Image and Video Segmentation using Graph cuts

Mayuresh Kulkarni

Supervised by Dr. Fred Nicolls

Submitted to the Faculty of Engineering, University of Cape Town,
in fulfillment of the requirements for the
Degree of Master of Science.

November 2010

Declaration of Authorship

I declare that this dissertation is my own work except where otherwise stated. It is being submitted for the degree of Master of Science in Engineering at the University of Cape Town and it has not been submitted before for any degree or examination at any other university.
I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own.

Signed: _____

Date: _____

Mayuresh Kulkarni.
November 2010.

"The brick walls are not there to keep us out. The brick walls are there to give us a chance to show how badly we want something. Because the brick walls are there to stop the people who don't want it badly enough."

Randy Pausch

Abstract

This dissertation explores the problem of binary segmentation of images and videos into two classes, foreground and background. Methods in this thesis are based on the graph cut algorithm.

Image segmentation for grayscale and colour images is discussed. A cost function based on the region and boundary properties of the image is minimized. Certain pixels marked by the user are used to constrain globally optimal solutions to the segmentation problem. Segmentation using shape priors is studied and the problems of initializing and aligning the shape prior to the object are addressed. The performance of different methods is evaluated using robust measures like precision, recall, F-score and accuracy.

Video segmentation is viewed as a 3D graph cut problem and different solutions are proposed. The method of shape priors is extended from images to videos. The cost function is modified to include a shape and a proximity term in addition to region and boundary terms. A circular shape is used as a prior and is defined using its center and radius. Powell's method is used to get the best alignment of the shape prior with the object. Segmentation using shape priors is compared to gradient and motion based methods. Although the shape prior is used to track faces in video data in this thesis, it can be used to track any other object with a parametrically described shape.

Accurate image and video segmentations are achieved with minimal user input. The average time taken for a segmentation is 0.2 seconds for images and 2 seconds per frame for videos. Conclusions and suggestions for future research are made as closing remarks to this dissertation.

Acknowledgements

I would like to thank the following people:

- My supervisor, Dr. Fred Nicolls, for providing a motivating environment for me to do this work. I would also like to thank him for patiently discussing ideas with me and cleaning up code. This work would not have been possible without his support.
- Mark and Nicole Ferris for putting up with my irregular work habits, the interesting conversations and all the fun over the period of this thesis. Also, thanks to Buffett and Daisy for their love and support.
- Baruch Lubinsky and Direshni Reddy for reading this document and suggesting improvements.

Contents

List of Figures

List of Tables

Abbreviations

AAM	**A**ctive **A**ppearance **M**odels
ASM	**A**ctive **S**hape **M**odels
CRF	**C**onditional **R**andom **F**ields
CLM	**C**onstrainted **L**ocalized **M**odels
EM	**E**xpectation **M**aximization
GMM	**G**aussian **M**ixture **M**odel
HMM	**H**idden **M**arkov **M**odel
LDP	**L**ayered **D**ynamic **P**rogramming
LGC	**L**ayered **G**raph **C**uts
LBP	**L**ocal **B**inary **P**attern
MRI	**M**agnetic **R**esonance **I**maging
MRF	**M**arkov **R**andom **F**ield
MAP	**M**aximum **a** **P**osteriori

To my parents, Milind and Megha...

Chapter 1

Introduction

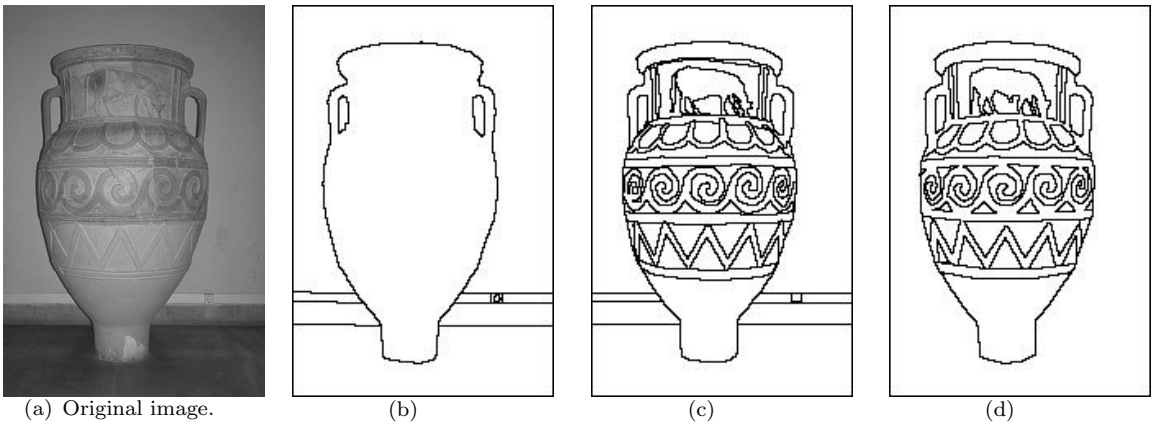
This thesis evaluates different variations of the graph cut algorithm described in Boykov and Jolly [2] applied to image and video segmentation. Images from the Berkeley Segmentation Dataset [1] and videos from the Microsoft i2i Dataset [3] are used to evaluate the performance of the segmentation. Segmentation using graph cuts and shape priors is discussed. Suggestions for future research are made.

The reader is assumed to have basic knowledge of image processing and linear algebra. Other concepts like graph cuts and Gaussian Mixture Models (GMMs) are described.

1.1 Image Segmentation

Image segmentation is the extraction of regions of interest from images. Fully automatic segmentation has inherent problems associated with it. This thesis focuses on interactive image segmentation into 'foreground' and 'background'. There can be many different segmentations of a given image, as shown in Figure 1.1. Some form of prior information is necessary to get a good and desirable segmentation.

The user marks certain pixels as 'foreground' and 'background', also known as *seeds*. Seeds are used as hard constraints for the segmentation. Hard constraints provide the clues to the desired segmentation. A graph is set up using each pixel as a node. Each pixel or node is connected to adjacent pixels in all directions to define the edges. A cost function based on region and boundary properties is defined. Weights for including regions or boundary elements in the solution are estimated using the properties of the hard constraints using GMMs. In this work we refer to these as soft constraints, since solutions with low total weight are preferred. Colour and texture features are used as components of the GMMs. The probability of each pixel being either 'foreground' or 'background' can



(a) Original image. (b) (c) (d)

FIGURE 1.1: Original image in (a) and possible segmentations in (b), (c) and (d). Image taken from [1].

be estimated using the logarithmic likelihood ratio. Edge detection methods are used to find the evidence of a boundary in each pixel in the image. A globally optimal solution is calculated using soft and hard constraints. The segmentation process can be made iterative to get the desired result. A globally optimal segmentation can be efficiently recalculated when the user adds or removes hard constraints at each iteration [4].

Intensity, colour and texture properties are used as features in GMMs to assign soft constraints to pixels. Different colour spaces like RGB and Luv are used to overcome the drawbacks of any single scheme. Combinations of colour and texture are used to analyse the best features. Edge detection methods like the Canny edge detector, gradient methods and a GMM-based edge model are used. A novel edge model based on GMMs is proposed to differentiate between relevant and irrelevant boundaries. The GMM to detect edges has two components, the first corresponding to 'foreground' pixels and the second to 'background' pixels. Pairs of neighbouring pixels are compared to the GMM to assign soft constraints. The output from the novel edge detector are compared to the Canny edge detector and gradient based methods.

Segmentations resulting from different methods are compared to ground-truth images using precision and recall. The accuracy of segmentations is also inspected visually and conclusions about the different methods are drawn. The run-time of the segmentation is 0.2 seconds on average. The majority of the time is taken to set up the GMMs and to compare the image data to them.

The segmentation technique is modified to include prior knowledge about the shape of the object. In many cases, the shape of the object to be segmented is known. An extra term is added to the cost function to bias the segmentation towards a shape. The shape term can be weighted to control its strength. Better segmentations can be accurately achieved using shape priors. The shape is aligned to the object in the image using Powell's method

of minimization [5] which searches for the best overall solution over all shape parameters. Segmentation using graph cuts and shape priors can be used to segment or retrieve for large databases of similar images. Application of segmentation using shape priors for medical images is discussed.

1.2 Video Segmentation

Videos are a collection of images over time. This thesis investigates different methods of segmenting videos using graph cuts with minimal user input. The techniques learnt from image segmentation are extended to videos while using the spatial and temporal information in the video sequence.

Video segmentation is more difficult than single-image segmentation because of the motion of the object in the video. Approximate and accurate motion models are tested to locate the object. The object can not only change position but can also change shape over time in the video. Motion, region, boundary and shape properties of the object need to be included in the cost function to make it robust and globally optimal.

The video sequence is viewed as a 3D graph constructed in a spatiotemporal data cube with all pixels being connected to their neighbours in the same frame (spatial) and to those in previous and subsequent frames (temporal). Each pixel is a node and connections between pixels are edges in the graph structure. A cost function based on region and boundary properties is used to segment the video. A globally optimal solution is achieved using hard and soft constraints. Hard constraints are set using the first frame of the sequence. The user marks certain pixels as 'foreground' and 'background'. The hard constraints are set using seeds, and soft constraints are assigned using properties of the data in the vicinity of the hard constraints.

Colour and texture properties are used as features in GMMs. Edge detection methods are used to assign weights to candidate spatial boundaries in the video sequence. The motion information of the video is extracted using background estimation and frame subtraction. The motion information is used to estimate the temporal boundary weights in the cost function.

Ways of extending segmentation using graph cuts and shape priors is investigated. A circular shape prior is defined using the center and radius as its parameters. The shape prior is imposed on the image and graph weights are modified using this prior. The distance transform from the shape is used to modify the region weights obtained from the GMMs. Powell's minimization algorithm [5] is used to optimize the position of the prior to give the smallest minimum cut in each frame. A proximity term is included in the cost function to penalize the cut against sudden changes in location of the prior in the

video sequence. This represents a single dynamical model for the shape model over time. It is observed that graph cuts with shape priors result in more accurate segmentations than other methods that do not use shape priors. The ideas discussed in PoseCut [6] are extended to videos.

The methods for video segmentation are tested on the 'Geoff', 'Antonio' and 'MS' videos from Microsoft i2i dataset [3]. A new video sequence, called the 'tennis ball' sequence, is recorded and introduced. The 'tennis ball' sequence is simple and easy to segment. It is used to test new methods and evaluate their performance. The performance of all methods is tested using visual inspection. The speed of the segmentation is approximately 2 seconds per frame on average.

Avenues for future research are discussed based on the work done in this thesis. Graph cuts can be extended to segmentation into more classes than just 'foreground' and 'background'. A feature selection stage can be included into the segmentation process to choose the features that separate 'foreground' and 'background' in the most effective way. Edge or boundary detection in images and videos can be explore further. A more flexible shape prior can be used to enable better estimation of the object. Further research can be done to simultaneously track and segment object that change shape over the video sequence.

1.3 Layout of this document

This document is organised as follows. Chapter 2 describes graphs and graph cuts. Chapter 3 gives a brief account of the previous research done in this field. Chapter 4 investigates the use of graph cuts in image segmentation as parameters that result in different segmentations are varied. Chapter 5 investigates ways of extending graph cuts for video segmentation. Chapter 6 concludes the dissertation by discussing the findings of this work and suggesting future improvements.

Chapter 2

Graphs and graph cuts

This chapter describes graphs and graph theory relevant to this thesis. It explains graph cuts and the use of graph cuts for image segmentation. Region and boundary properties are described and used to set edge weights in the graph. An example of the segmentation of a simple image is presented to show the link between edge weights and image properties. This chapter explains the terminology used in later chapters.

2.1 What is a graph?

A graph is a network of points connected with lines. The points are known as nodes or vertices and the lines are called edges or arcs. A formal definition of a graph is a pair of sets $(\mathcal{V}, \mathcal{E})$, where :

- $\mathcal{V}$ is a nonempty set of points called *vertices*.
- $\mathcal{E}$ is a collection of two-point subsets of $\mathcal{V}$ called *edges*.

Figure 2.1 shows pictorial representations of two graphs. These graphs can be written as:

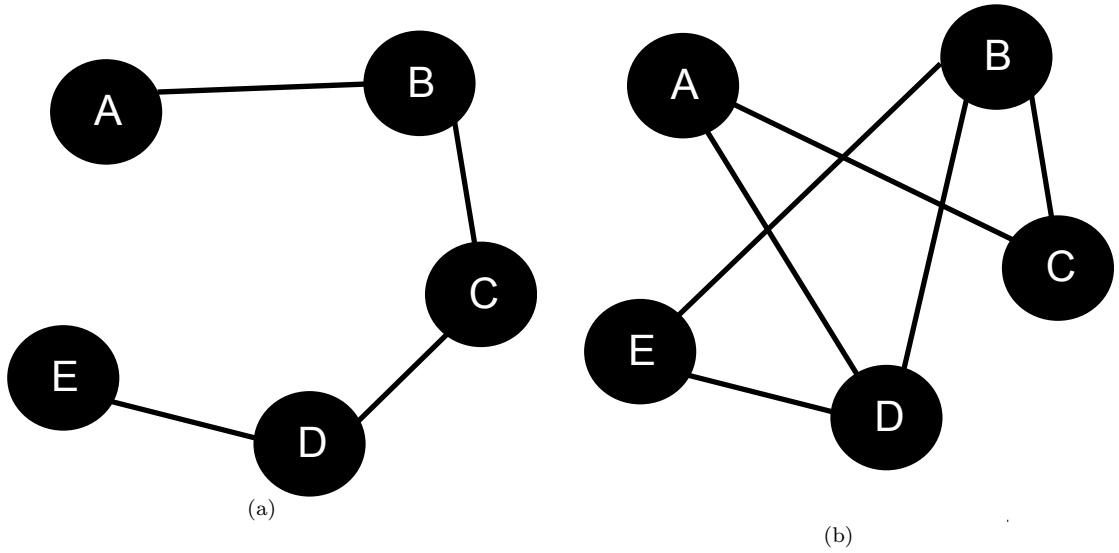
$$\mathcal{V}_a = \{A,B,C,D,E\} \tag{2.1}$$

$$\mathcal{E}_a = \{\{A,B\},\{B,C\},\{C,D\},\{D,E\}\}$$

$$\mathcal{V}_b = \{A,B,C,D,E\} \tag{2.2}$$

$$\mathcal{E}_b = \{\{A,C\},\{B,C\},\{A,D\},\{B,D\},\{B,E\},\{D,E\}\}$$

The sets $\mathcal{V}_a$ and $\mathcal{V}_b$ are equal as the number of vertices is the same in both graphs. But the size of the edge set increases from 4 to 6 because there are more edges in the second graph.



(a) (b)

FIGURE 2.1: Pictorial representation of two graphs.

2.1.1 Connected graphs

A graph is *connected* if every pair of vertices is connected by a path [7]. The two graphs shown in Figure 2.2 explain this definition where the graph in Figure 2.2(a) is not connected because it has two pieces, but the graph in Figure 2.2(b) is connected. A graph is *strongly connected* if every two vertices are reachable from each other. Connected graphs will be used in this thesis for image and video segmentation, although this is not a requirement for graph cut algorithms.

2.1.2 Directed and undirected graphs

A *directed graph* or (*digraph*) $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a finite set and $\mathcal{E}$ is a binary relation on $\mathcal{V}$ [7]. The edge set $\mathcal{E}$ therefore consists of ordered pairs. In an *undirected graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ the edge set $\mathcal{E}$ consists of unordered pairs of vertices, rather than of ordered pairs. Each edge has a weight associated with it, which is a number representing the cost between the vertices joined by that edge. Figure 2.3 shows the difference between directed and undirected graphs. A directed graph with edges directed from one vertex to another is shown in Figure 2.3(a). For instance, the edge between vertex 1 and 2 is directed from 2 to 1. Figure 2.3(b) gives an example of an undirected graph with no directionality to the edges.

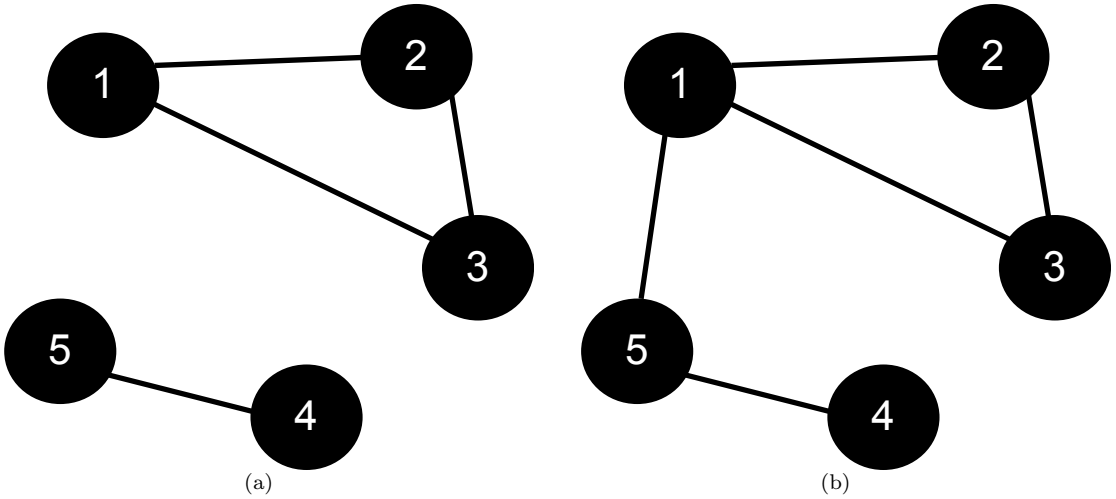


FIGURE 2.2: Unconnected graph in (a) and connected graph in (b).

