A Region-Based Filter for Video Segmentation

by

Micheal Yurick

HB.Sc., Queen's University, 2003

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

The Faculty of Graduate Studies

(Computer Science)

The University Of British Columbia

November 2006

Abstract

This thesis addresses the problem of extracting object masks from video sequences. It presents an online, dynamic system for creating appearance masks of an arbitrary object of interest contained in a video sequence, while making minimal assumptions about the appearance and motion of the objects and scene being imaged. It examines a region-based approach, in contrast to more recently popular pixel-wise approaches to segmentation to illustrate the advantages in the reduction of the complexity of the labeling problem.

The redundancy of information typically present in a pixel-wise approach is exploited by an initial oversegmentation of the current video frame. The oversegmentation procedure is based upon a modified version of the classic watershed segmentation algorithm. This oversegmentation produces a set of appearance/motion-consistent regions upon which a conditional random field is constructed. Observations at each region are collected based upon the colour statistics within a region and the motion statistics as determined by the optical flow over the region. An unparameterized model for both the object of interest and the remainder of the scene are constructed on a frame by frame basis.

The conditional random field model is used in conjunction with a first order hidden markov model over the frames of the sequence. Mean field approximations for variational inference in this model produce a region-based filter framework which incorporates both spatial and temporal constraints. This framework is used to determine an appropriate labeling for each region in each frame. The reduction in the complexity of the field model produced by the regions (as opposed to pixels) results directly in a reduced cost for the labeling problem with minor effects on accuracy.

Contents

List of Tables

List of Figures

Acknowledgements

Since, as a rule, I prefer not to exclude anyone, I will endeavor to remain unspecific. I will simply thank all of the people who offered moral, intellectual and financial support to me in any way, shape or form. Despite what it may seem at times, a Master's Degree is something that is accomplished far from individually, and I sincerely thank all of those who contributed to mine.

And now for the exceptions that prove the rule. I would like to thank Dr. Nando de Freitas for his theoretical clarifications and for agreeing to be my second reader. And I would like to specifically thank my supervisor, Dr. Jim Little, whose calming influence and assurances allowed me to keep things in perspective and to maintain my sometimes tenuous grip on sanity. Thanks Jim!

Chapter 1

Introduction

The task of image segmentation is well known and long studied. The ability to separate an object of interest from a dynamic scene has myriad applications, and is accomplished effortlessly in humans during virtually every instant of our waking lives. Like many perceptual vision problems, it is perhaps due to the entirely effortless nature with which this task is performed by humans that has caused the problem of image segmentation in computational vision to be somewhat ill-defined.

The goal of image segmentation has often been argued, namely because it is very hard to define what a "good" segmentation is. The simplest definition of the problem can likely be described as achieving a partitioning of the visual elements within an image so that similar elements are grouped together. Similarity could be determined in terms of such low level cues as colour or texture, or it could be determined through higher level notions of the conceptual consistency associated with the objects in a scene. A single image can produce several different segmentations that each optimize such different, but valid, criteria. The segmentation of a scene even by human subjects can vary from person to person depending on the individual interpretation of the scene (see Figure 1.1).

As suggested, at the core of segmentation is the idea of grouping perceptually consistent items. The perceptual grouping problem has been a long studied one in the fields of psychology and neuroscience [32]. Two key concepts from this literature are typically exploited in the segmentation problem; similarity and proximity (see Figure 1.2). One interpretation can simply be to group elements of a visual scene which are both contiguous and similar in appearance.

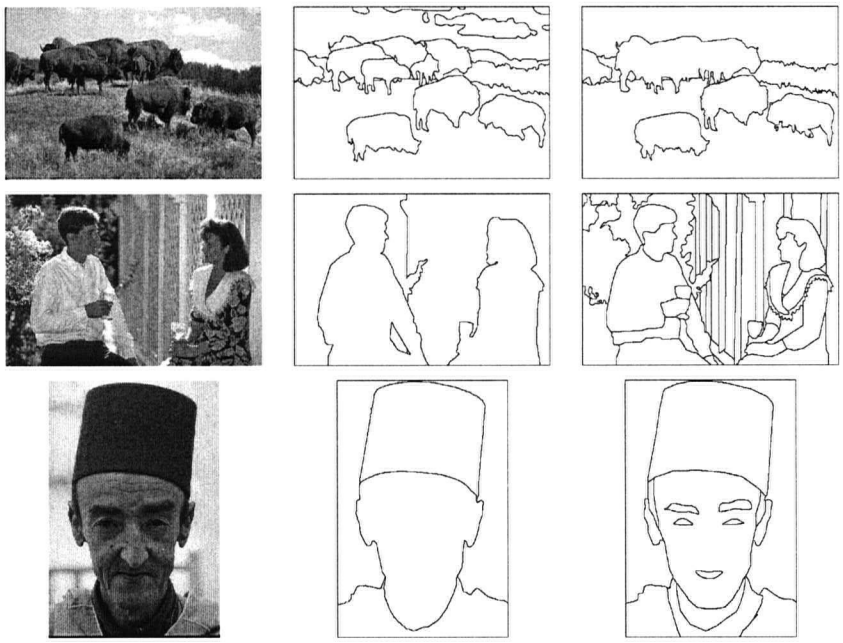

Figure 1.1: Different segmentation interpretations by different human subjects, courtesy of the Berkeley segmentation database [26]. On the left are the original images. In the centre and on the right are the hand segmentations provided by different subjects asked to segment the images. There are obvious and significant variations between subjects for identical images.

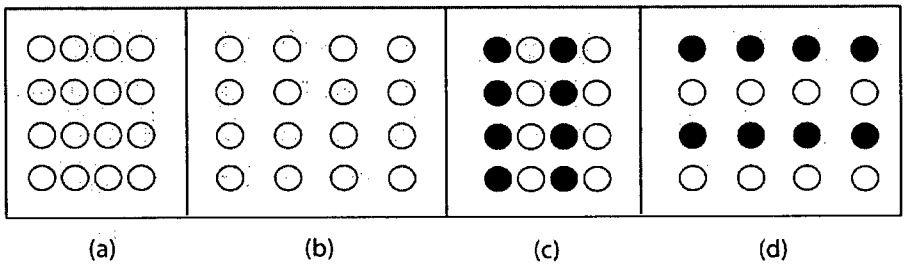

Figure 1.2: Classic examples of grouping by proximity and similarity. Upon observing the dot configurations (a) and (b), subjects typically report perception of four rows for (a), and four columns for (b). This effectively illustrates the tendency to group according to proximity. Conversely, the configurations (c) and (d) typically elicit the perception of four columns and four rows, respectively. This shows the tendency to group according to similarity, while at the same time illustrating the ability of one cue to supersede another.

Consequently, solutions to the task of image segmentation typically delineate a series of boundaries within an image which separate and distinguish these main conceptual and visual groups within the image. At a lower level, this can mean simply identifying boundaries across which there exist large changes in image intensities or visual textures, in order to group together those which are most similar and contiguous. However, at a higher level, the aim is typically to produce a segmentation which separates specific objects of interest within a scene from the background, rather than one that specifically optimizes lower level criteria successfully without achieving a consistent higher level interpretation.

As a result, a "good" segmentation of an image is often defined as one that would, in high likelihood, also be given by a human subject. This makes the definition of the problem somewhat clearer. Still, as illustrated, different human subjects can disagree to some extent on their choices for segmentation boundaries in an image, but overall some definite trends can be observed. At the very least, statistics can be collected on which elements of an image were included within one segmentation versus another in order to establish the ground

truth segmentation for a given image.

The static image segmentation problem still remains an extremely difficult one. Only recently has it been recognized that to perform such a task and achieve a result similar to that of a human typically requires some information about the object which is being segmented. As a result, approaches which combine both bottom-up and top-down cues for static image segmentation are currently seeing the most success. However, it is still apparent that humans can learn about and segment novel objects for which we do not have a previous experience. Indeed, we have been doing it since infancy.

One explanation for this might be the robust cues that the general motion similarity and temporal consistency of everyday objects provide in the dynamic scenes we experience throughout our daily lives. These concepts are intimately tied to the idea of common fate [32], another of the classic perceptual grouping criteria. Common fate represents the tendency to group together elements that exhibit similar behaviour over time. In this way, segmenting objects from video sequences is directly related to but distinct from static image segmentation.

Between the two problems lie the above mentioned important differences. Specifically, powerful temporal cues are present in the segmentation of a video sequence that are not present in the segmentation of a static image. There is an obvious interpretation of the segmentation of a video sequence, induced by the coherent nature of the visual elements appearing over the course of an image sequence. Such an interpretation may not be clear in a static image where such cues are absent. The consistent appearance and motion of an object over a series of frames as well as its tendency to occupy a contiguous space over time lends itself naturally to less ambiguity at both lower and higher levels. This leads us to define the problem of segmentation in video sequences.

Similarly to static image segmentation, the goal of object segmentation in video sequences is to robustly separate elements of each image which describe coherent objects of interest from the other elements and objects of the sequence. Specifically, this amounts to determining which elements in each frame of the sequence belong to an object of interest, and which do not. However, in the

case of an image sequence, this partitioning of visual elements into coherent structures must also necessarily consider the notion of consistency over time. There are many frustrating factors in performing such a task when using the grouping cues as described above.

Simply segmenting by similarity and proximity is often an inadequate approach. Several constructs with a similar appearance may be present in the scene. As a result, simply identifying those elements with a similar appearance and contiguous nature to the object of interest can easily result in drastic errors. On the other hand, simply grouping by common fate (which in many cases is interpreted simply as consistent motion) can easily lead to similar erroneous labeling of elements which only happen to exhibit similar motion. Combining these cues can certainly help to reduce confusion between similarly appearing or similarly moving objects. In addition, the nature of objects in a dynamic scene to occupy not only contiguous space but to do so over time is another cue to be exploited.

The question then becomes how to combine these cues to achieve the segmentation task. This question remains open, and it is this question which is central to this work. We present a novel approach to the video segmentation problem. Unlike some more popular recent approaches which operate by assigning image elements to the object of interest on the level of pixels, this approach works at the level of regions determined by an initial oversegmentation of the image into elements of consistent appearance (see Figure 1.3).

The aim here is two-fold. First, the most general of pixel based approaches, such as [46], note the undesirable effects of noise in image and motion data on the resulting segmentation. To counteract such effects, a probabilistic approach is employed to combine the above cues and achieve the partitioning of the visual elements of each frame by assigning a label (e.g., object or background) to each pixel in the image. Such models typically incorporate constraints to deal with such noise. However, defining observations by examining the data over an entire region instead of on a pixel by pixel basis is another way to further reduce the effects of pixel-wise noise in the data.

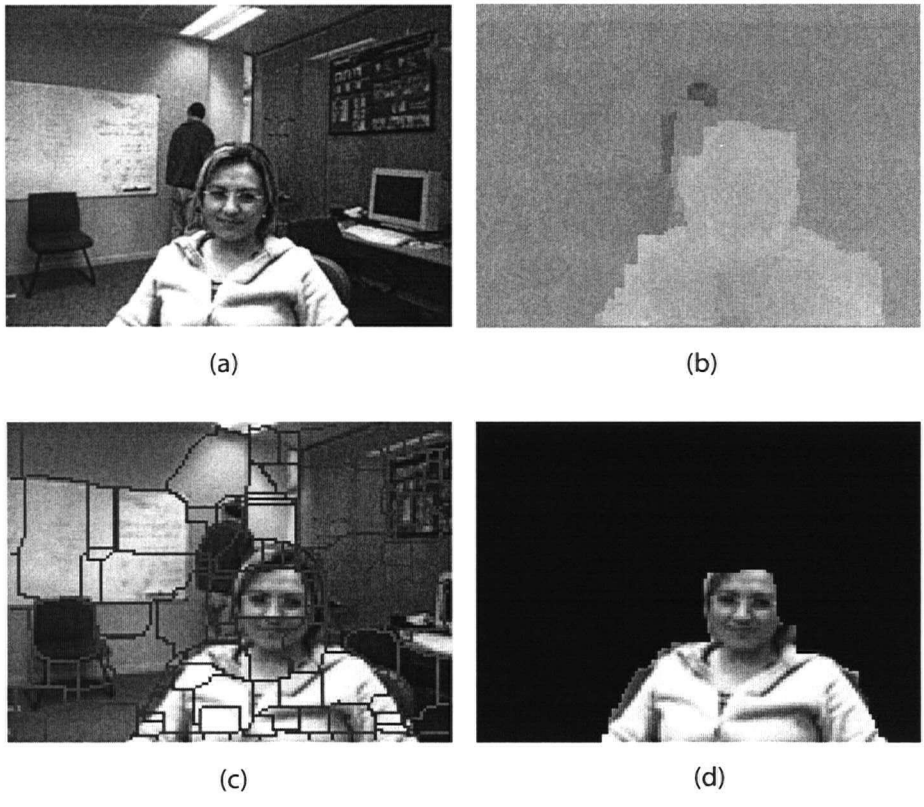

Figure 1.3: A brief illustration of the overall process. (a) An original image from a sequence. (b) Motion information from the previous frame to the current one in the form of optical flow magnitudes. (c) The oversegmentation of the image based upon appearance and motion information. (d) A labeling of the regions from (c) which incorporated motion and appearance information.

Second, to sufficiently reduce this noise, it was found necessary for the identity of the label at a given pixel to be affected by a large number of nearby pixels. This requires an examination of each of these neighbouring pixels for each given pixel in the image. Larger neighbourhood sizes result in a direct increase in the complexity of the probabilistic model being applied to achieve the segmentation as dependencies between each pixel and each of its neighbours must be

considered. This in turn results directly in an increase in cost for determining an optimal partitioning as defined by the model. If instead the model was based upon regions consisting of groups of pixels with very similar appearance, rather than the pixels themselves, the model complexity could be greatly reduced.

To this end, an initial oversegmentation step is performed to exploit the redundancy in the data at a pixel-wise level. Groups of pixels of relatively constant appearance and motion are identified and treated as a single region entity. The process by which this is accomplished as presented in a following section includes a novel modification to the classic watershed algorithm [44] to incorporate motion information. The resulting regions are then determined as either belonging to an object of interest within each video frame, or not.

The proposed method takes a probabilistic approach similar to that in [46]. The above cues are combined to achieve the partitioning of the visual elements in each image frame. A distribution is defined over the observed image data and the hidden variables which determine a label for each region in an image as determined by the initial oversegmentation. A discriminative framework is adopted, in contrast to the more classical generative approach of similar models, the advantages and disadvantages of which are discussed in a later section. Unlike previous region-based approaches, this method presents a unified filter framework, generalized from [46] to apply to situations where the structure of the probabilistic model used to determine the labeling from frame to frame may change. The framework imposes, on a region-based level, the spatial constraints associated with similarity, proximity and common fate, as well as the temporal constraints associated with the relatively consistent nature of an object over time.

An outline of the remaining pages in this document is as follows. A survey of related work to this problem is presented in Section 2. It examines the various approaches taken in the past, their strengths and weaknesses, and identifyies the situation of this work amongst them. Section 3 presents in detail the theory necessary to understand the operation of this approach. The implementational details are discussed in Section 4, and some results of this method applied to real

video sequences are presented in Section 5. Section 6 presents a brief discussion of possible paths available for future work.

Chapter 2

Related Work

2.1 Introduction

Segmentation in a sequence of images has been a subject of interest and research for a long time. In general, video segmentation techniques attempt to exploit the spatiotemporally coherent nature of the elements in a video scene. The aim then becomes to identify the elements within the scene that move and appear consistently. Many approaches have been taken to solving the problem posed in this way, and this section examines some of the major directions taken in the past, as well as their strengths and weaknesses.

Regardless of the exact approach taken, spatiotemporal segmentation techniques typically try to group together image features into meaningful and coherent visual structures. The features used in the grouping process vary from method to method, but they usually come in one of three forms. Many techniques operate directly on the level of image pixels [10, 25, 45, 46], and treat each pixel as a separate entity that needs to be grouped with other pixels to form an object. Others work at the level of regions or patches that result from an initial grouping of pixels based strictly upon low levels cues like colour and texture [29, 33, 42]. The regions are then combined in some fashion to form distinct objects. Lastly, some methods perform a grouping based primarily on geometric features or interest points computed from image data [9, 43]. Either the objects are represented simply as the collection of features, or regions can be grown in images around feature locations that are deemed to belong to a single object.

Video segmentation techniques have, in the past, been presented as falling into one of three categories [27]: (1) techniques that segment an image sequence with a priority on spatial cues (i.e., cues contained within a single frame of the sequence) and then integrate temporal consistency; (2) techniques that segment with a priority on temporal cues (i.e., cues that are spread across the frames of the sequence) and then search for spatial support; and (3) techniques that segment jointly over spatial and temporal cues simultaneously.

Each category has its own general approach to combining spatiotemporal information, each with its advantages and disadvantages. Temporally prioritized methods often rely upon the ability to track features or regions throughout a sequence, which can allow for robust estimates for the motion of an object. However, this can become a problem when these entities cannot be correctly corresponded across frames or simply disappear periodically over the course of a sequence. Joint spatial and temporal methods can incorporate more data to resolve such ambiguities. However, the amount of data involved in simultaneously processing of an entire video volume can become quite prohibitive.

Spatially prioritized methods enjoy much attention due to the close similarity they share with the well studied static image segmentation problem. Many methods have been developed to explore the different manners in which temporal cues can be incorporated into frame by frame spatial structures. Because the proposed method is most closely related to techniques which place a priority on spatial cues, the review will begin with the latter two categories.

