b)Initialized segmentation for frame 10. (d),(f),(h),(j) and (l) Corresponding segmentation results. 69

4.9 Selected segmentation results for frames 50−80 in the "Dog" sequence: (a),(c),(e),(g),(i) and (k) Frames 50,60,63,67,70 and 79. (b)Initialized segmentation for frame 50. (d),(f),(h),(j) and (l) Corresponding segmentation results. 71

4.10 Selected segmentation results for frames 1−20 in the "Jumping Girl" sequence: (a),(c),(e) and (g) Frames 1,5,15 and 20. (b)Initialized segmentation for frame 1. (d),(f) and (h) Corresponding segmentation results. 72

4.11 Comparison of segmentation results for the frames 1−30 of the "Table Tennis" sequence: (a),(d),(g) and (j) Segmentation masks for frames 1, 10, 20 and 30 of the Cost211 Analysis Model; (b),(e),(h) and (k) Corresponding segmentation results produced by Sifaki et al.; (c),(f),(i) and (l) Corresponding segmentation results produced by the proposed graph-based algorithm. The Cost211 results lost track on the pingpong ball for frame 20 (g). 75

4.12 Comparison of segmentation results for the frames 10−35 of the "Table Tennis" sequence: (a),(e) and (e) Segmentation masks for frames 10, 20 and 30 presented by Sifakis; (b),(d) and (f) Corresponding segmentation results produced by the proposed algorithm. The Cost211 Analysis Model could not identify any moving objects for the first 30 frames of the sequence, hence results are not available. 76

List of Tables

4.1 Average percentage of propagated, validated and new pixels for frames 1−30 of “Table Tennis” sequence. 60

4.2 Average percentage of propagated, validated and new pixels for frames 10−35 of “Coast Guard” sequence. 60

Chapter 1

Introduction

1.1 The Video Segmentation Problem

Video segmentation has attracted substantial research interests and effort in the past decade as it assumes a major role in many video-based applications, such as object-based compression and coding, and visual content retrieval. While human vision seems to achieve it effortlessly, the automatic segmentation of video sequences is one of the most challenging tasks in computer vision. Video segmentation is used in a wide range of vision applications. The exact meaning of the term video segmentation varies according to the context in which it is applied. Video segmentation refers to a decomposition of semantic entities in content-based video retrieval [1] and video epitomes [2], a segmentation of moving blocks in video coding [3] or a spatio-temporal grouping in scene interpretation [4, 5], etc. Loosely speaking, segmenting a video is to decompose it into objects, which may include semantic entities or visual structures, such as color patches. Except in restricted

domains, the semantic level is generally not computable automatically, since it requires some amount of scene interpretation. Therefore, segmentation methods rely on concrete and measurable segmentation criteria that define non-semantic entities.

Image segmentation is a well studied but ill-posed problem. Without task-specific requirements, there can be several 'correct' segmentation outputs for a given image. The notion of correctness is dependent on the application. Based on spatial grouping cues alone, single image segmentation can yield very different results for two very similar images. Unlike the image segmentation problem in which only spatial grouping cues (such as color and texture) are available, in video segmentation, both spatial and temporal information are available for solving the grouping problem. Points undergoing coherent motion indicate a strong likelihood to belong to the same rigid body. With the added temporal dimension, the video segmentation problem becomes a better constrained problem. The need to impose temporal consistency constraint makes video segmentation different from a series of single image segmentations. Video segmentation demands that for a given image, the segmentation achieved should relate to the segmentation of the previous image, as long as they belong to the same shot. Video segmentation not only spatially performs intra-frame pixel grouping but also temporally exploits inter-frame coherence and variations of the grouping. In fact, the inter-frame correlation provides strong constraints for an optimal intra-frame grouping.

In view of the high correlation between adjacent frames, most of the state-of-the-art video segmentation algorithms focus on the exploitation of temporal coherence.

However, these approaches usually enforce temporal coherence on a pixel level, without much exploitation of the intra-frame spatial coherence, i.e., pixels belonging to the same rigid object undergo similar motion. Hence, motion estimation and enforcement of temporal coherence can be done at the region level instead of at the pixel level.

1.2 Contributions

In this thesis, the video segmentation problem is addressed as an intra-frame grouping reinforced by inter-frame coherence. It is a problem of pixel labeling based on both temporal coherence and spatial grouping. The focus of this thesis is on exploiting temporal correlation for efficient spatio-temporal grouping of visual structures under a graph-based framework. Segmentation is accomplished by propagating and validating a preceding graph which encodes pixel labels for the previous frame, followed by spatial subgraph aggregation subject to the validated label information. Graph propagation is achieved by a global motion estimation relating the two frames. All propagated labels are carefully validated by similarity measures. Trustworthy labels are preserved and the erroneous ones rejected. The segmentation of the current frame is thus transformed into a highly constrained graph partitioning problem. Henceforth, the problem at hand becomes the label assignment for the unlabelled, invalidated nodes, given the labelled data. The proposed method demonstrates a unifying framework which combines the spatial and temporal constraints in segmenting a sequence of correlated images. Temporal constraints in turn serve as the spatial constraints for the segmentation of the

current frame. It is related to semi-supervised learning methods [6, 7] and transductive learning methods [8]. The fundamental difference is that the proposed temporal propagation yields a more constrained system as the ratio of labelled to unlabelled data is much higher since the propagated and validated nodes form the majority of the graph for the current frame. The proposed method has been presented at the Conference on Computer Vision and Pattern Recognition (CVPR), 2008 [9].

1.3 Organization of the Thesis

This thesis is is organized as follows. A review of previous work is presented in Chapter 2, where state-of-the-art image and video segmentation techniques are studied. Chapter 3 presents a detailed formulation of the proposed approach. Experimental results and performance evaluations are given in Chapter 4. Chapter 5 concludes this thesis with a discussion of future work.

Chapter 2

Background and Previous Work

2.1 Image Segmentation: Spatial Grouping

Image segmentation, the spatial grouping problem, aims at clustering the pixels of an image into homogeneous regions based on a variety of cues, e.g., color, texture and boundary continuity. Image segmentation lays the foundation for video segmentation, which is essentially an image segmentation problem constrained by temporal coherence. To devise an effective video segmentation algorithm, it is important to understand the fundamentals of the spatial grouping problem. In this section, various image segmentation techniques are studied.

2.1.1 The MRF-MAP Framework

Given an image of N pixels, let $\mathbf{S} = \{s_1, s_1, \ldots, s_N\}$ be a set of image pixels. Define $\mathbf{X} = \{X_s | s \in \mathbf{S}\}$ as a family of random variables, and $\mathbf{L} = \{1, \ldots, l_M\}$ as

a set of label states. To segment the image into l_M perceptual groups, each pixel is assigned one of the prescribed labels l_m so that $\forall s \in \mathbf{S}, X_s \in \mathbf{L}$. Using only constraints from image data, it is an ill-posed problem. With the prior distribution of image labels, Bayes' rule provides the best estimates of the likelihood of image labels by

$$P(\mathbf{X}|\mathbf{S}) \propto P(\mathbf{S}|\mathbf{X})P(\mathbf{X}) \tag{2.1}$$

Image labeling is the maximum *a posteriori* (MAP) estimation of $P(\mathbf{X}|\mathbf{S})$.

$$\mathbf{X}^* = \arg\max P(\mathbf{X}|\mathbf{S}) \tag{2.2}$$

In the MRF-MAP framework (see Appendix A.1), $P(\mathbf{X})$ is modelled as a Markov Random Field (MRF), which allows the incorporation of contextual constraints based on piecewise constancy [10]. Using a log likelihood of $P(\mathbf{X}|\mathbf{S})$, MRF-MAP is equivalent to the regularization of $\mathbf{X}$ by minimizing the energy function

$$E = E_d + \lambda E_s \tag{2.3}$$

where E_d is the energy of image data, E_s is the smoothness energy, and λ is a weighting factor.

The smoothness term, known as the Potts model, encodes the MRF prior. It does not over-penalize labelings with large label changes between neighboring pixels and hence preserves discontinuity at region boundaries. The elegance of MRF-MAP framework simplifies the image segmentation problem as an exact minimization of the above energy function by seeking a global solution for a non-convex energy in

a high dimensional space.

$$Y^* = \arg\min E \tag{2.4}$$

Unfortunately, such an approach is known to be difficult due to a large number of local minima.

2.1.1.1 Energy Minimization

In the last few decades, effective algorithms for solving equation (2.4) have been developed. They are either stochastic, deterministic or discriminative in nature. Being more effective than an exhaustive enumeration, Simulated Annealing (SA) is traditionally used to find the global solution by a stochastic optimization. However, it is notorious for its inefficiency and poor performance in degraded images. Iterative Conditional Mode (ICM) demonstrates a fast convergence by a deterministic greedy strategy, but it can only guarantee a local minima [11]. On the other hand, the discriminative approaches find natural clusters in the feature space by maximizing intra-cluster similarity while minimizing inter-cluster similarity. The standard Expectation-Maximization (EM) algorithm [1] fits a mixture of Gaussians to image data. Image pixels are assigned to the clusters using the posterior probabilities. It relies on a priori knowledge of cluster number, and often converges to a local optimum that depends on the initial conditions. The frequently used *Mean Shift* algorithm [12] recursively searches for kernel smoothed centroids based on the gradient of estimated kernel densities. It is sensitive to the parameter settings. A slight variation of color bandwidth can cause a large change in segmentation granularity.

2.1.2 Segmentation by Clustering

A large class of image segmentation algorithms is based on feature space analysis. In this paradigm the pixels are mapped into a color space and clustered. Clustering is the partitioning of a data set into subsets so that each subset share some common characteristics. The most commonly used clustering techniques are *k-Means* and *Mean Shift* clustering. The *k-Means* algorithm starts off with k random cluster centres and assigns each pixel to the nearest centre (also called centroid) according to some distance measure. After new clusters are formed, the cluster centres are re-computed as the mean of all members in the cluster. The total cost of membership assignment is defined as the cost (distance) of assigning all data points to their nearest centres. The whole procedure repeats until there is no significant change in the total cost in successive iterations. Although it can be proven that the procedure will always terminate, the *k-Means* algorithm does not necessarily find the most optimal configuration corresponding to the global minimum of the cost function. The algorithm is also significantly sensitive to the initialization of cluster centres.

The *Mean Shift* algorithm is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. *Mean Shift* is a simple iterative procedure that shifts each data point to the average of data points in its neighborhood. *Mean shift* analysis is a relatively new clustering approach originally advocated by Fukunaga [13] and recently extended and brought to the attention of the image analysis community by Comaniciu and Meer [12] who convincingly applied it to image segmentation

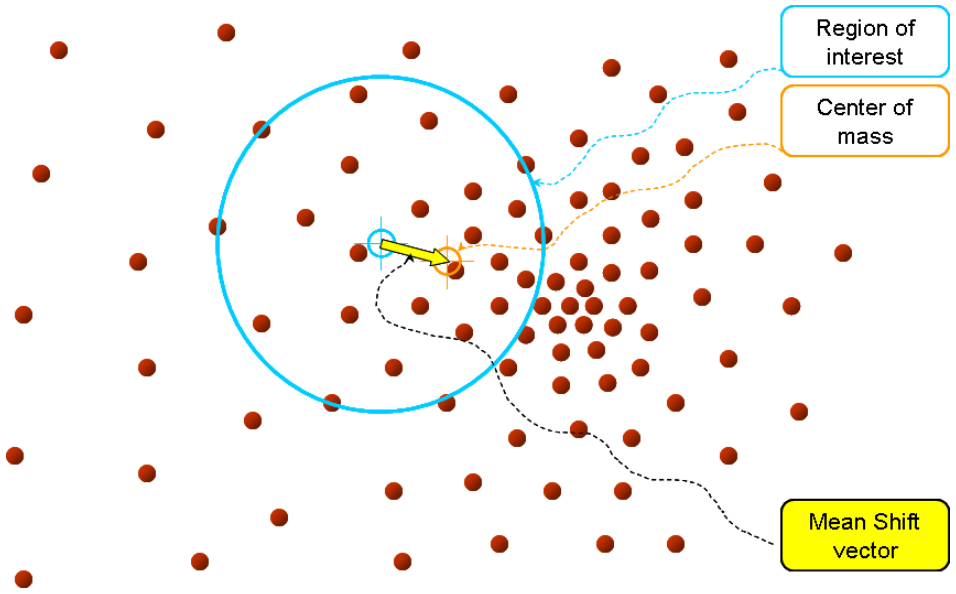


FIGURE 2.1: This diagram shows a distribution of data points in the feature space. Mean Shift vector points towards the denser region in the feature space and converges at the mode of the data set through density gradient estimation.

and frame-by-frame tracking. Figure 2.1 illustrates the *Mean Shift* mechanism in which the *Mean Shift* vector points towards a denser region in the feature space. Let $f(x)$ be the unknown probability density function underlying a p-dimensional feature space, and x_i, the available data points in this space. Under its simplest formulation, the mean shift property can be written as

$$\widehat{\nabla f(x)} \sim (ave_{x_i \in S_{h,x}}[x_i] - x) \tag{2.5}$$

where $S_{h,x}$ is the p-dimensional hypersphere with radius h centred on x. Equation (2.5) states that the estimate of the density gradient at location x is proportional to the offset of the mean vector computed in a window, from the centre of that window. Recursive application of the mean shift property yields a simple mode

detection procedure. The modes are the local maxima of the density, i.e.,$\nabla f(x) = 0$. They can be found by moving at each iteration the window $S_{h,x}$ by the mean shift vector, until the magnitude of the shifts becomes less than a threshold. The procedure is guaranteed to converge. When the mean shift procedure is applied to every point in the feature space, the points of convergence aggregate in groups which can be merged. These are the detected modes, and the associated data pints define their basin of attraction. The clusters are delineated by the boundaries of the basins, and thus can have arbitrary shapes. The number of significant clusters present in the feature space is automatically determined by the number of significant modes detected.

In contrast to the classical *k-means* approach, the clusters found by *Mean Shift* are separated by valleys in the point densities and not by artificially defined hyperplanes equidistant between the cluster centres. Finding the natural borders of clusters is important because such borders in feature space are mapped back to more natural segmentation borders in image space.

2.1.3 Graph-based Segmentation

In recent years, graph cuts have emerged as a powerful optimization technique for minimizing energy functions that arise in low-level vision problems. Graph cuts avoid the problems of local minima inherent in other approaches (such as gradient descent). Theses approaches generally represent the problem in terms of a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where each node $v_i \in \mathbf{V}$ corresponds to a pixel in the image, and the edges in $\mathbf{E}$ connect certain pairs of neighboring pixels. A weight is

associated with each edge based on some attributes of the nodes that it connects, such as pixel intensity. Normally, there are two types of edges in the graph: n-links and t-links. N-links connect pairs of neighboring pixels. Thus, they represent a neighborhood system in the image. The cost of the n-links corresponds to a penalty for discontinuities between pixels. T-links connect pixels with terminals (labels). The cost of a t-link connecting a pixel and a terminal corresponds to a penalty for assigning the corresponding label to the pixel. This cost is normally derived from the data term in (2.3).

Considering the set of label states $\mathbf{L}$ as the terminals of graph $\mathbf{G} = (\mathbf{S}, \mathbf{E})$, the energy minimization of MRF-MAP is equivalent to finding a minimum cost of multi-way cut for a graph, depending on some predefined label seeds in the image. With two terminals of source S and sink T, the Potts energy model of equation (2.3) can be exactly solved by a min-cut/max-flow (see Appendix A.2) of the s-t graph, i.e., searching for the maximum flow (minimum-cut) from S to T in the Ford-Fulkerson algorithm [14, 15] (see Appendix A.2.1). The minimum-cut is denoted as

$$C(S,T) = \min \sum_{u \in S, v \in T} W(u,v) \tag{2.6}$$

where $W(u,v)$ represents the weight of the edge connecting a node u in set S to another node v in set T.

The cost of a cut $C(S,T)$ is the sum of edge weights $W(u,v)$. To solve the minimum-cut problem is to find a cut that has minimum cost among all possible

cuts. The NP-hard problem in the multi-way cut is approximated by the α-expansion algorithm. In spectral graph partitioning, the cost of bi-partitioning G into subgraphs S and T is the sum of weights of all edges connecting the two subgraphs. The minimization of this cost is an NP-complete problem. Relaxing the membership indicator from discrete values to continuous values is equivalent to solving the eigen system $Lx = \lambda x$ (L is the Laplacian matrix of G). According to the Rayliegh quotient theorem, the minimum value of the cut is given by the second smallest eigenvalue of L. The eigenvector λ_2 (Fiedler vector) is the optimal solution of the cut. The minimum cut criterion is prone to cutting small isolated sets. This bias was later addressed with the *normalized cut* criterion [16] which considers self-similarity of regions. The min-max cut is able to perform more compact and balanced results for strongly overlapped clusters. The spectral graph cut has a high computation cost. For example, it is proportional to $O(N^{3/2})$ in normalized-cut, limiting its application on very large images. The Algebraic Multigrid (AMG) [17] is able to recursively achieve the minimization of normalized-cut by an adaptive graph coarsening with only $O(N)$ computation cost. However, in practice, the error in these approximations are not well understood and they are still fairly hard to compute, especially for the task of video segmentation when a large amount of data is to be handled.

These cut-based approaches provide only a characterization of each cut rather than of the final segmentation, and they often yield NP-hard problems for multi-way cuts. In view of these, a more efficient algorithm proposed in [18] defines a scale-adaptive predicate which measures the evidence for a boundary between two

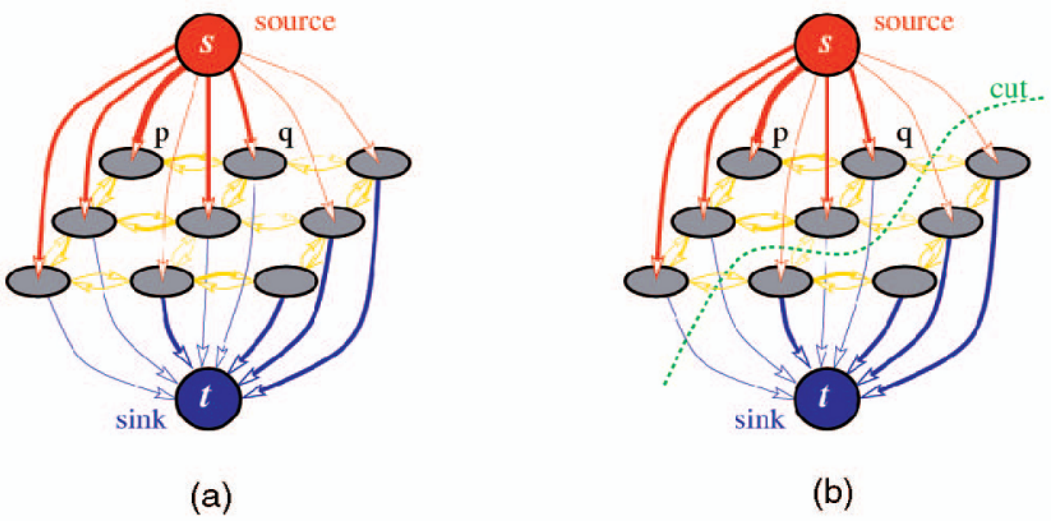


FIGURE 2.2: (a) A graph **G** with 2 terminals S and T. (b) A cut on **G**. Edge costs are reflected by thickness.

neighboring regions. Computation is simplified by representing regions in terms of *Minimum Spanning Trees* (MST).