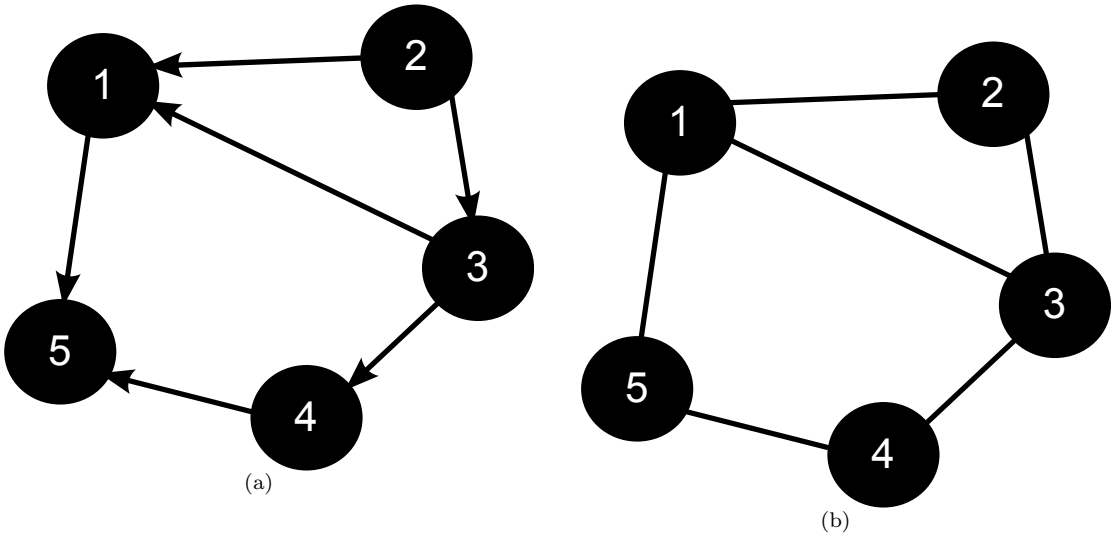


FIGURE 2.3: Directed graph in (a) and undirected graph in (b).

2.1.3 Max-flow min-cut theorem

A flow network is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which each edge $(u, v) \in \mathcal{E}$ has a nonnegative *capacity* $c(u, v)$. The capacity between two nodes u and v is assumed to be zero if $(u, v) \notin \mathcal{E}$. Two special nodes or vertices are defined as the *source* s and the *sink* t. Every vertex lies on some path from the source to the sink node. The *flow* is defined as a real-valued function that satisfies the following three constraints [7]:

- For all $u, v \in \mathcal{V}$, $f(u, v) \leq c(u, v)$,
- For all $u, v \in \mathcal{V}$, $f(u, v) = -f(v, u)$, and

- For all $u \in \mathcal{V} - \{s, t\}, \sum_{v \in \mathcal{V}} f(u, v) = 0$.

The last condition states that the total flow coming into a node should be equal to the total flow going out of a node. The value of the flow f is defined as

$$|f| = \sum_{v \in \mathcal{V}} f(s, v), \tag{2.3}$$

which is equal to the amount of flow passing from the source to the sink. The *maximum flow problem* is to maximize $|f|$, that is to push as much flow as possible from the source s to the sink t. A flow network from the source node s to the sink node t is shown in Figure 2.4(a). Each edge is labeled with its capacity. A flow in the network in Figure 2.4(a) with a flow value of $|f| = 19$ is shown in Figure 2.4(b). Each edge in Figure 2.4(b) is labeled with its positive flow. Note that the values of flow and capacity in the flow network meet the constraints listed above.

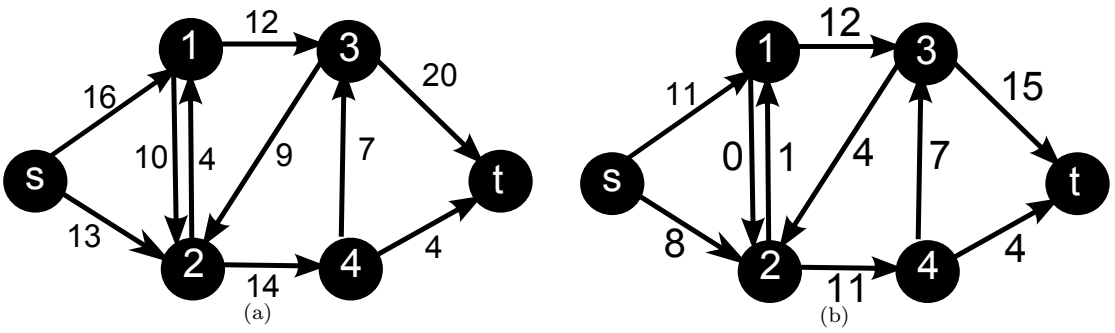


FIGURE 2.4: A flow network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is shown in (a). Each edge in (a) is labeled with its capacity. A flow f with value $|f| = 19$ is shown in (b). Each edge in (b) is labeled with its positive flow.

A *cut* (S, T) of a flow network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a partition of $\mathcal{V}$ into S and $T = \mathcal{V} - S$ such that $s \in S$ and $t \in T$. The net flow across the cut (S, T) is $f(S, T)$. The capacity function for the cut is $c(S, T)$. A *minimum cut* is defined as the partition (S, T) whose capacity is minimum over all cuts of the network. Hence the minimum cut problem is to find the cut where the capacity of the network is minimum. Figure 2.5(a) shows a flow network and Figure 2.5(b) shows a cut (S, T) in this flow network. The net flow across the cut $(\{s, 1, 2\}, \{t, 3, 4\})$ is

$$f(1, 3) + f(2, 3) + f(2, 4) = 12 + (-4) + 11 = 19,$$

and its capacity is

$$c(1, 3) + c(2, 4) = 12 + 14 = 26.$$

The net flow across a cut can include negative flows between vertices, but the capacity consists of nonnegative values. The value of the flow is less than or equal to the value of the capacity. In Figure 2.5(b) the vertices in S are shown in black and the vertices in T are shown in grey. The dotted line shows the cut. Thus a cut divides all the vertices into either a *source* class or a *sink* class.

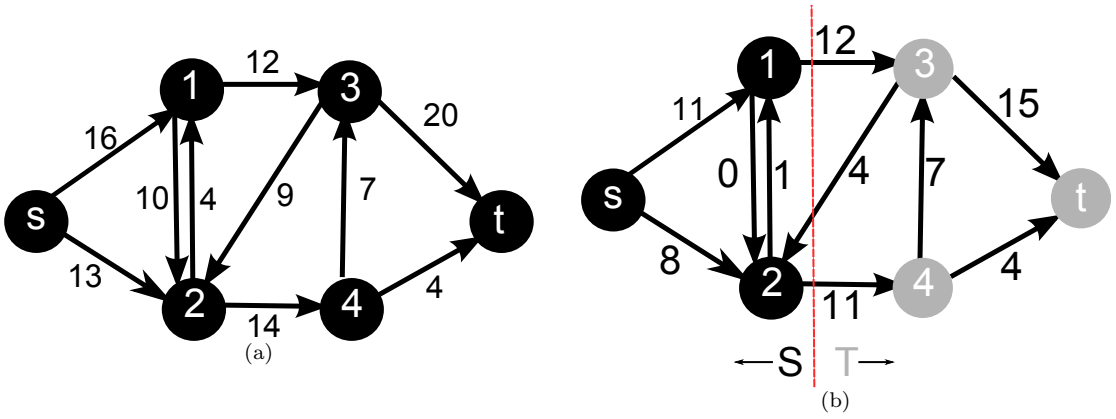


FIGURE 2.5: A flow network is shown in (a). A cut (S, T) in the flow network where $S = \{s, 1, 2\}$ and $T = \{t, 3, 4\}$ is shown in (b). The vertices in S are black and the vertices in T are brown.

The maximum flow problem and the minimum cut problem are linked by the duality theorem [7]. The max-flow min-cut theorem states that the maximum value of an $s - t$ flow is equal to the minimum capacity of an $s - t$ cut. Thus, solving the minimum cut problem will also solve the maximum flow problem.

The cost functions used in this thesis are constructed in a minimum cut framework. By the max-flow min-cut theorem these functions can be globally optimised using maximum flow algorithms, which are efficient and have fast implementations for the types of graphs occurring in vision problems [8].

2.2 Images and videos as graphs

2.2.1 Pixel and voxel connectivity

A graph is a network of nodes connected with edges. Images and videos can be viewed as directed graphs with each pixel or voxel being a node. The connections between pixels or voxels are the edges in the graph and can be corresponded with boundary elements in a segmentation. In this thesis, every pixel or voxel in an image is connected to the pixels or voxels adjacent to it in all directions. This connectivity gives rise to an 8-pixel neighbourhood for each pixel in the image. For videos a 26-voxel connectivity is used

where each voxel is connected to 8 surrounding voxels in the same frame (intra-frame spatial connections) and 9 voxels in each previous and subsequent frame (inter-frame spatiotemporal connections). The weights assigned to neighbourhood connections are called *n-weights*. Every pixel or voxel is also connected to the *source* terminal and the *sink* terminal. The weights assigned to terminal connections are called *t-weights*.

2.2.2 Region and boundary properties

The edge weights in the graph are set according to evidence for regions and boundaries in the image. In the segmentation problem, the source and sink terminals are viewed as foreground and background. The solution to the minimum cut problem leaves each node connected to either the source or the sink, in turn segmenting the pixels or voxels into foreground or background. The terminal weights are viewed as the probability of each pixel or voxel belonging to the source or the sink. The t-weights are set using region properties like colour or texture. The neighbourhood weights are set proportional to the boundary properties of the pixel or voxel. A boundary in the image provides the evidence for pixels or voxels on either side belonging to the different classes, foreground and background. Hence n-weights are set using the edge information in the image.

2.3 Graph cuts

This section describes the graph cut algorithm as used for segmentation. Even though images are used to explain the algorithm, the method can easily be extended to videos. Graph cuts for N-D segmentation was presented in Boykov and Jolly [2].

The goal is to compute a globally optimal segmentation using region and boundary evidence estimated from the image. A graph is set up using the 8-pixel neighbourhood connectivity as described in Section 2.2. The n-weights and t-weights are set using the region and boundary evidence. The desired segmentation is a binary one, a minimum cut dividing the image into two classes. These two classes are 'foreground' (or 'object') and 'background'. The cut that minimizes the capacity also maximizes the flow, as described in Section 2.1.3.

Let $\mathcal{V}$ be the set of pixels in an image and let $\mathcal{E}$ be a set of all unordered pairs of neighbouring pixels $\{p,q\}$. Under the assumptions described previously. The set $\mathcal{E}$ contains the 8-neighbourhood connectivity to each pixel in case of an image. Also, let $\mathcal{A} = (A_1, \cdots, A_p \cdots, A_{|\mathcal{V}|})^T$ be a binary vector whose components A_p denote background or foreground assignments to pixels in $\mathcal{V}$. Any instance of $\mathcal{A}$ defines one possible segmentation out of a set of all possible segmentations. Each element A_p in the vector $\mathcal{A}$ can

either be one or zero, 'foreground' or 'background'.

Let $B_{\{p,q\}}$ be the boundary term for two neighbouring pixels, R_p be the region term and $\mathcal{N}$ contain all the unordered pairs of neighbouring pixels or voxels. The cost function based on the estimated region and boundary properties of the image can be described by

$$E(A) = \lambda \cdot R(A) + B(A), \tag{2.4}$$

where

$$R(A) = \sum_{p \epsilon \mathcal{P}} R_p(A_p) \tag{2.5}$$

$$B(A) = \sum_{\{p,q\} \epsilon \mathcal{N}} B_{\{p,q\}} \cdot \delta(A_p, A_q) \tag{2.6}$$

and

$$\delta(A_p, A_q) = \begin{cases} 1 & \text{if } A_p \neq A_q, \\ 0 & \text{otherwise.} \end{cases}$$

The pixels marked as foreground or background by the user are hard constraints on the segmentation. The hard constraints provide clues to the desired segmentation.

The region term $R_p(A_p)$ reflects how well a pixel p matches foreground or background model based on region properties like colour, intensity or texture, where the models are estimated from image data in the vicinity of the hard constraints. The boundary term $B_{\{p,q\}}$ is set to be proportional to the evidence of a boundary between two neighboring pixels p and q, and can also be estimated from corresponding models. In equation (2.4), λ is a coefficient that weights region properties $R_p(A_p)$ to boundary properties $B_{\{p,q\}}$.

Boykov and Jolly [2] describe a graph construction to minimize $E(A)$ given $R_p(A_p)$ and $B_{\{p,q\}}$. Assuming that $\mathcal{O}$ and $\mathcal{B}$ denote pixels marked by the user as object ("OBJ") and background ("BKG"), where s and t are the source and sink terminals respectively, the weights of the edges in the graph are assigned as follows:

TABLE 2.1: Assignment of edge weights in Boykov and Jolly [2].

edge	**weight (cost)**	**condition**
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in \mathcal{E}$
$\{p,s\}$	$\lambda \cdot R_p(\text{"}bkg\text{"})$	$p \in \mathcal{V},\ p \notin \mathcal{O} \cup \mathcal{B}$
	K	$p \in \mathcal{O}$
	0	$p \in \mathcal{B}$
$\{p,t\}$	$\lambda \cdot R_p(\text{"}obj\text{"})$	$p \in \mathcal{V},\ p \notin \mathcal{O} \cup \mathcal{B}$
	0	$p \in \mathcal{O}$
	K	$p \in \mathcal{B}$

where

$$K = 1 + \max_{p \epsilon \mathcal{P}} \sum_{q:\{p,q\} \epsilon \mathcal{N}} B_{\{p,q\}}. \tag{2.7}$$

A similar graph structure is used in this thesis, but different methods are used to estimate edge weight values. A fast implementation of an algorithm for solving the problem is described by Boykov and Kolmogorov [8]. Chapter 4 describes and justifies the assignments of $R_p(A_p)$ and $B_{\{p,q\}}$ used in this thesis. The region term $R_p(A_p)$ is modified while using shape priors to use shape information for segmentation.

2.4 A example of graph cuts for image segmentation

This section presents an example of using graph cuts for image segmentation, based on Table 2.1 and work done by Boykov and Jolly [2]. Figure 2.6(a) shows a simple image consisting of 9 pixels. The top left pixel and the bottom right pixel are marked by the user as 'background' and 'object' respectively.

An undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$, is shown in Figure 2.6(b). Every pixel is a node and nodes are connected using the 8 pixel neighbourhood connectivity described in Section 2.2. Each edge $e \in \mathcal{E}$ is assigned a cost or weight, w_e. There are two special nodes called the source S and sink T terminals. The costs $R_p(A_p)$ and $B_{\{p,q\}}$ are chosen according to Equations 2.5 and 2.6. Every node is connected to the s and t terminals and costs are assigned to graph edges using Table 2.1. Note that the costs assigned to the user-marked pixels is high and are shown using thicker edges in Figure 2.6(b). A cut is a subset of edges $\mathrm{C} \subset \mathcal{E}$ such that the terminals become separated by $\mathcal{G}(\mathrm{C}) = \{\mathcal{V}, \mathcal{E} \backslash \mathrm{C}\}$. The cost of a cut is the sum of costs of the edges

$$\mid C \mid = \sum_{e \epsilon C} w_e. \tag{2.8}$$

A cut partitions the nodes in the graph and corresponds to a segmentation of the underlying image. A minimum cost cut generates a node partitioning that is optimal in terms of image properties that represent the t-weights and n-weights.

The cut that minimizes the cost is shown in Figure 2.6(c) using the dotted red line. The nodes are partitioned such that nodes that connect to the source terminal are on one side of the cut and the nodes that connect to the sink terminal are on the other side. The partitioning of the nodes segments the underlying image into 'object' and 'background'. The segmentation of Figure 2.6(a) using graph cuts is shown in Figure 2.6(d).