2.2 Temporally Prioritized Methods

Methods that rely primarily upon temporal segmentation typically operate first by extracting interest points or regions across a series of frames within a sequence. These methods then perform some manner of temporal grouping over the features into what are often referred to as motion trajectories (see Figure 2.1). The goal is typically to identify which features exhibit a coherent motion over time. Grouping over a longer time interval (as opposed to a single pair

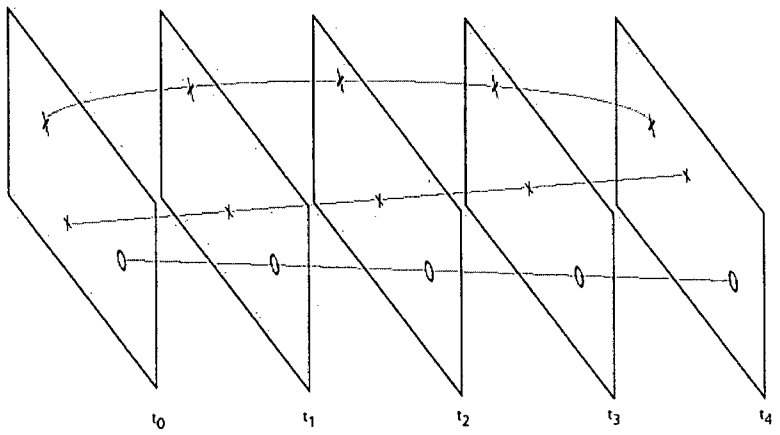

Figure 2.1: An illustration of grouping features into trajectories over time. Three distinct features on three separate trajectories. Trajectories can be subsequently grouped together into objects.

or triplet of frames) allows for more accurate estimations of motions, making it easier to discriminate between features belonging to distinct objects.

Many different approaches to the temporal grouping of features have been proposed. Originally, some methods simply relied on a direct comparison of motion trajectories that have been collected over a sequence [2, 28]. Trajectories can be represented simply as vectors in a high dimensional space, or by some combination of temporal derivatives [2]. Distances can then be measured to determine similarity between vectors and an appropriate clustering can be determined [28].

Other temporal methods rely on the assumption that under certain camera models, motion trajectories associated with different objects in a scene fall into separate parameter subspaces. Several methods have been developed to exploit this property and cluster trajectories into sets of consistently moving features which define an object [9, 34, 39]. These methods are intimately tied to the well known structure from motion problem [16]. Many such methods fail, however, when the data includes frames where no features are visible on some trajectories.

Incomplete or inconsistent data is a common problem in video sequences due to occlusions as objects move throughout the scene.

Vidal and Hartley [43] present a method termed generalized principal components analysis (GPCA) which elegantly deals with such problems. The segmentation of motion trajectories is again treated as a clustering problem where affine motion data is mapped into a higher dimensional space. The mapping employs a form of nonlinear function which automatically deals with the problem of missing data along motion trajectories. Subspaces corresponding to the motion of each specific object in the scene are automatically determined by fitting a set of homogeneous polynomials to the projected motion data and obtaining basis vectors using the polynomial derivatives. The result is a method robust to incomplete motion trajectories.

Some methods instead try to determine the number and parameters of probabilistic models which best fit the motion data in a sequence. They treat the segmentation problem as one of model selection and parameter estimation, which is in itself an entire field of research (the discussion of which is out of the scope of this review). Motions of the features tracked between frames are used as data points which are typically fit to a mixture model whose optimal parameters can be determined through any number of estimation techniques. A classic example is presented in [40] where a parametric mixture model is learned for all objects in a sequence. Each object is represented by a mixture component in the model. A classic probabilistic modeling algorithm termed expectation maximization (EM) [11] is used to cluster the trajectories in a sequence by assigning each a label which corresponds to one of the mixture components.

Drawbacks of such temporal-based methods include the necessity and ability to track an object, or more accurately a distinct subset of its features, through a volume of video frames in order to build a set of trajectories. This, of course, requires access to an entire video sequence ahead of time. Additionally, such methods typically result in groups of points or small regions belonging to an object which could be reliably tracked throughout a sequence, and not an actual appearance-based segmentation of the object within the sequence. Finally, often

implicitly assumed in such methods is some notion of object rigidity so that trajectories on an object are sufficiently similar to be clustered separately from other elements in the video scene, or a sufficiently simple model can be fit to the motion.

2.3 Joint Spatial and Temporal Methods

Similarly to temporal prioritization, methods which attempt to more equally prioritize spatial and temporal cues also typically use information throughout a video volume to perform a segmentation. At the same time, these methods consider spatial appearance and similarities within an image. In some cases, this means simply performing the parameter estimation of a model which includes not only motion information but spatial appearance information as well. For example, Greenspan et al. [15] use the aforementioned EM algorithm to learn a Gaussian mixture model in a 6 dimensional space, where measures for colour, spatial position and temporal position are included. Each mixture component in the model describes a coherent spatio-temporal region.

Other methods take a graph-based approach to segmenting the video volume [13, 35]. In general, graph-based approaches treat the segmentation problem as the minimization of an energy function represented in a graphical format. This energy function is typically defined by a cost for dissimilarity between the desired labeling of an image pixel and the agreement with the observed data at that pixel, while at the same time including some notion of cost for dissimilarity between pixels. A different labeling of the pixels results in different costs as determined by the energy function.

In the original graph cuts framework [6], a graph is constructed based upon a current labeling of the image pixels and the dissimilarity costs associated with such a labeling as defined by the energy function. Pixels are represented by nodes in the graph, and similarity measures are assigned to the edges connecting adjacent pixels. A source and sink node are included in the graph, representing a possible change in label for each pixel, and an edge representing the cost

associated with such a labeling is added. The optimal partitioning of the nodes in such a graph, as determined by the cost associated with the edges cut to achieve the partition, describes a new labeling. This new labeling is shown to be a step towards the optimal solution of the energy minimization problem. As described in [6], efficient techniques for computing optimal partitions in these types of graphs are already well-known in the graph theoretic community. These techniques can be applied in an iterative fashion to the graphs defined by the new labeling after each step in order to achieve an optimal configuration.

In [35], a similar graph structure is created where each node represents a pixel in the video volume, and edges are drawn both between pixels within an image (spatial constraints) and pixels across different frames within a sequence (temporal constraints). A motion-based similarity measure is used and costs are assigned to the edges depending on motion values at each pixel. The partition of the graph in this framework is achieved using the normalized cuts algortihm [36]. This involves solving an approximating system of linear equations determined by the structure of the graph and the associated costs as defined by the similarity functions. The solution to this system determines the "optimal" breaking of edges, resulting in spatiotemporal segments defined by those nodes of the graph which remain connected.

In [13], the pixel-based graph structure includes a feature vector at each node with information about motion and appearance at the corresponding pixel, as well as information about its spatial and temporal location. Different feature dimensions are weighted independently and normalized cuts is again applied, this time in an approximate framework.

In [48], an initial step is added to this process where an estimate for the appearance of spatiotemporal segments is obtained. Seed correspondences are determined over a short section of the video and regions are grown around them using a graph cuts approach integrated with a level sets framework [31]. The regions are merged to obtain an initial representation for the layers in the video volume. These representations are then refined in another graph cut over the entire video volume. Special considerations to occlusion constraints are

taken to account for inconsistency in layer appearance and improve the final segmentation.

An approximate method is absolutely necessary in such joint spatial/temporal methods to offset the prohibitive cost of computing the graph cut over the extremely large amount of data provided in even very short, relatively low resolution video sequences. Computation still remains relatively expensive. This is a major drawback of joint spatial and temporal segmentation methods: the vast amounts of data which need to be simultaneously analyzed. They also suffer from some of the same constraints as temporal methods which analyze sequences of images. Additionally, some notion of consistency in appearance is often necessary in order to allow for the learning of a single model of sufficient simplicity for objects.

2.4 Spatially Prioritized Methods

Finally, there remains the category of spatiotemporal segmentation methods which prioritize spatial cues within an image sequence and then integrate temporal consistency. This set of methods can probably be considered the richest of the three categories mentioned as many of these methods include ideas from the related and also long studied problem of static image segmentation. Many varied approaches have been taken to integrate the spatial and temporal cues in a manner which prioritizes the cues within a single frame primarily and then seeks for support across frames in the sequence.

Several techniques operate directly upon grouping the motion estimates of objects between images in the sequence [7, 33, 45]. Horn and Schunk [17] describe the well known optical flow constraint (that the intensity of a particular pixel in the image remains constant along its motion trajectory) and how it can be used to generate a 2D vector value for the motion at each pixel in a given frame, resulting in a motion field the same size as an image. Optical flow information can be highly unreliable, especially at object boundaries, and many efforts have been taken to improve the quality of the estimates of image motion

(e.g., [5, 38]).

None the less, optical flow information is often an integral part of video segmentation techniques. For example, Chang et al. [7] present a Bayesian framework that combines motion estimation and segmentation based on a representation of the motion field as the sum of a parametric field and a residual field. A piecewise smooth segmentation field is generated by learning the parametric models for each motion segment in an iterative Bayesian fashion taking into account how well they predict the appearance of successive frames in the sequence. The segmentation of the motion field for each frame is then applied directly to the intensity images of the sequence to achieve an appearance segmentation for each frame.

Another popular approach to video segmentation is often referred to as the extraction of video layers, as introduced in the seminal paper by Wang and Adelson [45], and has seen some continued interest. This representation is based largely upon ideas used in traditional cel animation, in which an image is composed by first creating a background upon which is laid a series of sheets of clear celluloid (or *cels*). Each cel layer has painted on it an image which occludes the part of the background it is laid upon. As the cel moves, it occludes and reveals different parts of the background layer (see Figure 2.2).

Thus, an image sequence can be decomposed into a set of layers, each of which contains three different maps: 1) an intensity (or texture) map describing the appearance of the layer; 2) an alpha map describing the visibility of the layer; and 3) the velocity or warp map, which describes how the other maps change over time.

In Wang and Adelson's original layer representation, appearance maps consist of a pixel-wise description of the gray level or colour intensity information associated with each layer. The alpha maps describe the pixel-wise association of each pixel with each layer. The warp map describes an affine transformation which is applied to the image on a pixel-wise basis to produce the observed appearance of the layer in each frame. The result is a single representation for the appearance of each layer and a set of warps that correspond to each individual

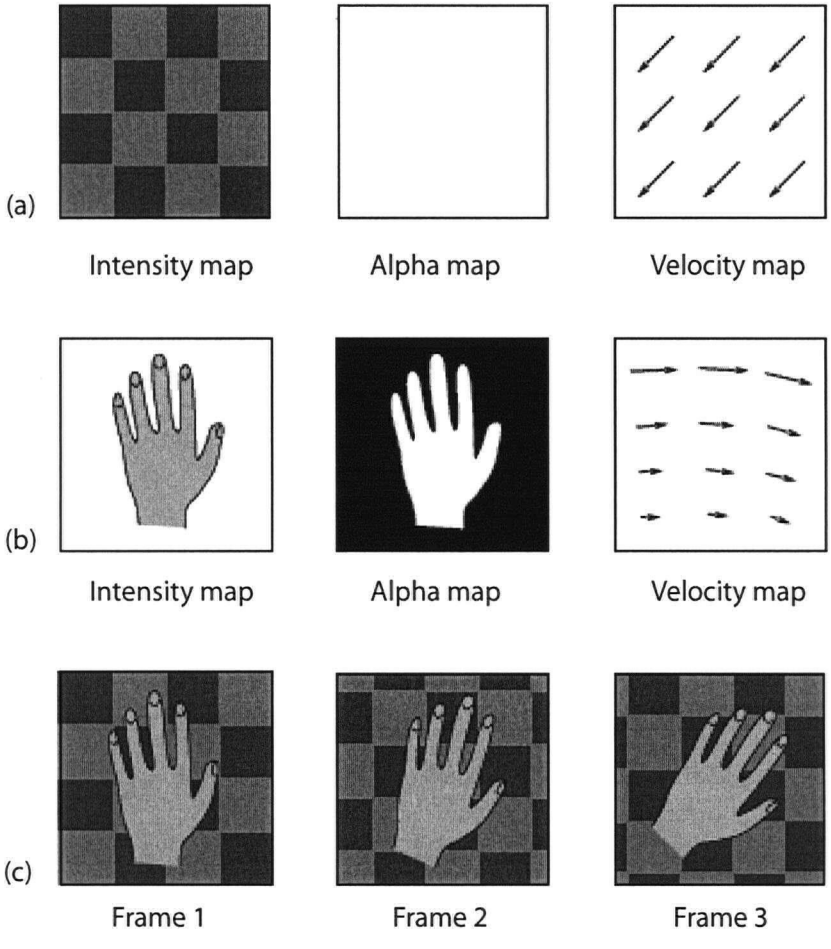

Figure 2.2: From [45]. The desired decomposition of a hand sequence into layers. The hand arcs across the sequence to the right as the background moves consistently down and to the left. (a) The background layer. The intensity map corresponds to the checkerboard pattern; the alpha map is unity everywhere (since the background is assumed to be opaque); the velocity map is constant. (b) The hand layer. The alpha map is unity where the hand is present and zero where the hand is absent; the velocity map is smoothly varying. (c) The resynthesized image sequence based on layers.

frame of the sequence.

Originally in [45], each image in the sequence is segmented by clustering image patches based on local motion estimates and assigning each patch to a consistent affine motion model. The entire sequence is then examined and information from each frame is combined over time so that the stable information within a layer can be accumulated and the warp for each image can be determined. The result is a consistent layered representation including a static depth ordering for each layer.

Such a process imposes some implicit limitations on the nature of the video scene being analyzed. Namely, for such a process to work, the layer appearances must remain consistent throughout the entire scene. Additionally, the number of layers within a scene must be known and cannot change, as no consistent representation could be arrived at with a fixed number of layers. Further, in order to facilitate the layer learning process, there must be a single occlusion ordering for the layers which must be discovered and can not change throughout the sequence.

Several other methods have been developed which build on this approach and address some of these issues. Jojic and Frey formally present the idea of sprites in [19]. A sprite is a vector of pixel intensities the same size as the input image, accompanied by a corresponding vector of mask values. Associated with each sprite, and with each image in a sequence, is a transformation that can be applied to the appearance and mask vectors in order to reproduce the sprite's appearance in a given frame.

So that the appearance of each sprite may vary from image to image, a probabilistic representation termed the "flexible sprite" is introduced. A flexible sprite allows for different instantiations from image to image, all of which are typical under the probability model that defines the flexible sprite. The sprite models are governed by parameters determining the mean and variation in both the appearance and mask vectors, as well as the prior probabilities for each sprite class and the transformations that need be applied to reproduce the sprite in each image. The problem then becomes one of learning the optimal parameters

to describe the sprites given an entire image sequence.

Although this approach extends [45] to allow for some variation in sprite appearance from image to image, and no longer requires the use of optical flow information, the parameter learning problem posed is intractable. Despite the assumption of simple translational/rotational motion throughout the sequence, the discretization of the transformation space yields an intractable number of possible configurations (e.g., 10^{15} for a 320x240 image containing 3 sprites in 3 layers) which need to be examined. Additionally, the form of the posterior distribution over the appearances and masks is not in a tractable closed form. As a result, a generalized EM approach is taken to obtain an approximation at each step and learn the parameters for each sprite model and FFT tricks are employed due to the simple transformational constraints to reduce computational costs.

To somewhat address this issue, Allan et al. [1] present an extension that uses a preprocessing stage to match invariant features across images in the sequence. This step clusters feature matches across an entire sequence to both automatically determine the number of layers in the scene, as well as provide an initial approximation for the transformations which need to be considered. The size of the transformation space is drastically reduced, allowing for a richer, affine transformation model to be considered.

However, the parameter learning performed over the entire image sequence is still intractable, and a variational EM approach is taken to approximate. Additionally, the preprocessing stage requires the objects in the scene to not only remain consistent in appearance throughout the sequence, but remain rigid so that their features can be associated with a single object.

In general, such layer methods based on the original Wang and Adelson proposal still require an entire image sequence ahead of time to learn about the appearances of and segment out the objects of interest within the sequence in an offline process. This is a drawback they share with most temporally prioritized methods and, of course, limits their applicability. Due to the complexity of the representation for each layer, the parameter estimation process is typically intractable, and approximate methods need to be taken to arrive at a solution.

Even still, such a solution is computationally expensive, and the models still require a somewhat strict notion of consistency in object appearance throughout the entire sequence.

Additionally, the layer-based methods described above do not make use of some constraints naturally present in video data. The sequential nature of images captured in a dynamic environment imposes a relationship between adjacent frames in a sequence. This temporal information is not typically exploited by these methods. Furthermore, although these methods operate primarily on spatial motion or appearance of elements within a frame, there is no notion of spatial cohesion of a layer. That is, connectedness between elements, which is another natural tendency of the data, is not included in the notion of the layer within the model. Objects tend to occupy a contiguous region in space, resulting in a contiguous 2D region when imaged. As a result, adjacent points in an image generally belong to the same object. Incorporating such additional spatial constraints, and additionally including temporal constraints, into the process of video segmentation has been accomplished in a variety of ways.

In [37], a system for object discovery is presented to identify independently moving objects within a dynamic scene. Normalized cuts is applied to appearance and depth information obtained from a stereo system to obtain a set of depth and appearance consistent regions. Regions are then grouped based upon features which appear within them between successive frames of the video sequence. A voting system based upon the consistency with which the features and regions are observed determines which regions and features are grouped together to form an object.

[29] presents a region merging method based on graph clustering. Each image in the sequence is subjected to an initial oversegmentation procedure. The images are then examined in a pairwise basis and the spatiotemporal coherence of regions is evaluated. Spatial and temporal similarity metrics are defined and values computed for each pair of regions. These metrics are based upon the significance value of a hypothesis test over temporal (in this case, simply motion characteristics) and spatial (in this case appearance and adjacency) similarity

between regions. Using these measures, a graph is constructed to represent the information and is in turn used in a graph-clustering framework to merge nodes into spatiotemporally consistent objects in the scene. A final post-processing step is performed to remove any remaining small regions by merging them with larger ones.