2.2 Video Segmentation: Spatio-temporal Grouping

Video segmentation makes the distinction from image segmentation by imposing temporal coherence on spatial feature groupings. Because of the availability of a third dimension, the temporal dimension which infers motion information from image features, the grouping problem is no longer ill-posed.

Video segmentation brings up an efficient way of spatial feature grouping using temporal correlations. Inter-frame correlation provides strong constraints for an optimal intra-frame grouping and implies that there exists some form of linkage/similarity between the segmentation results of consecutive frames. Such a correlation is justified by the causal relationship between frames. Numerous spatio-temporal segmentation approaches have been reported in the literature [4]. Many extend the MRF-MAP framework in time and treat temporal correlation as another prior. In this case, Bayes' rule in (2.1) is extended to

$$P(\mathbf{X}|\mathbf{S},\mathbf{T},\mathbf{X}^{-},\mathbf{S}^{-}) \propto P(\mathbf{S}|\mathbf{X},\mathbf{T},\mathbf{X}^{-},\mathbf{S}^{-})P(\mathbf{X}^{-}|\mathbf{X},\mathbf{T})P(\mathbf{X}|\mathbf{T}) \tag{2.7}$$

where $\mathbf{X}^{-}$ and $\mathbf{S}^{-}$ denote the sets of pixel labels, and image pixels in the previous frame. $\mathbf{T}$ refers to the inter-frame pixel displacements. The MRF-MAP estimation is the minimization of the energy function

$$E = E_d + \lambda E_s + \mu E_t \tag{2.8}$$

where λ and μ are weighting factors for smoothness and temporal coherence. This energy minimization has been suggested and solved by Iterative Conditional Modes (ICM) in [11, 5]. Under this framework, an over-segmentation has to be performed on each frame followed by enforcement of temporal coherence. Unfortunately, this approach tends to under-utilize the strong temporal correlation. Temporal correlation can be more efficiently exploited in video-based applications, such as [19].

Furthermore, MRF-MAP in (2.8) searches for an optimal combination of subgroups with different spatial scales. Such an approach will lead to an intractable problem of finding a model to handle variations in spatial scales. Simple pixel-based measures, such as intensity or color, are insufficient to characterize the subgroups with large scales. High-level scale measures, i.e. texture or shape, have to be incorporated since scale variations commonly occur among the segmented subgroups [17]. Lastly, estimation of pixel displacement has been a challenge. This is especially true for those pixels with independent motions. Motion estimation based on optical flow computation suffers from the aperture problem. The estimated flow is smooth and reliable within objects, but erratic and unreliable at the object boundaries. Often, such motion estimation is incorporated into the whole framework in a feed-forward manner which lacks regularization. With an erroneous motion prior, MRF-MAP in (2.8) can lead to extremely sub-optimal groupings.

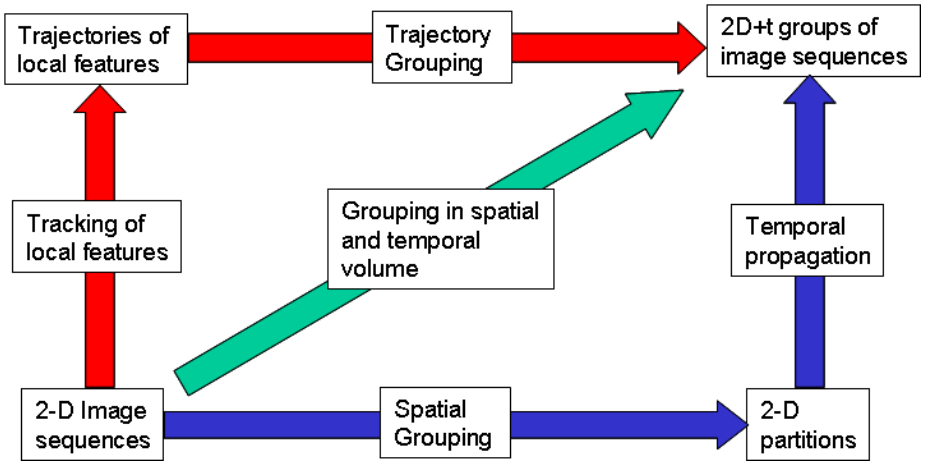


FIGURE 2.3: Structural flow of grouping along spatial and temporal axes.

2.3 Previous Video Segmentation Approaches

Spatio-temporal grouping manipulates features embedded in the spatio-temporal volume, the *video stack*, produced by the stacking of the individual consecutive video frames. The spatial and temporal dimensions of this volume can be handled either separately or simultaneously [4]. The structural flow of grouping along spatial and temporal axes is illustrated in Figure 2.3. Most state-of-the-art approaches handle these two types of dimensions separately, making a distinction between spatial segmentation, which groups features using spatial coherence criteria, and temporal tracking, which groups features using a temporal invariance hypothesis. Apart from these, a third approach, joint spatial and temporal grouping, avoids favoring one dimension over the other and instead operates directly in the spatio-temporal volume. These methods define the grouping criteria simultaneously in space and time, so that evidence for grouping is gathered at the same time in both dimensions. Based on the order of operations, spatio-temporal grouping approaches can be classified into three categories:

- Segmentation with spatial priority
- Segmentation by trajectory grouping
- Joint spatial and temporal grouping

2.4 Segmentation with Spatial Priority

This category of approaches favours spatial homogeneity when segmenting pixels in a video. It can be interpreted as an extension of single frame segmentation by adding temporal tracking. Such methods comprise two sub-modules, motion segmentation and spatial segmentation based on feature similarity. Figure 2.4 shows the structure of grouping approaches with spatial priority. Emphasis is placed on the spatial grouping based on similarity of image features (e.g., color or texture) or pixel motion. Spatial segmentation is carried out on every frame, followed by enforcement of temporal consistency.

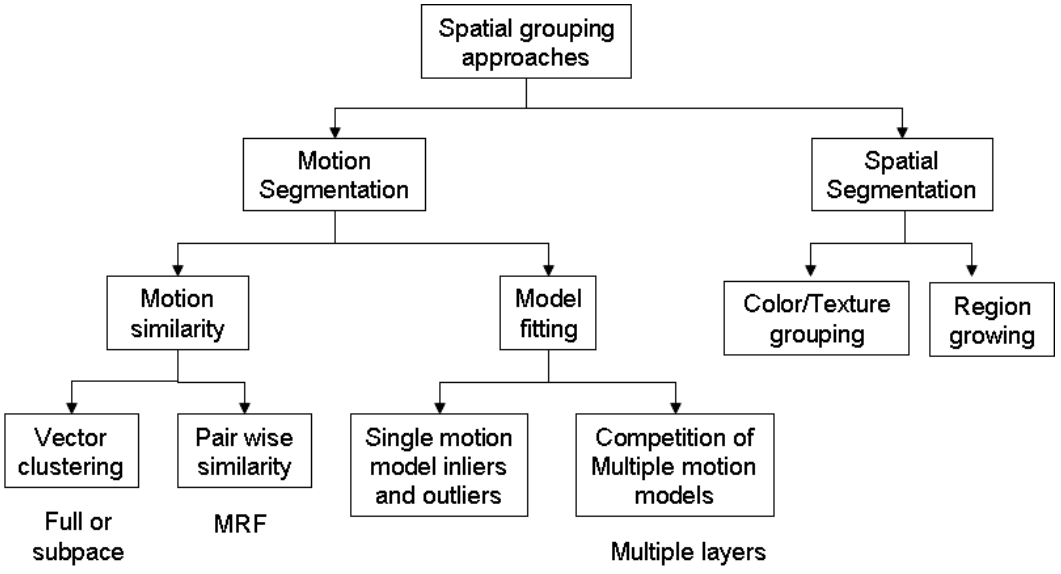


FIGURE 2.4: Structure of grouping approaches with spatial priority.

Motion grouping methods rely on an underlying motion model, be it a spatial motion smoothness model or a parametric model. Many approaches use optical flow

to estimate motion vectors and cluster pixels into the regions of coherent motion. However, motion fields may be unreliable or non-parametric under circumstances such as noisy data or non-rigid bodies. Motion similarity methods estimate motion parameters on a local basis. The grouping involves clustering features into regions of similar motion parameters. Motion model fitting methods compute motion parameters in groups of identically labeled elements. They involve the evaluation of the quality of fit of an element to a specified motion model. Temporal coherence can be enforced by two kinds of techniques: initialization from the previous frame and explicit temporal constraints. The former makes use of previous segmentation result to initialize the grouping for the current frame, while the latter acts as a stronger constraint to penalize large temporal change. Unfortunately, this is not realistic in the case of fast-changing motion or independent motion.

2.5 Trajectory Grouping

Trajectory grouping takes into account long-term information rather than short-term information as described in the motion segmentation category in Section 2.4. Less ambiguous displacement differences can be observed and motions are better discriminated. Main trajectory grouping techniques include grouping by motion similarity and grouping by explicit parametric models. Figure 2.5 shows the taxonomy of trajectory grouping approaches. The drawback of this approach is that since the spatial motion segmentation takes place afterward, tracking cannot use

any *a priori* spatial constraints. Furthermore, the need for long-term data precludes its use in online segmentation tasks where video data is obtained sequentially.

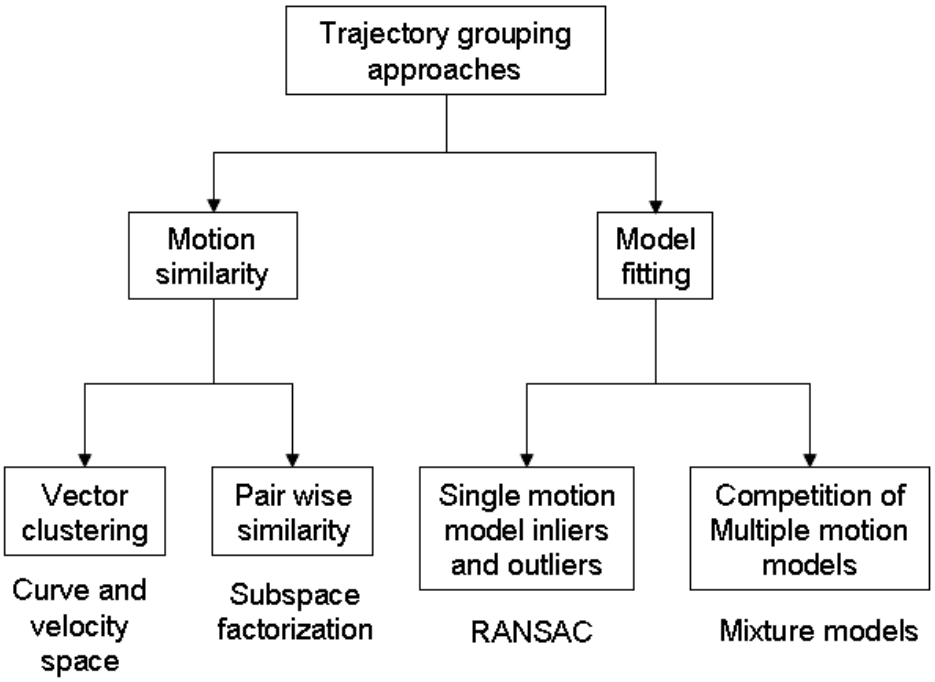


FIGURE 2.5: Taxonomy of trajectory grouping approaches.

2.5.1 Grouping by Motion Similarity

Motion is a prominent source of temporal variations in image sequences. Motion in image sequences acquired by a video camera is induced by movements of objects in a 3-D scene as well as by camera motion; hence, estimating motion is always a challenging task. Methods on direct comparison of trajectories define a similarity between two trajectories which is not influenced by the other trajectories. It can

consist in representing each trajectory as a point in a multidimensional vector space and then use Euclidean distances, or define a more general pairwise motion similarity such as spatio-temporal flow curves (by integrating local motion flow over time). Subspace methods represent a trajectory as the vector of the coordinates of its feature points over time, and stack them in a matrix C. With an affine camera, the tracks associated with rigid bodies moving differently lie in separate subspaces. Costeira and Kanade [20] factorize the matrix C using singular value decomposition (SVD).

2.5.2 Grouping by Model Fitting

Hypothesize and test methods are often used when explicit parametric models can be assumed. Hypotheses are obtained by fitting models to small data point sets chosen randomly. Each hypothesis is then validated by assessing the quality of fit. In RANSAC based methods, this is achieved by counting the number of inlier points. Hypotheses that have enough inliers are kept, and possibly compared to each other in order to merge similar ones. This method works well with outliers, when a small set of points are used, but not the case when all the data is used simultaneously.

Motion mixture models associate each trajectory with an object model; each object model consists of a parametric motion model, which describes the displacement of each point in the image over the whole sequence. Estimation of labels of trajectories (liking each trajectory to an object model) and motion parameter estimation are performed using an EM algorithm.

2.6 Joint Spatial and Temporal Segmentation

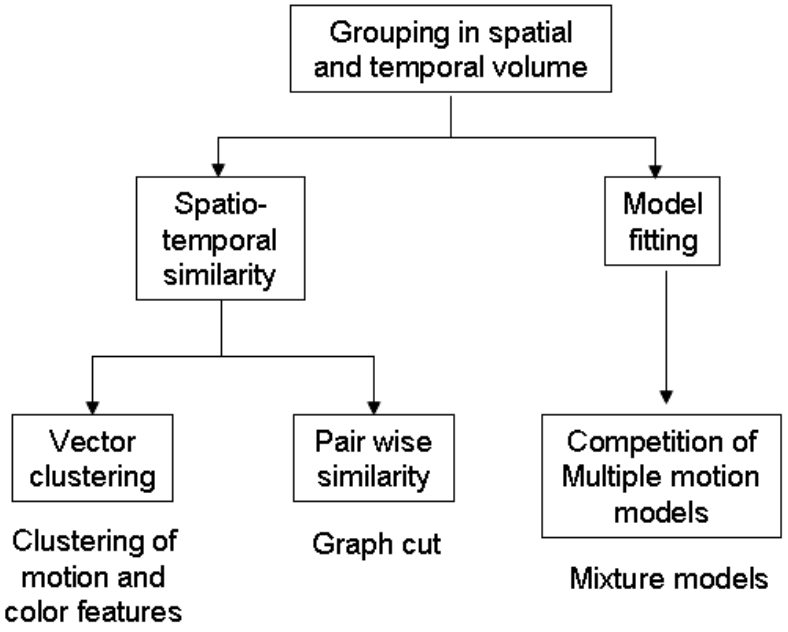


FIGURE 2.6: Taxonomy of joint spatial and temporal grouping approaches.

This category of methods avoids favoring one dimension over the other and instead operates directly in the spatio-temporal volume. The spatio-temporal grouping is simultaneously performed in a 3-D block of spatio-temporal pixels. The merit of this approach is supported by the evidence that human vision finds salient structures jointly in space and time [21]. Figure 2.6 presents the taxonomy of joint spatial and temporal approaches. Such segmentation approaches can be branched out into two subsections: grouping by similarity clustering, or fitting of a mixture model.

Clustering in feature space usually considers an n-dimensional space involving color, spatial and temporal positions. Each pixel is associated with a point in

feature space. Clustering a large volume of such points in feature space will incur high computational cost, and will also require the whole video shot to be available as input to construct the feature space. On the other hand, graph-based methods [22] consider a graph whose nodes are the image features, and whose edges are weighted according to some measure of similarity between nodes. To segment an image sequence, a weighted graph is constructed by taking each pixel as a node, and connecting nodes that are in a spatio-temporal neighbourhood of each other. The weight on a graph edge connecting two image pixels reflects the similarity between their motion profiles. A standard optimization technique such as normalized-cut [16] may be applied to partition the graph. Graph-based methods tend to model the groupings more accurately, but also require that all pixels in the entire video stack to be available prior to any graph-based processing. In model fitting methods such as [23], it is assumed that the image colors and their space-time distribution are generated by a model, such as a mixture of Gaussians. In general, a pixel is more likely to belong to a certain cluster if it is located near the cluster centroid. The EM algorithm is used to determine the maximum likelihood parameters of a mixture of Gaussians in feature space.

Recently, cosegmentation [24] has emerged as a new methodology for simultaneously segmenting the common parts in an image pair. It can be viewed as a joint spatial and temporal segmentation except that the grouping problem is solved in the spatial domain. Cosegmentation adopts a generative model for the histograms of the common parts and works towards maximizing the similarity between the generated histograms. Inference in the model leads to minimizing an energy with

an MRF term encoding spatial coherency and a global constraint. Such an optimization problem is usually NP-hard. Although more constrained optimization was later proposed to work around the NP-hard problem, the complexity of such a framework limits its application to binary segmentation, where the common foreground is to be segmented out of different backgrounds, while multi-label segmentation is often desired in generic video segmentation algorithms.

2.7 Summary of the Previous Approaches

The previous section presents three types of video segmentation methods. In segmentation with spatial priority, spatial grouping has to be inefficiently performed on every frame prior to temporal linking of regions. Temporal correlation is therefore under-utilized. On the other hand, the second method, segmentation by trajectory grouping, focuses on the separation of pixel trajectories and requires accurate tracking of features. Unfortunately, the intersection of motion trajectories would give rise to ambiguities in grouping. The third method, joint spatial and temporal segmentation, considers a video as a spatio-temporal block of pixels and performs clustering in the joint domain. However, both the trajectory grouping and joint spatio-temporal segmentation process all video frames in batches and hence are not applicable for applications where frames are acquired sequentially.