This thesis builds on the ideas of Boykov and Jolly [2] and explores different ways of assigning t-weights and n-weights. Segmentation of videos is investigated and changes in

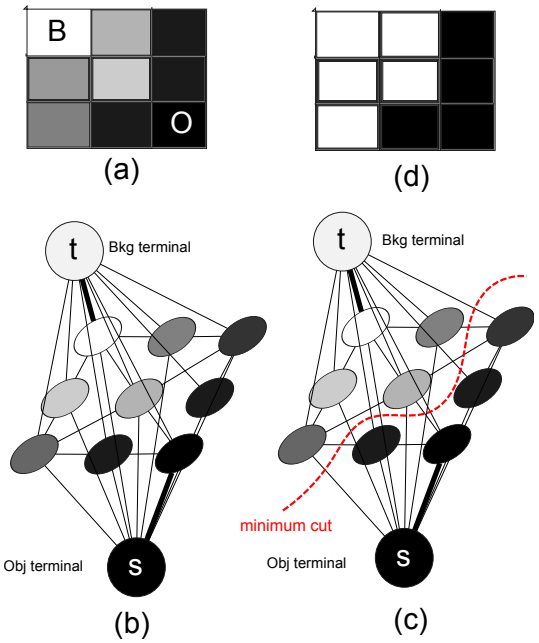


FIGURE 2.6: A simple 2D segmentation of 3 × 3 image. In (a) the top left pixel and bottom right pixel are seeds for background and object respectively. The cost is reflected by the edge thickness. The region cost and hard constraints define t-weights. The boundary cost defines n-weights [2]. The dotted red line in (c) represents the minimum cut and the resulting segmentation is shown in (d).

the cost function in Equation 2.4 are discussed. In the case of videos, the connectivity changes from 8 to 26 because adjacent pixels in the respective previous and next frames are also considered neighbours to every pixel.

Chapter 3

History and Literature Review

Graph cut optimisation has been used in many problems related to image processing and computer vision. Volumetric graph cuts were used by Campbell et al. [9] for automatic 3D object segmentation. Graphs cuts were used for multi-view stereo by Hernandez et al. [10], Vogiatzis et al. [11, 12] and Campbell et al. [13]. Greig et al. [14] used graph cuts for image restoration by estimating the maximum a posteriori (MAP) state of a binary image using the Ford-Fulkerson algorithm [15]. In this thesis, we only review the literature relevant to image and video segmentation.

There are many basic image segmentation techniques that can be combined and used in a graph cut formulation. Section 3.1 begins by reviewing the literature on these basic image segmentation techniques. It describes the work done on segmentation using graph cuts and discusses shape priors. Section 3.2 describes the general methods used in video segmentation with an emphasis on segmentation using graph cuts.

3.1 Image Segmentation

3.1.1 General Literature

Region and boundary information in images is the basis of the cost function used in this thesis. This section elaborates the use of region properties (colour, texture, intensity, etc.) and boundary properties (contours, edges, etc.) for image segmentation.

Comaniciu and Meer [16] use the mean-shift algorithm for colour image segmentation. Mean-shift is a simple, non-parametric technique for estimating the density gradient of the image. The image is automatically segmented into different classes based on user-marked pixels. A combination of mean-shift and graph cuts was used to segment images

in [16] and is discussed in the next section. Gray level images can also be segmented using their intensities as the colour component.

Permuter et al. [17] use Gaussian Mixture Models (GMMs) to model colour and texture information in images. GMMs are used to retrieve certain kinds of images from a database. In Permuter et al. [18] GMMs are also used for image classification and segmentation based on colour and texture. A background ('BKG') class is introduced and its parameters are estimated from a set of training images. This class is used to initialise the training of the texture models, and is also used to make classification decisions. Similar classes and GMMs are used in this thesis to model the user-marked pixels. Although [17, 18] do not use graph cuts, the use of GMMs based on color and texture is important in this work. These papers use GMMs based on different color spaces, like RGB and Lab, and texture feature filters based on the Discrete Cosine and Wavelet Transforms. They also compare the performance of GMMs to other statistical models like the 2D Hidden Markov Model (HMM), 2D Multi-resolution Hidden Markov Model (MHMM) and Classification and Regression Trees. For the cases considered it is shown that the average classification error rates are lowest for GMMs [17].

Texture descriptors and segmentation methods based on these descriptors are studied in [19]. Methods based on filter banks and local texture descriptors are investigated. Local methods like the SIFT descriptor, the Blobworld descriptor and Scale Space representations are used to segment images. The Brodatz Album mosiacs are used as data to test texture segmentation methods. The work produces a detailed analysis of various texture measures and evaluates their resulting segmentations.

Malik et al. [20] use contour and texture analysis to segment images. Images are partitioned into disjoint regions based on intensity and texture coherence. Contour information and texture differences in natural images is exploited by analysing the images using *textons*. Colour information is not used for segmentation. Evaluation of results is a challenging problem that is not addressed because there may not be a single correct segmentation. The work in [20] is based on the work done by Shi and Malik [21], who use normalized cuts to use global information of an image instead of limiting the segmentation to local image features. The cost function used is vaguely similar to the one in this thesis.

Martin et al. [22, 23] use local image measurements of brightness and texture to accurately localize boundaries. An explicit treatment of textures and intensity measures results in superior performance compared to conventional edge detection methods. Precision-recall curves are used to evaluate the performance of these methods.

A quantitative evaluation of the SE-MinCut segmentation algorithm is presented by

Estrada and Jepson [24]. They evaluate the performance of this algorithm using precision-recall curves and compare it to other methods like mean-shift, normalized cuts and local variation. The SE-MinCut algorithm is used to segment images from the Berkeley Segmentation Dataset [1]. Precision-recall curves are also used to compare and evaluate segmentation results in [25] and classification results in [26]. This motivates the use of precision-recall curves on images and ground truth from the Berkeley Segmentation Dataset [1] in this thesis.

3.1.2 Image segmentation using graph cuts

Boykov and Jolly [2] use interactive graph cuts for region- and boundary-based image segmentation. Globally optimal segmentation is achieved using the cost function described in Chapter 2 with hard constraints imposed by the user. The segmentation process is made interactive so that the segmentation desired by the user can be obtained. Applications of graph cuts for video and medical image segmentation are given.

Boykov and Jolly [27] use graph cuts for organ segmentation in medical images. This approach uses region and contour information from the 2D images and 3D volumes to segment them. The idea of segmenting multiple objects based on the background model is also discussed.

The problem of effective, interactive foreground/background segmentation is also investigated in GrabCut [28]. The results are compared to those obtained from approaches like Magic Wand, Intelligent Scissors, Bayes matting, Knockout 2 and Level Sets. Magic Wand, from Adobe Photoshop 7, starts with a user-specified point or region to find a set of connected pixels that fit the colour statistics to within a certain tolerance. The tolerance can be adjusted, but is difficult to specify correctly. Intelligent Scissors (also known as Live Wire or Magnetic Lasso) tracks the object boundary specified by the user. The algorithm computes the minimum cost contour but is likely to fail in the case of highly textured or untextured regions. After showing the limitations of Bayes matting, Knockout 2 and Level sets, [28] proposes using graph cuts for image segmentation. Colour data is modeled using GMMs to estimate foreground and background probabilities of each pixel.

The main aim of GrabCut [28] is to reduce user interaction by using techniques called "iterative estimation" and "incomplete labeling". GrabCut begins with the user drawing a rectangle around the desired object. Foreground statistics are estimated using the pixel data in the rectangle. A segmentation using graph cuts is done and the user is allowed to add background, foreground or matting information to improve the segmentation. Matting information is border information that is used to recover foreground colour

information, free of colour bleeding from the background. "Incomplete labeling" enables the user to only mark background pixels. There is no need to mark foreground pixels explicitly because of the rectangular bounding box provided by the user. "Iterative estimation" assigns provisional labels to some pixels (in the foreground) that can be retracted subsequently. Border matting is used to overcome the problem of blur and mixed pixels in the segmentation. Although a formal evaluation of the results is not performed, a visual inspection shows results better than other methods.

Several variations on GrabCut are implemented as a GIMP plug-in by [29]. GIMP is an image editing package and [29] explains the shortcomings of the existing segmentation tools in this package. Histograms and GMMs are used to model foreground and background pixels. It is observed that the process of calculating the GMM parameters is slow.

A combination of mean-shift and graph cuts is used in [30] to segment images. The image is mean-shifted and foreground and background seeds are clustered separately. The region properties are determined by the minimum distances from the unmarked pixels to foreground and background clusters. The boundary properties are estimated by the relationship between unmarked pixels and their neighbours. This approach was implemented in the course of this work but was considered ineffective as the graph cut algorithm should access the information masked by the mean-shifting process. GMMs or histograms are considered better options as they impose soft constraints and are more robust, based on [17].

Ratio regions are used for image segmentation by Cox and Zhong [31]. The segmented region has an exterior boundary cost and an interior benefit associated with it. It uses a similar cost function to the region and boundary cost function used in this thesis. It minimizes the cost

$$\mathcal{L} = \frac{cost(P)}{weight(P)} \tag{3.1}$$

where $cost(P)$ is the length of the segmentation boundary and $weight(P)$ is the weight imposed on the segmented region. Ratio regions minimize a cost function based on the ratio of the cost of the perimeter of the segmented region to the benefit assigned to its enclosed interior. This work is related to Active Contour Models [32] and similarities between the two are discussed. Active Contour Models (ACMs) are also known as *snakes*. Ratio cuts, based on ratio regions and the ratio cost function, are explored by Wang and Siskind [33]. The cost function in Equation 3.1 does not impose a size, shape, smoothness or boundary bias as it is based on the perimeter of the regions to be segmented. The ratio contour objective is also used by Wang et al. [34] to accurately extract boundaries in image. This approach guarantees a globally optimal solution without any bias on the area of the region or the length of the boundary.

The flow resulting from the max-flow calculation, as described in Chapter 2, is reused in dynamic graph cuts presented in [4]. The flow from a graph cut is used as an initialization for acquiring a segmentation dynamically and updating a previous segmentation. Dynamic graph cuts are used in image segmentation to make the system interactive. After one iteration of the algorithm, the user can reselect certain regions and foreground and background. Instead of recomputing the graph cut, the dynamic graph cut algorithm updates the solution from the previous instance until the desired segmentation is achieved. This solution update system makes dynamic graph cuts faster than the implementation of graph cuts in [8]. Dynamic graph cuts is fast and is used if multiple related graph cut solutions are required. Kohli and Torr [35] present a method for calculating uncertainty associated with graph cut solutions. The proposed algorithm, based on dynamic graph cuts, computes min-marginals for Markov Random Fields (MRFs) with a large number of latent variables.

In a graph cut formulation, the features and parameters are usually selected by the developer of the algorithm. Features are components in the GMM and parameters are variables that affect the segmentation. Parameters are like weights between foreground and background, and region and boundary properties in the cost function are explored. Parameter and feature selection can be a difficult task for any given image segmentation problem. The set of features and values of parameters that result in a good segmentation is difficult to determine. Peng and Veksler [36] try to develop an algorithm for automatic selection of the parameters of a graph cut. The weight given to parameters is controlled using a variable λ. The approach taken is empirical, based on given segmentation results for different values of λ. Various features like intensity, gradient, contour continuity and texture are investigated in this thesis.

Blake et al. [37] use an adaptive Gaussian Mixture Markov Random Field (GMMRF) for interactive image segmentation. A "pseudo-likelihood" is formed after analyzing the colour properties of an image, and this is used in a graph cut to find a segmentation. The pseudo-likelihood function is limited in its complexity compared to other models and is only approximately statistically correct.

3.1.3 Segmentation using graph cuts and shape priors

The graph cut method is a popular and powerful technique for image segmentation. It can be modified to fit problems where there is specific knowledge about the object to be segmented. For example, if the shape of the object to be segmented is known, then this information can be used to direct graph cuts to segment images accordingly.

Vicente [38] uses a natural assumption about the connectivity of objects to overcome the

shortcomings of graph cuts in segmenting elongated objects. An explicit connectivity prior is imposed on the segmentation. The user marks certain pixels that must be connected to the object being segmented, in addition to the pixels required to be foreground or background. The algorithm imposes this connectivity to get a detailed segmentation of elongated objects or thin parts of objects.

Lempitsky et al. [39] use a technique where the user draws a bounding box around the object to be segmented. This is an intuitive first step for the user. The bounding box not only excludes its exterior from consideration but also imposes a strong topological prior. This prevents the solution from shrinking, as discussed in [37]. The algorithm is driven towards a sufficiently 'tight' segmentation, which means that the segmented object should have parts sufficiently close to the edges of the bounding box. This work also defines the 'tightness' of shapes and globally optimizes a cost function similar to that given in Equation 2.4. Experiments are conducted and compared to the images used in GrabCut [28]. The algorithm is slower than GrabCut but it is more accurate.

PoseCut [6, 35] uses dynamic graph cuts to optimize a cost function based on Conditional Random Fields (CRFs) to simultaneously segment and estimate the pose of humans. A simply-articulated stickman model is used to ensure human-like segmentations. The distance transform of this stickman is used as a shape prior for segmentation. Region and boundary properties are represented by GMMs of pixel intensities and pose-specific stickman models respectively.

PoseCut is based on ObjCut [40]. ObjCut is based on a probabilistic approach which can deal with object deformation. Layered pictorial structures (LPS) are used as shape priors for segmentation. Pictorial structures are a combination of 2D patterns based on shape, appearance and spatial layout. ObjCut combines graph cut segmentation and object recognition techniques discussed in Felzenszwalb and Huttenlocher [41, 42]. The parameters of pictorial structures have to be estimated from the data and graph cuts are used to segment images. Likelihoods for parts are estimated using features and spatial locations of the parts. The desired configuration of parts of the object is given a lower cost than other unlikely configurations. Accurate object-specific segmentations are achieved by combining LPS and MRFs.

A star-shape segmentation prior is used for graph cut image segmentation in [43]. The star-shaped prior is used as a generic shape for all objects. In comparison to Equation 2.4, the cost function used in this case is

$$E(A) = \lambda \cdot \sum_{p \epsilon \mathcal{P}} R_p(A_p) + \sum_{\{p,q\} \epsilon \mathcal{N}} B_{\{p,q\}} \cdot \delta(A_p, A_q) + \sum_{p \epsilon \mathcal{P}} S_p(A_p), \tag{3.2}$$

where S_p is the shape prior. The shape prior is encoded using the distance transform of a learned shape. The shape prior tries to remove the shrinking bias of a graph cut segmentation and can be compared to other 'ballooning' terms. 'Ballooning' terms are used in [12] to inflate the segmented region since most graph cut methods tend to bias the result to small foreground objects. The inflation of the segmented region is used to accurately reconstruct thin protrusions and concavities in the 3D reconstruction problem. The value for the 'ballooning' term is set manually. The results using shape priors were promising but there were certain shortcomings. The major assumption in this work is that the center of the shape is known. The idea of using the star-shape prior for all objects gives rise to problems of shape alignment and of imposing the wrong shape prior.

Freedman and Zhang [44] incorporate level-set templates to introduce a shape energy into the overall cost function. The user is required to draw circles around the foreground and squares in the background, similar to the bounding box in [39]. The level-set templates are estimated by parameterizing the curve of the object boundary.

3.2 Video Segmentation

3.2.1 General Literature

Video segmentation is an extension to the image segmentation problem, as videos are a collection of images over time. Even though the following work does not use graph cuts for segmentation, the methods used are relevant to the methods developsed in this thesis. The basic video segmentation techniques are frame differencing, background prediction or estimation and tracking changes in pixels.

Heikkila et al. [45] use an algorithm based on Local Binary Patterns (LBP) to detect moving objects in video. It works on the assumption that the camera is stationary with a fixed focal length. The image is divided into blocks and each block is modeled with weighted LBP histograms. LBPs are based on the texture of the image blocks.

The motion in videos is classified using Bayesian networks by [46]. A dominant colour region is segmented in [47] using temporal variations in the dominant colour. This algorithm is tested on sports videos. Measures of evaluation of performance of video segmentation and tracking methods without ground-truth are introduced in [48]. These measures are based on the colour and motion along the boundary and temporal differences between the object and its neighbours. Even though this work is not directly related to graph cuts, the ideas used have influenced this thesis.

3.2.2 Video segmentation using graph cuts

Criminisi et al. [49] present an algorithm for the real time foreground/background segmentation in monocular video sequences. The algorithm uses Hidden Markov Models (HMMs) to model temporal changes and a spatial MRF to favour colour coherence. Spatial and temporal priors and likelihoods of colour and motion are used to get accurate results. The fusion of colour and motion for segmentation ensures the foreground being segmented even if it is similar in colour to the background.

Kolmogorov et al. [50] segment binocular stereo video using Layered Graph Cuts (LGC) and Layered Dynamic Programming (LDP). An extended 6-state space for foreground/ background separation, a colour-contrast model and the stereo-match likelihood are used to define the region and boundary measurements. The main contribution of their work is the fusion of stereo with colour and contrast, which results in good quality segmentation of temporal sequences without imposing any explicit temporal consistency between neighbouring frames.

Li et al. [51] present a system for cutting a moving object out of a video clip and inserting it into another video. It starts by performing a 3D graph cut, which pre-segments the video into foreground and background regions while preserving temporal coherence. The watershed transform is used for this pre-segmentation. The initial segmentation is refined locally by using a 2D graph cut on each frame, which utilizes the colour properties of the frame. Brush tools are provided to control the user boundary precisely, wherever needed. Coherent matting is used to smooth out the object boundary in a post-processing stage. Although this approach views the video as a 3D object, it requires a lot of interaction and can be cumbersome. The preprocessing, actual graph cut optimization and post-processing stages are slow. The approach of this thesis is loosely based on this work, but with many improvements.

Wang et al. [52] also suggest a min-cut based interactive system for efficiently extracting foreground objects from a video. A hierarchial mean-shift preprocessing stage is introduced to minimize the number of nodes. Alpha matting methods are used in 3D to segment video volumes. Colour and edge costs are used to set up the graph weights and 3D matting preserves spatiotemporal smoothness. The technique has advantages over the previous ones but the drawbacks are considerable: if the foreground is similar to the background, a minimum cut based approach has difficulties differentiating between the two. Also, the preprocessing stage, involving mean-shift, is not desirable as it does not provide the full information of the video sequence to the graph cut optimization. This thesis tries to build on these ideas and uses other techniques to overcome the drawbacks.