Such graph-based representation and approaches have been popular in these types of methods. One important step ahead in this direction was to formulate the segmentation problem as a statistical contextual labeling problem, as proposed in the seminal paper by Geman and Geman [14]. The problem then becomes one of determining the appropriate object label for each pixel in an image. Spatial coherence in this case can then be directly enforced by using a probabilistic graphical model termed a Markov random field (MRF) [24].

Similarly to the graph-based approaches mentioned above, a field is constructed where sites correspond to visual elements in the image, and edges are drawn from some notion of spatial adjacency. The MRF then allows for single site comparisons between observed data and the given label on a site by site basis, while at the same time including a pairwise comparison of labels at neighbouring sites within the graph. Favourable comparisons are rewarded with a higher probability, and unfavourable comparisons are penalized. The label configuration of the field yielding the highest probability when considering both single and pairwise site comparisons is then determined and the resulting labeling is taken as the segmentation.

Of such MRF-based methods, the most related to the proposed approach is the work of [33] and [42]. In [33], the initial oversegmentation is accomplished by a watershed segmentation over the gradient image of a frame. An MRF is constructed over the regions produced by the segmentation, where neighbouring sites are determined by adjacency of regions in the segmentation. Spatial constraints are enforced in the typical MRF label smoothness fashion. Temporal constraints are incorporated by projecting the region onto the previous frame labeling and rewarding the current label by counting the pixel wise labels in the previous frame that agree.

Temporal constraints are included through motion compensated intensity differences in the MRF energy functions to distinguish between objects. The motion is modeled as a 6D affine transformation, and the parameters are determined jointly in a framework that optimizes both the label configurations and the motion estimates of each model. Previous as well as subsequent frames have to be considered in order to resolve the ambiguity presented by the motion field in the case of occlusions.

In [42], a watershed segmentation is again used to obtain an initial oversegmentation of the image. This time, a static camera is assumed. This severely limits the applicability of the method, or requires an additional step to register frames to compensate for camera motion (which is known to be error prone and an area of study within itself). Additionally, a drastic simplification in the segmentation process is assumed where the problem becomes one of simply classifying regions as moving (part of the foreground) or stationary (part of the background).

Again, an MRF is employed to enforce spatial consistency and label regions as either object or background. Again, a parametric affine motion model for each region is estimated to facilitate the classification process. Again, special considerations need to be taken to insure that occlusion does not adversely affect the determination of the model parameters. Temporal consistency is enforced in a similar manner by examining the pixels the region is projected onto and how they were classified in previous frames. However, this is accomplished by warping the regions to the positions predicted by the motion estimates.

[23] introduces a supervised form of the MRF model, termed a conditional random field (CRF). This model relaxes some of the strict independence assumptions made between observations by the MRF model, as well as allowing for data-dependent interactions between sites. At the same time, an assumption is made during training of the model about the availability of the labels assigned to hidden variables. The MRF model requires no such assumption as it infers these values during the learning process. Regardless, the result in many cases is an improvement in the accuracy of labeling problems when using CRFs versus

MRFs [22].

The CRF model was first applied to the video segmentation task in [46]. It incorporates both spatial and temporal constraints at a pixel-wise level in an online fashion over an image sequence. Given the number of objects in a sequence and the first frame of the sequence, an initial segmentation is obtained through the motion clustering method previously mentioned in [45]. Parametric appearance models of the objects are constructed based upon the initial labeling and a pixel-wise CRF is used to model each subsequent frame of the sequence.

Sites with similar labels are rewarded to encourage spatial consistency, while sites whose labels are consistent with the previous frame are rewarded to encourage temporal consistency. At the same time, labels are assigned based upon the observations at each site and their similarity to each object model. No hard assignment concerning labels from previous frames is made to enforce the temporal consistency. Instead, a filter framework is adopted to recursively incorporate temporal dependencies through a state transition function defined on pixel-wise temporally neighbouring regions.

To address issues of noise and robustness, and to increase accuracy of the final labeling, very large neighbourhood sizes are adopted. This results directly in an increased cost computationally while at the same time imposing a patch-like structure to the problem. Additionally, a learning of parameterized generative models is performed at each step, including the usual problems associated with model selection and learning.

The proposed work combines ideas from the above field-based methods. In the proposed work, and unlike [42], no assumptions about camera motion are made, making the approach more general. As such, no global corrections need to be applied to the image data beforehand. Additionally, no estimation of affine motion parameters needs to be perrformed as in [33], reducing the cost and complexity of the optimization step. Only the previous frame in the sequence needs to be considered and no other considerations need to be taken as no motion ambiguity needs to be dealt with.

Unlike [33] and [42], spatiotemporal constraints are enforced by a CRF sim-

ilarly to [46] through the energy functions governing the interactions between neighbouring sites and the observed data. However, no parameterized model learning is required as discriminative non-parametric measures are used instead. An initial oversegmentation is used to construct the field which allows for more robustness to pixelwise noise, as well as a more simplified energy space for the label determining inference procedure. The filter is shown to be applicable to a general graph structure produced by the oversegmentation. The resulting method can segment a specified object from a video sequence without the expense of explicitly/implicitly learning models of the rest of the objects in a scene.

Chapter 3

Theoretical Background

3.1 Introduction

This chapter presents the theoretical background required to understand and evaluate the proposed approach. It includes a discussion of each of the major theoretical components involved in the synthesis of the overall system. The well-defined concepts of Markov [4] and conditional random fields [23] are discussed, and how they apply to the segmentation problem. The problem of intractable inference within random field models is also discussed, as well as a standard approach, termed mean field approximation [30], often taken to solve this problem. The well-known hidden Markov model framework is introduced to allow for the incorporation of temporal consistency into the model. Finally, a recursive filter is derived. It combines the above ideas to illustrate how the online update of an unfixed, general random field structure, a generalization to [46], amounts to inference in a general conditional random field at each time step.

3.2 Markov and Conditional Random Fields for Segmentation

Segmentation can be considered as a labeling problem. Each visual element in an image needs to be assigned a label that indicates to which segment and therefore to which real world visual construct the image element belongs. Although many schemes exist for the solution to the labeling problem, one popular approach in terms of image segmentation is to apply a Markov Random Field (MRF) model.

Applying the MRF model involves creating a graphical structure to represent a probability distribution. This probability distribution is over the labels that each site in an image can be assigned in a possible segmentation. The graph is created by designating a set of unordered nodes to represent image sites, each of which corresponds to a random variable which will determine the label for that site. As random variables, each site can take on one of a set of discrete labels, the likelihood of which depends on the data observed at the particular site. At the same time, a neighbourhood system is defined over all of the sites to enforce spatial interactions, representing a dependence between the corresponding variables. The assignment of labels which maximizes the probability of the distribution over the random variables represented in the field is sought as the solution to the segmentation.

More formally, a random field $\mathbf{X}$ is defined by a set of hidden random variables indexed by $S = \{1, 2, ..n\}$, each of which is associated with one of the image sites. A single site in the field can be indexed by $i \in S$ and is denoted $X(i)$. Each random variable can take on one of a discrete set of labels $L = \{1..m\}$. The joint event that $(X(1) = x(1), X(2) = x(2), ..., X(n) = x(n))$ is abbreviated $(\mathbf{X} = \mathbf{x})$, and the set of values $\mathbf{x} = \{x(1), x(2), ..., x(n)\}$ is referred to as a configuration of the field. The probability that a variable $X(i)$ takes on a particular value $x(i)$ is denoted $p(X(i) = x(i))$ or simply $p(x(i))$, while the probability of a particular configuration is denoted $p(\mathbf{X} = \mathbf{x})$ or simply $p(\mathbf{x})$.

With each site is associated a neighbourhood relating it to other sites in S. That is, each site i has associated with it a set N_i of other other sites from S. The neighbour relationship is determined through some notion of adjacency in the field, which can vary depending on the field's overall structure and is represented in the field by edges connecting neighbouring sites (see Figure 3.1). The neighbour relationship is self-exclusive, i.e., a site cannot neighbour itself ($i \notin N_i$) and symmetric, i.e., if site j is a neighbour to site i, then site i is a neighbour to site j (if $j \in N_i$ then $i \in N_j$).

$\mathbf{X}$ is a *Markov random field* if and only if the following two properties hold:

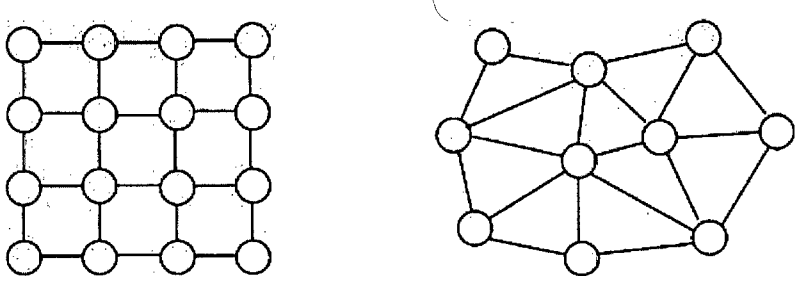

Figure 3.1: Examples of field structures over hidden variables. On the left is a standard 4-neighbour lattice, typically used to represent individual pixels in an image. On the right is a more general field configuration, typically used to represent a region-based image. The circles represent the variable sites and the edges represent neighbour relationships.

$$p(\mathbf{x}) > 0, \forall \mathbf{x} \tag{3.1}$$

and

$$p(x(i)|x(\{S-i\})) = p(x(i)|N_i). \tag{3.2}$$

The first relationship simply states the standard probabilistic assumption that any configuration must yield a probability value greater than zero, and is made for technical reasons [3]. The second relationship states the Markov assumption that the label at any given site depends only on those sites included in its neighbourhood, and is independent of the rest of the sites in the field. Here, $\{S-i\}$ represents the set of all sites S except site i.

According to the Hammersly-Clifford theorem [3], an MRF can be equivalently characterized by a Gibbs distribution, which is of the following form:

$$p(\mathbf{x}) = \frac{1}{Z} exp(-E(\mathbf{x})) \tag{3.3}$$

where

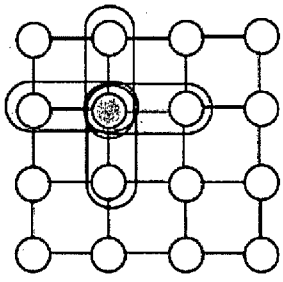

Figure 3.2: A depiction of graph cliques. Each clique associated with the shaded node is depicted as an outline around the clique. A clique is a set of nodes that can be be grouped together without including a node that does not have an edge to each other node in the set. Here, all cliques are pairwise, because no other node can be included with any pair without adding an edge.

$$Z = \sum_{\mathbf{x} \in X} exp(-E(\mathbf{x})) \tag{3.4}$$

and

$$E(\mathbf{x}) = \sum_{i \in S} \left[\phi(x(i)) + \sum_{j \in N_i, j<i} \psi(x(i), x(j)) \right]. \tag{3.5}$$

$E(\mathbf{x})$ is termed the *energy function*, and it is the sum of potentials over all cliques induced by the neighbourhood system and the Markov assumption (see Figure 3.2). In most cases, and is the case in this work, only up to pairwise clique potentials are assumed to be non-zero, yielding a term for single site ($\phi(x(i))$) and pairwise site ($\psi(x(i), x(j))$) interactions.

The single and pairwise site potential functions are used to express the fitness of labels at each site in the field. The value of the functions depends upon the local configuration of the field and the data observed at each site. These functions enforce adherence to a particular classification model and to spatial constraints. Typical single site potential functions reward labels that better match the observed data at the site with a lower value, resulting in a higher

value for configurations with that labeling as determined by Equations 3.3 to 3.5. Similarly, typical pairwise potential functions also reward a labeling that agrees with the labeling of neighbours with a lower value. When the form of the potential functions is independent of the location and orientation of the clique to which they correspond, the field is said to be homogeneous and isotropic. This is typically the case in segmentation problems and is the assumed case in this work.

The choice of potential functions is not restricted to those which will result in a specific probabilistic interpretation as the conditional or marginal distributions, although this can sometimes be the case. One consequence of this generality is that the resulting marginal distribution will not be normalized. This necessitates the inclusion of the normalization constant Z, which is often called the *partition function.* It is simply the sum of all possible configurations of the field, ensuring that the distribution in Equation 3.3 is normalized and sums to one.

To compute the $p(\mathbf{x})$ distribution, the partition function Z must be evaluated. This requires the enumeration and summing of all possible configurations of the field. Since each of the n sites in $\mathbf{x}$ can take one of m different discrete labels, this amounts to summing over m^n different configurations. It is easy to see that this evaluation becomes prohibitive even for models of moderate size. The next section discusses a class of techniques for approximating this constant and examines, in detail, the method used in this system.

The MRF is typically used in a probabilistic generative framework to model the joint probability of the labels at each site and the corresponding observed data. That is, given a set of observations $\mathbf{y}$ corresponding to the sites of $\mathbf{x}$, the posterior over the label field can be expressed as $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, where $p(\mathbf{x})$ is modeled as an MRF. A segmentation is achieved typically by seeking the configuration of the field that yields the maximum probability in the posterior distribution.

In the generative case, energy is expended modeling the joint distribution $p(\mathbf{x}, \mathbf{y})$ which implicitly involves modeling of the observations $\mathbf{y}$. In some cases,

the underlying generative model can be quite complex. This extra modeling effort is wasted in these cases when we are really only concerned with the classification task. The complex nature of any underlying observation models can also make any parameter learning task much more complicated.

Additionally, for computational tractability, the likelihood model $p(\mathbf{y}|\mathbf{x})$ is typically assumed to have strict independence at each site, and thus has a factorized form such as $p(\mathbf{y}|\mathbf{x}) = \prod_{i \in S} p(y(i)|x(i))$. Such strict independence assumptions have been found too restrictive for many image analysis tasks [8, 47].

To avoid such pitfalls, Lafferty et al. [23] introduced the conditional random field model which was extended to 2D by [22]. In contrast to generative approaches using an MRF, the CRF directly models the posterior over labels given the observations at each site. In addition, the CRF model allows for a relaxation of the strong independence of likelihood assumptions made at each site by the MRF model and allows for data interactions to occur in pairwise site potential functions.

In the CRF model, both the label sites $\mathbf{x}$ and the observation sites $\mathbf{y}$ are treated as random fields. When the label field $\mathbf{x}$ is conditioned on the observation field $\mathbf{y}$, the following Markov property holds:

$$p(x(i)|\mathbf{y}, x(\{S - i\})) = p(x(i)|\mathbf{y}, N_i). \tag{3.6}$$

The CRF can be thought of as a random field of labels globally conditioned on the observations. Again using the Hammersly-Clifford theorem, and again assuming up to only pairwise clique potentials, the posterior distribution can now be written as:

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{Z} exp\Big(\sum_{i \in S} \Big[\phi(x(i), \mathbf{y}) + \sum_{j \in N_i} \psi(x(i), x(j), \mathbf{y}) \Big] \Big). \tag{3.7}$$

Note that the single site potential function $\phi(x(i), \mathbf{y})$, and the pairwise site potential function $\psi(x(i), x(j), \mathbf{y})$, are in this case both a function of $\mathbf{y}$. That is, the *energy function* can now depend explicitly on all observations, instead

of allowing only a single site observation, or none at all, to influence the potentials. As in the MRF framework, the CRF framework seeks for the maximal configuration of labels for the posterior distribution to achieve the segmentation. Again, this requires the evaluation of the partition function Z through the use of approximate methods.

In summary, the CRF framework allows for a segmentation to be performed on each frame of a sequence based upon the labeling at each site. The probability of a particular configuration of labels is determined by the interactions between the labels and data in the single and pairwise potential functions indicated in Equation 3.7. Spatial influences between sites can be incorporated through favourable interactions defined by the potential functions resulting in a lower energy for a given labeling. This in turn results in a higher probability in the posterior distribution $p(\mathbf{x}|\mathbf{y})$. The segmentation for a given frame is determined by the configuration of labels which results in the maximal probability in the posterior distribution.

In the case of the approach taken by the proposed method, field sites are associated with regions of the image as opposed to individual pixels, while neighbourhoods are determined by the adjacency of regions within an image. The construction of the field from a given image is detailed later in the implementation section.

3.3 Inference in Random Fields

As mentioned in the previous section, the segmentation problem can be considered as one of labeling sites within an image using a random field model. The problem then becomes one of determining the configuration of labels for the field which results in the highest probability for the distribution represented by the field. In this case, we are interested in determining the configuration which produces the maximal probability in the distribution from Equation 3.7. Due to the nature of the partition function Z (i.e. that it requires a summation over all possible configurations of the field), evaluation of this distribution is

intractable. There are many approaches to address this intractability. This section is devoted to a discussion of variational methods, and in particular, mean field theory for approximating intractable distributions.

3.3.1 The Variational Approach

We have a distribution $p(\mathbf{x})$ which we wish to evaluate but it is intractable to do so. The variational approach seeks to approximate this intractable distribution with a suitable candidate $q(\mathbf{x})$ from a restricted family of tractable distributions. The best candidate approximate distribution from this family can be determined by using a common measure for similarity between two distributions; the Kullback-Liebler (KL) divergence [21]. The KL divergence is always nonnegative and equal to zero only when the two distributions being compared are identical.