On reviewing the previous methods, it is obvious that the key issue with video segmentation lies in the fusion of spatial and temporal information. Spatial and

temporal information interact in a complementary manner and are interchangeable. While spatial coherence serves as a strong cue for intra-frame grouping, temporal coherence ensures that such spatial grouping is consistent over time. Upon enforcement of temporal coherence, temporal constraints in turn act as spatial constraints for the segmentation of the current frame. In view of this, a novel method is proposed in this thesis to better utilize temporal correlation for more efficient grouping. Spatial grouping for the previous frame is temporally propagated according to a global transformation. The propagated results are validated based on similarity measures. Incorrect pixel labels will be disputed and relabelled subject to the validated labels. This method works for both sequential and batch data. Details of the proposed method are presented in the following chapter.

Chapter 3

Proposed Method

3.1 Efficient Fusion of Spatial and Temporal Information

The review on previous work in Chapter 2 highlighted the essence of the spatio-temporal segmentation problem. While most methods strive to maximize the use of previous frame segmentation, the strong temporal correlation between frames is still not optimized. Additionally, motion cues are often incorporated in a feed-forward manner which lacks regularization. In this thesis, the spatio-temporal video segmentation is cast as a temporally-constrained graph partitioning problem. It is a problem of pixel labeling based on both temporal coherence and spatial consistency. Instead of enforcing an unrectified motion prior in the MRF-MAP model, pixel labels from the previous frame are propagated to the current frame

by a global motion estimation using the affine model. Validation of the propagated pixels labels is based on similarity measures. Trustworthy propagated labels are preserved, while erroneous labels are removed to reduce the bias in the final grouping. All unlabeled pixels are initially grouped into subgraphs by a simple color clustering. These subgraphs are iteratively aggregated by a pairwise subgraph grouping to form the final segmentation [9]. The entire cycle repeats itself by using the obtained segmentation output as previous information for the segmentation of the next frame. In this way, both spatial and temporal information interact in a complementary manner.

3.2 System Overview

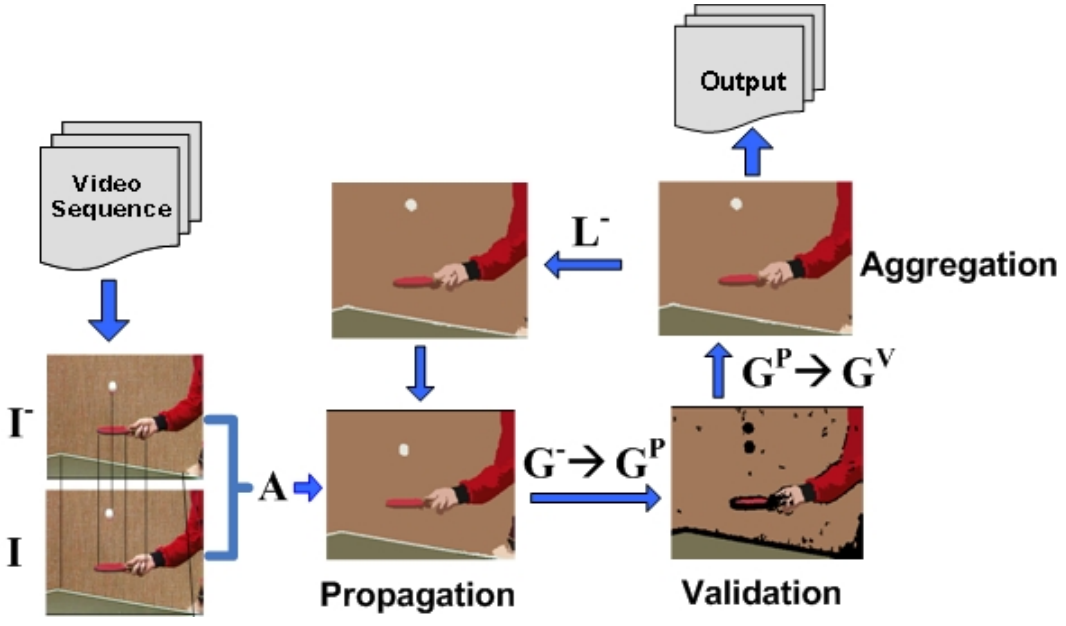


FIGURE 3.1: Spatio-temporal grouping by the propagation, validation and aggregation of a preceding graph.

Figure 3.1 shows the structure of the segmentation system. The proposed video segmentation is accomplished by propagating, validating and aggregating a preceding graph. Segmentation of the first frame of a given sequence is done using the *Mean Shift* segmenter. Minimum user intervention is required in tuning the parameters for the segmenter. This is done to set a meaningful scale for the segmentation. Applying *Mean Shift* usually yields an over-segmented result. This over-segmentation will be merged according to region similarity measures to form the initial segmentation.

While most of the state-of-the-art methods make use of temporal correlation on a pixel level to predict the label of a pixel in the next frame, the method proposed here attempts to predict pixel labels on a region level. It is justifiable to assume the scene in view to be piece-wise planar. A pair of adjacent frames in a video sequence can be regarded as a narrow baseline stereo system where only a small amount of translation and rotation are present, hence, the piece-wise planar assumption holds. Secondly, for the case of segmentation, the requirement on correspondence can be relaxed unlike in a registration system where point/curve features are precisely matched. Furthermore, predicting labels on a region level favours spatial consistency, which is much desired by a segmentation algorithm.

3.3 Notation

A preceding graph for the previous frame $\mathbf{I}^-$ can be specified by $\mathbf{G}^- = (\mathbf{S}^-, \mathbf{E}^-, \mathbf{L}^-)$, where $\mathbf{S}^-$ is the set of all nodes (pixels), $\mathbf{E}^-$ is the set of edges connecting the nodes,

and $\mathbf{L}^- = \{1, \ldots, l_M\}$ is the set of pixel labels. Temporal propagation through an affine estimation compensates global motion between two consecutive frames, thereby propagating $\mathbf{G}^-$ into $\mathbf{G}$ of the current frame $\mathbf{I}$. Let $\mathbf{G}^p = (\mathbf{S}^p, \mathbf{E}^p, \mathbf{L}^-)$ be the propagated graph from $\mathbf{G}^-$. $\mathbf{S}^p$ includes all nodes that can be projected to $\mathbf{I}$, $\mathbf{S}^p \subseteq \mathbf{S}^-$. $\mathbf{E}^p$ is the set of edges connecting the nodes in $\mathbf{S}^p$ in $\mathbf{I}$.

Considering the inter-frame pixel variations, all nodes in $\mathbf{S}^p$ are validated by measuring the color similarity between $\mathbf{I}^-$ and $\mathbf{I}$. A validated graph $\mathbf{G}^v = (\mathbf{S}^v, \mathbf{E}^v, \mathbf{L}^v)$ is formed by removing the nodes with wrong labels from $\mathbf{G}^p$, where $\mathbf{S}^v \subseteq \mathbf{S}^v$, $\mathbf{E}^v \subseteq \mathbf{E}^p$. Since $\mathbf{G}^v$ include the nodes with correct labels, it can be used to constrain the segmentation of the current frame $\mathbf{I}$. $\mathbf{G}^v$ is then implanted into the graph $\mathbf{G} = (\mathbf{S}, \mathbf{E})$ of the current frame $\mathbf{I}$, resulting in $\mathbf{G} = (\mathbf{G}^v, \mathbf{G}^x)$, where $\mathbf{G}^x = (\mathbf{S}^x, \mathbf{E}^x)$ is the set of unlabeled nodes, $\mathbf{S}^v \cap \mathbf{S}^x = \emptyset$, $\mathbf{S}^v \cup \mathbf{S}^x = \mathbf{S}$. Hence, the spatio-temporal grouping of the current frame $\mathbf{I}$ is equivalent to an optimal grouping of $\mathbf{G}^x$ subject to a labeled $\mathbf{G}^v$. The segmentation of a partially labeled image (with sparse labeled seeds) has been addressed in [14, 25] as an optimal cut on a partially labeled graph using min-cut/max-flow or random walker. In a two-label case, it is possible to find a global optimum because the energy function is convex. In comparison with a spatial grouping of $\mathbf{G}^x$ subject to $\mathbf{G}^v$ in video segmentation, $\mathbf{L}^x \subset \mathbf{G}^x$ may differ from $\mathbf{L}^v \subset \mathbf{G}^v$ due to dynamic scene changes. The existing labels in $\mathbf{L}^x$ may not fully appear in $\mathbf{L}^v$, and $\mathbf{L}^x$ can also contain some new labels. This fundamental difference makes the spatial grouping of $\mathbf{G}^x$ even more complicated. In this thesis, the segmentation problem is solved by a pairwise aggregation of subgraphs based on color heterogeneity, edge strength and

shape compactness.

A subgraph $\mathbf{g_m} = (\mathbf{s_m}, \mathbf{e_m}, \mathbf{l_m})$ of graph $\mathbf{G} = (\mathbf{S}, \mathbf{E}, \mathbf{L})$ is defined as a graph where $\mathbf{s_m} \subseteq \mathbf{S}, \mathbf{e_m} \subseteq \mathbf{E}$. All nodes in a subgraph $\mathbf{g_m}$ have a common label $\mathbf{l_m}$. A subgraph is used to describe a region. It may contain any number of nodes. Such a formulation thus offers flexibility to deal with variations in spatial scales.

3.4 Graph Propagation

The graph $\mathbf{G}^p$ is reconstructed in $\mathbf{I}$ from the labeled graph $\mathbf{G}^-$ based on the geometric transformation relating the two frames. Adjacent frames can be regarded as a narrow baseline stereo pair and the segmentation of the scene can be assumed to be piece-wise planar. A precise localization for correspondence is not required in the case of segmentation and hence a simple geometric transformation would suffice to relate segmentation results between two views. Without loss of generality, the inter-frame global motion can be approximated by an affine transformation $\mathbf{A}$. Then, $\mathbf{I}^-$ is warped to $\mathbf{I}$ by

$$\mathbf{A}\mathbf{I}^- = \mathbf{I} \tag{3.1}$$

The above linear system can be solved by using $N \geq 3$ corresponding pairs between $\mathbf{I}^-$ and $\mathbf{I}$. Motion estimation is embedded in this graph propagation process. With the transformation $\mathbf{A}$, $\mathbf{G}^p$ is constructed by projecting all labeled nodes in $\mathbf{S}^-$ into $\mathbf{I}$. The node edges $\mathbf{E}^p$ are reconnected in the topology of $\mathbf{I}$. It is worth-noting that some nodes in $\mathbf{S}^p$ may not be fully 4-connected due to the geometric transformation.

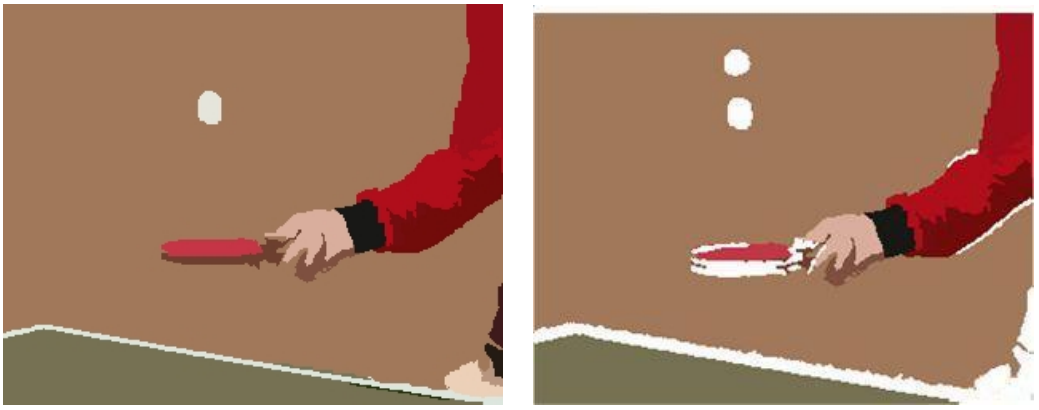
3.4.1 Scale Invariant Feature Detection

For accurate localization of corresponding feature points, the Scale Invariant Feature Transform (SIFT) algorithm in [26] is employed to detect scale-invariant features. This feature detection algorithm operates in scale space by convolving the image with Gaussian filters at different scales and taking the difference of successive Gaussian-blurred images. Extrema in this scale space are taken as distinctive features.

SIFT is applied to both $\mathbf{I}^{-}$ and $\mathbf{I}$ to detect and describe scale invariant features. Feature correspondence is based on the nearest neighbour. In fact, corresponding pairs undergoing independent motions can cause errors in the estimation of $\mathbf{A}$. For a robust solution, the Random Sample Consensus ($RANSAC$) [27] algorithm is used to reject the outliers and minimize the transformation error. $RANSAC$ randomly samples subsets from the data and calculates the affine model based on a randomly selected subset. The estimated affine transformation is used to verify against the remaining data points. The process is repeated until, within a certain error bound, there is at least one outlier-free sample subset. The advantage of $RANSAC$ is that it can estimate the parameters with a high degree of accuracy even when significant amount of outliers are present in the data set.

3.5 Validation

The graph propagation $\mathbf{G}^{-}$ to $\mathbf{G}^{p}$ relies only on the estimation of inter-frame global motion. Due to errors introduced by the affine approximation and independent



(a) (b)

FIGURE 3.2: A strong temporal correlation implies similar grouping in most corresponding regions between two frames. (a) Grouping results in the previous image frame. (b) Pixel labels in the previous frame are propagated and validated in the current frame. About 94.25 % of labels are reusable in segmenting the current frame.

motions, some nodes in $\mathbf{S}^p$ are wrongly labeled. The propagated node labels are validated based on color similarity. This step allows the verification of the motion estimates and it differentiates the proposed method from the state-of-the-art methods in which motion priors are often incorporated in a feed-forward manner without validation. An erroneous motion prior can severely affect the label prediction. For each label l_m^- in $\mathbf{G}^-$, color variances $\sigma_m^-(r)$, $\sigma_m^-(g)$, $\sigma_m^-(b)$ are calculated for all nodes with label l_m^-. Given a node s_n^p in $\mathbf{G}^p$ and its corresponding node s_n^- in $\mathbf{G}^-$, s_n^p is properly labeled if and only if these conditions are satisfied:

$$\begin{aligned} |s_n^p(r) - s_n^-(r)| &\leq 3\sigma_{l^-(s_n)}^-(r) \\ |s_n^p(g) - s_n^-(g)| &\leq 3\sigma_{l^-(s_n)}^-(g) \\ |s_n^p(b) - s_n^-(b)| &\leq 3\sigma_{l^-(s_n)}^-(b) \end{aligned} \tag{3.2}$$

Image noise often causes random color variations between two corresponding pixels. Instead of performing validation on a stand-alone node (3.2), a node label is validated by its local neighbors (e.g., 3×3 neighbors). With all properly labeled nodes in $\mathbf{S}^p$, a new graph $\mathbf{G}^v = (\mathbf{S}^v, \mathbf{E}^v, \mathbf{L}^-)$ is formed to retain correct labeling information from $\mathbf{G}^-$. Figure 3.2 shows an example of the propagated and validated segmentation results that are reusable for the segmentation of the current frame. In this example, 94.25 % of the propagated pixel labels are valid and retained. This large percentage of validated pixels suggests a profitable exploitation of temporal correlation.

Upon graph propagation and validation, the percentage of validated node labels are computed as

$$P_v = \frac{|\mathbf{S}^v|}{|\mathbf{S}|} \tag{3.3}$$

where $|\cdot|$ denotes the number of elements in the set. When the percentage of validated nodes falls below a certain threshold, i.e., $P_v \leq t_v$, it indicates that the propagated segmentation results are less reliable. Re-initialization of segmentation is thus performed on the current frame, taking into consideration the propagated segmentation results to ensure temporal consistency.

3.6 Independent Motions

The geometric relation in (3.1) can only recover the inter-frame global motion. It fails to compensate for pixel displacements due to independent motions. These independent motions can be identified by graph validation. Assume that one

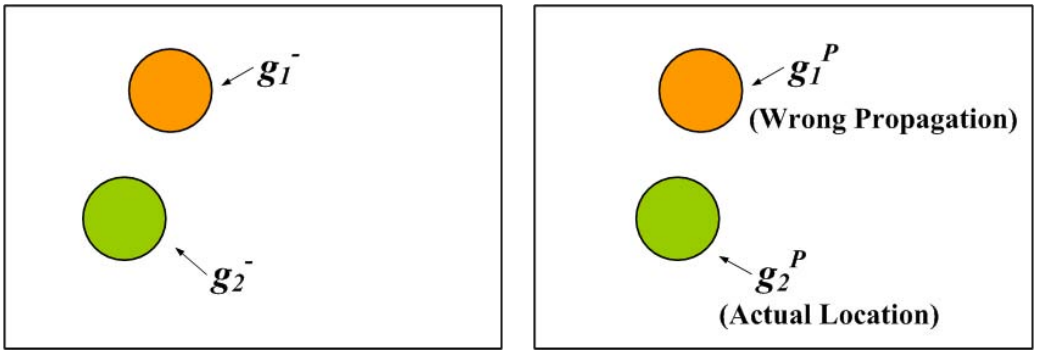
segmented region r experiences an inter-frame independent motion. Let $\mathbf{g}_r^- = (\mathbf{s}_r^-, \mathbf{e}_r^-, l_r^-)$ be the subgraph of r in $\mathbf{I}^-$. When $\mathbf{g}_r^-$ is propagated to $\mathbf{g}_r^p$ by $\mathbf{A}$, $\mathbf{s}_r^-$ is wrongly located in $\mathbf{I}$. As a result, l_r^- fails the validation check. Let the subgraph $\mathbf{g}_x^p = (\mathbf{s}_x^p, \mathbf{e}_x^p, l_x^-)$ represent the actual location of r in $\mathbf{I}$. Consequently, l_x^- is also invalidated due to color dissimilarity.

3.6.1 Regional Changes

Based on the spatial changes induced by independent motion, the invalidated regions can be classified into the following six cases:

1. **Case 1**: Whole region displacement

   This refers to the case where there is no overlap between the actual and the temporally predicted regions. It is often caused by fast independent motion of small objects.