Chapter 4

Image Segmentation using graph cuts

This chapter describes the experiments done in image segmentation using graph cuts. The different ways of calculating region and boundary evidence are defined first and then shape priors are discussed. Results and interpretation concludes the chapter. The images used are from the Berkeley Segmentation Dataset [1]. Netlab [53] was used for training Gaussian Mixture Models (GMMs) using Expectation Maximization (EM).

4.1 Graph cut components

The algorithm used in this chapter works as follows: the user marks certain pixels as foreground and background. Measurements related to region and boundary properties are extracted from these pixels using statistical models. The region properties, based on colour and texture, specify the probability of a pixel being background or foreground. The boundary measurements provide the evidence of boundaries or edges at different locations in the image. The region and boundary evidence is fed into the graph cut formulation for a pixel grid with each pixel being a node in the graph. After the graph cut, the user can edit the result to reassign pixels that have not been correctly segmented. This iterative process can help in getting the desired segmentation.

According to the cost function (Equation 2.4), the parameters of a graph cut are the region and boundary property estimates. Shape priors are included in this cost function (Equation 3.2) to improve the segmentation and direct it towards specific shapes. This section covers the different ways of calculating the region and boundary costs and explains how the shape prior is implemented.

4.1.1 Region properties

The label weights of regions in images are based on colour and texture in the regions. Gaussian mixture models (GMMs) are used to model region properties in the pixels marked by the user. A foreground model and a background model is estimated. Each pixel in the image can be compared to these models and be given a class-conditional probability for its classification.

A one-dimensional Gaussian distribution has the form

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}(x-\mu)^2}, \tag{4.1}$$

where μ is the mean and σ is the standard deviation. A multidimensional Gaussian distribution has the form

$$P_j(x_t \mid \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{D/2} \mid \Sigma_j \mid^{1/2}} \; e^{-1/2 \; (x_t-\mu_j)^T \; \Sigma_j^{-1}(x_t-\mu_j)}, \tag{4.2}$$

where $P_j(x_t \mid \mu_j, \Sigma_j)$ is the Gaussian distribution for the j-th class, with a mean vector μ_j and a covariance matrix Σ_j. The dimension of the feature vector is D.

A GMM, with feature vectors X^i for data points uses M Gaussians to model data as follows:

$$P(X^i \mid \theta^i_{GMM}) = \prod_{t=1}^{T^i} \sum_{j=1}^{M} P_j(x_t \mid \mu_j, \Sigma_j), \tag{4.3}$$

where θ^i_{GMM} includes the mean, covariance and mixture weights of the distribution and $X^i = \{x_t, \; 1 \leq t \leq T_i\}$. The complete set of parameters is $\{P_j, \; \mu_j, \; \Sigma_j\}$. Usually the covariance matrices Σ_j are set to be diagonal as the features are assumed to be independent to reduce the size of the parameter space. The probability of both the foreground and background classes can be calculated using these GMM equations.

The logarithmic likelihood ratio is used to assign terminal graph edge weights once each pixel is compared with the GMMs. The logarithmic likelihood is defined as

$$Log\ Likelihood\ Ratio\ (llr_p) = log(\mathcal{K} \cdot p_f / p_b) \tag{4.4}$$

where llr_p is the log likelihood ratio of a pixel p, and p_f and p_b are the probabilities of the pixel belonging to the foreground and background GMMs respectively. The sensitivity of the final segmentation depends on $\mathcal{K}$, which is a weighting parameter between foreground and background. The probability of a pixel being classified as foreground increases as $\mathcal{K}$ increases and vice versa. The effect of $\mathcal{K}$ is analyzed using segmented images in Section 4.2.3.

TABLE 4.1: Assignment of edge weights in this thesis.

edge	weight (cost)	condition
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in \mathcal{N}$
$\{p,s\}$	$\lambda \cdot max(0, llr_p)$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	$min(llr_p)$	$p \in \mathcal{O}$
	$max(llr_p)$	$p \in \mathcal{B}$
$\{p,t\}$	$-\lambda \cdot min(0, llr_p)$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	$max(llr_p)$	$p \in \mathcal{O}$
	$min(llr_p)$	$p \in \mathcal{B}$

The weights corresponding to region properties are set using the estimated GMMs. Many features of the image can be combined using these statistical models. In this thesis, features like colour, intensity and texture are used. For grayscale images the raw pixel intensities, along with Gabor and MR8 filter responses are used. For colour images, RGB values, Luv values, Gabor filter responses and MR8 responses and their combinations are used. The edge weights in this thesis are set differently from those in [2]. Table 4.1 shows how the edge weights are assigned. This table can be compared to Table 2.1 where s and t are the source and sink nodes respectively.

The Luv colour space was used in GMMs with the RGB colour space to reduce effects from brightness in images. Texture measures like Gabor and MR8 filters improve the performance of the segmentation since they can differentiate between foreground and background of the same colour.

(a) Original image. (grayscale) (b) Log likelihood ratio image.

FIGURE 4.1: Original image and the logarithmic likelihood ratio of each pixel in the image based on foreground and backgroun GMMs. Intensity values of (a) are used to assign terminal costs in the graph cut formulation.

Examples of images and their logarithmic likelihood ratios are shown in Figures 4.1 and 4.2. The brighter regions in Figure 4.1(b) correspond to pixels that have more evidence of being foreground than background. The properties of the colour image in Figure 4.2, such as RGB values and Luv values, can be used for GMM classification. In

the same way as in Figure 4.1(b), the brighter regions in Figure 4.2(b) indicate pixels that have high evidence of being foreground. This likelihood map can be used to assign graph edge costs to the pixel in the image. As seen in Figure 4.2(b), many of the background pixels have probabilities that are similar under background and foreground hypotheses, especially on the edge of the object. GMMs give an approximate indication of classification that can be refined with graph cut optimization.

(a) Original image. (colour) (b) Log likelihood ratio image.

FIGURE 4.2: Original image and the logarithmic likelihood ratio of each pixel in the image based on GMMs of colour using RGB and Luv colour spaces.

4.1.2 Boundary properties

Boundary weights, corresponding to values $B_{\{p,q\}}$ in Table 4.1, are assigned based on the evidence of a boundary or an edge at any location in an image. The cost is low when there is evidence of an edge in the image at the corresponding location. This will encourage the segmentation to include that edge in the boundary and improve segmentation. This section describes the different methods that are used. It also introduces a novel edge model based on GMMs.

4.1.2.1 Edge detection method

Conventional edge detection methods can be used to find the evidence of boundaries in an image. Edge detection techniques like the Sobel, Prewitt or Canny [54] edge detectors can be used. The Canny edge detector is used in this thesis to find the edges. This detector first blurs the image and then finds the intensity gradient of the image. This is followed by a ridge-tracking stage that includes hysteresis thresholding. The output is a binary image where a 1 indicates a detected edge. There are many parameters in a Canny detector that can be adjusted to get different results. The distance transform of

the Canny output is used to assign costs $B_{\{p,q\}}$ such that boundary elements near the edges in the image should be assigned a low cost. The drawback of such a method is that it detects all edges in the image. For foreground/background segmentation, only the edges that separate object from the background are needed. These are more significant than the edges in the background or foreground that do not bound the object. Thus a strong edge that is unrelated to the object boundary can be detrimental to the performance of the algorithm.

Figures 4.3 and 4.4 show the performance of the edge detection technique. Figures 4.3(b) and 4.4(b) show the output of the Canny edge detector of the images in Figures 4.3(a) and 4.4(a). The outputs of the distance transforms are shown in Figures 4.3(c) and 4.4(c).

(a) Original image. (b) Canny edge detector output. (c) Distance transform of (b).

FIGURE 4.3: Assignment of boundary costs for (a) the original image. The edges detected by the Canny detector are shown in (b). The distance transform of (b) is shown in (c). A high value in (c) indicates that a large cost should be assigned to the boundary elements at the corresponding location.

(a) Original image. (b) Canny edge detector output. (c) Distance transform of (b).

FIGURE 4.4: Assignment of boundary costs for (a) the original image. The edges detected by the Canny detector are shown in (b). The distance transform of (b) is shown in (c). A high value in (c) indicates that a large cost should be assigned to the boundary elements at the corresponding location.

4.1.2.2 Gradient-based method

The derivative of an image can be used to provide evidence for edges. The gradient at each point in an image is a 2D vector with the partial horizontal and vertical derivatives as its components. The gradient vector can also be represented as a magnitude and an angle. If D_x and D_y are the derivatives in the x and y directions respectively, Equations 4.5 and 4.6 show the magnitude and angle of the gradient vector ∇D. This is a measure of the rate of change of intensity at a point in the image. The direction of the gradient vector at a point shows the direction of the largest increase in the intensity of the image while the magnitude of the gradient vector denotes the rate of change in that direction:

$$|\nabla D| = \sqrt{D_x^2 + D_y^2} \tag{4.5}$$

$$\Theta = \arctan(D_y/D_x). \tag{4.6}$$

The gradient magnitude can be used to assign costs to boundary elements in a graph cut formulation. Edge costs can be set in many ways. Two possibilities are to use:

$$w = e^{-\left(\frac{|\nabla D|^2}{\sigma^2}\right)}, \tag{4.7}$$

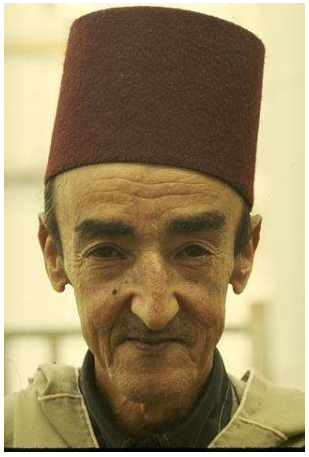
or

$$w = -\log(|\nabla D|^2 + 1), \tag{4.8}$$

where w is the edge cost, σ is a smoothing parameter that determines the scale of the edges, D_x and D_y are the derivatives in x and y directions and $|\nabla D|$ is the magnitude of the gradient, as defined in Equation 4.5. Figures 4.5 and 4.6 show weights for each location in the image based on the two different gradient-based methods. Figures 4.5(c) and 4.6(c) follow Equation 4.8 and Figures 4.5(b) and 4.6(b) are based on Equation 4.7.

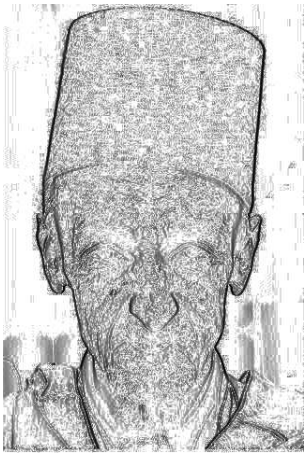
4.1.2.3 Edge model based on GMMs

Edge detection and gradient based methods are effective but have certain drawbacks. In Figures 4.5(b), 4.5(c), 4.6(b) and 4.6(c) it can be seen that all the edges in the images are detected. The desired object to be segmented in these images is the man in Figure 4.5(a) and the tiger in Figure 4.6(a). The gradient-based methods detect the evidence for all possible boundaries in the image. Ideally, the significant edges that separate the object from the background are needed. A method for detecting the edges of the object accurately, with no other edges in the background, is proposed in this section. This new edge model is based on GMMs of foreground and background derived from the user-marked



(a) Original image.

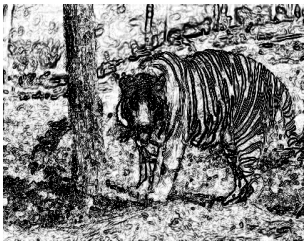
(b) Costs described by Equation 4.7 with $\sigma = 4$.



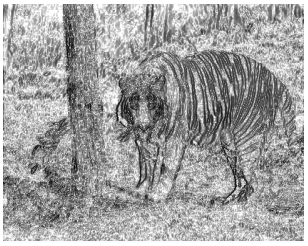
(c) Costs described by Equation 4.8.

FIGURE 4.5: Assignment of boundary weights for (a) the original image using (b) absolute value of the negative log of gradient magnitude in Equation 4.8 and (c) exponential of the gradient in Equation 4.7 with $\sigma = 4$. The value of σ is manually chosen.

(a) Original image.



(b) Costs described by Equation 4.7 with $\sigma = 10$.



(c) Costs described by Equation 4.8.

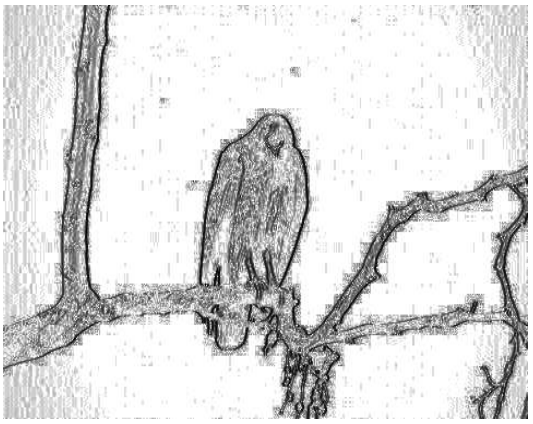
FIGURE 4.6: Assignment of boundary weights for (a) the original image using (b) absolute value of the negative log of gradient magnitude in Equation 4.8 and (c) exponential of the gradient in Equation 4.7 with $\sigma = 10$. The value of σ is manually chosen.

pixels.

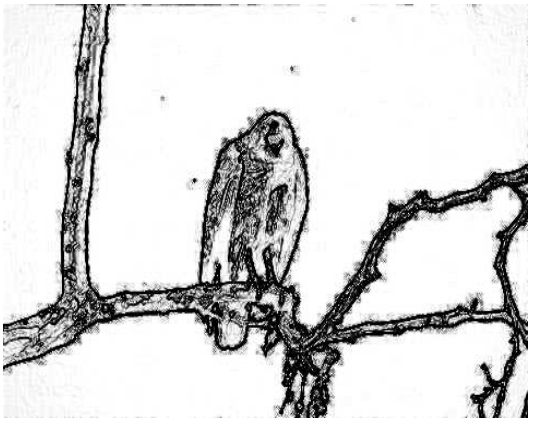
The user-marked pixels are used to train a edge model that is similar to the model used to assigned region costs. The edge model is trained using foreground and background seeds to model the pairwise appearances of pixels across the desired object boundary. This is done using pairs of pixels, one from the foreground and one from the background. This training models the joint relationship of foreground and background pixels. The pairs are randomly chosen to provide training data to the model. Once the model is trained, pairs of adjacent pixels in all directions in the image are compared to the edge GMM. Probabilities are assigned to each pair of pixels depending on how well they match this GMM. The edge costs are set inversely proportional to the GMM probabilities, assigning low costs for high probabilities. The joint GMM of foreground and background results in high probabilities at the boundary of objects and lower probabilities elsewhere, because the boundaries of the object closely match the model.

The new edge model based on GMMs is effective in three ways. Firstly, it is easy to implement and does not increase the execution time of the graph cut. Secondly, the user input is used for assigning boundary costs and differentiating between relevant and irrelevant edges. It provides a better edge detection measure, as shown in Figure 4.7, without any additional user input. This has not been done in any previous work. Thirdly, this method results in both the region and boundary properties being defined by GMMs. Thus all the components of the graph cut are costs derived from probability maps, which makes the whole system more robust than gradient-based and edge detection-based methods.

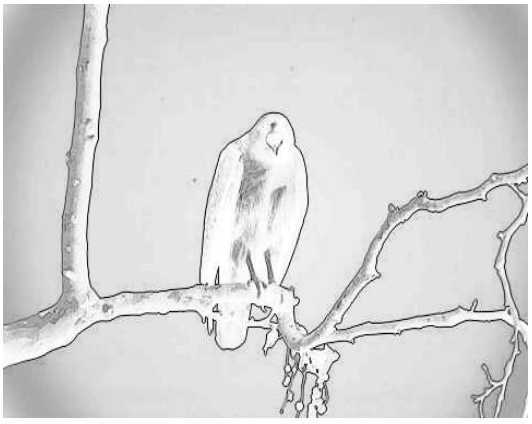
(a) Original image.



(b) Costs described by Equation 4.7 with $\sigma = 10$.



(c) Costs described by Equation 4.8.



(d) Edge model based on an edge GMM.

FIGURE 4.7: Assignment of boundary properties for (a) the original image using (b) edge costs defined by Equation 4.8, (c) edge costs defined by Equation 4.7 with $\sigma = 10$ and (d) the edge model based on an edge GMM. The value of σ is manually chosen.

Figure 4.7 compares the two gradient-based methods in Figures 4.7(b) and 4.7(c) to the edge model based on GMMs in Figure 4.7(d). The edge model is a probability map in contrast to the values obtained using gradient-based methods. The edge model can differentiate between relevant and irrelevant edges, as seen in Figure 4.7(d). The graph edge weights obtained, using the new model on the eagle image are more significant than those found using gradient-based methods.

4.2 Results and discussion

4.2.1 Defining a performance measure

Giving a numerical figure of merit to the result of a segmentation is difficult. The number or percentage of misclassified pixels is an intuitive measure of error. However, it may be incomplete as it does not convey information about false positives and negatives.