As mentioned, in the case of field models where only up to pairwise potentials are considered, $p(\mathbf{x})$ is of a standard Gibbs form $p(\mathbf{x}) = exp\{-E(\mathbf{x})\}/Z$. In this case, the KL divergence yields:

$$\begin{aligned} KL(q||p) &= \sum_{\mathbf{x}} q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} \qquad (3.8) \\ &= \sum_{\mathbf{x}} q(\mathbf{x})[\ln q(\mathbf{x}) + \ln Z + E(\mathbf{x})] \\ &= \sum_{\mathbf{x}} q(\mathbf{x}) \ln q(\mathbf{x}) + \sum_{\mathbf{x}} q(\mathbf{x}) \ln Z + \sum_{\mathbf{x}} q(\mathbf{x}) E(\mathbf{x}) \\ &= \ln Z + \sum_{\mathbf{x}} q(\mathbf{x}) \ln q(\mathbf{x}) + \sum_{\mathbf{x}} q(\mathbf{x}) E(\mathbf{x}). \qquad (3.9) \end{aligned}$$

Some of the terms in Equation 3.9 may look familiar to those from a statistical physics background. The negative of the log of the partition function, $-\ln Z$, is often referred to as the *Helmholtz free energy*, while the remaining terms will be referred to as the *Gibbs free energy*. They consist of a *variational energy* term $\sum_{\mathbf{x}} q(\mathbf{x})E(\mathbf{x})$, and an *entropy* term $\sum_{\mathbf{x}} q(\mathbf{x}) \ln q(\mathbf{x})$.

The variational approach seeks to minimize these energy terms. Since the KL divergence measure is always nonnegative and equal to zero only when the distributions $p(\mathbf{x})$ and $q(\mathbf{x})$ are identical, it will reach its minimal value of 0

when the Gibbs free energy reaches its minimal value of $-\ln Z$ and the distributions are identical. Allowing the approximate distribution to vary (hence the name variational for this type of approach) while minimizing the KL divergence between the two distributions will result in the best candidate distribution from the restricted family.

3.3.2 Mean Field Theory for CRF Inference

In the case of a pairwise CRF model, the intractable distribution is over hidden variables given a set of corresponding observed variables as given by Equation 3.7. The variational mean field method [30] seeks to approximate $p(\mathbf{x}|\mathbf{y})$ with a candidate from a family of fully factorized distributions of the form

$$q(\mathbf{x}) = \prod_i b_i(x(i)). \tag{3.10}$$

Here, b_i are referred to as the variational mean field parameters or beliefs which correspond to marginal probabilities which sum to one over each node $x(i)$. The parameters are obtained by minimizing the KL divergence between $p(\mathbf{x}|\mathbf{y})$ and $q(\mathbf{x})$. Since the minimal KL divergence is obtained when the Gibbs free energy is minimized, minimizing KL divergence is equivalent to minimizing the Gibbs free energy. In the case of the fully factorized family of distributions used for mean field approximation, the Gibbs free energy can be written as:

$$\begin{aligned} G_{MF} &= \sum_{\mathbf{x}} q(\mathbf{x}) \ln q(\mathbf{x}) + \sum_{\mathbf{x}} q(\mathbf{x}) E(\mathbf{x}) \\ &= \sum_{\mathbf{x}} \left(\prod_i b_i(x(i)) \ln \prod_i b_i(x(i)) \right) \\ &\quad + \sum_{\mathbf{x}} \prod_i b_i(x(i)) \left(- \sum_{<ij>} \ln \psi(x(i), x(j)) - \sum_i \ln \phi(x(i)) \right) \\ &= - \sum_{<ij>} \sum_{<x(i),x(j)>} b_i(x(i)) b_j(x(j)) \psi(x(i), x(j)) \\ &\quad + \sum_i \sum_{x(i)} b_i(x(i)) [\ln b_i(x(i)) - \ln \phi(x(i))]. \end{aligned} \tag{3.11}$$

where $< ij >$ and $< x(i), x(j) >$ refer to all possible combinations of i, j and $x(i), x(j)$ respectively.

Minimizing this term concurrently for all marginals is complicated, but it can be minimized for a given marginal b_k by taking the partial derivative with respect to that marginal. Using a Lagrange multiplier to enforce the constraint that $\sum_{x(k)} b_k(x(k)) = 1$ and setting the derivative of the Gibbs free energy to zero results in:

$$\begin{aligned} 0 &= \frac{\partial}{\partial b_k(x(k))}\Big(-\sum_{ij}\sum_{x(i),x(j)} b_i(x(i))b_j(x(j))\psi(x(i),x(j)) \\ &\quad + \sum_i \sum_{x(i)} b_i(x(i))[\ln b_i(x(i)) - \ln \phi(x(i)) - \lambda(\sum_{x(k)} b_k(x(k)) - 1)\Big) \\ 0 &= -\sum_{kj}\sum_{x(j)} b_j(x(j))\psi(x(k),x(j)) + \sum_{x(k)}[\ln b_k(x(k)) - \ln \phi(x(k))] - \lambda \\ b_k(x(k)) &= \alpha\phi(x(k))exp\{\sum_{j\in N_k}\sum_{x(j)} b_j(x(j)) \ln \psi(x(k),x(j))\}. \end{aligned} \tag{3.12}$$

The term in Equation 3.12 allows us to minimize the free energy with respect to a particular marginal. This requires simply computing its single site potential and its pairwise potentials with its neighbours, while renormalizing with some constant α. Of course, since the resulting value depends upon that of the neighbouring nodes whose marginals are also to be varied to minimize the free energy, an iterative scheme must be used. All nodes are initialized to a preset value and nodes are updated using Equation 3.12 until convergence of the Gibbs free energy.

3.4 Hidden Markov Models

To incorporate temporal consistency into the segmentation framework, the notion of the standard hidden Markov model (HMM) is introduced [4]. An HMM is a graphical model that is appropriate for modeling sequence data because

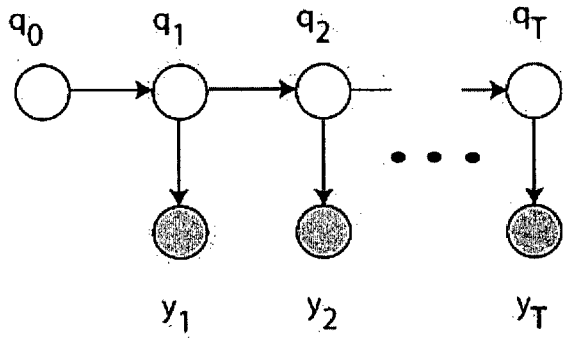

Figure 3.3: The graphical depiction of a hidden Markov model. Each state q_t is dependent upon the previous state q_{t-1} with the exception of the initial state q_0. The observation y_t is assumed to have been generated based upon the current state of the system and is therefore dependent upon the current state q_t.

it incorporates temporal dependencies between different states in a sequence. It consists of a finite set of states, each of which is associated with a separate probability distribution. Transitions between states are governed by a set of transition probabilities. With each state is associated an observation which is assumed to have been influenced by the probability distribution for that state. The state is not observable (that is it remains hidden, hence the name) and it can only be inferred from the observed outcomes. This makes the inference problem for an HMM one of taking a sequence of observed events and producing a probability distribution for each underlying state as output.

More formally, an HMM is defined by the following set of criteria. It consists of some number of states n, and a set of state transition probabilities $p(q_i|q_j)$ from states j to i, where $1 \leq i, j \leq n$. At any given time t, the state of the system is denoted by the random variable q_t and the observation of the system at this time is denoted by the observed output variable y_t. These relationships are typically represented graphically as illustrated in Figure 3.3.

The first state in the sequence has no parent and is represented with an unconditional distribution $p(q_0)$, while every other state in the sequence depends

on previous states. This is the main property of interest in the HMM, the conditional nature of a state at time t upon the states immediately proceeding it in the sequence. This dependence is defined by the order of the model. For example, a first order HMM conditions only upon the immediately preceding state at time $t-1$, while a second order HMM conditions upon the two previous states at times $t-1$ and $t-2$.

As is often the case, and as is assumed in this work, the HMM is of first order and thus the state q_t only depends on its sole parent q_{t-1}. Similarly, each observation variable y_t depends only upon its sole parent q_t. As a result, conditioning on q_t renders all states $q_u, u \leq t$ and $q_v, v \geq t+1$ independent of each other.

The joint probability of a sequence of observations and the states that generated them, $(\mathbf{q}, \mathbf{y}) = (q_0, q_1, ..., q_T, y_0, y_1, ..., y_T)$ is expressed as

$$p(\mathbf{q}, \mathbf{y}) = p(q_0) \prod_{t=1}^{T} p(q_t|q_{t-1}) \prod_{t=1}^{T} p(y_t|q_t). \tag{3.13}$$

There are several types of existing problems to which HMMs are applied, but of particular interest in this case is the *filtering problem.* Consider a sequence of observations $\mathbf{y}$ that arrives in an online fashion, where it is desired to compute the probability of the state at time t, without waiting for future data. That is, at time t it is desired to compute the probability $p(q_t|y_0, y_1, ..., y_t) = p(q_t|y_{1:t})$. Making use of the conditional independencies in the model and applying Bayes rule while conditioning upon the observations up to the previous time $t-1$, this can be written as

$$\begin{aligned} p(q_t|y_{1:t}) &= p(q_t|y_{1:t-1}, y_t) \\ &= \frac{p(y_t|q_t)p(q_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}, \end{aligned} \tag{3.14}$$

where $p(q_t|y_{1:t-1})$ is often referred to as the *prediction*, and can be further expressed as

$$p(q_t|y_{1:t-1}) = \sum_{q_{t-1}} p(q_t|q_{t-1})p(q_{t-1}|y_{1:t-1}). \tag{3.15}$$

Equations 3.14 and 3.15 illustrate how the posterior at time t can be computed recursively using the posterior input from the previous step and the transition probabilities in conjunction with the likelihood at the current step. This approach can then be used in conjunction with the CRF model to incorporate temporal dependencies between successive segmentations in a sequence.

3.5 The General Spatiotemporal CRF

As mentioned previously, the problem of object segmentation in a video involves obtaining a segmentation for each image in a sequence. Given such a sequence of images, the goal is to determine the best assignment of labels to each site in a random field representing each image in the sequence by computing the field posterior at time T. This section illustrates how the HMM and CRF ideas can be combined similarly to [46] into such a structure which incorporates both spatial and temporal constraints. Here, no assumptions are made about field structure, and the structure of the field is allowed to vary from step t to $t+1$.

$$p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) = \frac{p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T})}{p(\mathbf{y}_{1:T})} \tag{3.16}$$

Here, $\mathbf{x}_t$ is a field over variables from a set S_t, consisting of some number of sites representing regions in an image t of the sequence, and connected on the basis of adjacency within the image as described further in the implementation section. Similarly, $\mathbf{y}_t$ are the observations corresponding to these regions. Specific locations in the field are indexed by i, such that $x_t(i)$ and $y_t(i)$ refer to the label and observation at site i in fields $\mathbf{x}_t$ and $\mathbf{y}_t$ respectively, for the fields at time t. The joint distribution over segmentation fields and observation fields for the sequence can be expressed in the standard HMM framework described above.

$$
\begin{aligned}
p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) &= p(\mathbf{x}_0) \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{y}_t|\mathbf{x}_t) \\
&= Z_0^x exp(\sum_{i \in S_0} [\phi_i^{x_0} + \sum_{j \in N_i} \psi_{ij}^{x_0}]) \cdot \\
&\quad \prod_{t=1}^{T} \{p(\mathbf{x}_t|\mathbf{x}_{t-1}) \cdot Z_t^y exp(\sum_{i \in S_t} [\phi_i^{y_t} \sum_{j \in N_i} \psi_{ij}^{y_t}])\} \qquad (3.17)
\end{aligned}
$$

Here, ϕ and ψ refer to the single and pairwise site potential functions, respectively. In particular, $\phi_i^{x_t}$ and $\phi_i^{y_t}$ refer to the single site potential for the fields representing the transition probability and observation likelihood functions at location i for field t. $\psi_{ij}^{x_t}$ and $\psi_{ij}^{y_t}$ refer to the pairwise site potentials in the transition and observation functions for sites i and j, where j is drawn from the set of neighbours N_i of site i, again for field t (see Figure 3.4).

The distributions over the initial chain variable and the observation likelihoods have been replaced by Gibbs distributions with single and pairwise potentials as shown along with the appropriate normalization constants. In this case of fields being constructed from regions produced by an initial oversegmenation, the structure and dimension of the fields in each frame can change. Due to this fact, the transition function $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ must consider the possible dimension change from the field at step $t-1$ to the field at step t. However, the transitional distributions in this case will also be represented simply as Gibbs distributions with appropriate potentials.

$$
\begin{aligned}
p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) &= Z_0^x exp(\sum_{i \in S_0} [\phi_i^{x_0} + \sum_{j \in N_i} \psi_{ij}^{x_0}]) \cdot \qquad (3.18) \\
&\quad \prod_{t=1}^{T} \{Z_t^x exp(\sum_{i \in S_t} [\phi_i^{x_t} + \sum_{j \in N_i} \psi_{ij}^{x_t}]) \cdot Z_t^y exp(\sum_{i \in S_t} [\phi_i^{y_t} + \sum_{j \in N_i} \psi_{ij}^{y_t}])\}
\end{aligned}
$$

The single site transition potential $\phi_i^{x_t}$ makes use of temporal neighbours in order to enforce temporal coherence between segmentations throughout the sequence. Temporal neighbours are sites from the previous segmentation field $\mathbf{x}_{t-1}$ that have been designated as neighbours to site i in field $\mathbf{x}_t$, and are denoted

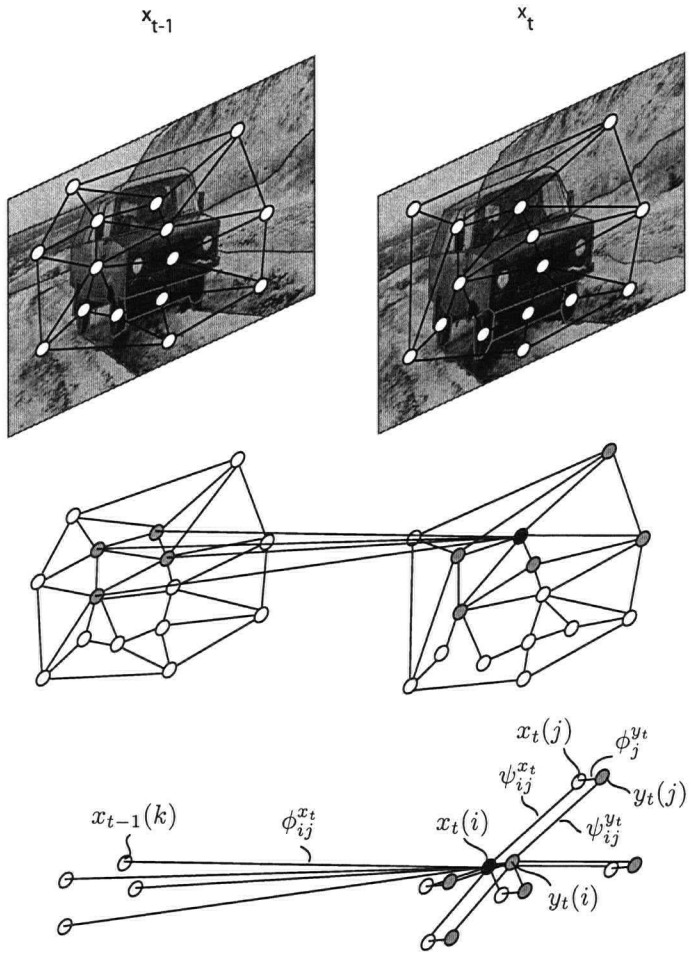

Figure 3.4: First row: An illustration of two consecutive frames in a sequence with their region-based representations and resulting segmentation fields. Second row: a depiction of a possible set of spatial and temporal neighbours for a node of interest in field $\mathbf{x}_t$. The shaded nodes in frame $\mathbf{x}_t$ are the spatial neighbours, N_i, of the black node i. The shaded nodes in field $\mathbf{x}_{t-1}$ are the temporal neighbours, T_i. Third row: A closer look at a node i and its associated potentials. Omitted for clarity from the previous illustrations are the observation field nodes, which are here represented by shading and connected to their applicable label nodes. The potential functions to which each edge corresponds are also shown.

by the set T_i (see Figure 3.4). The temporal neighbour relationship represents the dependency that the state at time t, in this case the field $\mathbf{x}_t$, has on the previous field at time $t-1$. The single site transition potential incorporates this information through a summation over contributions from the site's temporal neighbours, determined by a temporal neighbouring potential function $\phi_{ij}^{x_t}$.

$$\phi_i^{x_t} = \sum_{j \in T_i} \phi_{ij}^{x_t} \tag{3.19}$$

Each $\phi_{ij}^{x_t}$ is a function of the current site i and one of its temporal neighbours j. The precise formulation is discussed in the implementation section.

3.6 The Filter

The filtering algorithm is used to recursively update the posterior distribution of the segmentation field. A first-order Markov assumption is made on segmentation fields. At time t, observation field $\mathbf{y}_t$ is used to update the posterior probability distribution of the segmentation field $\mathbf{x}_t$.