(a) (b)

FIGURE 3.3: A pair of invalidated subgraphs due to whole region displacement. (a) The circle $\mathbf{g_1^-}$ in $\mathbf{I}^-$ and the pre-propagated location of its wrong prediction $\mathbf{g_2^-}$. (b) The predicted location of the circle is now at $\mathbf{g_1^P}$, while the correct location should be at $\mathbf{g_2^P}$.

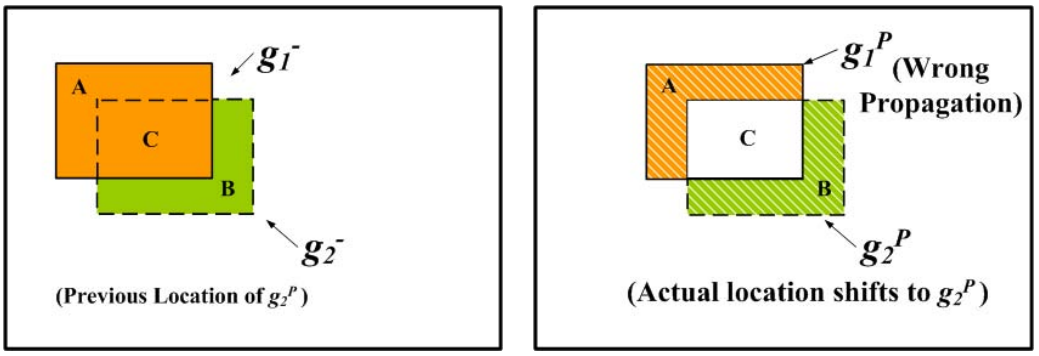
.

Figure 3.3 shows an example in which a circle moves independently with respect to the inter-frame global motion. Given the subgraph of the circle $\mathbf{g}_1^-$ in frame $\mathbf{I}^-$, graph propagation predicts an improper location $\mathbf{g}_1^p$ for it in $\mathbf{I}$. The label of subgraph $\mathbf{g}_1^p$ is invalidated, because the circle colors in $\mathbf{I}^-$ are different from the observed colors in $\mathbf{I}$. The proper location of the circle is indicated by the subgraph $\mathbf{g}_2^p$ and its pre-propagated location in $\mathbf{I}^-$ is indicated by $\mathbf{g}_2^-$. Therefore, the label of subgraph $\mathbf{g}_2^p$ is also invalidated. The independent motion of a segmented region causes the label of its propagated subgraph to be invalidated in $\mathbf{I}$. The label of subgraph at its actual location in $\mathbf{I}$ is also invalidated. In the later graph aggregation process, the two invalidated subgraphs are matched based on shape similarity and their labels are exchanged.

2. **Case 2**: Partial region displacement

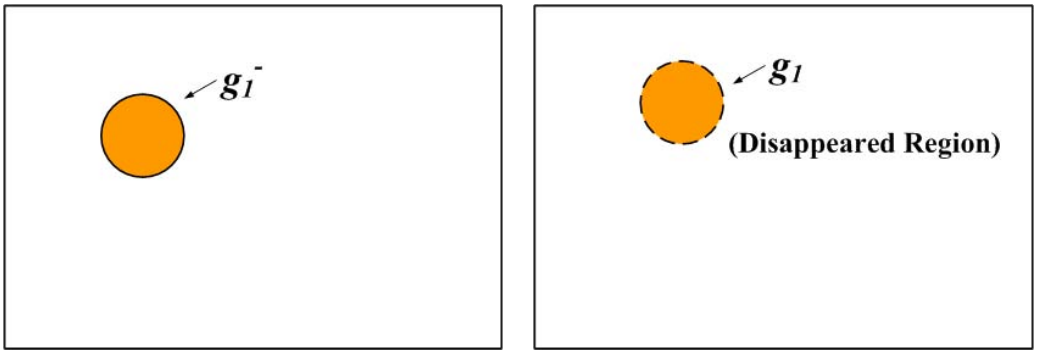
   Partial region displacement refers to the case where there is partial overlap between the actual and the temporally propagated regions. This could arise because of small independent motion. Two disputed subregions will be formed, labeled 'A' and 'B' respectively, as shown in Figure 3.4.

   Figure 3.4 shows an example where independent motion causes the rectangle to shift away from its predicted location, but there is still a partial overlap between the two. Given the subgraph of the rectangle $\mathbf{g}_1^-$ in frame $\mathbf{I}^-$, graph propagation predicts a partially correct location $\mathbf{g}_1^p$ for it in $\mathbf{I}$. The overlapping portion C is validated, while the non-overlapping portions A and B are invalidated.



(a) (b)

FIGURE 3.4: Two invalidated subregions due to partial region displacement. (a) A rectangle $\mathbf{g_1^-}$ (orange) and the pre-propagated region of its wrong prediction in frame $\mathbf{I}^-$, annotated as $\mathbf{g_2^-}$ (green). (b) The actual location of the rectangle shifts to $\mathbf{g_2^p}$ and it partially overlaps with the predicted region $\mathbf{g_1^p}$. The non-overlapping subregions A and B are invalidated while the overlapping subregion C is validated.



(a) (b)

FIGURE 3.5: An invalidated subgraph due to a disappearing object. (a) A circle $\mathbf{g_1^-}$ in frame $\mathbf{I}^-$. (b) The circle disappears in $\mathbf{I}$, causing $\mathbf{g_1^p}$ to be invalidated.

3. **Case 3**: Disappearing region

   Due to camera motion or independent motion, a region observed in $\mathbf{I}^-$ may disappear in $\mathbf{I}$. Thus, a propagated subgraph $\mathbf{g_1^p}$ will be an erroneous prediction of $\mathbf{g_1^-}$ in frame $\mathbf{I}$. See Figure 3.5 for illustration. The total number of label states for the current frame will be decreased by 1, i.e., $|\mathbf{L}| = |\mathbf{L}^-| - 1$.

4. **Case 4**: Newly appeared region

   This is the reverse of Case 3. Figure 3.6 shows a newly appeared circle. To classify a disputed region $\mathbf{g_x}$ as a newly appearing one, its dissimilarity with respect to all neighbouring labelled regions in $\mathbf{G}^-$ should reach a large value. Note that $\mathbf{g_x}$ is not a result of temporal propagation and therefore the superscript 'P' is dropped. Such a region will be seen as an "unmerged" subgraph after the pair-wise subgraph aggregation. This will be elaborated on in Section 3.7.

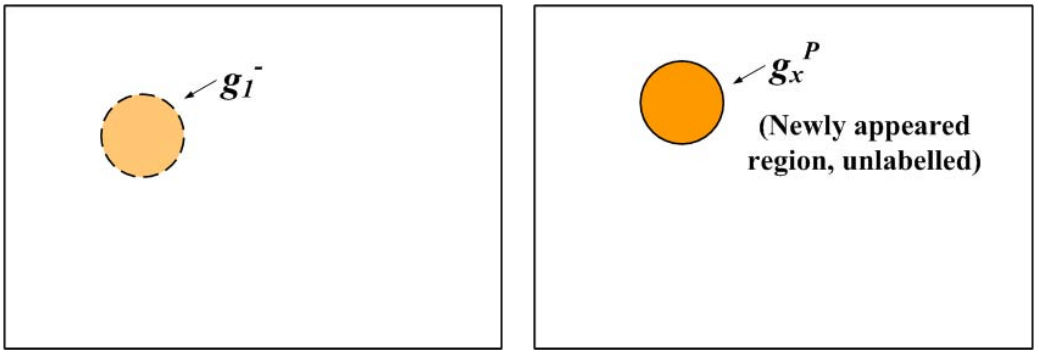
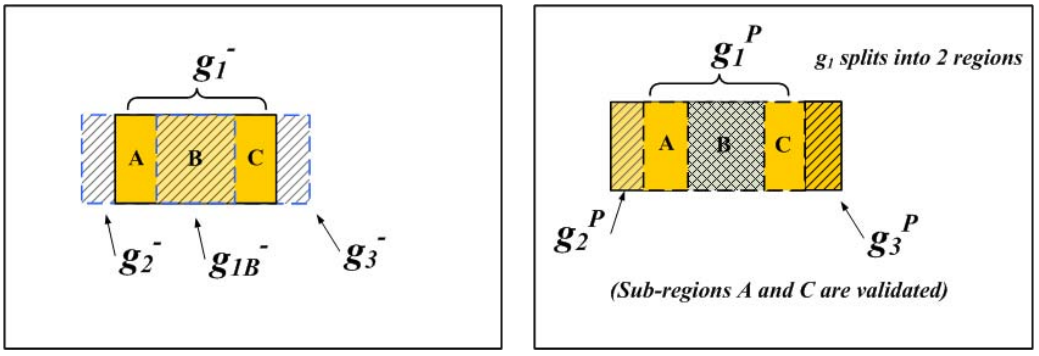


FIGURE 3.6: An invalidated subgraph due to a newly appearing object. (a) $\mathbf{g_1^-}$ denotes the pre-propagation of a newly appeared circle. (b) A new circle appears in frame $\mathbf{I}$, causing $\mathbf{g_x}$ to be invalidated. Note that the subscript 'x' indicates that $\mathbf{g_x}$ is not a result of temporal propagation and it is yet to be grouped and labelled, whereas its pre-propagated subgraph $\mathbf{g_1^-}$ is labelled.

5. **Case 5**: Splitting region

   The is caused by the splitting of regions or by occlusion. In the case of occlusion, if a single region seen previously is occluded by another object that blocks only its middle part but not its ends, it will be seen as a region split into two disconnected parts in the current frame. As illustrated in Figure 3.7, the shaded regions $\mathbf{g_2^p}$ and $\mathbf{g_3^p}$ as well as the gap $\mathbf{g_{1B}^p}$ between the two separated regions are invalidated. The total number of label states $|\mathbf{L}|$ will be increased by 1.



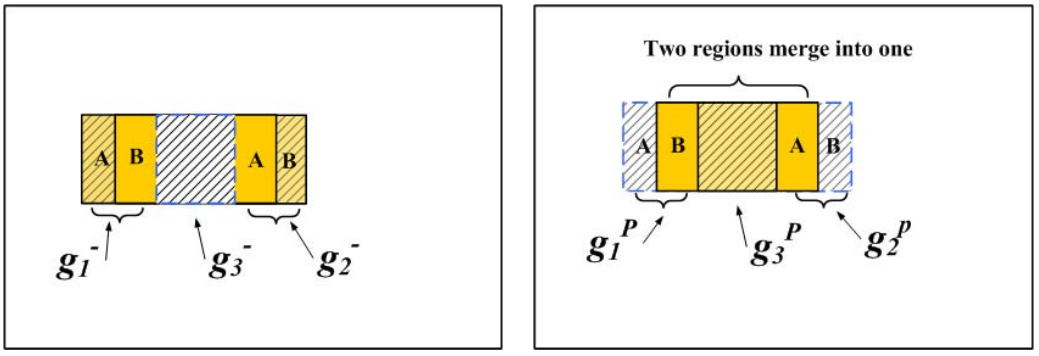
(a) (b)

FIGURE 3.7: Three invalidated subregions due to region splitting. (a) A rectangle $\mathbf{g_1^-}$ in frame $\mathbf{I}^-$ and the pre-propagation of its separated parts denoted by $\mathbf{g_2^-}$, $\mathbf{g_3^-}$ and $\mathbf{g_{1B}^-}$ (the shaded regions). (b) The rectangle splits into two regions in $\mathbf{I}$, causing $\mathbf{g_2^p}$, $\mathbf{g_3^p}$ and $\mathbf{g_{1B}^p}$ to be invalidated. Only portions that still overlap with the split regions (solid yellow regions), $\mathbf{g_{1A}^p}$ and $\mathbf{g_{1C}^p}$, are validated.

6. **Case 6**: Merging regions

   The is the opposite of Case 5. It happens when two previously split regions merge or when an occluding object moves away. Subgraphs $\mathbf{g_1^p}$ and $\mathbf{g_2^p}$ in Figure 3.8(b) represents the predicted regions that are merging in frame $\mathbf{I}$. Here, these two regions will be combined to form a single region during *Mean*

Shift segmentation and the propagated portions $\mathbf{g^p_{1A}}$ and $\mathbf{g^p_{2B}}$ that do not overlap with the actual merged region will be invalidated. The merged region will take the label of either participating region. The total number of label states $|\mathbf{L}|$ will be decreased by 1.



(a) (b)

FIGURE 3.8: A pair of invalidated labels of a single region due to region merging. a) Two rectangles $\mathbf{g_1^-}$ and $\mathbf{g_2^-}$ in frame $\mathbf{I}^-$ and the pre-propagation of the centre part of their merged version is denoted by $\mathbf{g_3^-}$. (b) The two rectangles merge into one region in $\mathbf{I}$, causing $\mathbf{g^p_{1A}}$, $\mathbf{g^p_{2B}}$ and $\mathbf{g^p_3}$ to be invalidated. Only portions that still overlap with the merged regions (solid yellow regions), $\mathbf{g^p_{1B}}$ and $\mathbf{g^p_{2A}}$, are validated.

3.7 Aggregation

The aggregation of subgraphs performs a spatial grouping for all unlabeled nodes in $\mathbf{G}^x$ based on $\mathbf{G}^v$. The challenge here is that some new groups may be formed in $\mathbf{G}^x$. Instead of using a seeded segmentation as in [14, 25], we conduct a pairwise subgraph grouping on $\mathbf{G}^x$, which is similar to [18], but with different grouping criteria. Prior to the aggregation of subgraphs in $\mathbf{G}^x$, the unlabeled nodes in $\mathbf{S}^x$ are grouped into small subgraphs by a low-level color clustering (*Mean Shift* [12]).

This pre-grouping is conducted to serve two purposes. Firstly, it accelerates the grouping of $\mathbf{G}^x$ and secondly, it initializes reasonable scales for the subsequent groupings. In a pairwise subgraph grouping, each subgraph $\mathbf{g^x}$ corresponds to an intermediate group in $\mathbf{I}$. Grouping criteria include edge relationship, color, and shape measures.

3.7.1 Edge Information

The color gradient between two pixels is characterized by the weight associated with the edge connecting their respective nodes. Let e_{ij} be the edge of two neighboring nodes s_i and s_j. The edge weight is defined by

Definition 3.1. The edge weight $w(e_{ij})$ between two neighboring nodes s_i and s_j is the norm of L*u*v* color difference between two pixels connected by the edge

$$w(e_{ij}) = \sqrt{(l_i - l_j)^2 + (u_i - u_j)^2 + (v_i - v_j)^2} \tag{3.4}$$

A strong edge connecting two subgraphs discourages the grouping of the said subgraphs. In [18], the grouping predicate checks if the minimum edge weight connecting a pair of subgraphs is large relative to the internal difference within at least one of the subgraphs. The internal difference is defined as the largest edge weight of the minimum spanning tree, which tries to find a maximum gradient from a low gradient path. This measure is very sensitive to image noises. Given a subgraph $\mathbf{g}_i = (\mathbf{s}_i, \mathbf{e}_i)$, $\mathbf{e}_i^B$ is used to denote the edges crossing the region boundary, $\mathbf{e}_i^B \subset \mathbf{e}_i$. A pictorial illustration of this region boundary is given in Figure 3.9.

The boundary edges are marked as black dotted lines. Let $w_B(\mathbf{e}_i^B)$ be the strength of the boundary of subgraph $\mathbf{g}_i$, which is given by

Definition 3.2. The strength of the boundary of a subgraph $\mathbf{g}_i$ is the mean of all edge weights in $\mathbf{e}_i^B$.

$$w_B(\mathbf{e_i^B}) = \frac{1}{N_B} \sum_{e \in \mathbf{e_i^B}} w(e) \tag{3.5}$$

where $N_B = |\mathbf{e_i^B}|$ is the number of elements in the set $\mathbf{e_i^B}$.

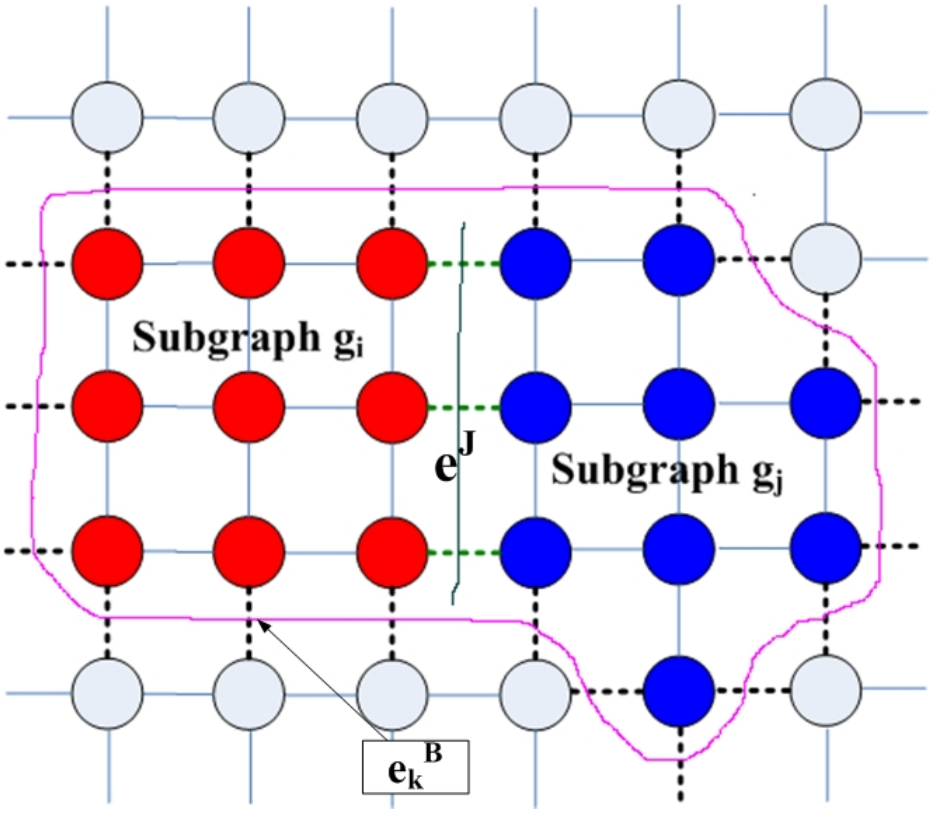


FIGURE 3.9: If subgraphs $\mathbf{g}^i$ and $\mathbf{g}^j$ are to be merged to form $\mathbf{g}^k$, the strength of the boundary of these two subgraphs is the mean of all edge weights in $\mathbf{e}_k^B$ (denoted by black dotted lines). The strength of the joint between subgraphs $\mathbf{g}^i$ and $\mathbf{g}^k$ is computed as the mean of all edge weights in $\mathbf{e}^j$ (denoted by green dotted lines).