Precision-recall curves and F-scores have been proposed as performance measures [24]. These error measures can be defined as follows:

$$Precision = \frac{tp}{tp + fp}, \; Recall = \frac{tp}{tp + fn} \tag{4.9}$$

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}, \; F\text{-}score = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}. \tag{4.10}$$

where tp is the number of true positives, tn is the number of true negatives, fp is the number of false positives and fn is the number of false negatives. These measures require a ground-truth segmentation. The output of the segmentation is a binary image of the same size as the input image. A true positive is when a pixel is classified as foreground when the ground truth is foreground, a true negative is when a pixel is classified as background when the ground truth is background, a false positive is when a pixel is classified as foreground when the ground truth is background and a false negative is when a pixel is classified as background when the ground truth is foreground.

Accuracy is the ratio of misclassified pixels to the total number of pixels. Precision is low when there is significant over-segmentation. Low recall is a result of under-segmentation and indicates failure to retrieve relevant image information. For a perfect segmentation, precision and recall will both equal 1. F-score is a combination of precision and recall that provides a single measure for the system, which will be 1 for a perfect segmentation and very low (close to zero) for a bad segmentation.

4.2.2 Parameters $\mathcal{K}$ and λ

This section discusses the impact of the parameters $\mathcal{K}$ and λ on the segmentation. Equation 2.4 introduces λ and Equation 4.4 contains $\mathcal{K}$. The parameter $\mathcal{K}$ is a weighting parameter between foreground and background probabilities, and can be used to model prior information about the class of a pixel. A high value of $\mathcal{K}$ implies a greater probability of foreground pixels. The costs assigned to the foreground pixels are low when $\mathcal{K}$ is high and the segmentation includes a lot more foreground. Foreground and background are equally weighted when $\mathcal{K}$ is 1.

The parameter λ is used as a weighing factor between the region and boundary costs. When λ is 0, the cost function in Equation 2.4 only measures of boundary weights. Region and boundary costs are equally weighted when λ is 1. As λ increases, the cost associated with the region properties will increase. The coefficient λ is an important parameter in the graph cut segmentation.

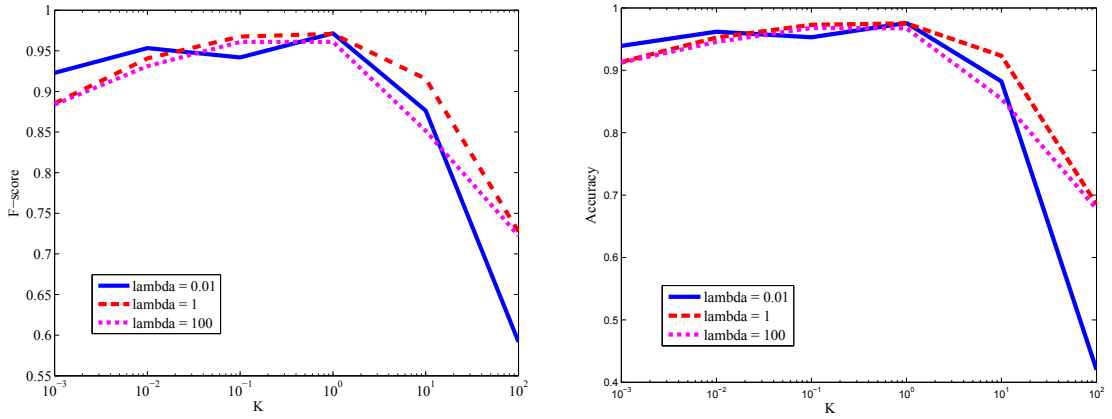
(a) $\mathcal{K} = 0.01$. (b) $\mathcal{K} = 1$.

FIGURE 4.8: The effect of changes in $\mathcal{K}$ on the final segmentation ($\lambda = 0.02$).

Figure 4.9 shows the changes in F-score and accuracy for different values of $\mathcal{K}$ using colour GMMs to assign region weights and gradient-based methods to assign boundary weights, while segmenting the image in Figure 4.8. Values of $\mathcal{K}$ are plotted on the x-axis and the curves for $\lambda = 0.01$, $\lambda = 1$ and $\lambda = 100$ are shown. F-score is a combination of precision and recall and accuracy is the ratio of misclassified pixels to the total number of pixels. Low and high $\mathcal{K}$ values result in under-segmentation and over-segmentation respectively, so accuracy and F-score will be low for very low or very high values of K. As seen in Figure 4.9, accuracy and F-score are highest when $\mathcal{K}$ is close to 1 and λ varies. High accuracy and F-score are indications of a good segmentation.

F-score is considered a better performance measure than accuracy. This is because accuracy is the percentage of misclassified pixels whereas F-score has information about precision and recall.

Figure 4.10 shows a family of precision-recall curves for different values of λ as $\mathcal{K}$ varies. For the ideal segmentation, precision and recall will be 1. As $\mathcal{K}$ increases, it is seen that the precision and recall values vary. For a low λ value ($\lambda = 0.01$) the recall is low, indicating in under-segmentation. Low precision and over-segmentation is observed when λ increases. The sensitivity of the segmentation depends on $\mathcal{K}$. When $\mathcal{K}$ is 0 all pixels are classified as background and when $\mathcal{K}$ is 1 all are classified as foreground. As the value of $\mathcal{K}$ increases the precision starts decreasing and the recall starts increasing. Figure 4.10



(a) F-score versus $\mathcal{K}$.

(b) Accuracy versus $\mathcal{K}$.

FIGURE 4.9: The effect of changes in $\mathcal{K}$ on F-score and accuracy of the segmentation.

shows that the best segmentation is achieved when $\mathcal{K}$ is 1 and the background and foreground probabilities are equally weighted. Over-segmentation is observed as $\mathcal{K}$ goes above 1. This means that some background pixels are classified as foreground because of the increase in $\mathcal{K}$. Then all the foreground pixels are in the segmentation (high recall), but some background pixels are also in the segmentation (low precision). The variable $\mathcal{K}$ is important because it weights foreground and background. If the graph cut output is an under-segmentation compared to the desired segmentation, $\mathcal{K}$ can be increased to get a better segmentation and vice versa.

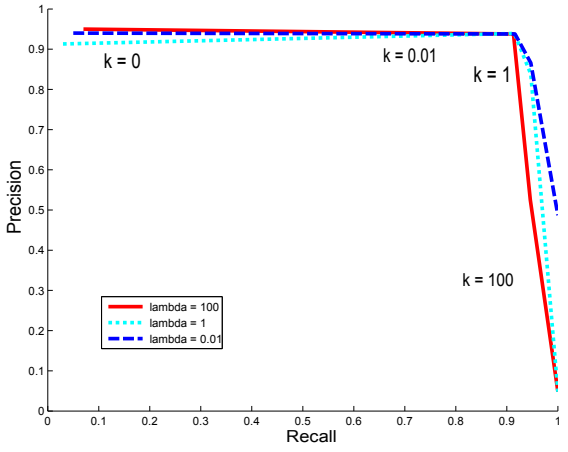


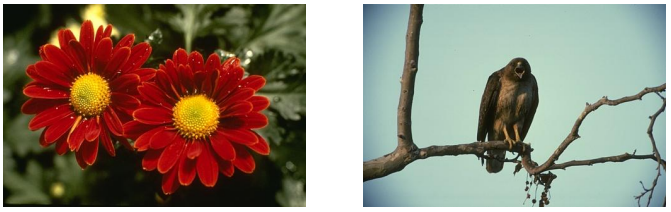
FIGURE 4.10: A family of precision-recall curves as $\mathcal{K}$ and λ vary.

Both λ and $\mathcal{K}$ can be varied to get the desired segmentation. The optimal values for these parameters may be different for different images.

4.2.3 Results and interpretation

F-score, accuracy, precision and recall are numerical indicators that qualify the performance of segmentation methods. Visual inspection can also be considered as a good measure to separate the methods that work best. This section discusses different features and methods used to define the region and boundary costs and how they affect the result of the segmentation. Experiments were conducted on images named 'birds', 'grass', 'plane', 'flowers' and 'eagle'. These images are shown in Figure 4.11. These images, both colour and grayscale, were taken from the Berkeley dataset [1]. The 'A' and 'F' that appear in the captions of images stand for accuracy and F-score respectively. Segmentation results for other images are included in Appendix A.

(a) 'birds' image. (b) 'grass' image. (c) 'plane' image.



(d) 'flowers' image. (e) 'eagle' image.

FIGURE 4.11: Colour images, 'birds', 'grass', 'plane', 'flowers' and 'eagle', used to evaluate the performance of graph cuts for image segmentation. Grayscale versions of the same images are used to evaluate the performance of grayscale algorithm variants.

4.2.3.1 Grayscale images

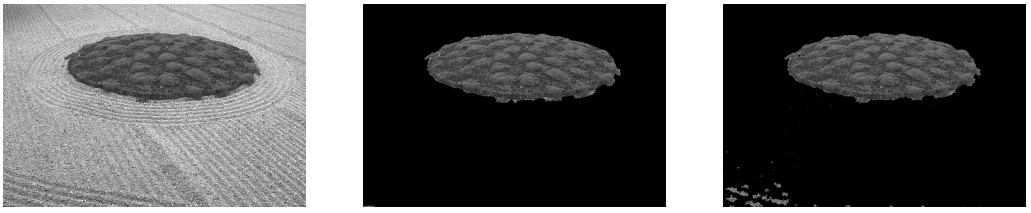
While testing the algorithm on grayscale images, intensity values and MR8 filter responses were used to assign region costs. Boundary costs were set using the edge-based, gradient-based and GMM-based methods described in Section 4.1.2.

As seen in Figure 4.12, a combination of MR8 filter responses and raw intensity values of the image for determining region costs results in over-segmentation of the image. Intensity values alone are sufficient to model the region properties in these images. In more textured images MR8 filter responses may prove useful. Boundary costs, calculated using gradient-based methods, are kept constant in order to study the effects of changes in the method for calculating region costs on the final segmentation.

(a) Original 'birds' image.

(b) Segmentation using intensity values.

(c) Segmentation using intensity values and MR8 filter responses.



(d) Original 'grass' image.

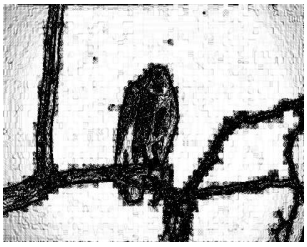
(e) Segmentation using intensity values.

(f) Segmentation using intensity values and MR8 filter responses.

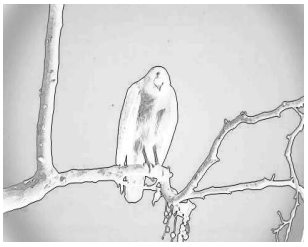
FIGURE 4.12: Segmentations based on different region costs. Boundary costs are kept constant and are calculated using gradient-based methods. (a) Original 'birds' image is segmented using (b) intensity values only and (c) intensity values and MR8 filter responses. (d) Original 'grass' image is segmented using (e) intensity values only and (f) intensity values and MR8 filter responses.

Experiments are performed to evaluate different boundary assignments (described in Section 4.1.2) and their impact on the final segmentation. Figures 4.13 and 4.14 show the results of these experiments. Figures 4.13(a) and 4.14(a) show the edges costs calculated using the gradient-based method in Equation 4.7. Since this method gave good segmentations, the GMM-based edge model proposed is compared to this gradient-based method. Figures 4.13(b) and 4.14(b) show the boundary costs according to the GMM-based model. The segmentations achieved from all these methods are displayed in Figures 4.13(c), 4.13(d), 4.13(e) and 4.13(f) and 4.14(c), 4.14(d), 4.14(e) and 4.14(f). The Canny edge detector with the distance transform is also used as a way to detect the evidence of boundaries. A GMM with 3 components is used got acquire the logarithmic likelihood ratios. The values of the parameters of λ and $\mathcal{K}$ are 0.1 and 1 for the 'eagle' image respectively and 0.1 and 0.1 for the 'plane' image. The values for λ and $\mathcal{K}$ can be manually set to get the desired segmentation.

The results show the performance of all methods and justify the use of the new edge model. It can be seen that the Canny edge detector with distance transform and the negative log of the gradient give under-segmentation and over-segmentation of the image respectively, in both images. The segmentations based on the GMM-based edge model are better as the model distinguishes between significant and insignificant edges based on user input. The 'A' (accuracy) and 'F' (F-score) indicators show this in both cases. These figures not only justify the need for a GMM-based edge model, but also prove its



(a) Edges calculated using gradient-based method.

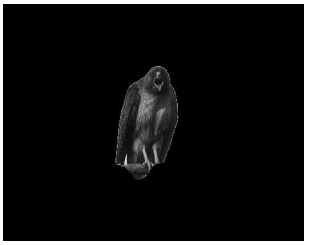


(b) Weights obtained for edge model based on GMM.

(c) Segmentation using the distance transform of the Canny edge detector to assign boundary costs (A = 0.823, F = 0.253).

(d) Segmentation using negative log of the gradient to set boundary costs (A = 0.688, F = 0.466).

(e) Segmentation using exponential of the gradient (A = 0.705, F = 0.411).



(f) Segmentation using GMM-based edge model (A = 0.860, F = 0.496).

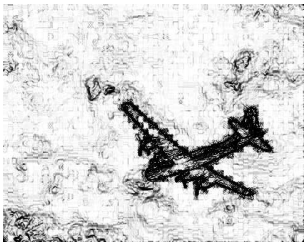
FIGURE 4.13: Different methods to set the boundary costs are evaluated. Gradient-based methods are used in (a) and a GMM-based edge model is used in (b). Segmentations achieved using boundary properties based on (c) the Canny edge detector, (d) negative log of gradient, (e) gradient-based methods and (f) GMM-based edge model are shown. From (b) and (f), it can be clearly seen why the proposed edge model works better than other methods.

superior performance to conventional methods.

4.2.3.2 Colour images

The experiments done using colour images use the methods discussed in Section 4.1.2 to assign weights to boundary elements. GMMs with the combination of the following features are used to assign the region costs:

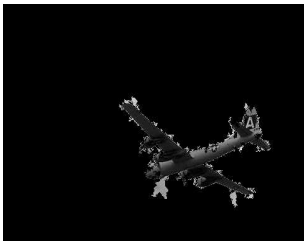
- R, G and B values of the image,
- G, (G-R) and (G-B) values of the image,
- L, u and v values of the image,
- G, (G-R), (G-B), L, u, v values and MR8 filter responses,
- R, G, B, L, u, v values and MR8 filter responses,
- G, (G-R), (G-B), L, u and v values, and



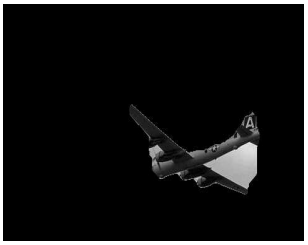
(a) Edges extracted using gradient-based method.

(b) Weights obtained for edge model based on GMM.

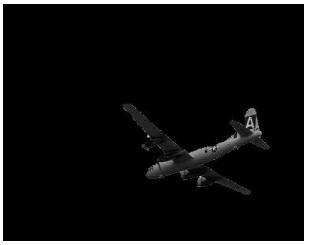
(c) Segmentation using the distance transform of the Canny edge detector to assign boundary costs (A = 0.989, F = 0.891).



(d) Segmentation using negative log of the gradient to set boundary costs (A = 0.982, F = 0.851).



(e) Segmentation using exponential of the gradient (A = 0.969, F = 0.756).



(f) Segmentation using GMM-based edge model (A = 0.993, F = 0.940).

FIGURE 4.14: Different methods to set the boundary costs are evaluated. Gradient-based methods are used in (a) and a GMM-based edge model is used in (b). Segmentations achieved using boundary properties based on (c) the Canny edge detector with distance transform, (d) negative log of gradient, (e) gradient-based methods and (f) GMM-based edge model are shown. From (b) and (f) it can be clearly seen that the proposed edge model works better than other methods.

- L, u, v values and MR8 filter responses.

Different colour spaces are used to model the colour information marked by the user. L, u and v values are used with R, G and B values to get the colour and the intensity content of the image. MR8 filter responses are used to obtain texture information. GMMs are built based on the user-marked pixels and then region-based costs are set based on these GMMs.

Figure 4.15 shows the segmentation of the 'birds' image using the different methods listed above. It can be seen that the use of MR8 filter responses results in over segmentation (Figures 4.15(c), 4.15(g) and 4.15(h)), and the colour features work well (Figures 4.15(b), 4.15(e), 4.15(d) and 4.15(f)) because the foreground is very different from the background. The best segmentation, according to performance measures and visual inspection, is achieved by using G, (G-R), (G-B), L, u, and v values. The performance of different ways of assigning region costs is tested and the boundary properties calculated using gradient-based methods are kept constant.

Foreground and background are well separated in the 'plane' image. The brightness is

(a) Original 'birds' image.

(b) Segmentation using G, (G-R) and (G-B) values (A = 0.994, F = 0.940).

(c) Segmentation using G, (G-R), (G-B), L, u, v values MR8 filter responses (A = 0.971, F = 0.771).

(d) Segmentation using L, u and v values (A = 0.991, F = 0.918).