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \tag{3.20}$$

The prediction step involves determining the posterior distribution of the segmentation field at time t given the observation fields up to the previous step, $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$. This can be written as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}). \tag{3.21}$$

This computation involves summing over the possible configurations of the field $\mathbf{x_{t-1}}$. This is, of course, intractable. However, $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ can be approximated using a factorization according to mean field theory, making the computation tractable. Similarly to [46], substituting the transition function from Equation 3.18 and the mean field approximation for the posterior from

the previous step yields

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \propto exp[-\sum_{i\in S_t}\sum_{j\in N_i}\psi_{ij}^{x_t}]\cdot \sum_{\mathbf{x}_{t-1}}\{exp[-\sum_{i\in S_t}\sum_{j\in T_i}\phi_{ij}^{x_t}]\cdot\prod_{i\in S_{t-1}}\widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1})\}. \quad (3.22)$$

The first term is the spatial smoothness of labels from the transition probability function, which does not depend explicitly on the previous frame. The second term, which sums over all possible configurations of field $\mathbf{x}_{t-1}$, contains the mean field approximation, $p(\mathbf{x}_{t-1}(i)|\mathbf{y}_{1:t-1}) \approx \prod_{i\in S_{t-1}}\widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1})$, of the segmentation field from the previous step. The terms can be rearranged so that the exponential sum is in terms of the sites in the field for step $t-1$ and their temporal neighbours in the field from step t. This neighbouring relationship will be referred to as *reverse temporal neighbours* and such neighbours to a node $x_{t-1}(i)$ will be denoted T_i^{-1}. The second term can be manipulated to produce:

$$\begin{aligned}
&\sum_{\mathbf{x}_{t-1}}\{exp[-\sum_{i\in S_t}\sum_{j\in T_i}\phi_{ij}^{x_t}]\cdot\prod_{i\in S_{t-1}}\widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1})\}\\
=&\sum_{\mathbf{x}_{t-1}}\{exp[\phi_{11}^{x_t}(x_t(1),x_{t-1}(1))]\cdot exp[\phi_{12}^{x_t}(x_t(1),x_{t-1}(2))]\cdots\\
&\prod_{i\in S_{t-1}}\widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1})\}\\
=&\sum_{\mathbf{x}_{t-1}}\{exp[-\sum_{i\in S_{t-1}}\sum_{j\in T_i^{-1}}\phi_{ij}^{x_t}]\cdot\prod_{i\in S_{t-1}}\widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1})\}\\
=&\sum_{\mathbf{x}_{t-1}}\prod_{i\in S_{t-1}}exp[\sum_{j\in T_i^{-1}}\phi_{ij}^{x_t}]\widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1}). \quad (3.23)
\end{aligned}$$

Due to this arrangement of terms and the full factorization of the approximate distribution provided by mean field, the intractable sum of products term can be converted to a tractable product of sums.

$$\sum_{\mathbf{x}_{t-1}}\prod_{i\in S_{t-1}}exp[\sum_{j\in T_i^{-1}}\phi_{ij}^{x_t}]\widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1})$$

$$= \prod_{i \in S_t} \{ \sum_{x_{t-1}(i)} exp[- \sum_{j \in T_i^{-1}} \phi_{ji}^{x_t}] \cdot \widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1}) \} \tag{3.24}$$

The resulting term is nearly a Gibbs distribution. However, since the exponential function is convex, we can apply Jensen's inequality to pull it out of the sum and therefore approximate the term using its lower bound.

$$\begin{aligned} & \prod_{i \in S_t} \{ \sum_{x_{t-1}(i)} exp[- \sum_{j \in T_i^{-1}} \phi_{ji}^{x_t}] \cdot \widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1}) \} \\ \geq & \prod_{i \in S_t} \{ exp[- \sum_{x_{t-1}(i)} \widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1}) \sum_{j \in T_i^{-1}} \phi_{ji}^{x_t}] \} \end{aligned} \tag{3.25}$$

The terms can again be rearranged so that the sums are now with respect to single sites in the field from step t, as before, in the standard Gibbs form. As a result, the expression for the filter prediction step becomes the following Gibbs distribution:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \quad \propto \quad & exp[- \sum_{i \in S_t} \sum_{j \in N_i} \psi_{ij}^{x_t}] \cdot \\ & \prod_{i \in S_t} \{ \sum_{j \in T_i} \sum_{x_{t-1}(j)} \widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1}) \phi_{ij}^{x_t} \}. \end{aligned} \tag{3.26}$$

This result can then be used in an update step that requires the application of this distribution to the likelihood for the current frame. Recall that the likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ consists of simply a Gibbs distribution with the appropriate single and pairwise site potentials.

$$p(\mathbf{y}_t|\mathbf{x}_t) = exp(\sum_{i \in S_t} [\phi_i^{y_t} \sum_{j \in N_i} \psi_{ij}^{y_t}]) \tag{3.27}$$

Collecting all terms from Equations 3.27 and 3.15 into one equation for the posterior yields

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \quad = \quad Z_t \cdot p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t})$$

$$\propto \; exp\{-\sum_{i\in S_t}[\sum_{j\in N_i}\psi_{ij}^{x_t} + \sum_{j\in T_i}\sum_{x_{t-1}(j)}\widetilde{p}(x_{t-1}(i)|\mathbf{y}_{1:t-1})\phi_{ij}^{x_t} + \sum_{j\in N_i}\psi_{ij}^{y_t} + \phi_i^{y_t}]\}. \tag{3.28}$$

As can be seen from Equation 3.28, the combination of Gibbs distributions in the HMM framework has resulted in simply another field equation. It incorporates both the temporal and spatial labeling constraints (the ϕ_t and ψ_t terms), as well as the single and pairwise likelihoods (the ϕ_y and ψ_y terms). With the inclusion of data interactions in the pairwise potentials, the field equation thus represents a CRF over the current frame $\mathbf{x}_t$.

Computation of the distribution is again intractable. However, a mean field approximation can again be applied to the field equation resulting from the update step, where each site is iteratively updated using the mean field update as illustrated above. The resulting approximation can be used to determine which labels should be applied to each node. Namely, due to the factorization by the mean field approximation, the label with the maximal posterior probability at each site is used to determine its label at time t. The approximated distribution is then stored and can be used as the input to the next filter step.

Chapter 4

Implementation

This section discusses the implementational details of the proposed method. This includes a discussion of the optical flow calculation used to obtain motion information, the oversegmentation approach and the construction of the resulting field from an image. Also discussed are the observable quantities extracted from the image and the precise formulation of the energy functions in which this observed information is applied.

4.1 Object Motion and Optical Flow

The notions of object motion and optical flow are intimately tied within the image analysis framework. During the observation of a dynamic scene, motion of an object can be observed due to either the movement of the object itself, or the movement of the camera in relation to the object. The location of the object in the resulting sequence of images changes depending on the nature of the motion the object undergoes. The manifestation of this motion in the image sequence is typically referred to as optical flow. It is this optical flow information that is taken advantage of in this approach.

Optical flow is a vector field which can be loosely defined as the apparent motion of brightness patterns in a sequence of images (see Figure 4.1). It is an approximation of the motion field which represents the projection of the true motion in a scene onto the 2D viewing plane. It can be computed from time-varying image sequences under the simplifying assumptions that the surfaces are generally Lambertian, the light sources are pointwise at infinity and that no photometric distortions in the scene are present. The error of this approximation

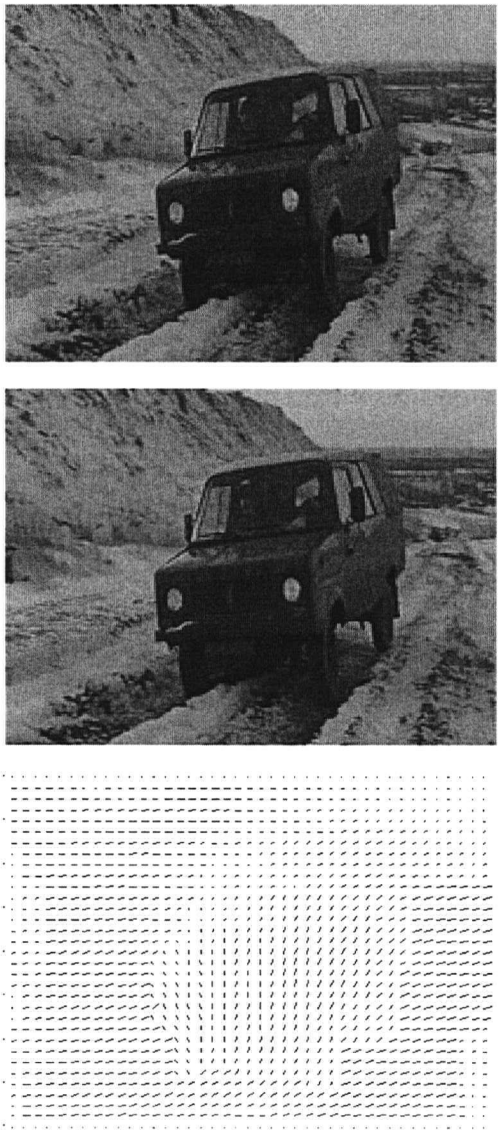

Figure 4.1: An illustration of optical flow. The top two images are from a sequence where the camera pans to the left to follow a jeep driving down a bumpy snow-covered road. The slight difference in location of the the jeep in each image is evident as due to it's motion between frames. On the bottom is a vector field illustrating the optical flow estimations for individual pixel locations in the image. The dominant motions in the image are apparent. As the camera pans to the left, the background moves relatively to the right. This is evidenced by the horizontal vectors in background areas. While the jeep moves towards the camera, bouncing up and down, the optical flow vectors capture this information as well. In regions of little or no texture, such as the sky, the flow information is either spurious or nonexistent.

is small at points with high spatial gradients, and exactly zero only when the motion of an object is such that the illumination direction is parallel to the angular velocity [18].

Optical flow is typically calculated by examining the motion of the brightness patterns in the image. These brightness changes are usually induced by the motion of the scene, when the assumptions hold true, although this is not necessarily the case. Certain circumstances can result in an optical flow which is not at all indicative of the motion in the underlying scene. Such examples can include a light being turned on or off during an image sequence, the rotation of an untextured sphere, or the classic example of a rotating barber pole. In any such case, the brightness change present in the image is not actually informative about the true motion in the scene. In general, however, such cases are unusual and in this work the optical flow is generally assumed to be indicative of the motion of the objects in the scene relative to the camera.

Computation of optical flow also relies on the assumption that apparent brightness of moving objects remains constant, in conjunction with the assumption that there is also some constancy of the apparent brightness of the observed scene. This can be written as the stationarity of the image brightness E over time: $\frac{dE}{dt} = 0$. Here, the image brightness E is a function of both the spatial coordinates of the image plane, and of time. In turn, the spatial coordinates are both functions of time, and the total temporal derivative can be written using the chain rule. This results in what is known as the *brightness constancy equation*[18].

$$\frac{dE(x(t), y(t), t)}{dt} = \frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = 0 \tag{4.1}$$

Virtually all approaches to estimation of optical flow rely on this relationship [41]. Due to the fact that the underlying assumptions of this relationship can sometimes be violated, the resulting estimation of optical flow can be in error. In addition to this, since many methods rely on corresponding image patches across frames to determine the flow vectors at a given location, optical flow

estimation can often suffer from noisy results when the matches are in error. This often occurs around object boundaries in an image when occluded and unoccluded areas may not have a true match, or in areas without sufficient texture to determine a reliable match.

Many approaches have been taken to address these issues to varying degrees of success. Any number of methods for computation of optical flow would be suitable for use in this approach. The method presented in [5], often termed *Robust Optical Flow*, was chosen for its relatively favourable accuracy and availability, although it too can often be susceptible to the above problems.

4.2 Oversegmentation

An oversegmentation seeks to exploit redundancy in image information in order to reduce the complexity of the labeling task. Oversegmentation of a scene before labeling is becoming a popular approach to visual segmentation problems and many methods exist for both image segmentation and oversegmentation with varying degrees of success [12, 36, 37], but the two should not be confused.

A segmentation of an image typically seeks to identify boundaries between the major visual components of the scene. Each image element is labeled as either belonging to a particular object or not. An oversegmentation, on the other hand, typically seeks to simply identify contiguous regions of relatively constant appearance. These regions may or may not belong to the same object in an image but the intent is to simply identify regions such that any given region does not likely belong to more than one object in the image. That is to say, a region's boundaries should not violate an object's boundaries within the image, although there may be many more region boundaries than object boundaries (see Figure 4.2). The result is a set of regions over an image, each of which needs to be assigned a label instead of a pixelwise assignment of labels.

This is advantageous for a number of reasons. As mentioned, the labeling task in the CRF framework, as presented above, is much less expensive since there is a large reduction in the number of nodes in the model. This is desirable

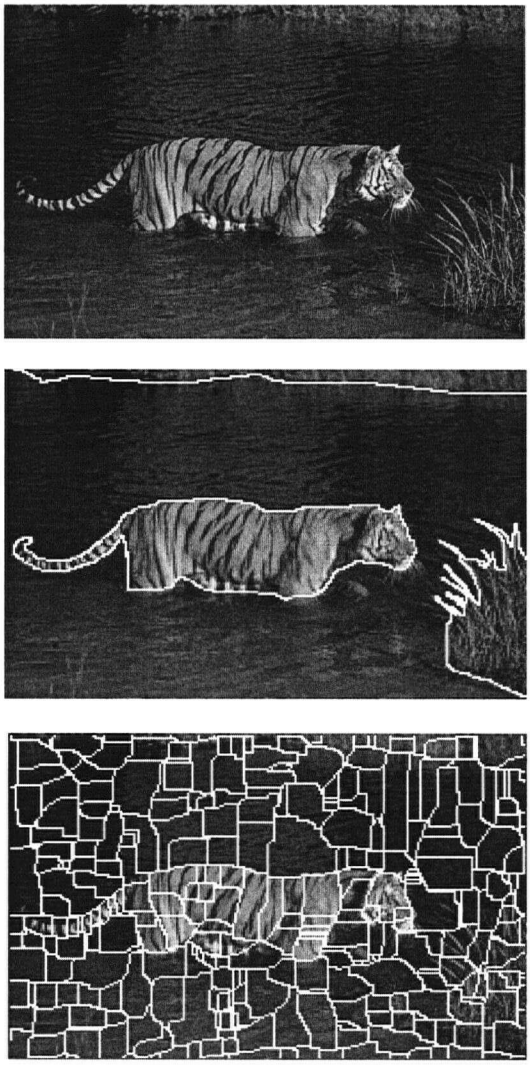

Figure 4.2: An illustration of oversegmentation on an image. On the top, the original image. In the middle, a hand segmented version of the image into its main visual components. On the bottom, an oversegmentation of the image. For the most part, the contours from the segmented image are preserved in the oversegmentation, although many more contours are present in the oversegmented image.

since methods for inference in CRFs often require an iterative updating of the nodes in a model until convergence is reached, and is certainly the case in mean field inference. Furthermore, a reduction of nodes in the model is often accompanied by a reduction in complexity of the error/energy space, hopefully resulting in a faster convergence to a non-local minimum.

The segmentation in the proposed approach is accomplished by a modification to the watershed algorithm as presented by Vincent and Soille [44]. The general idea of watershed segmentation is to identify regions of relatively constant appearance. This is accomplished by either operating directly upon image intensity values or upon some transformed version of these values. The values are treated as heights in a landscape over the extent of the 2D image. The topography of this landscape is then used to determine the appropriate location for boundaries which separate regions of consistent appearance.

The image to be segmented is first subjected to a Gaussian smoothing to reduce the effects of image noise. Next, it is convolved with the derivative of a Gaussian to produce a gradient intensity image. The result can be thought of as a three dimensional gradient intensity landscape, where large changes in gradients which likely correspond to boundaries in the image are represented by high intensity values and therefore peaks in the landscape. Areas of relatively constant intensity in the original image, on the other hand, correspond to small changes in gradient and result in valleys in the gradient intensity landscape (see Figure 4.3).

The peaks separating the valleys in the gradient intensity landscape are referred to as watershed lines, while the valleys themselves are referred to as catchment basins. The terms come from the field of geographic topography where they are typically used to describe the path that water will take when falling on a three dimensional landscape. These ideas are central to the operation of the watershed segmentation algorithm. Each catchment basin has associated with it a single minimal value, or minimal altitude on the landscape. All points on the landscape (which correspond to pixels in the gradient intensity image) associated with a catchment basin have a strictly non-decreasing path to the

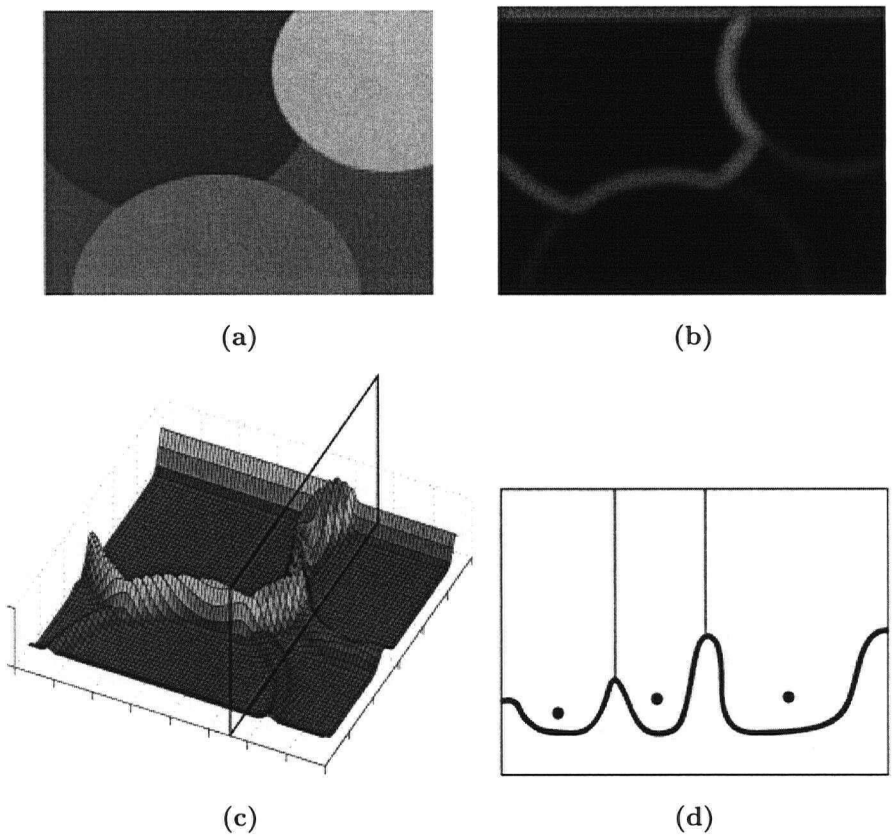

Figure 4.3: An illustration of the watershed process. Image (a) shows the original image and (b) a gradient intensity image resulting from a Gaussian smoothing and then a convolution with a derivative of a Gaussian. Image (c) depicts the gradient intensity image as a three dimensional landscape. Image (d) shows a two dimensional slice of the landscape for clarity. It illustrates the gradient intensity curve along this slice, as well as the watershed lines and the catchment basins that they separate the landscape into.

basin minimum.