In the case of two neighboring subgraphs $\mathbf{g}_i$ and $\mathbf{g}_j$, let $\mathbf{e}^J = \mathbf{e}_i \cap \mathbf{e}_j$ be the edges connecting boundary nodes between $\mathbf{g}_i$ and $\mathbf{g}_j$ (this set of edges is called the "joint"). Edges in this joint are represented by green dotted lines in Figure 3.9. Let $w_J(\mathbf{e}^J)$ be the strength of this joint $\mathbf{e}^J$. Then, $w_J(\mathbf{e}^J)$ is estimated by

Definition 3.3. The strength of the joint between $\mathbf{g}_i$ and $\mathbf{g}_j$ is the mean of all edge weights in the set $\mathbf{e}^J$.

$$w_J(\mathbf{e^J}) = \frac{1}{N_J} \sum_{e \in \mathbf{e^J}} w(e) \tag{3.6}$$

where $N_J = |\mathbf{e^J}|$ is the number of elements in the set $\mathbf{e^J}$.

In fact, weaker edges in $\mathbf{e}^J$ are preferred when merging $\mathbf{g}_i$ and $\mathbf{g}_j$ into $\mathbf{g}_k$, i.e., $\mathbf{g}_k = \mathbf{g}_i \cup \mathbf{g}_j$ which means a smaller $w_J(\mathbf{e}^J)$ than $w_B(\mathbf{e}_k^B)$. Therefore, the cost of merging $\mathbf{g}_i$ and $\mathbf{g}_j$ can be formulated as follows

$$C_E(\mathbf{g}_i, \mathbf{g}_j) = \begin{cases} 1 & \text{if } w_J(\mathbf{e}^J) \geq w_B(\mathbf{e}_k^B) \\ \frac{w_J(\mathbf{e}^J)}{w_B(\mathbf{e}_k^B)} & \text{otherwise} \end{cases} \tag{3.7}$$

3.7.2 Color

The color heterogeneity of a subgraph $\mathbf{g}_i$ is computed as the sum of color variances for all color channels, i.e., $C_H(\mathbf{g}_i) = \sigma_L(\mathbf{g}_i) + \sigma_u(\mathbf{g}_i) + \sigma_v(\mathbf{g}_i)$. Given two neighboring subgraphs $\mathbf{g}_i$ and $\mathbf{g}_j$, the merging cost in terms of color heterogeneity is computed by

$$C_H(\mathbf{g}_i, \mathbf{g}_j) = \begin{cases} 1 & \text{if } C_H(\mathbf{g}_k) \geq avg(i,j) \\ \frac{C_H(\mathbf{g}_k)}{avg(i,j)} & \text{otherwise} \end{cases} \tag{3.8}$$

where $avg(i,j) = (C_H(\mathbf{g}_i) + C_H(\mathbf{g}_j))/2$, $\mathbf{g}_k = \mathbf{g}_i \cup \mathbf{g}_j$.

3.7.3 Shape

The merging of two subgraphs $\mathbf{g}_i$ and $\mathbf{g}_j$ into $\mathbf{g}_k$ results in a more compact representation of subgraph $\mathbf{g}_k$. The compactness of a subgraph $\mathbf{g}_i$ is used as a generic shape measure. It is defined as $C_S(\mathbf{g}_i) = 4\pi A(\mathbf{g}_i)/L(\mathbf{g}_i)^2$, where $A(\mathbf{g}_i)$ is the area of $\mathbf{g}_i$, and $L(\mathbf{g}_i)$ is the perimeter of $\mathbf{g}_i$. When $\mathbf{g}_i$ is a circle, $C_S(\mathbf{g}_i) = 1$. If $\mathbf{g}_i$ is infinitely long and narrow, $C_S(\mathbf{g}_i) = 0$. Given two neighboring subgraphs $\mathbf{g}_i$ and $\mathbf{g}_j$, the cost of merging $\mathbf{g}_i$ and $\mathbf{g}_j$ in terms of shape compactness is given by

$$C_S(\mathbf{g}_i, \mathbf{g}_j) = 1 - \frac{4\pi A(\mathbf{g}_k)}{L(\mathbf{g}_k)^2} \tag{3.9}$$

3.7.4 Cost

The total cost of merging two subgraphs $\mathbf{g}_i$ and $\mathbf{g}_j$ is the weighted sum of the following measures: color heterogeneity, edge strength and shape compactness. This is given by

$$C(\mathbf{g}_i, \mathbf{g}_j) = k_E C_E(\mathbf{g}_i, \mathbf{g}_j) + k_H C_H(\mathbf{g}_i, \mathbf{g}_j) + k_S C_S(\mathbf{g}_i, \mathbf{g}_j) \tag{3.10}$$

where k_E, k_H and k_S are weighting factors for edge, color and compactness costs respectively.

A pairwise subgraph aggregation is conducted by searching the best fitting pair of adjacent subgraphs by the rule of mutual best fitting. Let C_{max} be the maximum merging cost. For the subgraph $\mathbf{g}_i$, a neighboring subgraph $\mathbf{g}_j$ is regarded as a

merging candidate iff,

$$C(\mathbf{g}_i, \mathbf{g}_j) <= C_{max} \tag{3.11}$$

For the subgraph $\mathbf{g}_i$, $\mathbf{g}_j$ is treated as the best fitting subgraph among all neighbors of $\mathbf{g}_i$ if a lowest merging cost exists between $\mathbf{g}_i$ and $\mathbf{g}_j$. According to the rule of mutual best fitting, the subgraph $\mathbf{g}_i$ has to be the best fitting neighbor of $\mathbf{g}_j$ as well. The algorithm of a pairwise subgraph aggregation is summarized in Algorithm 1. For the subgraph $\mathbf{g}_j$, it should be ensured that the merging cost between $\mathbf{g}_i$ and $\mathbf{g}_j$ is lowest among all neighbors of $\mathbf{g}_j$.

During the subgraph grouping, some small subgraphs (with irregular shapes) are quite resistent to the merging. In this case, a simple smoothing is performed on them by grouping them into their nearest neighbours based on color similarity after the above grouping procedure. A threshold is set to control the minimum similarity score for a merge to take place.

Note that during the merging stage, the participating subgraphs can either be labelled or unlabelled. Mutually best fitting neighbours that satisfy Equation 3.11 are to be merged. At the end of the iterative pairwise subgraph aggregation, if there still exists isolated unlabelled subgraphs which differ considerably from their nearest neighbours, they are likely to be caused by newly appearing objects. New labels will be assigned to such regions. These labels will be appended to the existing set of labels to denote the addition of new regions/objects.

3.7.5 Complexity Analysis of Subgraph Aggregation

The computational complexity is determined by the number of initial subgraph N_s, and the maximum number of adjacent segments per subgraph N_a. In step 2 of Algorithm 1, the computation to construct the adjacency relations is at most $O(N_s N_a \log(N_s N_a))$. The initial number of possible subgraph pairs $N_s N_a$ gradually reduces as step 2 proceeds. To update the adjacency relations of one subgraph in step 8, the computation is at most $O(\log(N_s N_a))$. The maximum number of updated subgraphs is $2N_a$. Steps 4-10 are repeated for at most N_s times. The computational complexity is $O(N_s 2N_a \log(N_s N_a))$.

Algorithm 1 A pairwise subgraph aggregation

Require: $\mathbf{G}^x$, $\mathbf{G}^v$

1: Start with the initial subgraphs in $\mathbf{G}^x$.
2: Construct the adjacency relations of these subgraphs.
3: Calculate the merging cost for all adjacent pairs of subgraphs using (3.10).
4: **repeat**
5: Search the adjacent subgraphs that satisfy (3.11)
6: Find the best pair of subgraphs $(\mathbf{g}_i, \mathbf{g}_j)$ with the minimum merging cost.
7: Merge $\mathbf{g}_i$ and $\mathbf{g}_j$ into a new subgraph $\mathbf{g}_k = (\mathbf{g}_i, \mathbf{g}_j)$
8: Update the adjacency relations of $\mathbf{g}_k$.
9: Extend the label to $\mathbf{g}_k$ if $\mathbf{g}_i$ or $\mathbf{g}_j$ is labeled.
10: **until** No more pairs of subgraphs satisfy (3.11).
11: Assign the new labels to the unlabeled subgraphs.

3.7.6 Algorithm

The proposed spatio-temporal segmentation involves the propagation and validation of a preceding graph, followed by the aggregation of unlabeled subgraphs. The former ensures the correctness of pixel groupings propagated from the previous frame, while the later performs a pairwise subgraph grouping for unlabeled subgraphs. The algorithm of the proposed spatio-temporal segmentation is summarized as follows

Algorithm 2 Spatio-temporal segmentation using a preceding graph

Require: $\mathbf{I}^-$, $\mathbf{I}$, $\mathbf{G}^-$

1: Estimate the affine transformation $\mathbf{A}$ using SIFT.
2: Propagate $\mathbf{G}^-$ to $\mathbf{G}^p$ based on (3.1).
3: Validate the labels $\mathbf{L}^p$ in $\mathbf{G}^p$ using (3.2). Construct the graph $\mathbf{G}^v$ that contains all trustable labels propagated from $\mathbf{G}^p$.
4: Correct labels of independent moving regions.
5: Implant $\mathbf{G}^v$ into $\mathbf{G}$. Group unlabeled nodes $\mathbf{S}^x$ into small regions using *Mean Shift*
6: Perform subgraph aggregation for unlabeled subgraphs using algorithm 1.
7: Return the labeled $\mathbf{G}$.

3.8 Connections to Transductive Learning

The proposed algorithm is similar to transductive learning method [6, 7, 8] which aims to learn from partially labelled data. The transductive method is used in image segmentation where partial grouping is known *a priori* and label assignment for the unlabelled data is done through statistical transductive inference. The transductive learning problem can be solved by spectral graph partitioning, such as min-cut and normalized-cut. However, the fundamental difference between the proposed method and the transductive method is that the number of label states for the video segmentation problem changes dynamically due to independent motions and appearance/disappearence of objects, whereas in a transductive segmentation framework, the number of labels is usually fixed. Furthermore, the ratio of unlabelled to labelled data is much less in the proposed method and hence it is faster to converge to a global minima. In the case where the propagated labels are invalidated, temporal constraints have no effects on the partitioning of the new unlabelled data if they are not temporally related. Only spatial constraints induced by the validated temporal constraints have effects on the partitioning of such data.

Chapter 4

Experimental Results and Discussion

To assess the validity and to evaluate the performance of the proposed video segmentation algorithm, a series of tests and comparisons has been performed on several standard test sequences. This chapter describes details of the experiment and analyzes the segmentation results qualitatively and quantitatively, using both the standalone and relative evaluation methodologies. To highlight the advantage of the graph-based temporal propagation and validation scheme, examples are shown to demonstrate the proposed algorithm's strength in handling independent moving objects, as well as the efficiency achieved by propagating and preserving reliable results for the segmentation of later frames.

4.1 Experiment Settings

The proposed algorithm is applied on several typical test sequences containing different spatial complexity and temporal activity characteristics, namely, the "Table Tennis", the "Coast Guard", the "Jumping Girl" and the "Dog" sequences. Results are presented for the video sequences in which different challenges arise. Details of the sequences are as follows.

- **Table Tennis**, images 1 to 30 − Sequence with high temporal activity due to rapidly-changing independent motions of the pingpong ball and the player. There are appearing and disappearing objects entering and leaving the scene.
- **Coast Guard**, images 10 to 35 − Sequence with independently moving boats and a static background. The camera follows the boat in the middle, while another boat is entering the scene. The water of the river globally appears to move to the right, but deviation from the dominant motion pattern occur locally. The small sizes of the independent moving objects and their blurry edges make it difficult to contrast against the background.
- **Jumping Girl**, images 1 to 30 − Sequence with two girls, one jumping towards another stationary girl. There is considerable independent motion caused by the jumping girl, though the background is uniform and easy to segment.
- **Dog**, images 60 to 80 − Sequence with a moving dog on the lawn. The background is uniform and therefore easy to segment, but the dog in the foreground has large and fast motions in certain frames.

Default parameters used in the total cost function (3.10) are: $k_E = 0.27$, $k_H = 0.55$ and $k_S = 0.18$. More weightage is given to the color heterogeneity for the subgraph grouping. Results for the four test sequences will be presented in later sections. For the purpose of a comparative evaluation, due to lack of reference segmentation results for some of the test sequences, discussions will be focused on the "Table Tennis" and the "Coast Guard" sequences.

4.1.1 First Frame Initialization

The solution proposed in Chapter 3 deals mainly with the propagation, validation and aggregation of previous segmentation result, assuming a segmented first frame is given. Initialization of the first frame of the video sequences was done by applying the *Mean Shift* [12] segmenter. A five-dimensional feature space was used. The three components of the CIE L*u*v* color space were used as color features while the remaining two dimensions were the lattice coordinates. The L*u*v* color space was employed since its metric is a satisfactory approximation to Euclidean, hence allowing the use of spherical windows. A cluster in this 5D feature space thus contains pixels which are not only similar in color but also contiguous in the spatial domain. The quality of segmentation is controlled by the size of the spatial h_s, and the color h_r, resolution parameters defining the radii of the (3D/2D) windows in the respective domains. The segmentation algorithm has two major steps. First, the image is filtered using mean shift in 5D, replacing the value of each pixel with the 3D (color) component of the 5D mode it is associated to. Note that this filtering is discontinuity-preserving. In the second step, the basins

of attraction of the modes, located within $h_r/2$ in the color space are recursively fused until convergence. The resulting large basins of attraction are the delineated regions, and the value of all the pixels within are set to their average [28]. It is important to emphasize that the segmenter processes gray level and color images in the same way.

In the fusion step, extensive use is made of region adjacency graphs and graph contraction with the *union-find* algorithm [29]. The initial RAG was built from the filtered image, the modes being the vertices of the graph and the edges were defined based on four-connectivity on the lattice. Fusion was performed as a transitive closure operation on the graph, under the condition that the color difference between two adjacent nodes should not exceed $h_r/2$. At convergence, the color of the regions was recomputed and the transitive closure was again performed. After at most three iterations the final labelling of the image was obtained. Small regions (the minimum region size, M, is defined by the user) were then allocated to the nearest neighbour in the color space. This postprocessing step can be refined by employing a look-up table which captures the relation between the smallest significant color difference and the minimum region size.

Minimum user intervention is required in tuning the parameters for the *Mean Shift* clustering. Default parameters for a 240×320 image is $(h_s = 13, h_r = 11, M = 30)$. Applying *Mean Shift* yields an over-segmentation of the first frame. This over-segmentation was merged according to region color similarity in the L*u*v* color space. The procedure is similar to the bottom-up agglomerative clustering. The merging cost is the Euclidean distance between any two cluster centroids. Two

adjacent regions are merged if the cost is below a certain threshold, according to the rule of mutual best fitting. The iterative steps are analogous to the pairwise sub-graph aggregation discussed in Chapter 3. Merging terminates when all regions are considered and no more pairs of regions satisfy the above-mentioned conditions. Results of the first frame initialization for the four test sequences are shown in Figures 4.7(b), 4.8(b), 4.9(b) and 4.10(b), respectively.

4.2 Segmentation Evaluation Methodology

The challenge faced when evaluating a video segmentation algorithm lies in the lack of ground truth and an objective and reliable quantitative metric. There is no ground truth segmentation result readily and clearly available in most segmentation problems. Manually segmented results are often used as ground truth for performance evaluation, but there is no unique ground truth for a fair comparison [30]. The optimal segmentation varies according to the context in which it is applied. Depending on the availability of a reference, the evaluation can be carried out in two ways: the standalone evaluation and the relative evaluation [31]. The former is used when a reference segmentation is not available, while the latter compares the segmentation results against some ground truth. The ground truth can either be the segmentation defined by human subjects or representative results produced by state-of-the-art algorithms.

Given the subjectivity of this topic, to avoid favoring either one of the above-mentioned evaluation methods and to arrive at a fair conclusion, results of the

proposed algorithm is evaluated under both standalone and relative schemes. Although it is generally accepted that the relative evaluation is expected to produce more reliable assessment of segmentation quality, the reference segmentation performed by human subjects may induce some degree of subjectivity. Furthermore, owing to the lack of ground truth segmentation results, certain test sequences can only be evaluated by the standalone method. For the case of relative evaluation, segmentation results generated by the COST211 Analysis Model [32] and [33] are used to benchmark the performance of the algorithm proposed in this thesis.

Past work on no-reference quality metrics for image and video has been used to measure performance in the absence of reference. In a no-reference quality metric, instead of approaching the result to a truth reference, one aims at defining the characteristics of a good quality image (in the case of segmentation, a good segmentation result). The quality of segmentation is then assessed by examining the degree in which the algorithm approaches the good characteristics mentioned above. Past work both on image quality assessment and segmentation quality assessment show there is a good potential behind this approach. Results obtained however are quite preliminary and unacceptable in terms of fidelity and correlation with a subjective metric performed by a human being. This is especially true for the case of segmentation quality metrics [31].

Two types of measurements were targeted when performing video segmentation quality evaluation: 1) Individual object segmentation quality evaluation − when

one of the objects identified by the segmentation algorithm is independently evaluated in terms of its segmentation quality; 2) Overall segmentation quality evaluation − when the complete set of objects (the scene partition) identified by the segmentation algorithm is globally evaluated in terms of its segmentation quality.

4.3 Standalone Segmentation Quality Evaluation

Metrics for individual object standalone segmentation quality evaluation can be established based on the expected feature values computed for each object (intra-object metrics), as well as on the observed disparity of some key features relative to the neighbours (inter-object metrics). The former is used in this thesis to examine the effects of temporal constraints on spatial segmentation.

4.3.1 Spatial Uniformity

Since the proposed algorithm focuses on temporal propagation of previous segmentation results, the resulted object-level segmentation is temporally coherent. Spatial segmentation is a result of the temporal propagation. To examine the validity of this temporal constraints on the spatial segmentation, *Spatial Perceptual Information*(SI) [31] is adopted as a quality metric for spatial uniformity. This metric is defined as

$$SI_r(R_n) = \sqrt{\frac{1}{N_n}\sum_i\sum_j(Sobel(i,j))^2 - \left(\frac{1}{N_n}\cdot\sum_i\sum_j(Sobel(i,j))\right)^2} \qquad (4.1)$$

where R_n denotes the n^{th} region of a frame and N_n is the number of pixels in that region.