(e) Segmentation using L, u, v, G, (G-R) and (G-B) values (A = 0.995, F = 0.951).

(f) Segmentation using R, G and B values (A = 0.994, F = 0.945).

(g) Segmentation using R, G, B, L, u, v values and MR8 filter responses (A = 0.971, F = 0.771).

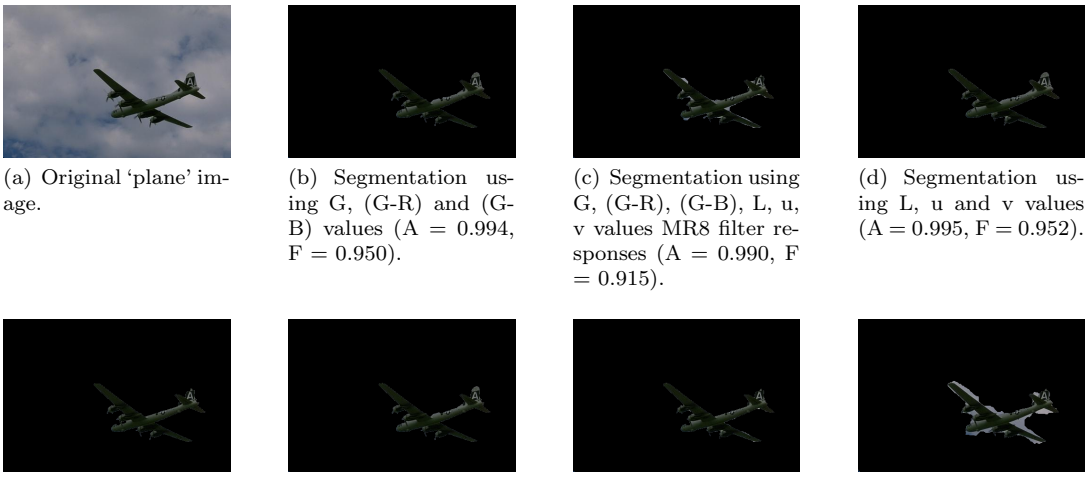
(h) Segmentation using L, u, v values and MR8 filter responses (A = 0.950, F = 0.662).

FIGURE 4.15: Segmentation of the 'birds' image using different region costs. Original image is shown in (a). Segmentations are shown for (b) G, (G-R) and (G-B) values, (c) G, (G-R), (G-B), L, u, v values and MR8 filter responses, (d) L, u and v values, (e) G, (G-R), (G-B), L, u and v values, (f) R, G and B values, (g) R, G, B, L, u, v values and MR8 filter responses and (h) L, u, v values and MR8 filter responses.

variable, so features like L, u and v values (Figure 4.16(d)) work better than R, G and B values (Figure 4.16(f)). MR8 filter responses work better when used with R, G, B, L, u and v values as seen in Figure 4.16(g). The best segmentation is achieved using L, u and v values.

The 'grass' image is highly textured and is brighter in some regions than in others. Features like L, u and v values and MR8 filter responses will therefore result in better segmentations (as in Figures 4.17(c) and 4.17(d)) than only colour-based features like R, G and B values (Figure 4.17(f)) and G, (G-R), (G-B), L, u and v values (Figure 4.17(e)). The best segmentation is achieved by using G, (G-R), (G-B), L, u, v values and MR8 filter responses in Figure 4.17(c).

The foreground in the 'eagle' image is spread out and includes some thin branches. The 'shrinking' bias of the graph cut formulation is tested as the required segmentation includes thin object elements: only part of the eagle is marked by the user as foreground, so segmenting the branches far away from the eagle depends on setting the region weights properly. The branches are a similar color to the eagle, so colour-based methods work well (Figures 4.18(b) and 4.18(d)). The L, u and v values give the best segmentation in Figure 4.18(d). In some images, the sky on the edges is wrongly segmented as foreground. This is because pixels at the image border are darker than at the center. In most segmentations the fine detail of the branches and the eagle are segmented properly. The reason for such good detail in the segmentation is the accuracy of GMMs based on colour and



(a) Original 'plane' image.

(b) Segmentation using G, (G-R) and (G-B) values (A = 0.994, F = 0.950).

(c) Segmentation using G, (G-R), (G-B), L, u, v values MR8 filter responses (A = 0.990, F = 0.915).

(d) Segmentation using L, u and v values (A = 0.995, F = 0.952).

(e) Segmentation using L, u, v, G, (G-R) and (G-B) values (A = 0.992, F = 0.926).

(f) Segmentation using R, G and B values (A = 0.994, F = 0.946).

(g) Segmentation using R, G, B, L, u, v values and MR8 filter responses (A = 0.992, F = 0.926).

(h) Segmentation using L, u, v values and MR8 filter responses (A = 0.977, F = 0.822).

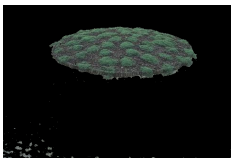
FIGURE 4.16: Segmentation of the 'plane' image using different region costs. Original image is shown in (a). Segmentations are shown for (b) G, (G-R) and (G-B) values, (c) G, (G-R), (G-B), L, u, v values and MR8 filter responses, (d) L, u and v values, (e) G, (G-R), (G-B), L, u and v values, (f) R, G and B values, (g) R, G, B, L, u, v values and MR8 filter responses and (h) L, u, v values and MR8 filter responses.

the difference between foreground and background in this image. The 'shrinking' bias may be seen in other images, where the background and foreground have similar colours and the foreground has fine detail.

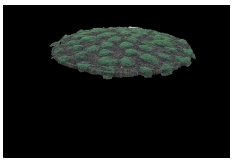
In the 'flowers' image in Figure 4.19(a) the foreground and background both contain yellow pixels. The image is highly textured and the foreground has fine detail around the petals of the flower. Many segmentation results include the yellow in the background as foreground, but good results are achieved using a combination of R, G, B values and L, u, v values (Figures 4.19(c), 4.19(e) and 4.19(g)). As predicted, many (R, G, B)-based GMMs do not perform well because of the yellow content in the background. More information and different features are required to model the data better when foreground and background are similar. The texture in the image also adversely affects the segmentation using G, (G-R) and (G-B) values in Figure 4.19(b). Using R, G, B, L, u, v values and MR8 filter responses as features in the GMM lead to the best segmentation in Figure 4.19(g).

It can be seen from these experiments that different methods and different subsets of features give good segmentations for different images. A good segmentation depends on the desired result of a user, so the features used for segmentation can be changed accordingly. A feature selection phase could perhaps be added to the segmentation process, choosing certain features to get good results depending on each image. Appendix A provides detailed Tables A.1 and A.2 of the precision, recall, F-score and accuracy of all the methods listed in this section.

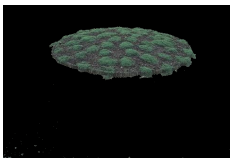
(a) Original 'grass' image.



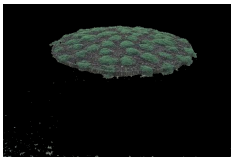
(b) Segmentation using G, (G-R) and (G-B) values (A = 0.988, F = 0.964).



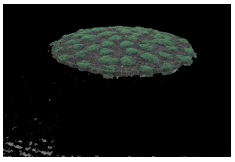
(c) Segmentation using G, (G-R), (G-B), L, u, v values MR8 filter responses (A = 0.995, F = 0.984).



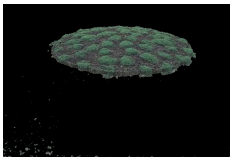
(d) Segmentation using L, u and v values (A = 0.994, F = 0.980).



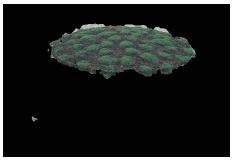
(e) Segmentation using L, u, v, G, (G-R) and (G-B) values (A = 0.991, F = 0.972).



(f) Segmentation using R, G and B values (A = 0.984, F = 0.954).



(g) Segmentation using R, G, B, L, u, v values and MR8 filter responses (A = 0.989, F = 0.968).



(h) Segmentation using L, u, v values and MR8 filter responses (A = 0.986, F = 0.956).

FIGURE 4.17: Segmentation of the 'grass' image using different region costs. Original image is shown in (a). Segmentations are shown for (b) G, (G-R) and (G-B) values, (c) G, (G-R), (G-B), L, u, v values and MR8 filter responses, (d) L, u and v values, (e) G, (G-R), (G-B), L, u and v values, (f) R, G and B values, (g) R, G, B, L, u, v values and MR8 filter responses and (h) L, u, v values and MR8 filter responses.

4.3 Shape Priors for image segmentation

The segmentation of images based on their region and boundary properties has been discussed in this chapter. Statistical models are trained using seeds marked by the user. These seeds provide clues to the desired segmentation. The segmentation of the image is done based on features like colour, texture and spatial edge evidence in the image.

4.3.1 Shape prior in the segmentation

In addition to features like colour and texture, a prior knowledge of the shape of the desired object is known in many cases. For example, in case of x-ray segmentation the shape of the bone may be known. This shape information can be used modify the edge costs and direct the segmentation towards the shape, as shown in Equation 3.2. A circular shape prior defined by the center and radius is used to demonstrate the use of the shape prior. The distance transform exterior to the shape prior is used to weight the region and boundary costs that are described in Sections 4.1.1 and 4.1.2. Costs in the interior of the shape prior are set to zero. The alignment of the shape prior to the object in the image is done using Powell's method of minimization.

Powell's method of minimization, also known as Powell's conjugate gradient descent

(a) Original 'eagle' image.

(b) Segmentation using G, (G-R) and (G-B) values (A = 0.966, F = 0.915).

(c) Segmentation using G, (G-R), (G-B), L, u, v values MR8 filter responses (A = 0.969, F = 0.923).

(d) Segmentation using L, u and v values (A = 0.974, F = 0.934).

(e) Segmentation using L, u, v, G, (G-R) and (G-B) values (A = 0.968, F = 0.922).

(f) Segmentation using R, G and B values (A = 0.966, F = 0.916).

(g) Segmentation using R, G, B, L, u, v values and MR8 filter responses (A = 0.969, F = 0.922).

(h) Segmentation using L, u, v values and MR8 filter responses (A = 0.968, F = 0.921).

FIGURE 4.18: Segmentation of the 'eagle' image using different region costs. Original image is shown in (a). Segmentations are shown for (b) G, (G-R) and (G-B) values, (c) G, (G-R), (G-B), L, u, v values and MR8 filter responses, (d) L, u and v values, (e) G, (G-R), (G-B), L, u and v values, (f) R, G and B values, (g) R, G, B, L, u, v values and MR8 filter responses and (h) L, u, v values and MR8 filter responses.

method, is an algorithm used to find the local minimum of a function of a vector variable without using derivatives. This method minimizes the function using a bi-directional search along each search variable component, in turn. In the segmentation problem, the function to be minimized is the cost of the graph cut. The variables are the center and radius of the circular shape prior. The co-ordinates used as an initial guess for Powell's method are the average co-ordinates of the seeds provided by the user. Powell's method accurately aligns the shape prior with the object in the image by minimizing the cost of the cut, over segmentations and shape parameters. The alignment optimizes the parameters of the shape (the radius and the co-ordinates of the center) while minimising the cost of the cut as an inner loop optimization. In this way the objective function is jointly minimised over both the shape prior parameters and the desired segmentation. The location of the seeds marked by the user is used as a starting point for Powell's method. Powell's method of minimization minimizes the cost of the cut using a bi-directional search along each search vector of variables, in turn. Powell's method accurately aligns the shape prior to the object and provides a segmentation that minimizes the cost. Powell's method is also used in Chapter 5 for video segmentation with shape priors.

Figure 4.20 shows the use of shape priors to segment faces and demonstrates their effect on the boundary costs. Figures 4.20(a) and 4.20(d) show two images from the Microsoft i2i dataset [3]. The output of the GMMs based on colour using the RGB and Luv colour spaces is shown in Figures 4.20(b) and 4.20(e). It is observed that some regions in the

(a) Original 'flowers' image.

(b) Segmentation using G, (G-R) and (G-B) values (A = 0.873, F = 0.862).

(c) Segmentation using G, (G-R), (G-B), L, u, v values MR8 filter responses (A = 0.975, F = 0.971).

(d) Segmentation using L, u and v values (A = 0.959, F = 0.953).

(e) Segmentation using L, u, v, G, (G-R) and (G-B) values (A = 0.973, F = 0.969).

(f) Segmentation using R, G and B values (A = 0.967, F = 0.961).

(g) Segmentation using R, G, B, L, u, v values and MR8 filter responses (A = 0.976, F = 0.972).

(h) Segmentation using L, u, v values and MR8 filter responses (A = 0.958, F = 0.952).

FIGURE 4.19: Segmentation of the 'flowers' image using different region costs. Original image is shown in (a). Segmentations are shown for (b) G, (G-R) and (G-B) values, (c) G, (G-R), (G-B), L, u, v values and MR8 filter responses, (d) L, u and v values, (e) G, (G-R), (G-B), L, u and v values, (f) R, G and B values, (g) R, G, B, L, u, v values and MR8 filter responses and (h) L, u, v values and MR8 filter responses.

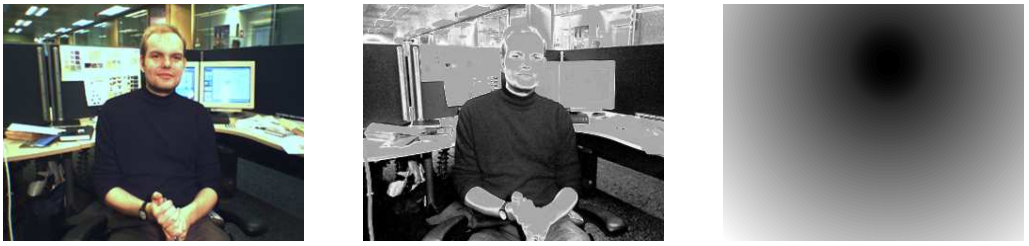
background are assigned probabilities similar to foreground regions as they match each other in colour. Some parts of the face, which is the desired object, are assigned a high probability of being background. The distance transforms of the correctly aligned shape prior are shown in Figures 4.20(c) and 4.20(f). The parameters of the circular shape prior, the radius and the center co-ordinates, are determined by Powell's method. The graph edge weights corresponding to region costs are set proportional to the distance transform from the correctly aligned shape. This penalises segmentations that contain foreground regions far from the shape prior.

Although a circular shape prior is used in this thesis, any other shape prior can also be used. The shape of the prior can depend on the shape of the object to be segmentation. Alignment can be done automatically using gradient descent methods like Powell's method. The ideas of using shape priors for video segmentation is discussed in Chapter 5. The segmentation results when using shape priors are compared to those without using shape priors in Section 4.3.2.

4.3.2 Results

The segmentation results of images given in Figure 4.20 are shown in Figure 4.21. Figures 4.21(a) and 4.20(d) show the original images. The distance transforms from the aligned shape priors are shown in Figures 4.21(d) and 4.20(e). The segmentations resulting from

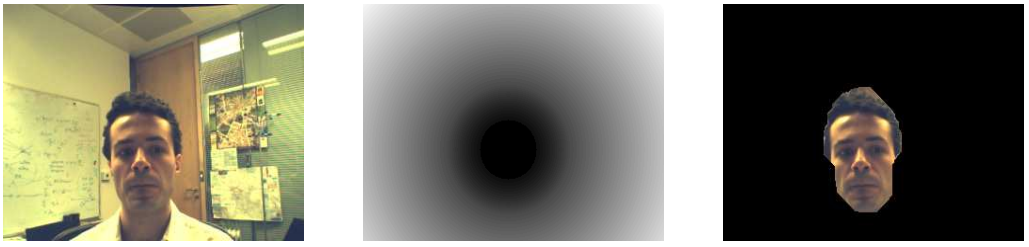
(a) Original image. (b) Output of GMMs. (c) Shape prior.



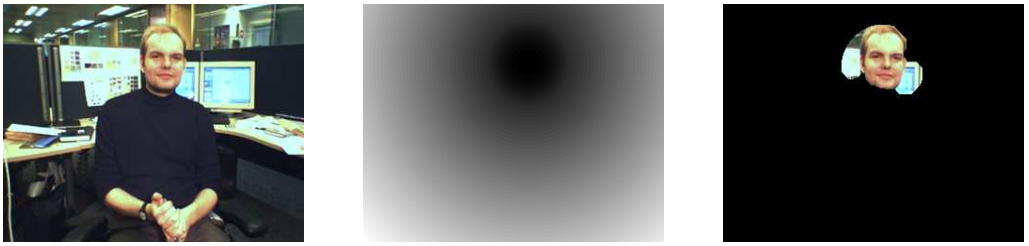
(d) Original image. (e) Output of GMMs. (f) Shape prior.

FIGURE 4.20: Image segmentation using shape priors and graph cuts. The figure shows (a) and (d) the original images, (b) and (e) probability estimation using GMMs, (c) and (f) distance transform from the shape prior aligned using Powell's method.

using shape priors to modify region and boundary costs are displayed in Figures 4.21(c) and 4.21(f). It is observed that the shape prior does have an effect on the segmentation and directs it towards the desired shape and location. Figures 4.22 and 4.23 compare the



(a) Original image. (b) Shape prior. (c) Segmentation using shape prior.