Watershed algorithms typically deal with determining which minima and therefor which basin each pixel in an image belongs to by determining its downstream path. In the Vincent and Soille version [44], this is accomplished by "filling" the catchment basins from the bottom. Imagine that there is a hole in the bottom of each catchment basin, and that the gradient intensity landscape was plunged into water. The basins would begin to fill from the bottom, and regions of water would start to grow in the two dimensional plane as observed from above. Each water region would correspond to the connected pixels of an area of low gradient intensities, which in turn correspond to regions of relatively constant appearance in the original image.

As the basins continued to fill, they would begin to merge as the water level overtook the peaks separating them. In the case of the watershed algorithm, a "dam" is built where the regions would intersect at a watershed line. Once the entire landscape is under water, the flooding stops. The resulting watershed lines then represent the oversegmentation boundaries within an image.

The operation of the actual algorithm is very similar to this intuitive explanation. Pixels in the gradient intensity image are sorted in terms of their intensity values and an intensity histogram is created. At each flooding step, the flood value, f, is increased in intensity and the bin of the histogram corresponding to the new flood value is examined. Any pixels of intensity lower than f have already been assigned to a basin, while pixels of intensity equal to f need to be assigned to a basin.

Influence zones are computed for each basin determined thus far. The influence zone for a basin is defined by the set of non-labeled pixels of current flood intensity f that are contiguous with the basin and closer to it than to any other basin (see Figure 4.4). Each pixel of value f is examined and if a neighbouring pixel already carries a basin label, the pixel is assigned the same label. Remaining pixels are assigned labels based upon the basin influence zones. If a pixel cannot be assigned to a bin by the above procedure, then it must be the member of a new basin beginning at the altitude f. A new basin label is created

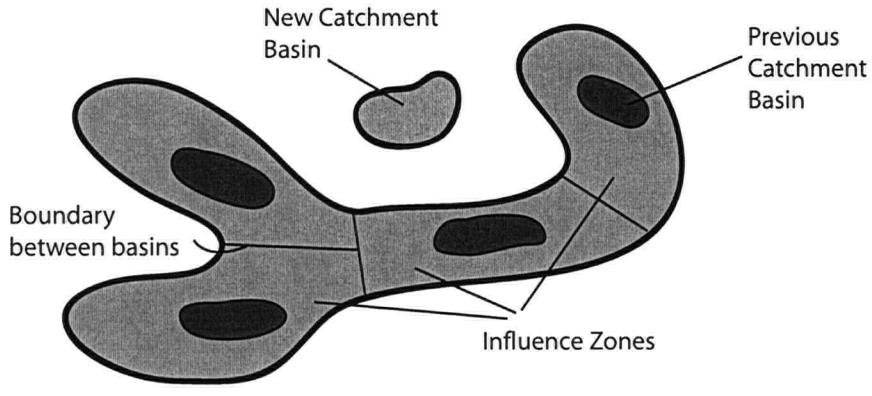

Figure 4.4: An illustration of assignments to catchment basins. The darker gray regions represent basins determined from previous iterations of intensity values lower than the current flood intensity f. Lighter gray regions represent the influence zones associated with each catchment basin, where the boundaries between zones are delineated with a line. A new catchment basin is also depicted where the assignment rules fail to associate the region with any previous basin. The unshaded regions within the bounding box represents all the pixels of higher intensity than f that have yet to be considered.

and the appropriate pixels are assigned. When all pixels have been assigned, the current flood intensity f is incremented and the process repeated.

Watershed segmentation, while being relatively fast and easy to compute, typically relies entirely upon information from a gradient intensity image taken of a grayscale image. This can result in incorrect boundaries between regions where the gradient may not be particularly strong. To increase the chance that watershed boundaries more often correspond to real image boundaries, colour information can also be used in the watershed process. An image in the HSV colour space can be used and the gradient of each colour channel can be taken, treating each channel as a grayscale image. The gradient values can be compared across planes and the maximal value for each location can be adopted for a final

gradient intensity image which is then fed to the watershed algorithm (similar to [42]).

Additionally, the optical flow images can also be used to further improve the correspondence between oversegmentation boundaries and image boundaries. The gradient of the magnitude of the vectors from the xy optical flow planes can be computed and again values can be compared. In the proposed method, both the colour channel gradients and optical flow gradient images can all be computed and the maximal intensity value across these images can be taken and stored in a final gradient intensity image. The gradient intensity image is then given to the watershed algorithm. An example of a watershed segmentation from a frame in one of the test sequences is provided in Figure 4.5.

4.3 Construction of the Region-based CRF Model

Once the optical flow has been calculated and the oversegmentation of an image from the sequence has been performed, a general planar CRF structure must be constructed from the image. This involves creating the graph structure based upon the oversegmentation, as well as extracting the observable information to be associated with the observation field. This process is performed as follows.

Given an image which is subjected to an initial oversegmentation, a CRF is constructed with each label site $x(i)$ and observation site $y(i)$ corresponding to a region in the image. Site adjacency in the field is determined by the adjacency of the regions they correspond to in the image. Thus, edges are drawn in the CRF between sites which correspond to regions in the image that share a border (see Figure 4.6). For each region, the length of the border between it and neighbouring regions is noted as a percentage of the total border length of the region.

The process for determining temporal neighbours as introduced in the HMM framework is a little more complicated. The average motion vector for each re-

Figure 4.5: An example of the watershed segmentation from a frame of a test sequence. The smoothing parameter used for the initial Gaussian smoothing step generally controls the resulting number of segments in the oversegmentation. In this case, σ was set to 2, which was found to achieve an oversegmentation of approximately 200-300 segments in the above test sequence. On the left is the original image. In the centre, the gradient image obtained from the hybrid colour/flow information. On the right, the segmentation produced from the watershed algorithm.

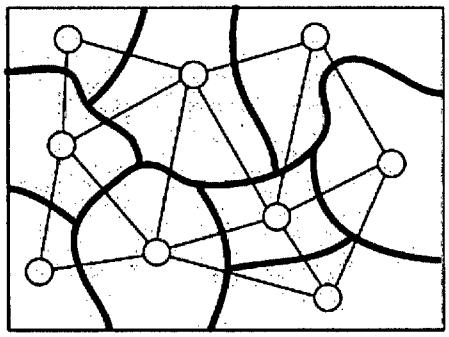

Figure 4.6: An example of a field built upon region adjacency in a planar map. A node is created to represent each region in the map and an edge is drawn between any nodes whose corresponding regions share a border. Such a planar map can be produced by an initial oversegmentation procedure as described in the implementation section, and an appropriate field constructed to represent it.

gion, as determined by the optical flow over the region, is computed and the reverse vector is applied to all the pixels in the region. Since the initial oversegmentation can include optical flow information, the motion within a region should be fairly consistent, and an average should be a reasonable estimate for the motion of the region. The transformed region is then projected onto the previous frame and the overlapping regions are determined. Each region overlapped is assigned as a temporal neighbour of the original region to which the transformed region corresponds. Again, a weight is computed for each region-based on the percentage of pixels which overlap each of the temporally neighbouring regions (see Figure 4.7).

4.3.1 The Observables

The observation at each site consists of both a measure of appearance and motion for the corresponding region. The HSV colour model is used to populate an appearance histogram for each region. The values from the hue and saturation planes of the HSV colour space which are obtained from each pixel in the region

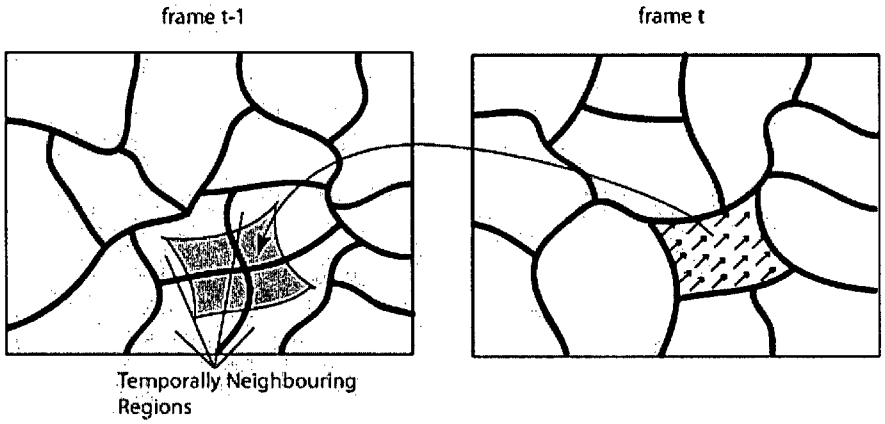

Figure 4.7: An illustration of the determination of temporal neighbours. The optical flow over a region is determined and the region is transformed according to the average flow. The transformed region is projected onto the oversegmentation of the previous frame and the overlapped regions are determined to be the temporal neighbours.

are used to populate an n by n two dimensional histogram. The remaining intensity information from the value plane is then used to populate an n bin one dimensional histogram. The HS histogram is then strung out to produce a single dimension $n \cdot n$ bin histogram which is concatentated with the V histogram to produce a one dimensional $n \cdot n + n$ bin histogram. This histogram is then stored as the appearance observation for a region (Figure 4.8).

Optical flow information for each frame is used to populate a motion histogram for each region in a similar manner. A two dimensional histogram of n by n bins is populated with the x and y vector values obtained from the optical flow at each pixel. The two dimensional histogram is again strung out into an $n \cdot n$ bin single dimension histogram. The resulting histogram is stored as the motion observation for a region.

Models for the foreground object of interest, and the remaining background information in a scene, are learned from the MAP labeling of the segmentation field from the previous frame. The appearance and motion histograms constructed for each region are used to construct the model for each of the fore-

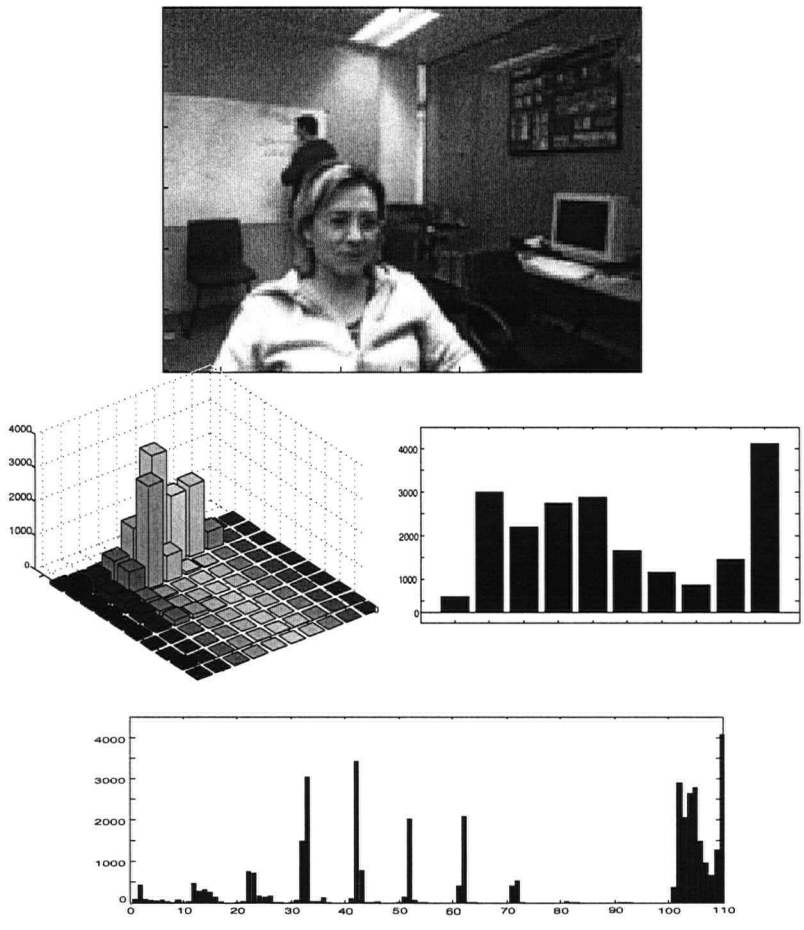

Figure 4.8: A depiction of the histogramming process for a size of $n = 10$ bins over an entire image. On the top, the image with its final one dimensional HSV histogram below. In the middle, the two dimensional HS histogram of the image with the one dimensional V histogram beside. On the bottom, the final concatenated HSV histogram.

ground and background labels. The histograms from all regions whose maximal probability corresponds to a particular label are added together to create an aggregate appearance and motion histogram model for that label.

An appearance history is kept over the last five frames to reduce the effects of spurious labelings in a single frame on the labeling of subsequent frames. At the same time, the adaptive appearance model allows for changes in the appearance of the object being tracked. The histograms from these past five frames are added together and subsequently normalized to create the appearance model used for comparison. No similar such model history is used for motion as similar assumptions cannot be made about object movement, since it is more unpredictable and less likely to be consistent over time.

4.4 The CRF Energy Functions

Once the optical flow has been calculated, the image has been oversegmented and the segments have been processed for appearance and motion data, it remains to define the CRF energy functions and performance of the inference step. These energy functions, used to represent both spatial and temporal smoothness over labels, as well as likelihoods for observed data on both a single and pairwise site basis, are detailed below.

The single site potential function for temporal label smoothness at site i is defined as follows:

$$\begin{aligned} \phi_i^{x_t} &= \sum_{j \in T_i} \phi_{ij}^{x_t}(x_t(i), x_{t-1}(T_i)) \\ &= \sum_{j \in T_i} \left[\alpha_1(1 - \delta(x_t(i) - x_{t-1}(j)))\right]. \end{aligned} \tag{4.2}$$

Again, T_i is the set of temporal neighbours from field $\mathbf{x}_{t-1}$ of site i in field $\mathbf{x}_t$. As can be seen, this function is used to enforce a pairwise interaction between the labels of a site i and its temporal neighbours $j \in T_i$. The Kronecker delta function, $\delta(\cdot)$, allows this potential function to reward a site with a labeling

that matches that of its temporal neighbours by giving it a lower value, which of course results in a higher probability for a configuration containing that labeling. The pairwise site potential function for spatial label smoothness is implemented as

$$\begin{aligned} \psi_{ij}^{x_t} &= \psi_{ij}^{x_t}(x_t(i), x_t(j)) \\ &= \alpha_2(1 - \delta(x_t(i) - x_{t-1}(j))). \end{aligned} \tag{4.3}$$

Here, the effect of the Kronecker delta function is again used to reward labelings that result in similar labels between spatial neighbours with a lower energy. In a sequence of images oversegmented as described by the process in a previous section, this should generally apply to both situations. The parameters α_1 and α_2 allow us to weight the importance of spatial vs. temporal smoothness between labels. In this work, $\alpha_,\alpha_2 > 0$, resulting in what is referred to a the "ferromagentic" case in the statistical physics literature.

The single and pairwise site potentials for the likelihood are separated into those for motion ($\mathbf{m}_t$) and appearance ($\mathbf{a}_t$). The reasonable assumption that motion and colour, given the segmentation field, are conditionally independent is made (the colour of an object should not affect the way it moves). This allows the likelihood to be factored into separate motion and appearance likelihoods.

$$\begin{aligned} p(\mathbf{y_t}|\mathbf{x_t}) &= p(\mathbf{a_t}, \mathbf{m_t}|\mathbf{x_t}) \\ &= p(\mathbf{a_t}|\mathbf{x_t})p(\mathbf{m_t}|\mathbf{x_t}) \end{aligned} \tag{4.4}$$

This in turn results in additive single and pairwise potentials. The single site potentials are based on a distance measure for the observed data from the motion and appearance model for each label.

$$\begin{aligned} \phi_i^{y_t} &= \phi^{y_t}(y_t(i)|x_t(i)) \\ &= \phi^{y_t}(m_t(i)|x_t(i)) + \phi^{y_t}(a_t(i)|x_t(i)) \end{aligned}$$

$$= b(m_t(i), x_t(i)) + b(a_t(i), x_t(i)) \tag{4.5}$$

Here, $m_t(i)$ refers to the motion observed at the region corresponding to site i in the field at time t, while $a_t(i)$ refers to the appearance. The term $b(m_t(i), x_t(i))$ refers to the similarity of the observed motion to what is expected of the motion model for a given labeling at site i, while the term $b(a_t(i), x_t(i))$ refers to the appearance similarity with the appearance model. Labels for which the observations are closer to its model are rewarded with a lower energy through the assignment of a smaller similarity value. Since the motion and appearance observables and models are all represented as histograms, the similarity function $b(\cdot)$ was chosen in this case to be a metric employing the Bhattacharya coefficient [20].