The above metric computes the standard deviation over all pixels in a Sobel-filted segment. The Sobel filter is implemented by convolving two 3×3 kernels over the video frame and taking the square root of the sum of the squares of the results of these convolutions.

The SI metric was originally used to measure the spatial detail in an image, taking higher values for the more spatially complex scenes. The SI metric is specified in ITU-T Recommendation P.910 [34] and it is based on the amplitude of the Sobel edge detector. Here the SI is adapted to measure the spatial homogeneity of a segmented region. It is normalized to produce results in the interval [0,1], with the lower value associated to the more homogeneous segmentation result. Note that $SI_r = 0$ corresponds to a perfect segmentation on a textureless image region. A textureless and uniform region itself is actually a perfect grouping without any segmentation. This rarely happens in real images with random variations in intensity and hence one should expect the SI_r value to be above zero.

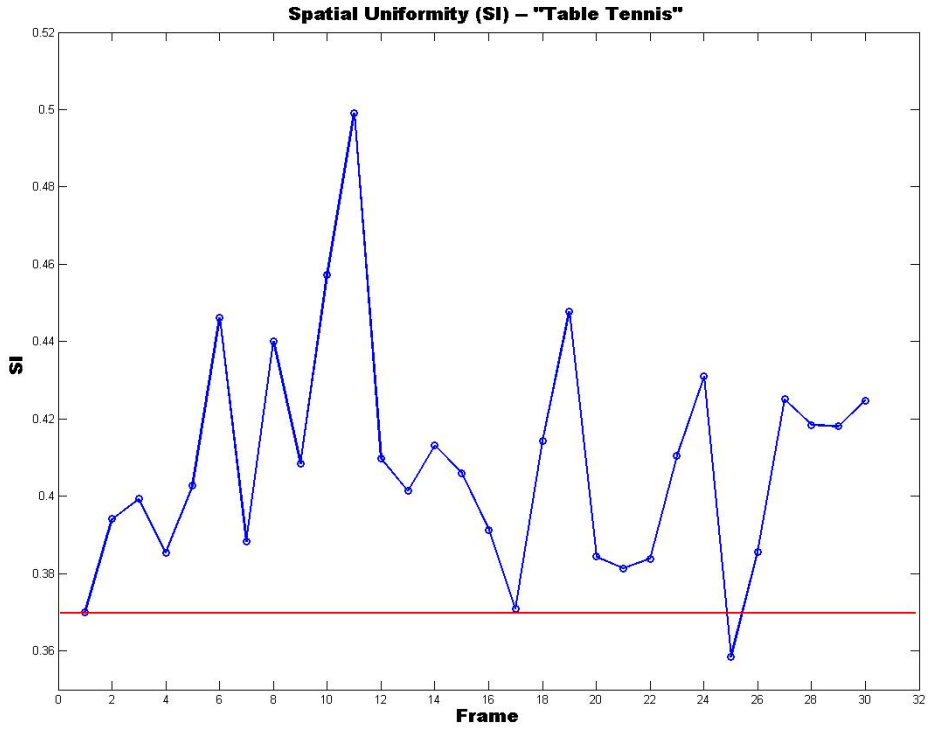
$$\widehat{SI_r(R_n)} = \left(\frac{1}{1 + \frac{SI_r(R_n)}{256}} - 0.5 \right) \times 2 \tag{4.2}$$

Combining all scores for individual regions, the aggregated frame-wise SI is computed as

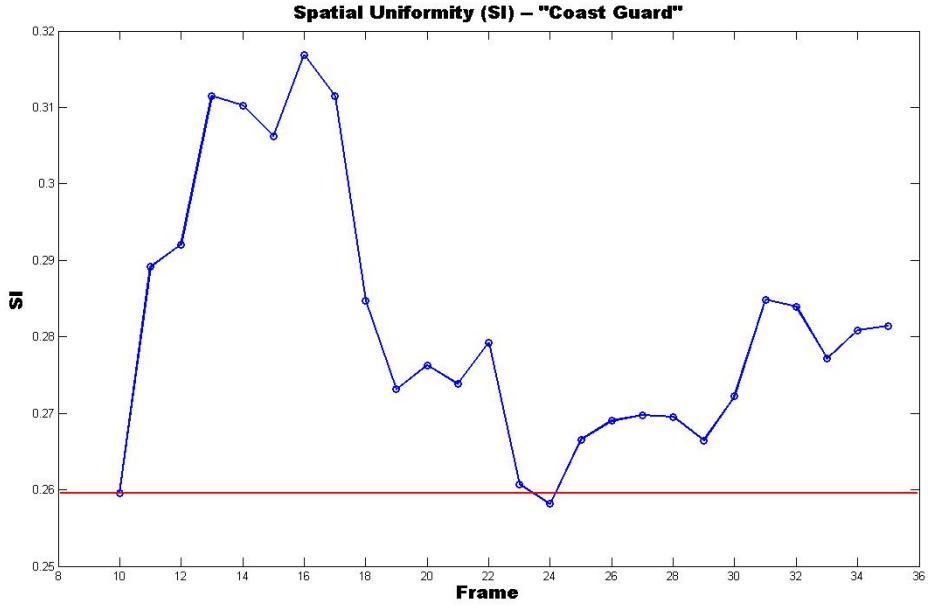
$$SI(I_t) = \frac{1}{N}\sum_{n=1}^{N}(SI_r(R_n)) \tag{4.3}$$

where N is the total number of regions in the segmentation result.

Figures 4.1 and 4.2 present the evaluated spatial uniformity scores for sections of the four test sequences. It can be observed that despite some temporal fluctuations, most of the SI values of subsequent frames do not deviate much from that of the initialized segmentation, suggesting spatially coherent results. For the "Table Tennis" and the "Dog" sequences where there exist more independent motions, the maximum deviation from the initialized frame amounts to 0.123 and 0.052, respectively. Due to independent motions, especially significant change in spatial position, such as the pingpong ball and the hand in the former sequence, tracking and localization of object boundaries tend to be more difficult over time. Using the Sobel response, the SI metric measures how well the segmentation results, the delineation, agree with the observed boundaries (edges) found on the image itself. The presence of an edge-like feature within a segmented region is penalized. Hence, it is understood that the performance for sequences with more temporal activity is lower than the more static ones. Note that for a few frames in the test sequences except for the "Dog" sequence, the SI values obtained are actually lower than that of the initialization. This suggests that the temporal propagation and validation to a certain extent has the self-correcting effect to achieve more spatially coherent segmentation. Overall, for the four test sequences, the SI scores for subsequent segmentation are quite close to the initializations, hence demonstrating the effectiveness of temporal propagation on spatial grouping.

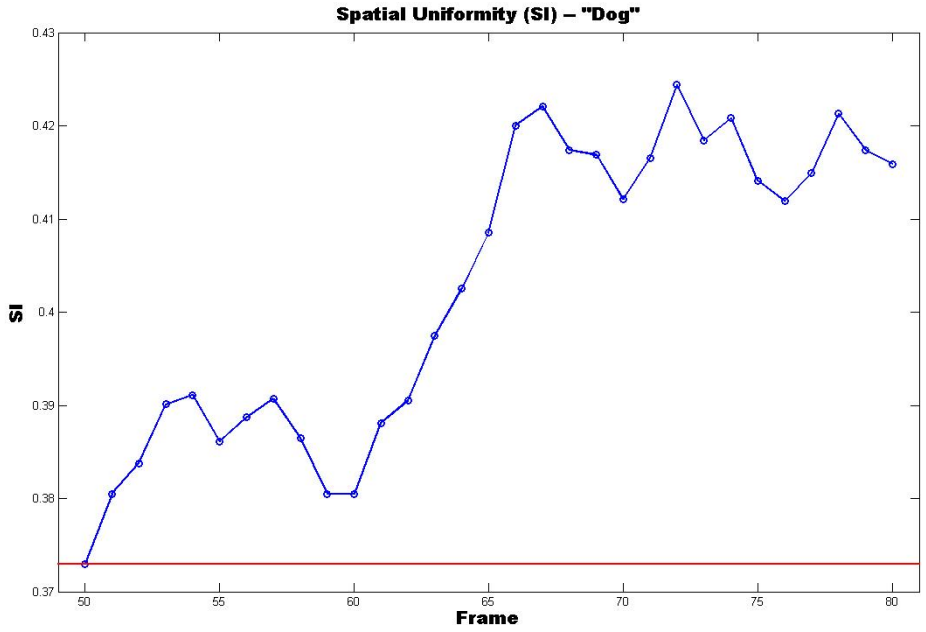


(a)

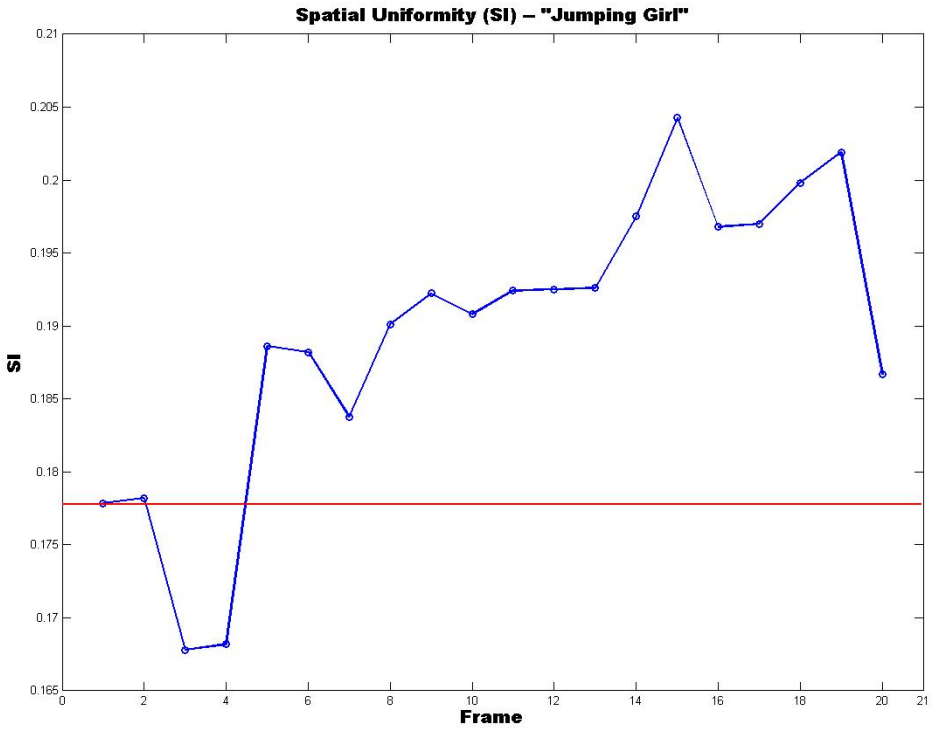


(b)

FIGURE 4.1: (a) and (b) Spatial uniformity (SI) of frames 1−30 of the "Table Tennis" Sequence and frames 10−35 of the "Coast Guard" Sequence respectively. The horizontal line marks the *SI* value of the initialized segmentation. The majority of the segmentation results have *SI* values close to that of the initialized segmentation.



(a)



(b)

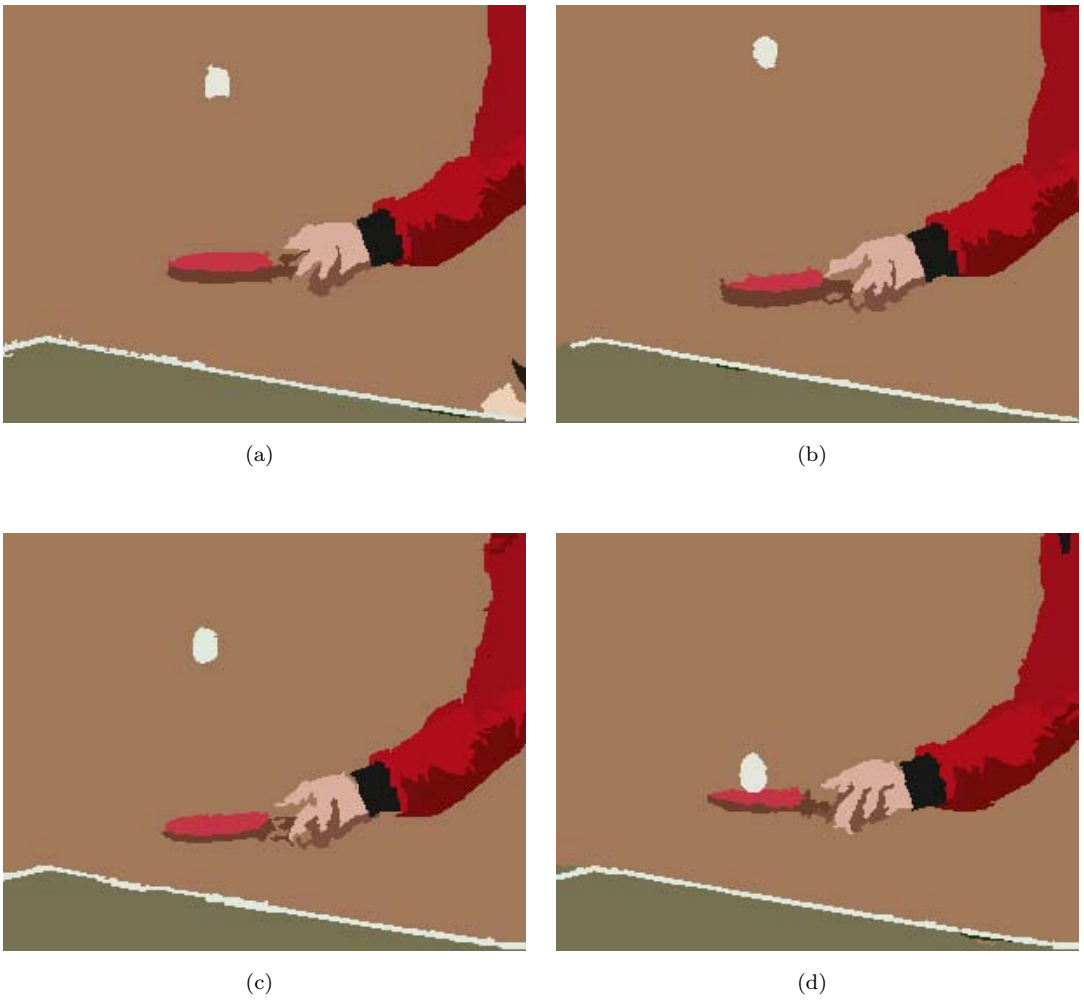
FIGURE 4.2: (a) and (b) Spatial uniformity *(*SI) of frames 50−80 of the "Dog" Sequence and frames 1−20 of the "Coast Guard" Sequence respectively. The horizontal line marks the *SI* value of the initialized segmentation. The majority of the segmentation results have *SI* values close to that of the initialized segmentation.

4.3.2 Independent Motion

As discussed in Section 3.3, the proposed algorithm is designed to handle independent motion. Figure 4.3 illustrates a case of fast independent motion. The video section that was used for testing (frames 1−30 of "Table Tennis" sequence) contains fast independent motion. The pingpong ball bounces up and down and the human arm, an articulated model, swings back and forth. Traditional approaches based on motion parameter estimation suffer from their inability to handle fast-moving objects, while the proposed algorithm is able to track both the pingpong ball and the arm accurately. In segmenting the pingpong ball in this example, the proposed algorithm is able to handle fast moving objects that do not overlap if adjacent frames are superimposed by warping one onto another according to an affine transformation estimated from SIFT features. As for the human arm, there is some overlap between the projected and the actual regions upon warping, and reassignment of region labels is handled by graph aggregation.

4.3.3 Newly Appeared Objects

Newly appeared objects are detected during graph aggregation as "unmerged" regions. Figure 4.4 shows a case where a poster hanging on the wall enters the scene. The proposed algorithm is able to detect this newly appeared object. Despite its color similarity to the pingpong ball and the table edge, proximity constraint (only neighboring subgraphs are merged during pair-wise subgraph grouping) is still able to identify this object as a new comer. A new label state will be appended to the



(a) (b)

(c) (d)

FIGURE 4.3: (a)−(d) Segmentation results of frames 2, 5, 9 and 12 in the "Table Tennis" sequence by the proposed algorithm. The pingpong ball and human hand are segmented as independent moving objects. Note that pingpong ball is correctly associated despite no temporal overlapping after propagation.

existing set of labels **L**. Segmentation for subsequent frames will carry out with the updated label set.

The disappearance of existing objects is handled during the graph validation process. During later subgraph grouping, the invalidated region belonging to a disappearing object will be pre-grouped into subgraph(s) and merged into its best

TABLE 4.1: Average percentage of propagated, validated and new pixels for frames 1−30 of "Table Tennis" sequence.

Class	Propagated(%)	Validated(%)	New(%)
Table	96.20	85.10	0.20
Ball	98.35	0.29	0
Hand	97.50	12.18	6.57

TABLE 4.2: Average percentage of propagated, validated and new pixels for frames 10−35 of "Coast Guard" sequence.

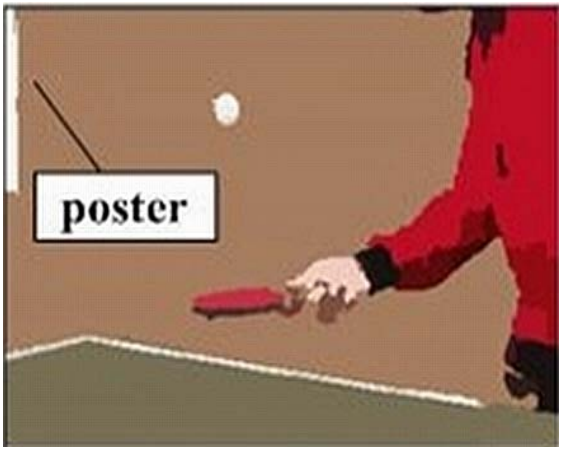
Class	Propagated(%)	Validated(%)	New(%)
Boat	97.50	87.10	0.11
Water	98.35	84.20	5.22
Land	97.21	95.50	5.43

fitting neighbouring labeled subgraph. This graph-based validation and aggregation processes is flexible in the handling of appearance and disappearance of objects.