(d) Original image. (e) Shape prior. (f) Segmentation using shape prior.

FIGURE 4.21: Image segmentation using shape priors and graph cuts. The figure shows (a) and (d) the original images, (b) and (e) the distance transform from the shape prior aligned using Powell's method, and (c) and (f) the output of the graph cut.

segmentations with and without using shape priors. The segmentation in Figure 4.22(b)

does not classify some parts of the face as foreground, because those parts are of a different colour than the rest of the face. The results from the use of shape priors in Figure 4.22(c) are better than when using only GMMs for regions and edges. The distance transform from the shape prior reduces the probability of pixels far away from the shape as being classified foreground. The effectiveness of shape priors can also be seen in Figure 4.23(c). The output when using the shape prior with GMMs for regions and edges is better than the segmentation using only GMMs for regions and edges.

(a) Original image. (b) GMMs and edges. (c) Shape priors.

FIGURE 4.22: Comparison of segmentation methods. The original image shown in (a) with its segmentations using GMMs and edges in (b) and GMMs, edges and shape priors in (c).

(a) Original image. (b) GMMs for regions and edges. (c) Shape priors.

FIGURE 4.23: Comparison of segmentation methods. The original image shown in (a) with its segmentations using GMMs for regions and edges in (b) and GMMs for regions, edges and shape priors in (c).

While the graph cut solution is globally optimal, the gradient descent over shape parameters is only locally optimal. A good starting estimate of the shape parameters is therefore necessary. The starting estimate is derived using the locations of the seeds marked as foreground.

Image segmentation using shape priors to modify region costs is a powerful way to improve segmentation. The problem of aligning the shape to the object can be solved using Powell's method of minimization. Different shape priors can be chosen depending on the shape of the object. A way of using shape priors for video segmentation is investigated in Chapter 5.

Shape priors for image segmentation can be very effective when the shape of the object

is known. A possible application for graph cuts and shape priors for segmentation can be found in medical imaging. The shapes of the objects in x-rays and MRI images are known or can be specified before segmentation. This strong shape prior can be used to segment x-rays and MRI images accurately. Segmentation using shape priors will not only make use of the intensity values in the image but also impose a shape cost on the segmentation.

Chapter 5

Video Segmentation using graph cuts

This chapter discusses the problem of video segmentation. Videos are a sequence of images over time. Different segmentation methods based on graph cuts and the idea of finding a globally optimal solution are explored. A video sequence has a lot of inherent structure and segmentation techniques can be guided to use this information. There is a considerable amount of work done on the problem of video segmentation but the graph cut based methods are simple yet powerful. Most of the work based on graph cuts requires a lot of user information and constant input from the user. This not only increases the time taken for the segmentation, but also does not use the complete information from the video.

Visual inspection is used for checking the accuracy or adequacy of the segmentation. This chapter is divided into sections that describe the methods used and show some results. The experiments conducted, segmentations achieved, the interpretation of results, and directions for future research are discussed.

Before the different methods of video segmentation can be discussed, a new video dataset needs to be introduced. A short video was recorded to test the ideas that are discussed in this chapter. The purpose of this simple video sequence is to provide good test for segmentation methods. The video sequence consists of 15 frames, 640×480 and captures the motion of a tennis ball moving from right to left. The motion is approximately linear and the tennis ball is a completely different colour from the background. This new video sequence is referred to as the 'tennis ball' video and some frames are displayed in Figure 5.1.

The methods are also tested on the videos called 'Antonio', 'Geoff' and 'MS' that are part of the Microsoft i2i dataset. Antonio contains 200 frames, Geoff has 48 frames and MS

(a) Frame 1. (b) Frame 3. (c) Frame 6.

(d) Frame 9. (e) Frame 12. (f) Frame 15.

FIGURE 5.1: Frames from the 'tennis ball' video sequence.

has 325 frames. All video sequences are 320×240 images. All three videos are stereo, but for the purposes of this thesis only one of the views is used. Figures 5.2, 5.3 and 5.4 show some sample frames from these video sequences.

(a) Frame 15. (b) Frame 40. (c) Frame 42.

FIGURE 5.2: Sample frames from the 'Antonio' video sequence of the Microsoft i2i [3] dataset.

(a) Frame 18. (b) Frame 41. (c) Frame 58.

FIGURE 5.3: Sample frames from the 'MS' video sequence of the Microsoft i2i [3] dataset.

(a) Frame 5. (b) Frame 14. (c) Frame 47.

FIGURE 5.4: Sample frames from the 'Geoff' video sequence of the Microsoft i2i [3] dataset.

5.1 Individual frame-wise segmentation

The idea behind this method is that a video sequence is a collection of images, known as frames, and segmenting each frame will in turn segment the video. All the methods discussed in Chapter 4 can be used for each individual image in the sequence. An accurate segmentation for each image can be achieved using those methods.

Every image can be individually initialized by selecting 'foreground' and 'background' seed pixels and a segmentation can be found. A cost function based on region and boundary properties can be used.

Even though this may be considered a good method, not all the information in the video is used: every frame is treated as if it is not related to other frames, which is not the case. In a video, each frame is related to the next and the previous frames through time. Thus each pixel is related to the pixels around it in the same frame (intra-frame connectivity) and to the adjacent pixels in the previous and next frames (inter-frame connectivity). Even though individual frame segmentation is viable, it was discounted because it does not make use of all the information in the video. The video is viewed as a 3D object and segmentation methods are proposed in Section 5.2.

5.2 Video as a 3D object

A video is a set of frames related to each other through time. Thus any video sequence can be viewed as a 3D spatiotemporal object, with the first two dimensions being the image dimensions and the third dimension being time. The connectivity of the nodes in the graph can be described in this way. Instead of having an 8-pixel neighbourhood, as in the case of images, a 26-pixel neighbourhood is used.

Figure 5.5 shows the connectivity that is used for videos. The figure shows the connectivity of a sample pixel, the center pixel in the frame at time t. The 8 intra-frame connections in frame t are shown with cyan edges. The 9 connections between t-*1* and t, and the 9

connections between t and $t+1$ are shown using blue edges. The terminal connections between each pixel and the source node s and sink node t are assumed to be present but are not shown. This figure shows the connectivity of a sample pixel and all pixels are connected in an analogous way.

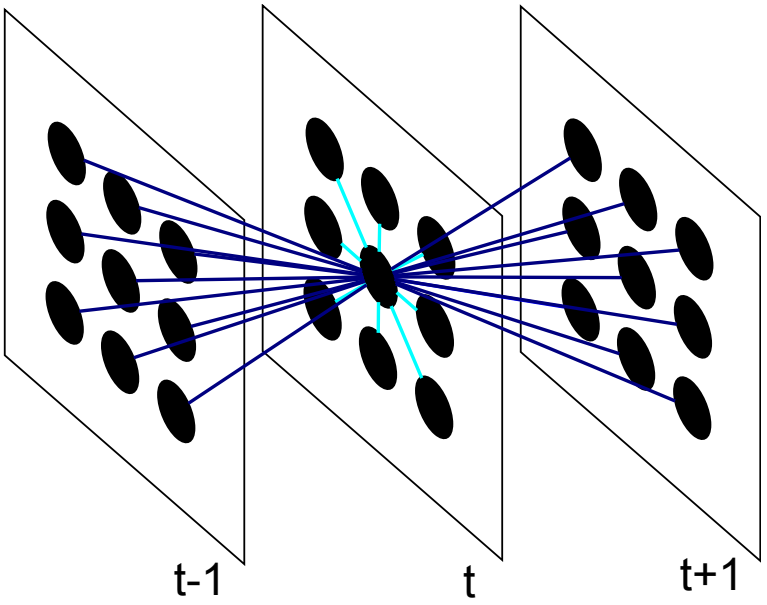


FIGURE 5.5: The 26-pixel neighbourhood connectivity of pixels in a video. The figure shows a connectivity of the center pixel in the frame at time t. Intra-frame connections are cyan and inter-frame connections are blue. Other pixels are connected in an analogous way.

The connectivity of the pixels described can be used with the region and boundary properties of all the frames to set up a 3D graph cut formulation, where each pixel in each frame is a node. This not only ensures that all the information in the video sequence is used, but also finds a globally optimal solution. The foreground and background GMMs are estimated based on annotations in the first frame. All the frames can be compared to these GMMs and probabilities of pixels being foreground or background are assigned. Regions and boundary properties can be derived using the GMMs and edge detection methods.

5.3 Segmentation without shape priors

The region and boundary based cost function used for image segmentation is used for video segmentation without shape priors. The user selects seeds using the first frame in the video sequence. A 3D graph using the pixel connectivity described in Section 5.2 is generated. GMMs based on colour are trained using these seeds. Each pixel in the video sequence is compared to these GMMs and logarithmic likelihood ratios are used

to assign 'foreground' and 'background' terminal weights for regions. The weights are assigned using the same techniques that are discussed in Sections 4.1.1 and 4.1.2. The boundary weights are defined using edge detection methods (spatial direction) and frame subtraction (temporal direction).

The region properties in the graph cut are defined using GMMs based on colour. R, G, B values and L, u, v values are used set the terminal weights. Figures 5.6, 5.7, 5.8 and 5.9 show the original frames in the sequence (subfigures (a)), probability maps based on GMMs (subfigures (b)) and the output of the graph cut (subfigures (c)). The 'tennis ball' video sequence is a simple dataset, but still the segmentation has some errors as seen in Figures 5.8(c) and 5.9(c). Some sections of the background are wrongly classified as foreground.

(a) Frame 2. (b) Logarithmic likelihood ratios. (c) 3D graph cut output.

FIGURE 5.6: Region properties using GMMs in 3D graph cuts. The (a) original frame 2 is segmented using graph cuts in (c) based on probability maps in (b).

(a) Frame 4. (b) Logarithmic likelihood ratios. (c) 3D graph cut output.

FIGURE 5.7: Region properties using GMMs in 3D graph cuts. The (a) original frame 4 is segmented using graph cuts in (c) based on probability maps in (b).

Boundary evidence can be determined in different ways, but now spatial and temporal boundaries are treated differently. Spatial boundaries are related to the intra-frame connectivity between pixels and temporal boundaries correspond to the inter-frame connectivity. The methods described while segmenting images can be used for videos. Finding edges in the spatial directions can be done using gradient based or conventional edge detection methods on each frame in the sequence.

Motion in the video is used for finding boundary evidence in the temporal directions.

(a) Frame 10. (b) Logarithmic likelihood ratios. (c) 3D graph cut output.

FIGURE 5.8: Region properties using GMMs in 3D graph cuts. The (a) original frame 10 is segmented using graph cuts in (c) based based on probability maps in (b).

(a) Frame 13. (b) Logarithmic likelihood ratios. (c) 3D graph cut output.

FIGURE 5.9: Region properties using GMMs in 3D graph cuts. The (a) original frame 15 is segmented using graph cuts in (c) based based on probability maps in (b).

When the background is reasonably stationary, these costs can be specified using motion in the video sequence extracted using frame subtraction. Subtracting a given frame from its previous one will show the changes between the frames provides evidence for an object boundary in the temporal direction. The absolute value of frame difference is used to set the graph edge weights in the temporal direction. So if the object to be segmented is moving in the video, its edges are accurately detected using motion.

Similar experiments are conducted on the videos from the i2i [3] dataset. The region costs are derived from the logarithmic likelihood ratios from colour GMMs. The boundary evidence in the spatial direction is extracted using the gradient of each frame in the video sequence. The costs of the intra-frame connected pixels are set using the boundary evidence in the spatial direction. In the temporal direction, frame subtraction is used to find the evidence of boundaries. The motion in the frames is used to set costs in the temporal direction. Figure 5.11 show three sample images from the 'Antonio' video sequence and the evidence of motion in those frames is shown using frame subtraction. Figures 5.7, 5.8 and 5.9 show the segmentation of the tennis ball video sequence using GMMs and edge detection techniques. Figure 5.12 shows the segmentation of a sample frame from 'Antonio' using colour GMMs and the motion of the frame calculated by using frame subtraction. The segmentation classifies a lot of the background as foreground and hence is not accurate. The desired object is the face of the person. But because

(a) Frame 1. (b) Frame 3. (c) Frame 6.

(d) Frame 9. (e) Frame 12. (f) Frame 15.

FIGURE 5.10: Motion in the frames using frame subtraction.

(a) Frame 15. (b) Frame 40. (c) Frame 42.

(d) Motion from frame 15. (e) Motion from frame 40. (f) Motion from frame 42.

FIGURE 5.11: Motion in the frames using frame subtraction. The video sequence 'Antonio' from the i2i dataset [3] is used.

some background pixels have a similar colour to the face, they are wrongly classified as foreground. A detailed analysis of the segmentations achieved using different methods is provided in Section 5.5.

5.4 Segmentation with shape priors

The motivation for using shape priors is seen from the results for image segmentation in the previous section. The video is not accurately segmented if the background and

(a) Frame 5. (b) GMM probabilites. (c) Segmented output.

FIGURE 5.12: Segmentation without shape priors of the 'Antonio' video sequence. The original frame (a) is segmented using the logarithmic likelihood ratio derived from the GMM probabilities in (b) and motion information shown in Figure 5.11. The output of the 3D graph cut is shown in (c).

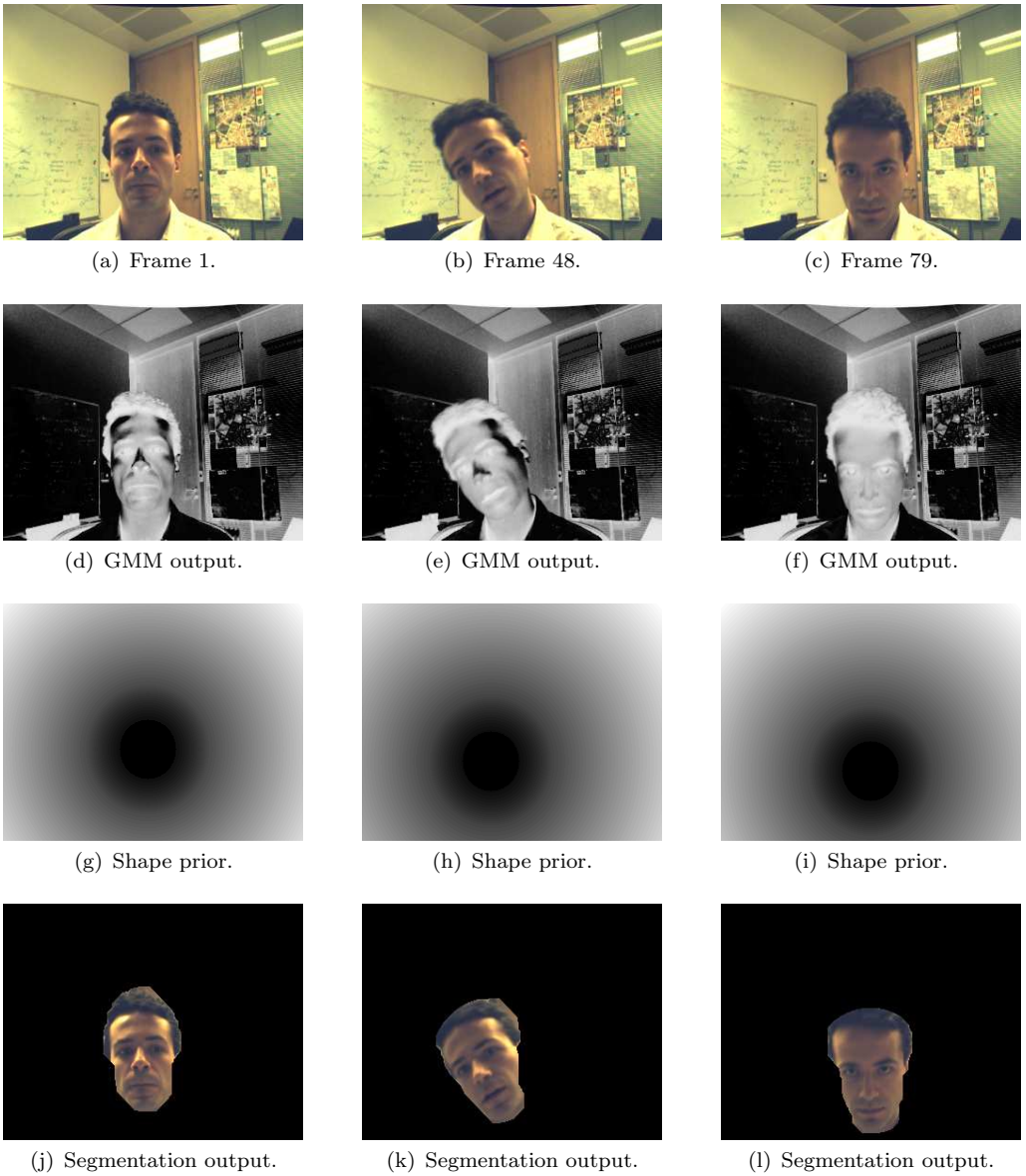
foreground pixels are similar in colour. Some background pixels will be wrongly classified as foreground because of the similarity in colour. The motion in the video sequence is used to detect the evidence of temporal boundaries. The segmentation output will be affected if the object to be segmented is stationary and the background is not stationary. Thus some prior knowledge of the location and shape of the object to be segmented will improve the segmentation. In this section, the use of shape priors for video segmentation is discussed.