The Bhattacharya coefficient is an approximate measurement of the amount of overlap between two statistical samples. The coefficient often used as a measure of similarity between the two samples being considered. Typically, the samples being compared are continuous probability distribution functions. Determining the coefficient requires that these PDFs first be discretized into a prespecified number of partitions so that a sum can be performed in the place of an integration. In this case, the PDFs have already been discretized into histogram-based appearance/motion observations and models.

Calculating the Bhattacharya coefficient essentially amounts to a rudimentary form of integration of the overlap of the two samples. The number of members of each sample in each bin of the discretization is used in the following formula:

$$D(A, B) = \sum_{i=1}^{n} \sqrt{(A_i \cdot B_i)}. \tag{4.6}$$

Here the sample histograms are A and B, where n is the number of bins, and A_i, B_i are the number of entries of A and B in the i'th bin.

As can be seen, this quantity is larger when corresponding bins in each histogram have more entries. It can be subject to discretization error, and as a result, the choice of number of bins for the histogram can effect the its

performance. Too few bins will lose accuracy by over-estimating the entries in corresponding bins. Too many bins will lose accuracy by creating bins with no entries despite being next to well populated bins just a single step away. The the quantity will be 0 if there is no overlap at all due to the multiplication by zero in every bin. For normalized histograms, this quantity is exactly 1 for a perfect match. The the metric we use is simply $\sqrt{1 - D(A, B)}$ for normalized histograms.

The pairwise potentials are based on the same distance measure between observations given the similarity between labels.

$$\begin{aligned} \psi_{ij}^{y_t} &= \psi^{y_t}(y_t(i), y_t(j)|x_t(i), x_t(j)) \\ &= \psi^{y_t}(m_t(i), m_t(j)|x_t(i), x_t(j)) \qquad (4.7) \\ &= \beta_1 \cdot b(m_t(i), m_t(j)) \cdot \delta(x_t(i) - x_{t-1}(j)) \qquad (4.8) \end{aligned}$$

The Kronecker Delta function is again used to reward label configurations for neighbouring sites which have similar motion when their labels are similar. Since motion is represented as a histogram, the Bhattacharya coefficient is again used as a similarity measure between the motion observations of each region. Since appearance similarities have already largely been exploited by the over-segmentation process, no appearance term is included in the data dependent pairwise potentials. Similarly to the label smoothing potentials, parameters β_1 can be included to weight relative importance of pairwise motion characteristics. Again, as for α_1 and α_2, $\beta_1 > 0$.

In addition to the above potential functions and parameters is the weight associated with the border between each pair of regions. Each pairwise potential is subsequently multiplied by this border weight determined during the segmentation step to affect the interaction between neighbouring regions appropriate to the extent to which they border. Similarly, the temporal smoothness potential (eq 4.2) is weighted according to the extent to which the region in question overlaps each of its temporally neighbouring regions as determined during the field construction.

Node and pairwise potentials are calculated and used in the mean field inference step as described in the theory section. The initial distribution over the labels for each node is set to be equal across labels. The mean field update is performed until convergence of the Gibbs Free energy to a tolerance of 10^{-3}, with a maximal allowance of 200 iterations. The resulting label distributions at each node are used to determine an appearance mask for the object being segmented. The regions represented by nodes whose object label is maximal in its distribution are flagged as belonging to the object and the pixels comprising the region can maintain their original appearance. The pixels belonging to regions whose label distributions indicate otherwise are set to 0.

Chapter 5

Results

This section presents some of the results obtained through experiments with various implementations of the proposed system. It shows a demonstration of the ability of the proposed method to distinguish between interacting foreground objects, or moving objects within a scene in the presence of a dynamic background. It also compares different segmentation methods in the presegmentation step, illustrating the effects of choosing more expensive oversegmentations over cheaper ones. It concludes with a comparison against a leading pixelwise method with respect to the accuracy of the classification of pixels in the image as belonging to an object of interest.

5.1 Presegmentations

The presegmentation step is essentially separate from the labeling step, and allows for different methods to be plugged in to this part of the system. The purpose of this section is to validate the choice of the watershed segmentation presented in the implementation section. It shows a comparisons of different oversegmentation options over a test sequence; the well-known normalized cuts for image segmentation [36], the relatively inexpensive watershed segmentation, and a simple grid segmentation based on 10x10 image patches. Each of these segmentation methods was substituted into the presegmentation step and a field was constructed over the regions as outlined in the implementation section.

A challenging test sequence was created with interacting foreground objects and illumination artifacts (shadows and specularities). Each frame was labeled with the ground truth segmentation containing the object of interest. In this

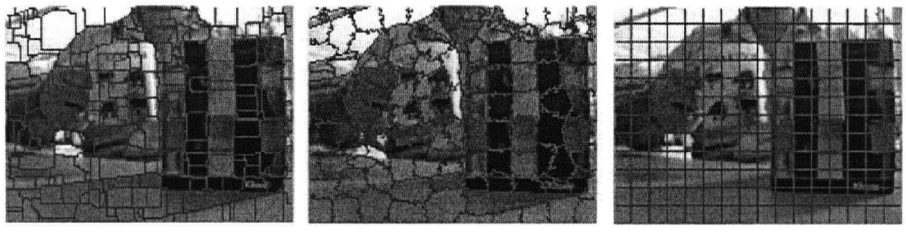

Figure 5.1: Examples of framewise oversegmentations from the mug test sequence. On the left, a watershed segmentation of approximately 200 segments. In the middle, a normalized cuts segmentation of 200 segments. On the right, a patchwise segmentation of the image into 10 by 10 pixel squares.

case, the object of interest was a hand gripping a coffee mug as it occluded and was occluded by a box during its movement throughout the sequence.

To allow for the fairest comparison, approximately the same number of segments were aimed for from each of the oversegmentation methods. A standard normalized cuts oversegmentation was performed on the grayscale image of each frame resulting in 200 segments. A small image size of 160x120 was adopted to reduce computational cost, but the available MATLAB implementation still required on the order of minutes per frame. The patch-based oversegmentation divided the image into 192 equal size square patches of 10x10 pixels each, so as to produce a comparable number of regions. The number of segments produced by the watershed segmentation algorithm can not be controlled directly as it is determined by the shape of the gradient landscape generated on an image by image basis. However, the amount of smoothing over the initial image directly affects the number of segments produced. The watershed algorithm was run with an initial smoothing parameter of $\sigma = 2$, which was found to also result in approximately 200 regions per frame. The entire watershed process, in our matlab implementation, took a small fraction of a second. In the interests of a fair comparison, only the grayscale image information was used in the watershed segmentation, instead of the full colour and motion data. Examples of the segmentations are shown in Figure 5.1.

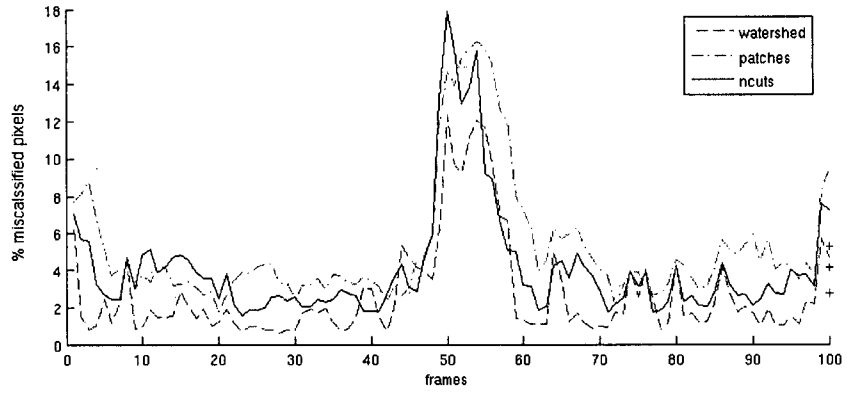

Figure 5.2: Frame by frame percentage misclassification of pixels error for the 100 frame mug test sequence. The watershed segmentation is generally the best performer, followed by normalized cuts and the patch-based segmentations. On the right end of the plot are the mean errors for each segmentation variation illustrated in crosses of the corresponding colour.

With the grid and normalized cuts approaches, the number of segments appropriate for an image must be forced upon the resulting segmentation. In the case of normalized cuts, this will produce superfluous boundaries when this number is too large, and will result in missed boundaries when this number is to small. In the patch based approach, the grid size must be chosen as well. Regardless of the precise discretization chosen, there will necessarily be patches spanning boundaries as image information is completely ignored. The watershed approach, on the other hand, allows for the appropriate number of segments to be chosen based upon the available image information.

All weights affecting the single and pairwise potentials as described in the implementation section were set at unity across all segmentation variations. As can be seen from Figure 5.2, there is not much difference in the overall accuracy of the final segmentation based on the different segmentation methods. However, the magnitude of the pixel-wise misclassification errors in the video segmentation are quite small (typically less than 5% as can be noted from Figure 5.2, and

	error	false positive	false negative
Appearance/Motion Only	46.1	36.7	9.33
Spatial Smoothness	41.1	32.4	8.73
Temporal Smoothness	6.85	0.25	6.60
Full Spatiotemporal CRF	4.00	1.00	3.01

Table 5.1: This table shows the average error over the test sequence for the different variations of the model. Error is in the percentage of misclassified pixels. Each row describes the error as more constraints are added to the model.

in the following subsections), and as a result, even small improvements in the error on the order of 1% or 2% should be noted. Not surprisingly, the coarser patch-based segmentation which ignores any boundary information in the image is generally the worst performer. The normalized cuts segmentation, although much more expensive, performs very closely to the relatively cheap watershed segmentation.

5.2 Model Complexity

This section illustrates the improvements in accuracy afforded by the increases in complexity of the video segmentation model. A comparison over a test sequence is made between different variations of the model. In each case, the bin size for the histogram models of appearance and motion was set to $n = 10$ bins. In each case the sequence was initialized with the ground truth labeling for the first frame. Labeling accuracy is presented in Figure 5.2 and table 5.1.

The first variation involves only colour and motion observations as compared to the colour and motion models learned for the foreground and background each frame. That is, any label or data dependent interactions between sites were omitted and the temporal label smoothness constraint was not included in the potential functions. The test sequence shows that error can be extreme when appearance and motion are not necessarily discriminative in particular

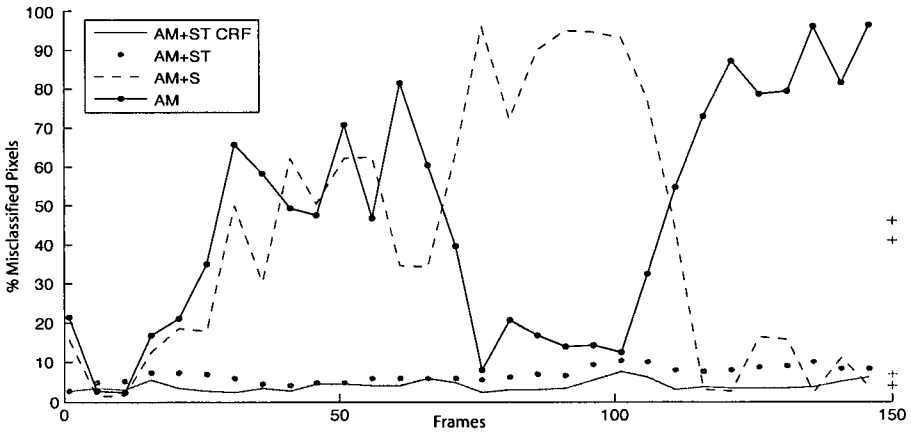

Figure 5.3: Frame by frame percentage misclassification of pixels error for the 150 frame IU test sequence. The appearance/motion only (AM) and appearance/motion with incorporated spatial smoothness (AM+S) versions of the model can be seen to perform quite poorly on a realistic test sequence. The next version of the model incorporating the temporal constraints (AM+ST) can be seen to perform quite favourably, with the full CRF (AM+ST CRF) version described in the previous section outperforming all others. On the right end of the plot are the mean errors for each model variation illustrated in crosses.

frames of the scene.

The second variation incorporates label smoothing constraints to encourage similar labeling of adjacent sites as described in the implementation section, but data dependence in pairwise interactions was still omitted. An equal weighting is given to both the observables and the label smoothness between regions in terms of the energy functions. The results from the test sequence again show that the accuracy is still greatly affected when appearance and motion alone cannot be counted on to discriminate foreground from background. In fact, in some cases, encouraging label smoothness among incorrect labeling can propagate incorrect labelings to other sites in the image, drastically increasing the error.

The third variation incorporates temporal label smoothness into the model in sufficiency to discourage incorrect labeling due to ambiguous appearance and motion of regions. In this case, temporal consistency was given twice the weight of the observable's adherence to object models and the spatial label smoothness, as this was found in practice to produce favourable results. The result is a marked improvement in accuracy. This can be attributed to the tendency of an object to remain consistent temporally between frames and the incorporation of the above temporal constraints into the model which encourage such labeling behaviour, even in the face of uncertainty concerning appearance and motion.

The final variation includes the full model as described in the theory and implementation sections. The result is a further improvement in accuracy. The data-dependent interactions allow for increased support for the similar labeling of consistent neighbouring regions. This helps to overcome label smoothness in spatial and temporal constraints near the boundaries of objects where the tendency might otherwise be to label as background.

5.3 Robust Object Segmentation

This section illustrates the ability of the proposed method to handle the interaction of foreground objects, as well as a dynamic background, and still segment only an object of interest. This is in contrast to other video segmentation tech-

niques that rely primarily on the presence of movement to distinguish an object of interest from the rest of a scene (e.g.,[10, 42]).

The model parameters were set equally except for temporal smoothness, which was given twice the weight, and the system was given an office scene sequence as input where movement not belonging to the object of interest is present. The object of interest in this case was the woman in the foreground of the scene. A seed region was selected on her face as an initialization to the sequence and the subsequent segmentation recorded.

As can be seen from Figure 5.4, the proposed method is robust to the dynamic background in such sequences. The joint motion and appearance models used to describe the woman and the rest of the scene allow for a discrimination between the object of interest and objects moving in the background. Although two other people enter and leave the scene throughout the course of the sequence, the segmentation remains generally on the woman in the foreground. This can be attributed to the inclusion of the optical flow information and its discriminatory power across models when motion differs. This is in contrast to methods which rely simply on a learned background model or motion subtraction to segment a dynamic scene.

5.4 Comparison to a Pixel-Wise CRF

This section illustrates the difference in misclassification error between a pixelwise CRF and the region-based CRF proposed in this work. The most appropriate method for comparison is detailed in [46] and was implemented as described except for one detail. The models for each frame for the object and background were not mixtures of gaussians learned via the adaptive mixtures algorithm based upon an EM update step. Details in the paper were insufficient to recreate the learning process, so instead, non-parametric lookup tables were used to learn a discretized probability distribution for each model in each frame. This approach has been supported recently in [10], where such models were found to give the same performance in a similar setting without having to

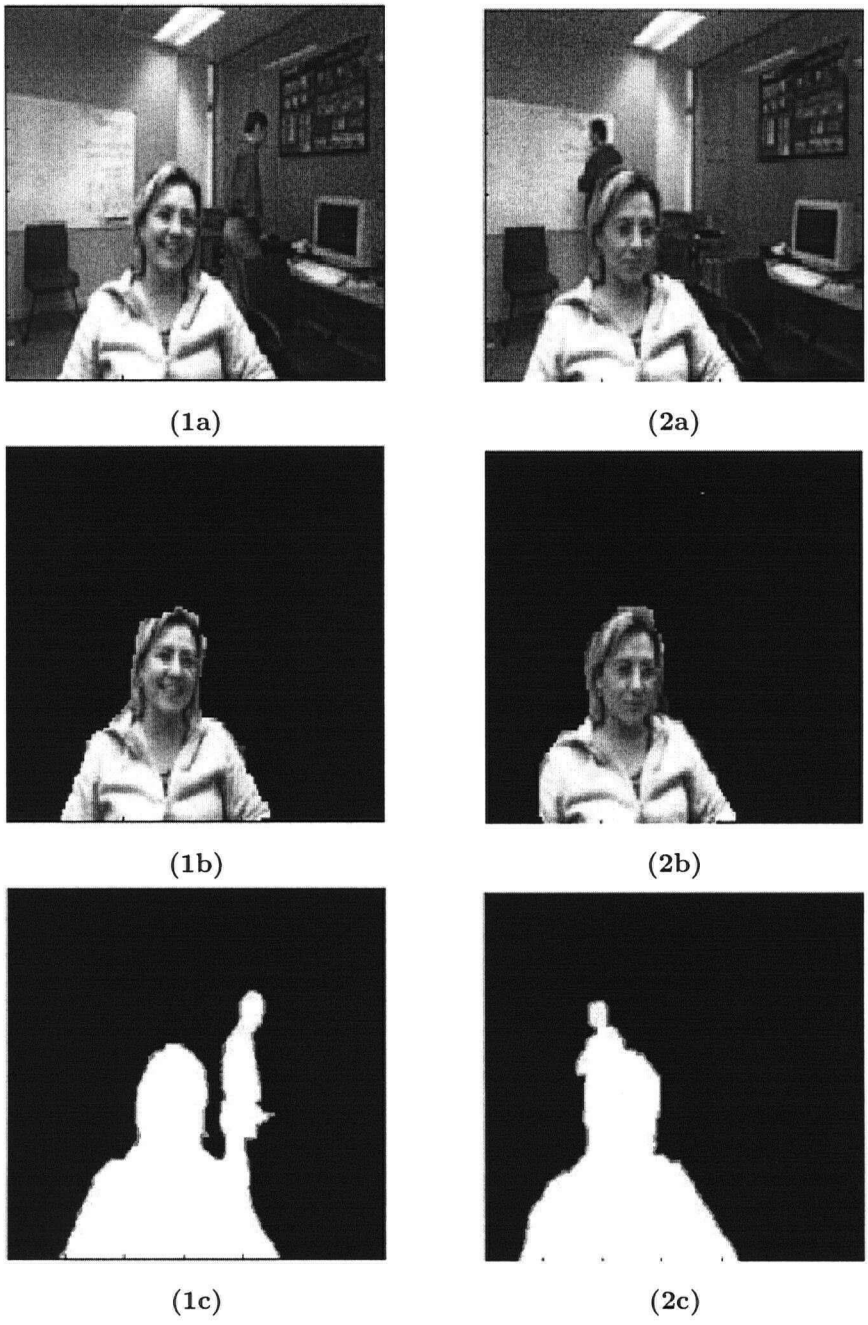

Figure 5.4: An illustration of robustness to dynamic backgrounds by using a few example frames from a test sequence. The images 1a and 2a show frames from the original sequence. The images 1b and 2b show frames from the segmented sequence where the woman is the object of interest. The images 1c and 2c show a ground truth labeling based on the motion of objects throughout the scene.

address to problems typical to learning a generative model.