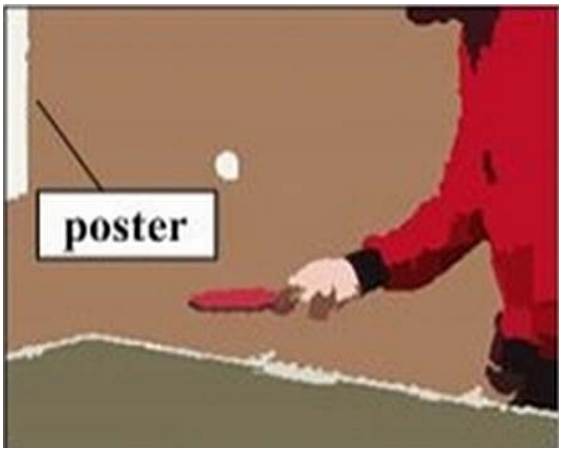
4.3.4 Benefit of Temporal Propagation

To highlight the profitable exploitation of temporal redundancy in video segmentation, Table 1 and Table 2 show the average percentage of propagated, validated and newly appeared pixels for both video sections. Segmentation results of the major objects in the two sequences are analyzed. For every pair of adjacent frames, the percentage of propagated, validated and newly appeared pixels for these major objects are evaluated. For a sequence of N frames, there are $N-1$ adjacent pairs. The average values are computed over the $N-1$ results. On comparing the

(a)



(b)



(c)

FIGURE 4.4: (a)−(c) Segmentation results for frame 35, 37 and 39 of the "Table Tennis" sequence. The poster on the wall is successfully detected and segmented as a newly appeared object.

percentage of validated pixel labels for both sections, it can be seen that the percentage of validated labels for a particular object is more than 84.20% when there is little or no independent motions as in the case of the "Coast Guard" sequence (Table 2).

4.4 Relative Segmentation Quality Evaluation

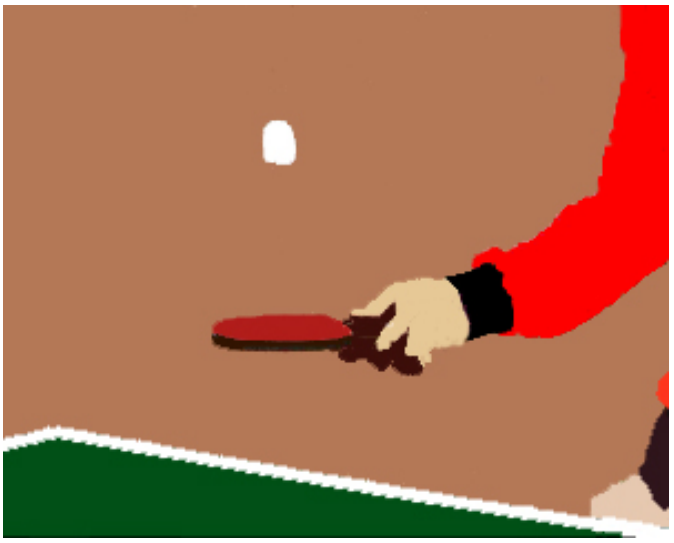
4.4.1 Overall Segmentation Evaluation

To examine the segmentation quality with respect to a reference segmentation, experimental results are compared by overlaying the segmentation with their manually segmented ground truths. Examples of these ground truths are shown in Figure 4.5. Figure 4.6 shows the overall segmentation accuracy for frames 1−30 of the "Table Tennis" sequence and that for frames 10−35 of the "Coast Guard" sequence. This overall segmentation accuracy is defined as,

$$AC(\mathbf{S}) = \sum_{\mathbf{s}=\mathbf{s}_1}^{\mathbf{s}_N} \frac{N_{accu}(\mathbf{s})}{N_{total}(\mathbf{s})} \tag{4.4}$$

where $N_{accu}(\mathbf{s})$ is the number of correctly labeled pixels in $\mathbf{s}$, and $N_{total}(\mathbf{s})$ is the number of pixels in $\mathbf{s}$.

Segmentation results are analyzed holistically for sections of the test sequences. Figures 4.7 and 4.8 show selected segmentation results for sections of the test sequences. Results for the "Dog" and the "Jumping Girl" sequences are shown

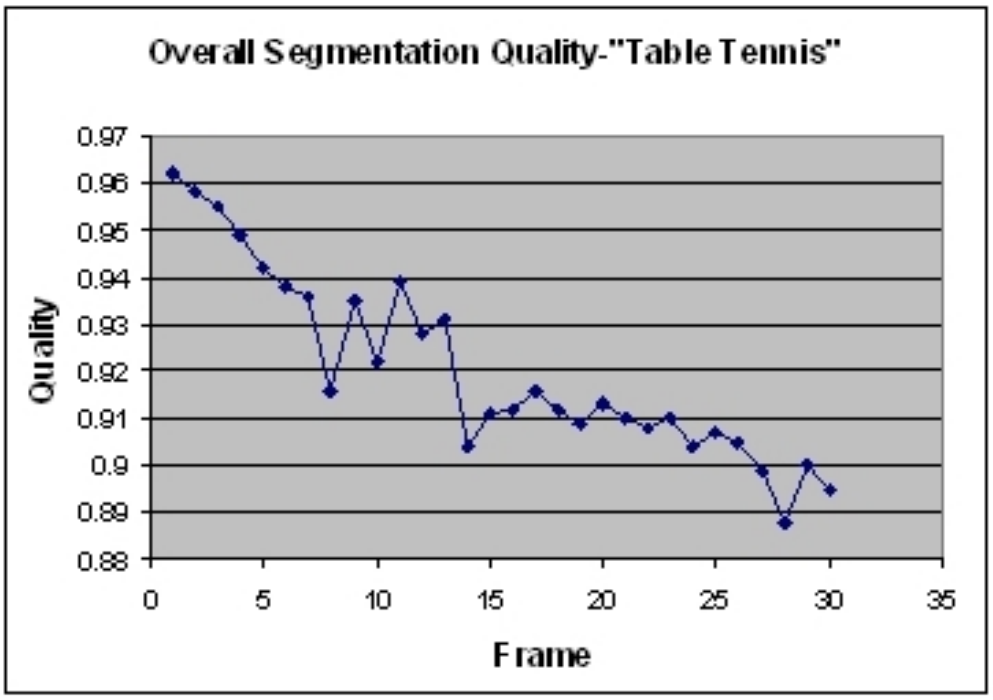


(a)

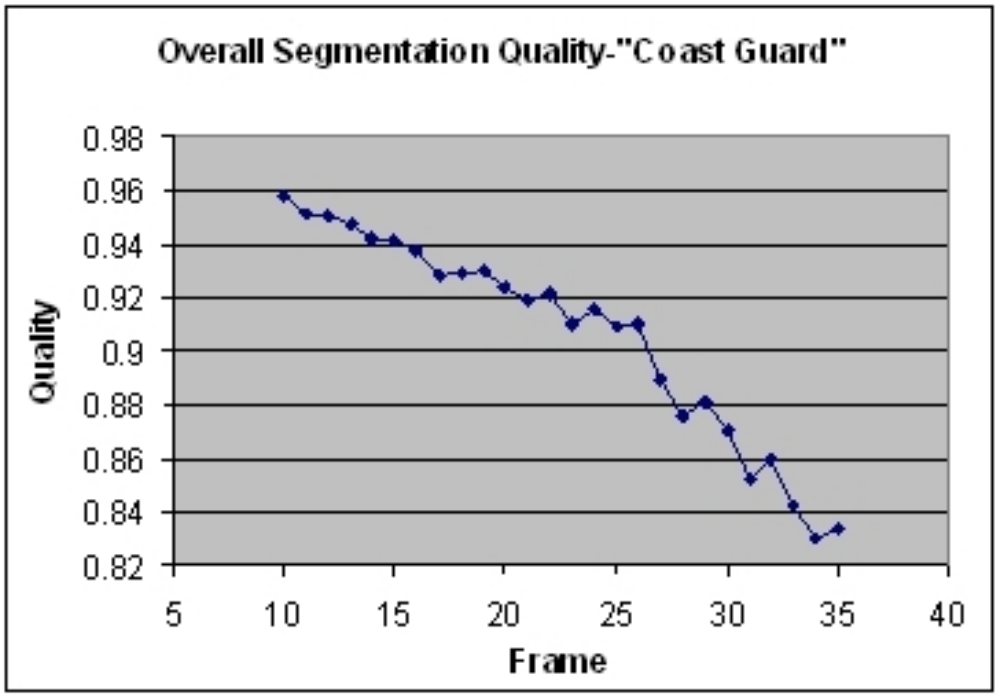
(b)

FIGURE 4.5: (a) Manually segmented ground truth of frame 1 of the "Table Tennis" sequence; (b) Manually segmented ground-truth of frame 10 of the "Coast Guard" sequence.

in Figures 4.9 and 4.10. Without re-initialization, the overall accuracy for the "Table Tennis" sequence drops from an initial value of 96.35% to the lowest value of 88.9% as the temporal section approaches its end. Similar results are observed for the "Coast Guard" sequence, with a maximum drop of 13%. The decline in segmentation accuracy is due to accumulation of propagation error. The results shown also reflect a tolerance limit for acceptable deterioration. Temporal graph validation verifies the predicted pixel labels after propagation, but it does not guarantee an error-free graph aggregation. Some residual error will still be carried over to subsequent frames. Empirically, it is found that to limit the temporal error propagation to within 10%, the maximum propagation time span allowed is about 20 frames. A re-initialization is required to avoid further accumulation of propagation error. As previously discussed in Section 3.5, the decision to re-initialize segmentation can be made based on the percentage of validated pixel labels, but the rejected pixel labels can also be attributed by large independent motions. Such a decision is also dependent on the reliability of the validation.



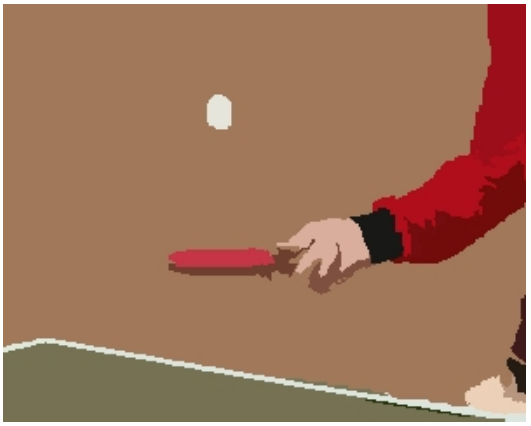
(a)



(b)

FIGURE 4.6: Overall segmentation quality: (a) Overall quality for frames 1−30 of the "Table Tennis" sequence. (b) Overall quality for frames 10−35 of the "Coast Guard" sequence.

(a) Frame 1



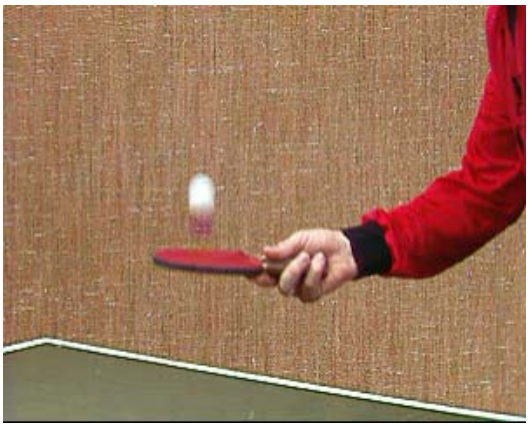
(b) Initialization (16 segments)

(c) Frame 3

(d)

(e) Frame 7

(f)



(g) Frame 13

(h)

(i) Frame 25

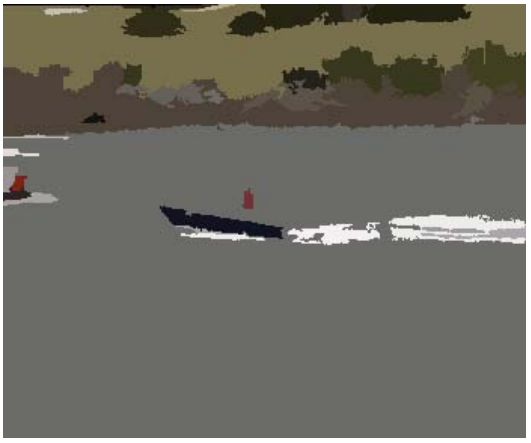
(j)

(k) Frame 30

(l)

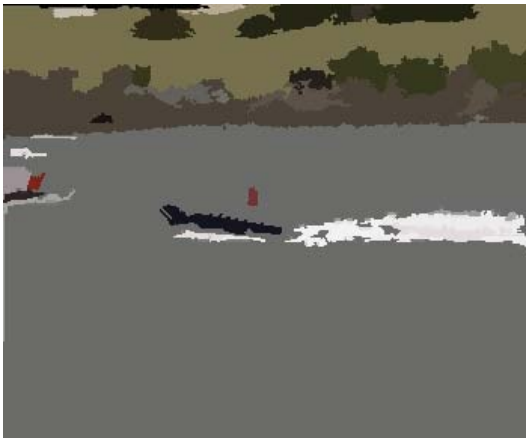
FIGURE 4.7: Selected segmentation results for frames 1−30 in the "Table Tennis" sequence. (a),(c),(e),(g),(i) and (k) Frames 1,3,7,13,25 and 30. (b)Initialized segmentation for frame 1. (d),(f),(h),(j) and (l) Corresponding segmentation results.

(a) Frame 10



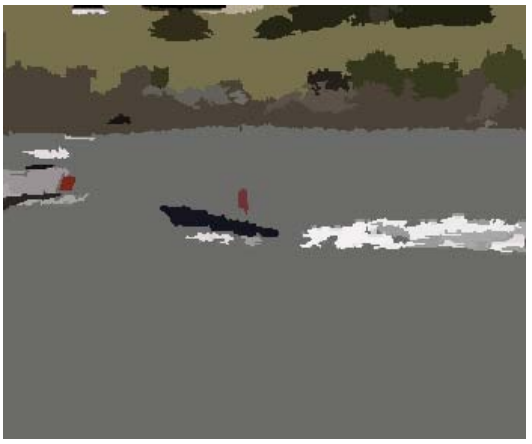
(b) Initialization (33 segments)

(c) Frame 13



(d)

(e) Frame 19



(f)

(g) Frame 22

(h)

(i) Frame 27

(j)

(k) Frame 33

(l)

FIGURE 4.8: Selected segmentation results for frames 10−35 in the "Coast Guard" sequence: (a),(c),(e),(g),(i) and (k) Frames 10,13,19,22,27 and 33. (b)Initialized segmentation for frame 10. (d),(f),(h),(j) and (l) Corresponding segmentation results.

(a) Frame 50

(b) Initialization (13 segments)

(c) Frame 60

(d)

(e) Frame 63

(f)

(g) Frame 67

(h)

(i) Frame 70

(j)

(k) Frame 79

(l)

FIGURE 4.9: Selected segmentation results for frames 50−80 in the "Dog" sequence: (a),(c),(e),(g),(i) and (k) Frames 50,60,63,67,70 and 79. (b)Initialized segmentation for frame 50. (d),(f),(h),(j) and (l) Corresponding segmentation results.

(a) Frame 1 (b) Initialization (14 segments)

(c) Frame 5 (d)

(e) Frame 15 (f)

(g) Frame 20 (h)

FIGURE 4.10: Selected segmentation results for frames 1−20 in the "Jumping Girl" sequence: (a),(c),(e) and (g) Frames 1,5,15 and 20. (b)Initialized segmentation for frame 1. (d),(f) and (h) Corresponding segmentation results.

4.4.2 Comparison against State-of-the-art Video Segmentation

Apart from comparing the segmentation results with the manually segmented ground truth, they are also compared against benchmark results such as segmentation produced by the COST211 Analysis Model [32] as well as Sifakis'

algorithm [33, 35]. This comparison caters mainly to the segmentation quality evaluation of individual objects.

The Cost211 Analysis Model is a collection of image analysis tools which can be flexibly combined to achieve fully automatic segmentation and tracking of moving objects in a video sequence. Both scenes with static textured background and scenes where the background can be described by global motion parameters are considered. The algorithm proposed by Sifakis et al. adopts statistical and level set approaches in formulating moving object detection and localization. For the change detection problem, the inter-frame difference is modelled by a mixture of two zero-mean Laplacian distributions. Statistical tests using criteria with negligible error probability are used for labelling as changed or unchanged as many sites as possible. A multi-label fast marching algorithm was introduced for expanding competitive regions. The solution of the localization problem is based on the map of changed pixels previously extracted. The boundary of the moving object is determined by a level set algorithm. Sifakis' result serves as a reference for the segmentation of scenes containing independent moving objects.

As seen in Figure 4.12, the results for the "Table Tennis" sequence produced by the proposed algorithm compare favorably to the results presented by Sifakis. In Sifakis' results, the independently moving pingpong ball and paddle tend to merge with their neighbouring regions, resulting in inaccurate object boundaries, especially for frames 20 and 30 (Figures 4.12(h) and (k)). On the other hand, the proposed segmentation algorithm successfully tracks and segments these independent objects by the graph propagation and aggregation. The proposed video

segmentation algorithm also compares favorably to the Cost211 Analysis Model which fails to segment out the pingpong ball (Figure 4.12 (i)).

As for the "Coast Guard" sequence, as the Cost211 Analysis Model could not identify any moving objects for the first 30 frames, the proposed algorithm is only compared against Sifakis'. The proposed algorithm performs better than Sifakis', in terms of segmentation quality of the boat and the water tail. Note that part of the water tail is cut off in Sifakis' results.