Both methods were initialized with the ground truth labeling for the first frame. Again the parameters for the region-based approach were set to equal except for the temporal weighting which was set to twice the weight of the other parameters. The overall weighting scheme described in [46] was adopted to determine the relative weights for the appearance vs motion and the spatial vs temporal.

Results of the comparison are detailed in Figures 5.5 and 5.6. The reader may notice a pronounced difference between the performance of the method on the test sequences used here in comparison to those used in the original paper [46]. As the same test sequences were unavailable and the authors were unreachable, we were unable to reproduce the exact results due to some parametric details absent in the original publication. The results produced for this comparison are the product of our best guess (and limited patience). While the proposed method may clearly outperform [46] on the given test sequences, the comparison may not be entirely fair for the above reasons.

A more qualitative comparison of the segmentations can be seen in Figure 5.7, where the MAP labelings for two separate frames of the coast sequence taken from [46] are illustrated. Included for comparison are the corresponding frames for the proposed method using each of the boats in the sequence as an object of interest. The MAP labeling between the each object of interest and the background were taken and the segmentation applied accordingly. [46] reports errors on average of 2.1% for such frames. The proposed method reports errors of 4.3% for the depicted frames. Our labeling is able to distinguish the wake from the boat, while the pixel-wise method includes the wake as part of the smaller boat object. [46] treats such a labeling as correct, however.

A firm argument can be made to the contrary however. While the wake exhibits some motion characteristics that distinguish it from the background, its motion and appearance are not really the same as that of the smaller boat. Its adjacency to the boat and its dissimilarity from the background have apparently caused it to be classified as being part of the boat, when it is in fact not.

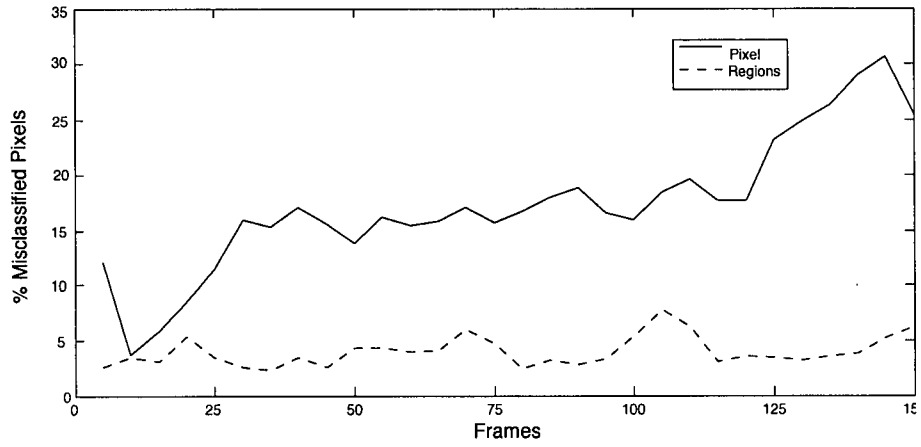

Figure 5.5: A comparison of the frame by frame error between the pixelwise and region-based approaches for the IU test sequence.

Barring an optimal parameter set in terms of the classification accuracy, the rate of convergence of the two methods can still be examined. As can be seen, the proposed region-based method requires many less iterations of the mean field update to converge to a solution. Considering the complexity of a mean field update depends directly on the number of nodes and edges in a graph, this can be a substantial cost. Recall, a mean field update requires an examination of each node, and the effects on it from each of its neighbours:

$$b_k(x_k) = \alpha\phi(x(k))exp\{\sum_{j\in N_k}\sum_{x(j)} b_j(x(j))\ln\psi(x(k),x(j))\} \tag{5.1}$$

For a graph with n nodes and a neighbourhood size of k, over l possible labels, this is nkl. In the case of [46], this involves a 320 by 240 node graph (a node for each pixel in a 320 by 240 image), with a 25 node neighbourhood. This results in close to 2 million comparisons per iteration. In comparison, the proposed region-based approach resulted in somewhere around 300 nodes graph for a 320 by 240 image, and and average neighbourhood size of around 5 neighbours (these numbers fluctuate from frame to frame and can be affected by the parameters used in the oversegmentation). This results in about 3000

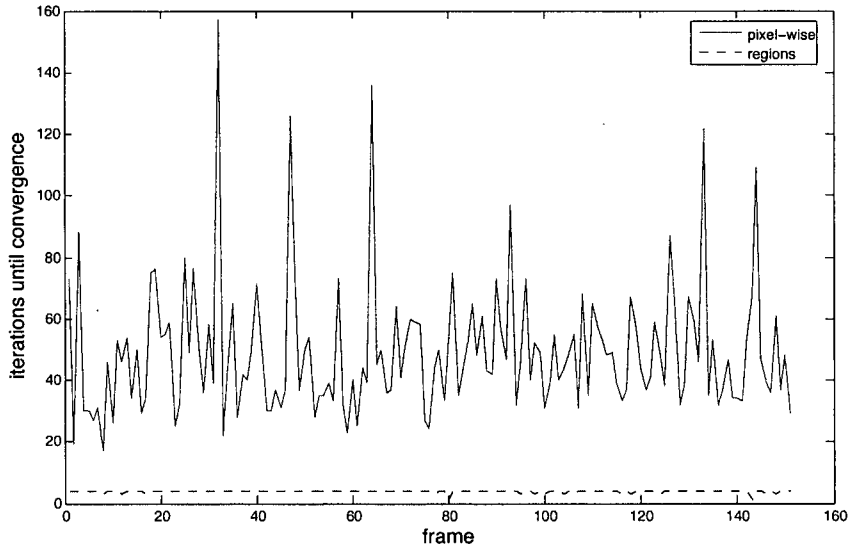

Figure 5.6: A comparison of the number of iterations of mean field updates before convergence for both the pixelwise and region-based approaches for the IU test sequence. The dashed line indicates the number of iterations before convergence for the proposed region-based method. The solid line indicates the number of iterations for the pixel-based method. It is clear that the pixel-based method requires many more iterations for convergence.

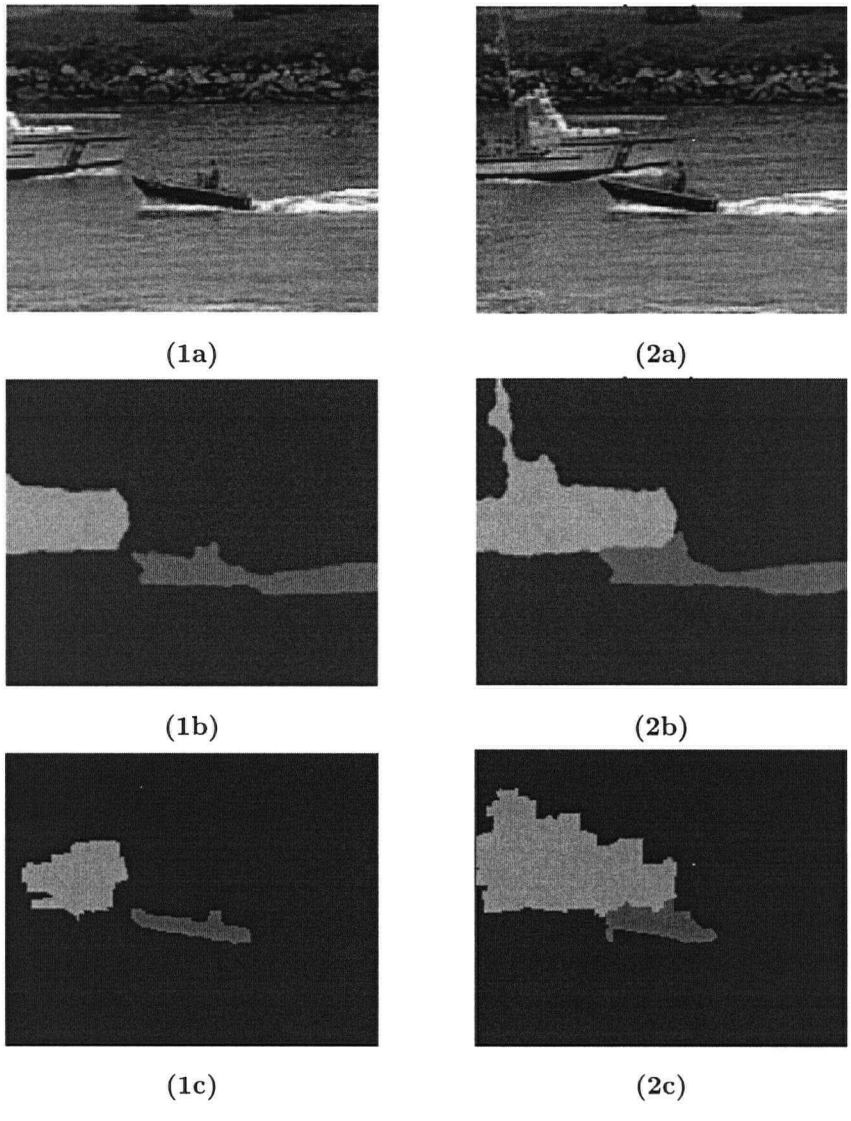

Figure 5.7: A comparison of frame labelings for the pixel-wise method of [46] and the proposed region based approach. 1(a) and (b) present two frames of the coast sequence. 2(a) and (b) depict the labelings for each boat and the background as determined by [46]. 3(a) and (b) show the labelings for the proposed approach.

comparisons per iteration.

In conjunction with the reduced number of iterations required for convergence, the savings are obvious. This reduction in the cost of mean field iterations is, of course, directly paid for by the cost of the oversegmentation. However, depending on the precise oversegmentation scheme (such as the proposed watershed segmentation), this cost can be relatively small.

Chapter 6

Conclusions and Future Work

We have presented a novel online approach for labeling regions in an image sequence to achieve an object based segmentation of video. It combines the spatiotemporal filtering ideas presented in [46] within a region based framework. Spatial and temporal consistencies are enforced through interactions between segments produced by an initial oversegmentation process in each frame of the video sequence. The solution is expressed in terms of an optimal labeling as determined by a mean field approximation in a conditional random field based upon these interactions. In the process, we have also introduced a modification to the classic watershed algorithm to take advantage of motion information, in the form of optical flow, during the initial oversegmentation process.

We have shown results which validate the choice of such a simple oversegmentation scheme even when much more complex and popular schemes are available. We have also shown the validity of the choices in the model in terms of an increased accuracy with increased complexity. Finally, a comparison to an appropriate pixel-based segmentation method is made to illustrate the comparable error rates and where a region-based approach can be advantageous.

We have illustrated a generalization to the filter presented in [46] that applies to a general field model based upon regions in an image obtained from an initial oversegmentation step. This model is shown to have much less complexity than a corresponding pixel-wise model, and has the freedom to change in structure from frame to frame to appropriately fit the data. The cost of inference in the

resulting filter is greatly reduced, both in the number of iterations required for convergence, as well as the cost of each iteration. The resulting segmentation of an image sequence is shown to have comparable accuracy to that produced by similar pixel-wise methods.

Avenues exist for exploration of future work. Parameter learning for the weights affecting the potential functions could be employed. However, a static parameter weighting learned over the duration of multiple sequences might not be appropriate for the challenges presented in novel seqeunces. More appropriate might be some form of adaptive weighting scheme based upon the data observations each frame. Frames in which motion or appearance is uninformative can have the appropriate parameter weighting adjusted.

Additionally, while labels for regions in the interior of an object are seldom misclassified, the regions occupying object boundaries are much more susceptible to misclassification. This is a direct side effect of the label smoothness constraints adopted in such approaches. While this effect can be addressed somewhat through the adjustment of the parameter that weights the importance of label smoothness, some notion of shape similarity between frames might also be introduced. This might help to avoid the "flicker" around object boundaries in the video segmentation resulting from the misclassification of background bordering regions.

There is always a choice over which observable features to use. Currently, a simple histogram-based approach to region descriptors has been employed and found to be quite effective. Other options for region-based measures and models can be explored which may be more discriminative. However, the more complex the descriptors, the higher the cost of computation. This is of concern when dealing with the amounts of data involved with video segmentation.

Finally, the possibility exists of employing other approximation techniques to the inference problem. Currently, mean field approximation is used as the inference engine and the filter is derived to accommodate it. Other approximation techniques for inference exist, such as Loopy Belief Propagation, expectation propagation, Monte Carlo, conditional mean field, and higher order

approximate distributions for use in the variational scheme. The application of such approximation schemes could be explored and re-derivations of the filter performed in order to accommodate them where necessary.

Bibliography

[1] Moray Allan, Michalis K. Titsias, and Christopher K. I. Williams. Fast learning of sprites using invariant features. In *Proceedings of the British Machine Vision Conference*, 2005.

[2] Mark C. Allmen and Charles R. Dyer. Computing Spatiotemporal Relations for Dynamic Perceptual Organization. *Computer Vision, Graphics and Image Processing: Image Understanding*, 58:338–351, 1993.

[3] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36:192–236, 1974.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[5] Michael J. Black. Robust incremental optical flow, 1992.

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

[7] Michael M. Chang, A. Murat Tekalp, and M. Ibrahim Sezan. Simultaneous motion estimation and segmentation. *IEEE Transactions on Image Processing*, 6(9):1326–1333, 1997.

[8] H. Cheng and C. Bouman. Multiscale Bayesian segmentation using a trainable context model. *IEEE Transactions on Image Processing*, 10(4):511–525, April 2001.

[9] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[10] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 53–60, 2006.

[11] Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

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

[13] Charless Fowlkes, Serge Belongie, and Jitendra Malik. Efficient spatiotemporal grouping using the Nyström method. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 231–238, 2001.

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

[15] Hayit Greenspan, Jacob Goldberger, and Arnaldo Mayer. Probabilistic space-time video modeling via piecewise GMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):384–396, 2004.

[16] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[17] B. K. P. Horn and E. J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76, 1988.

[18] B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[19] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 199–206, 2001.

[20] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, 1967.

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

[22] Sanjiv Kumar and Martial Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proceedings of the IEEE International Conference on Computer Vision*, page 1150, 2003.

[23] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[24] Stan Z. Li. *Markov random field modeling in image analysis.* Springer-Verlag New York, Inc., 2001.

[25] Kia-Fock Loe and Jian-Kang Wu. A dynamic conditional random field model for foreground and shadow segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):279–289, 2006.

[26] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 416–423, July 2001.

[27] R. Megret and D. DeMenthon. A survey of spatio-temporal grouping techniques. In *Technical report: LAMP-TR-094/CS-TR-4403, University of Maryland, College Park*, 2002.

[28] R. Megret and Jean-Michel Jolion. Tracking scale-space blobs for video description. *IEEE MultiMedia*, 9(2):34–43, 2002.

[29] Fabrice Moscheni, Sushil Bhattacharjee, and Murat Kunt. Spatiotemporal segmentation based on region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):897–915, 1998.

[30] Manfred Opper and David Saad. *Advanced Mean Field Methods.* MIT Press, Cambridge., 2001.

[31] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[32] Stephen E. Palmer. *Vision Science: Photons to Phenomology.* MIT Press, Cambridge, 1999.

[33] Ioannis Patras, E. A. Hendriks, and Reginald L. Lagendijk. Video segmentation by MAP labeling of watershed segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):326–332, 2001.

[34] Conrad J. Poelman and Takeo Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218, 1997.

[35] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1154–1160, 1998.

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

[37] Tristram Southey. ODMAS: Object discovery through motion appearance and shape, 2005.

[38] A. Murat Tekalp. *Digital Video Processing.* Prentice Hall, NJ, 1995.

[39] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[40] Philip H. S. Torr and Andrew Zisserman. Concerning Bayesian motion segmentation, model, averaging, matching and the trifocal tensor. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 511–527, London, UK, 1998. Springer-Verlag.

[41] E. Trucco and A.Verri. *Introductory Techniques for 3D Computer Vision.* Prentice Hall, NJ, 1998.

[42] Y. Tsaig and A. Averbuch. Automatic segmentation of moving objects in video sequences: a region labeling approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(7):597–612, 2002.

[43] R. Vidal and R. Hartley. Motion segmentation with missing data using power factorization and GPCA. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 310–316, 2004.

[44] L. Vincent and O. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1992.

[45] J. Y. A. Wang and E. H. Adelson. Representing Moving Images with Layers. *IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, September 1994.

[46] Yang Wang and Qiang Ji. A dynamic conditional random field model for object segmentation in image sequences. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 264–270, 2005.

[47] Roland Wilson and Chang-Tsun Li. A class of discrete multiresolution random fields and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):42–56, 2003.

[48] Jiangjian Xiao and Mubarak Shah. Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1644–1659, 2005.