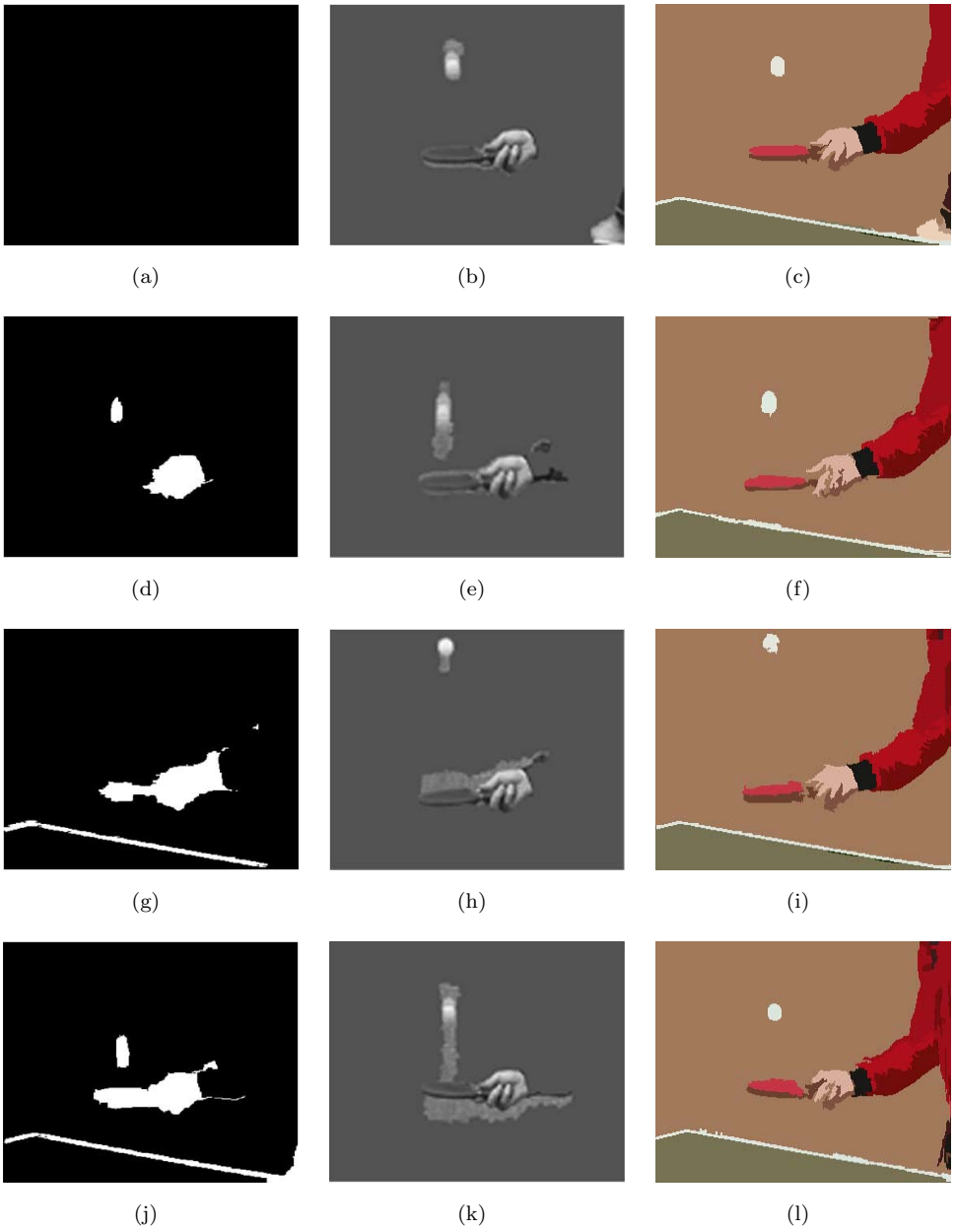
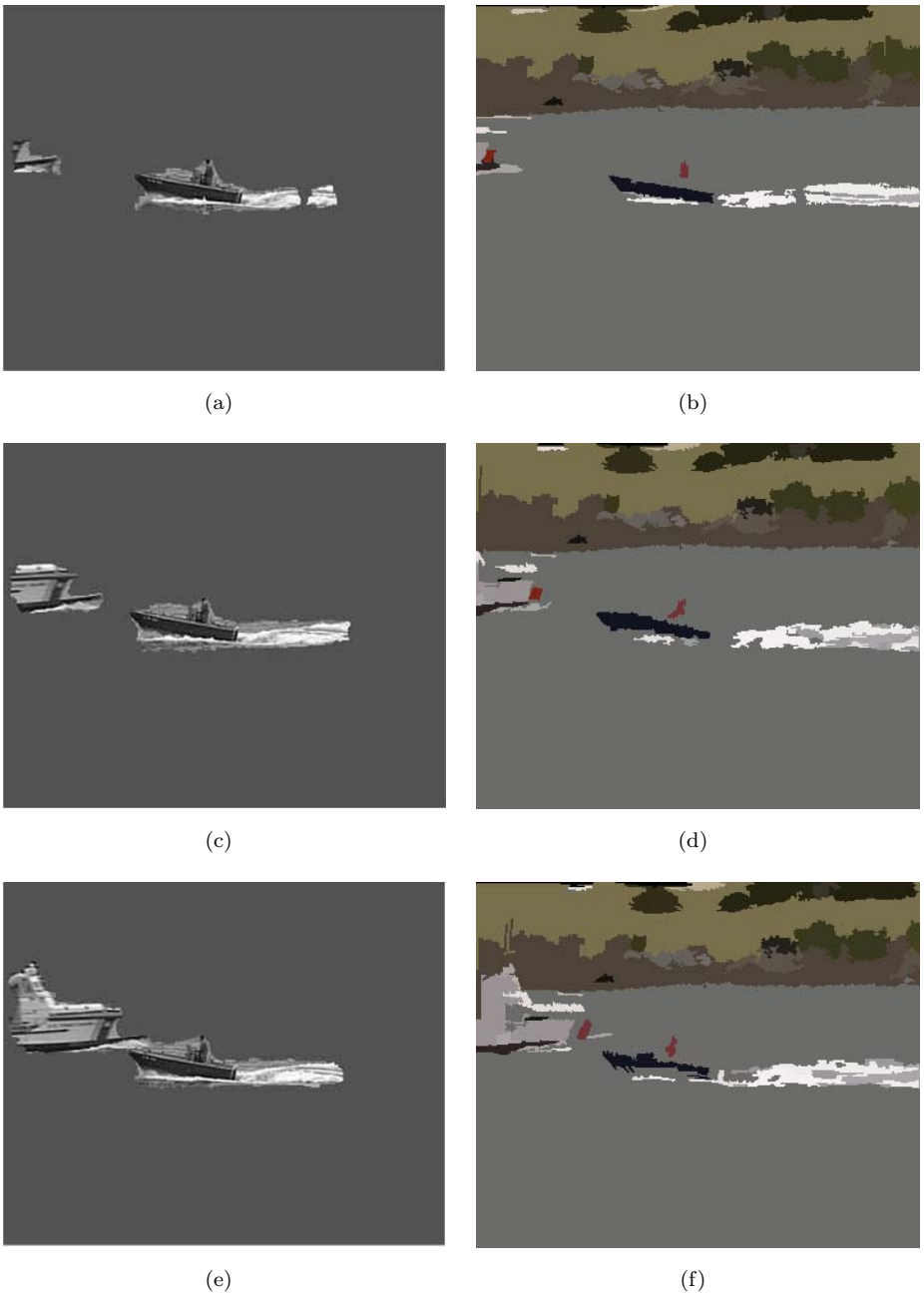


FIGURE 4.11: Comparison of segmentation results for the frames 1−30 of the "Table Tennis" sequence: (a),(d),(g) and (j) Segmentation masks for frames 1, 10, 20 and 30 of the Cost211 Analysis Model; (b),(e),(h) and (k) Corresponding segmentation results produced by Sifaki et al.; (c),(f),(i) and (l) Corresponding segmentation results produced by the proposed graph-based algorithm. The Cost211 results lost track on the pingpong ball for frame 20 (g).



(a) (b)

(c) (d)

(e) (f)

FIGURE 4.12: Comparison of segmentation results for the frames 10−35 of the "Table Tennis" sequence: (a),(e) and (e) Segmentation masks for frames 10, 20 and 30 presented by Sifakis; (b),(d) and (f) Corresponding segmentation results produced by the proposed algorithm. The Cost211 Analysis Model could not identify any moving objects for the first 30 frames of the sequence, hence results are not available.

Chapter 5

Future Work and Conclusions

In this work, an efficient algorithm is proposed to gain leverage on temporal redundancy in video sequences. The proposed algorithm exploits the inter-frame correlation to propagate trust-worthy grouping from the previous frame to the current. A preceding graph is constructed and labeled for the previous frame. It is temporally propagated to the current frame, validated by the similarity measures, and spatially aggregated for the final grouping. In doing so, one can retain maximally the propagated segmentation results and hence lessen the computational burden of re-segmenting every frame. Experimental results demonstrated the proposed algorithm's strength in handling fast independent motion and appearance of new objects through graph validation and aggregation processes. To evaluation the performance of the proposed video segmentation algorithm, both standalone and relative methodologies were adopted. Results show that, for the standalone evaluation, the proposed graph propagation and aggregation method

is able to preserve spatial uniformity. For the relative comparison, an overall segmentation evaluation based on manually segmented ground truth suggests that the segmentation accuracy declines over time due to accumulation of propagation error, but a re-initialization can be easily incorporated to tackle this problem. Results of the proposed algorithm also compare favorably to benchmark results which include segmentation by the COST211 Analysis Model and that produced by Sifakis et al., especially in the handling of fast moving and independent moving objects.

The current algorithm validates pixel labels based on color information and it may not be sufficient to handle lighting variations. For future work, a more robust subgraph validation approach is aimed to be achieved, such as correlation matching which considers multiple low-level cues in a local neighbourhood on top of the currently adopted color similarity check. In addition, an automatic scheme to re-initialize the segmentation output to minimize propagation error is also desirable. The percentage of validated pixel labels may not be a reliable indicator for re-initialization because large independent motions can also cause a significant drop in this measure. In the presence of large independent motion or abrupt motion, one has to strike a balance between temporal correlation (when correlation is low for some objects) and spatial coherence (re-initialization) to avoid compromising region label consistency.

The proposed video segmentation algorithm has a wide range of potential applications. It is applicable for content-based video coding or compression, or a content-based multimedia application such as video object querying. The generic

segmentation algorithm can also be made more task-specific by incorporating prior knowledge for tasks such as target object segmentation and background/foreground modelling.

Appendix A

Mathematical Models

A.1 Markov Random Field (MRF)

Consider a set of random variable $X = X_1, X_2, \cdots, X_n$ defined on the set S, such that, each variable X_i can take a value x_i from the set $L = l_1, l_2, \cdots, l_n$ of all possible values. Then X is said to be a MRF with respect to a neighbourhood system $N = \{N_i | i \in S\}$ if and only if it satisfies the positivity property $P(x) > 0$, and Markovian property $P(x_i|x_{S-i}) = P(x_i|x_{N_i})$, $\forall i \in S$. Let $P(x)$ represent $P(X = x)$ and $P(x_i)$ represent $P(X_i = x_i)$. Refer to the joint event $(X_1 = x_1, \cdots, X_n = x_n)$ as $X = x$ where $x = \{x_i | i \in S\}$ is a configuration of X corresponding to a realization of the field. The MRF-MAP estimation can be formulated as an energy minimization problem where the energy corresponding to the configuration x is the negative log likelihood of the joint posterior probability of the MRF and is defined as

$$E(x) = -logP(x|D) \tag{A.1}$$

where D is the observation (such as pixel intensities).

A.1.1 MRF for Image Segmentation

In the context of image segmentation, S corresponds to the set of all image pixels, N is a neighbourhood defined on this set, the set L comprises of labels representing the different image segments, and the random variables in the set X denote the labelling of the pixels in the image. Note that every configuration x of the MRF defines a segmentation. The image segmentation problem can thus be solved by finding the least energy configuration of the MRF. The energy corresponding to a configuration x consists of a likelihood and a prior term as

$$\Psi_1(x) = \sum_{i \in S} \left(\phi(D|x_i) + \sum_{j \in N_i} \psi(x_i, x_j) \right) + const \tag{A.2}$$

where $\phi(D|x_i)$ is the log likelihood which imposes individual penalties for assigning label l_i to pixel i and is given by

$$\phi(D|x_i) = -\log P(i \in S_k|H_k) \text{ if } x_i = l_k \tag{A.3}$$

where H_k is the RGB distribution for S_k, the segment denoted by l_k. Here, $P(i \in S_k|H_k) = P(I_i|H_k)$, where I_i is the color intensity of the pixel i. The prior $\psi(x_i, x_j)$

takes the form of a Generalized Potts model

$$\psi(x_i, x_j) = \begin{cases} K_{ij} & \text{if } x_i \neq x_j \\ 0 & \text{if } x_i = x_j \end{cases} \tag{A.4}$$

In MRFs used for image segmentation, a contrast term is added which favours pixels with similar color having the same labels. This is incorporated in the energy function by reducing the cost within the Potts model for two labels being different in proportion to the difference in intensities of their corresponding pixels.

A.2 Max-flow/Min-cut Algorithm

One of the fundamental results in combinatorial optimization is that the minimum s-t cut problem can be solved by finding a maximum flow from the sources s to sink t. The theorem of Ford and Fulkerson [15] states that a maximum flow from s to t saturates a set of edges in the graph dividing the nodes into two disjoint parts S, T, corresponding to a minimum cut. Thus min-cut and max-flow problems are equivalent.

Theorem A.1. (Max-flow Min-cut Theorem) *In every network, the maximum value of a feasible flow equals the minimum capacity of a source/sink cut.*

A.2.1 Ford−Fulkerson Algorithm

The *Ford−Fulkerson Algorithm* computes the maximum flow in a flow network. As long as there is a path from the source (start node) to the sink (end node), with available capacity on all edges in the path, flow is sent along one of these paths. Then another path is sought, and so on. A path with available capacity is called an augmenting path. The detailed algorithm is as follows.

Algorithm A.1. *(Ford−Fulkerson Labelling Algorithm)*

Input: A feasible flow f in a network

Output: An f-augmenting path or a cut with capacity $val(f)$

Idea: Find the nodes reachable from s by paths with positive tolerance. Reaching t completes an f-augmenting path. during the search, $\mathbf{R}$ is the set of nodes labelled *Reached*, and $\mathbf{S}$ is the subset of $\mathbf{R}$ labelled *Searched*.

Initialization: $\mathbf{R} = s$, $\mathbf{S} = \emptyset$

For each existing edge vw with $f(vw) < c(vw)$ and $w \notin \mathbf{R}$, add w to $\mathbf{R}$.

For each entering edge uv with $f(uv) > 0$ and $u \notin \mathbf{R}$, add u to $\mathbf{R}$ Label each vertex added to $\mathbf{R}$ as "reached", and record v as the vertex reaching it. After exploring all edges at v, add v to $\mathbf{S}$.

If the sink t has been reached (put in $\mathbf{R}$), then trace the path reaching t to report an f-augmenting path and terminate. If $\mathbf{R} = \mathbf{S}$, then return the cut $[\mathbf{S}, \overline{\mathbf{S}}]$ and terminate. Otherwise, iterate.

Bibliography

[1] J. Goldberger and H. Greenspan. Context-based segmentation of image sequences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(3):463–468, 2006.

[2] E. Shechtman Y. Wexler and M. Irani. Space-time video completion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(3):463–476, 2007.

[3] I. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. Wiley, 2003.

[4] R. Megret and D. DeMenthon. A survey of spatio-temporal grouping techniques, 1994. Technical report: LAMP-TR-094/CS-TR-4403, University of Maryland, College Park.

[5] Y. Wang, K. F. Loe, T. Tan, and J. K. Wu. Spatio-temporal segmentation based on graphical models. *IEEE Trans. Image Processing*, 14(7):937–947, 2005.

[6] M. Culp and G. Michailidis. Graph-based semisupervised learning. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(1):174–179, 2008.

[7] F. Wang and C. Zhang. Label propagation through linear neighbourhoods. *in Proc. International Conference on Machine Learning*, pages 290–294, 2006.

[8] T. Joachims. Transductive learning via spectral graph partitioning. *in Proc. International Conference on Machine Learning (ICML)*, 12(2):2003, 2003.

[9] S. Liu, G. Dong, C. H. Yan, and S. H. Ong. Video segmentation: Propagation, validation and aggregation of a preceding graph. *in Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[10] S. Z. Li. *Markov Random Field Modeling in Image Analysis, 2nd Edition.* Springer, 2001.

[11] I. Patras, E. A. Hendriks, and R. L. Lagendijk. Video segmentation by map labeling of watershed segments. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(3):326–332, 2001.

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

[13] L. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory*, 21(1):32–40, 1975.

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

[15] L. Fukunaga and L. Hostetler. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[17] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. *in Proc. IEEE International Conference on Computer Vision*, pages 70–77, 1999.

[18] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[19] B. Li and R. Chellappa. Face verification through tracking facial features. *Journal of the Optical Society of America*, 18:2969–2981, 2001.

[20] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. *in Proc. International Conference on Computer Vision*, pages 1071–1076, 1995.

[21] S. Gepshtein and M. Kubovy. The emergence of visual objects in space-time. *Proceedings of the National academy of Science*, pages 8186–8191, 2000.

[22] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. *in Proc. International Conference on Computer Vision*, pages 1151–1160, 1998.

[23] H. Greenspan, J. Goldberger, and A. Mayer. A probabilistic framework for spatio-temporal video representation and indexing. *in Proc. European Conference on Computer Vision*, pages 461–475, 2002.

[24] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. *in Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 993–1000, 2006.

[25] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. *in Proc. IEEE International Conference on Computer Vision*, pages 290–294, 2007.

[26] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[27] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *ACM Trans. Communications*, 24(6):381–395, 1981.

[28] B. Georgescu C. Christoudias and P. Meer. Synergism in low level vision. *in Proc. International Conference on Pattern Recognition*, 4:150–155, 2002.

[29] R. Sedgewick. Algorithms in c. 1990.

[30] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(6):929–944, 2007.

[31] P. L. Correia and F. Pereira. Objective evaluation of video segmentation quality. *IEEE Trans. Image Processing*, 12(2):186–200, 2003.

[32] A. A. Alatan, R. Mech E. Tuncel L. Onural, M. Wollborn, and T. Sikora. Image sequence analysis for emerging interactive multimedia services–the european cost211 framework. *IEEE Trans. Circuits, System and Video Technology*, 8:19–31, 1998.

[33] E. Sifakis and G. Tziritas. Moving object localization using a multi-label fast marching algorithm. *Signal Processing: Image Commuications*, 16:963–976, 2001.

[34] Recommendation p.910–subjective video quality assessment methods for multimedia applications. *Recommendations of the ITU (Telecommunication Standardization Sector)*, 1996.

[35] I. Grinias E. Sifakis and G. Tziritas. Moving object localization using a multi-label fast marching algorithm. *EURASIP Journal on Applied Signal Processing*, 2002:379–388, 2002.