Shape priors are used in addition to the colour GMMs and edge information. The first frame is used to train the GMMs based on RGB and Luv color spaces. The temporal boundary information is acquired using the motion in the video and the spatial boundary information is extracted using the gradient of each frame in the sequence. A circular shape prior is defined using its center and radius. Thus the shape prior has three parameters, the x and y co-ordinates of the center and the radius of the circle. The shape prior is aligned to each frame using Powell's method to give the minimum cost. An additional proximity term is added to the cost function to penalize discontinuity in the segmentation which essentially constitutes a random walk model for the object dynamic. This ensures that the alignment of shape priors for each image is sufficiently close together to maintain continuity of the object between frames. The proximity term is calculated using the distance between two shape priors in consecutive frames, using the parameters of the shape prior. The graph cut is performed on the spatiotemporal 3D object and each pixel is assigned to 'foreground' or 'background'. Powell's method, described in Chapter 4 is used to accurately align the shape prior with the object in the frames by minimizing the cost of the cut, over different segmentations and shape parameters.

The cost function for video segmentation using shape priors is

$$E(A) = \lambda \cdot \sum_{p \epsilon \mathcal{P}} (R_p(A_p) + S_p(A_p)) + \sum_{\{p,q\} \epsilon \mathcal{N}} (B_{\{p,q\}} + P_{\{p,q\}}) \cdot \delta(A_p, A_q) \qquad (5.1)$$

where R_p and $B_{\{p,q\}}$ are the region and boundary terms and S_p is the shape prior term parameterised as described. The shape prior term is calculated from the distance transform from the proposed shape. The shape prior term is used to modify the region costs. Thus the pixels far away from the shape prior have a higher probability of being classified as 'background'. The proximity term $P_{\{p,q\}}$ is included to penalize the distance between shape priors in consecutive frames.



(a) Frame 1. (b) Frame 48. (c) Frame 79.

(d) GMM output. (e) GMM output. (f) GMM output.

(g) Shape prior. (h) Shape prior. (i) Shape prior.

(j) Segmentation output. (k) Segmentation output. (l) Segmentation output.

FIGURE 5.13: Video segmentation of 'Antonio' video sequence using shape priors. The first row contains the original frames (a-c). The output of the GMMs (d-f) are shown in the second row. The distance transform from the aligned shape priors (g-i) is shown in the third row. The segmentations using shape priors (j-l) are shown in the final row.

Figure 5.13 shows results for video segmentation using the circular shape prior. The first

row contains three frames from the video sequence. The second row shows the logarithmic likelihood ratios of the images in the top row based on region GMMs trained on the face. The aligned shape priors are shown in the third row of images. Powell's method simultaneously minimizes the shape prior parameters for all frames. The segmentation of the three frames is displayed in the last row. The frames are chosen to contain different orientations of the face. It can be seen that the face is accurately segmented using the circular shape prior even if the face is rotated and translated. The optimal alignment of the shape prior also changes according to the position of the face in the different frames.

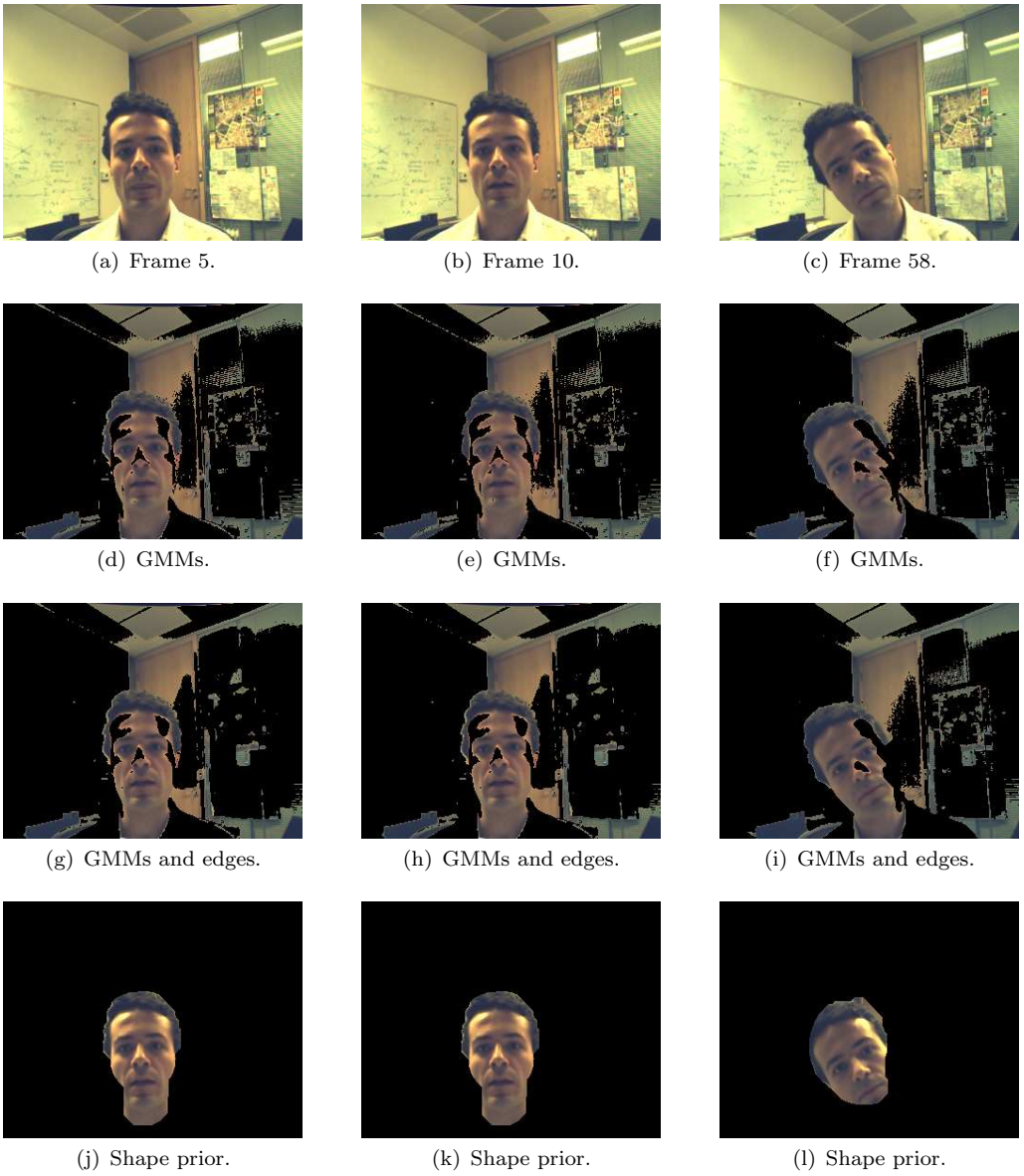
5.5 Results

This section compares segmentation using shape priors to segmentation using just GMMs and edge detection methods. Segmentation methods are tested using data that has background similar to foreground, the object changes orientation, and the background is not stationary. It shows the advantage of using a shape prior in segmentation. Segmentations using GMMs only, GMMs and edge detection, and GMMs and edge detection with shape priors are compared for using video sequences from the Microsoft i2i dataset [3].

Figures 5.14, 5.15 and 5.16 are organized in the same way by displaying different methods in different rows. The first row shows the original frames in the sequence. The segmentations of those frames using only colour-based GMMs are shown in the second row. The third row displays segmentations using GMMs and edge detection methods. The segmentations from the shape prior, with GMMs and edge detection, are shown in the final row. Circular shape priors of different radii and centers are used.

Figure 5.14 shows results for the 'Antonio' video sequence. The performance of different methods is tested when the background and foreground are similar and the face changes orientation. Figures 5.14(k) and 5.14(l) show that shape priors provide accurate segmentations even if the orientation of the object changes. The face has been tilted to the side, but is accurately segmented using shape priors while other methods fail. It is observed that using only GMMs, as in Figures 5.14(d), 5.14(e) and 5.14(f), results in many pixels being wrongly classified because the background and foreground have similar colours. GMMs classify some background pixels as foreground because of colour and texture similarities between the two classes. Edge detection methods are inaccurate as they detect all the boundaries in the image. It is desired the only detect the boundaries that separate the foreground and the background.

Figure 5.15 shows the 'Geoff' video sequence and segmentations using the different methods. 'Geoff' is a difficult sequence to segment because the background is not stationary. The frame differencing motion estimates in the video are affected and hence the temporal



(a) Frame 5. (b) Frame 10. (c) Frame 58.

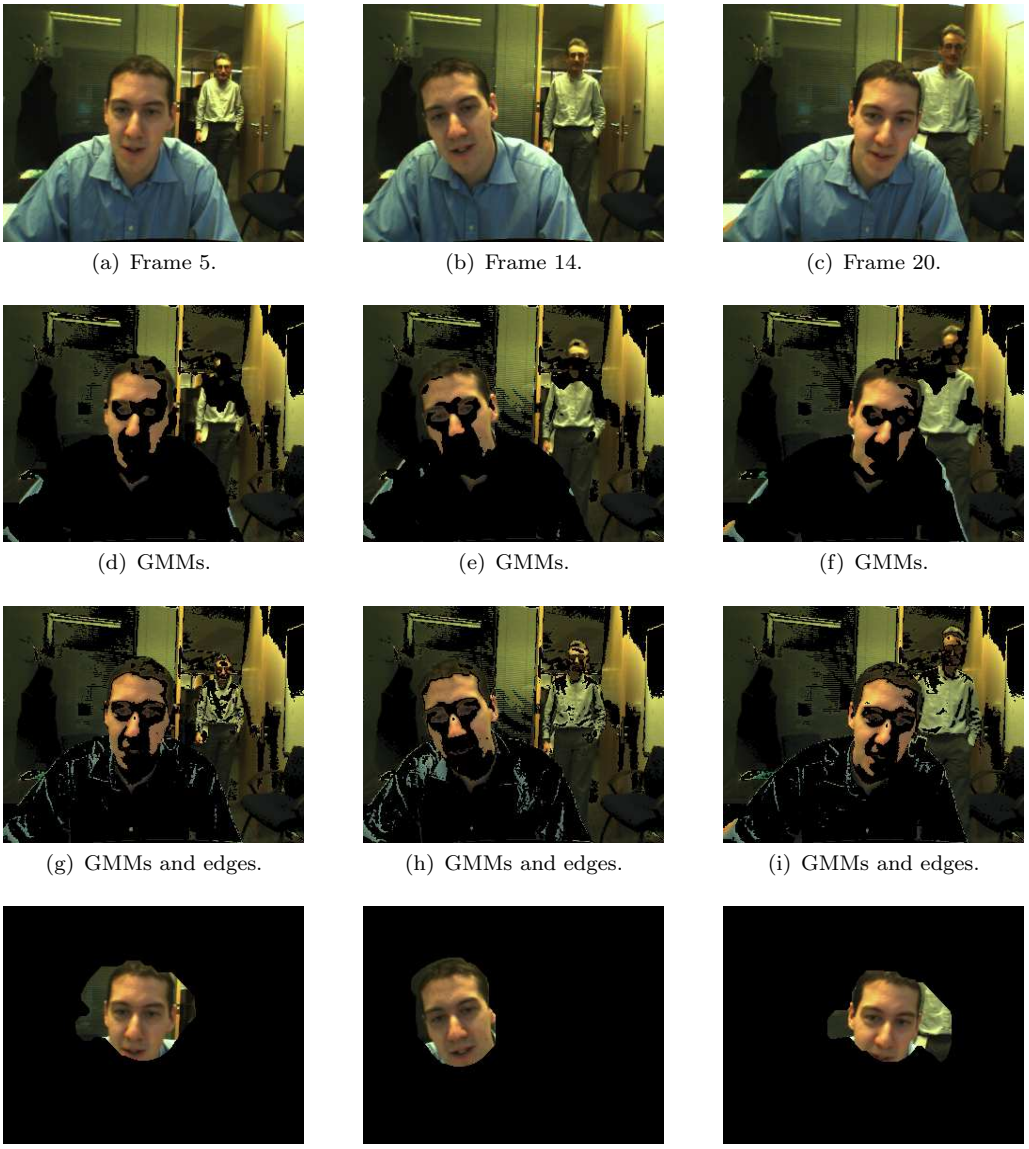
(d) GMMs. (e) GMMs. (f) GMMs.

(g) GMMs and edges. (h) GMMs and edges. (i) GMMs and edges.

(j) Shape prior. (k) Shape prior. (l) Shape prior.

FIGURE 5.14: Comparison of segmentation methods on the 'Antonio' video sequence. Some frames (a-c) from the original sequence are shown in the first row. Segmentations using graph cuts and colour GMMs (d-f), GMMs with edge detection methods (g-i) and GMMs with shape priors (j-l) are shown.

boundary weights include unimportant boundaries. The segmentation using edges and GMMs is adversely affected as shown in Figures 5.15(g), 5.15(h) and 5.15(i). The face to be segmented also changes position from left to right. This affects the brightness of the face in different frames and affects the GMMs. Figures 5.15(d), 5.15(e) and 5.15(f) do not provide accurate segmentations because of these limitations. Figures 5.15(j), 5.15(k) and 5.15(l) show the effect of changes in the position of the object and background motion on the segmentation using shape priors. It is observed that the shape prior is aligned correctly to the face using Powell's method. The performance of segmentation using shape priors is better than other methods even though the background is not stationary and the



(a) Frame 5. (b) Frame 14. (c) Frame 20.

(d) GMMs. (e) GMMs. (f) GMMs.

(g) GMMs and edges. (h) GMMs and edges. (i) GMMs and edges.

(j) GMMs and edges with shape prior. (k) GMMs and edges with shape prior. (l) GMMs and edges with shape prior.

FIGURE 5.15: Comparison of segmentation methods on the 'Geoff' video sequence. Some frames (a-c) from the original sequence are shown in the first row. Segmentations using graph cuts and colour GMMs (d-f), GMMs with edge detection methods (g-i) and GMMs with shape priors (j-l) are shown.

colour and position of the face changes.

The 'MS' video sequence and its segmentations are shown in Figure 5.16. Graph cuts and shape priors provide more accurate segmentations than other methods, even though the background is similar in colour to the object. The segmentation in Figures 5.16(c) and 5.16(d) classifies the hands of the person as foreground because they are the same colour as the face. The hands are also in motion, which affects segmentations in Figures 5.16(c) and 5.16(d) that use the motion in the video. Many pixels from the background are also wrongly classified as foreground. The segmentation using shape priors

in Figures 5.16(g) and 5.16(h) provide accurate segmentations in these cases, when it is difficult to segment using other methods.

In general it can be seen that shape priors result in more accurate segmentations compared to other methods. The shape prior is correctly aligned to each frame using Powell's method of minimization. Shape priors provide a clue to the location of 'foreground'. Hence they help in overcoming certain drawbacks of other methods, like background motion, changes in the position and orientation of the object and the object and background being similar in terms of colour. The correct alignment of the shape prior with the object ensures a good segmentation even if the foreground and background are similar in colour or the object is stationary relative to the background.

(a) Frame 5.

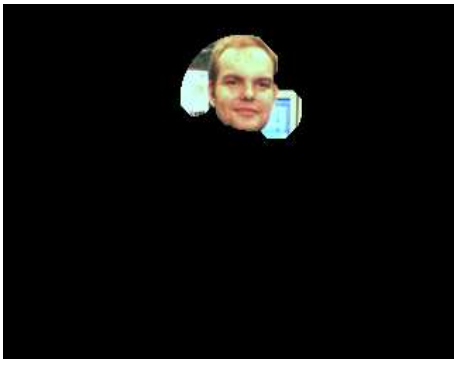
(b) Frame 39.

(c) GMMs.

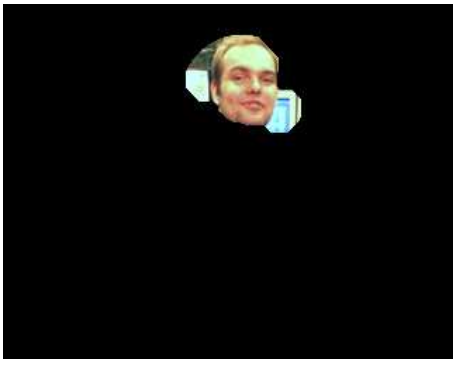
(d) GMMs.

(e) GMMs and edges.

(f) GMMs and edges.



(g) GMMs and edges with shape prior.



(h) GMMs and edges with shape prior.

FIGURE 5.16: Comparison of segmentation methods on the 'MS' video sequence. Some frames (a-b) from the original sequence are shown in the first row. Segmentations using graph cuts and colour GMMs (c-d), GMMs with edge detection methods (e-f) and GMMs with shape priors (g-h) are shown.

Chapter 6

Conclusions

This chapter reviews the work done and summarizes the contributions of this thesis. It lists the limitations of some of the previous work and proposes improvements. The conclusions drawn lead to suggestions for future research.

6.1 Conclusions of this thesis

The problems of image and video segmentation using graph cuts are studied in this thesis.

Images are segmented into 'foreground' and 'background' using a cost function based on region and boundary properties. Each pixel is considered a node and adjacent pixels are connected to form a graph structure. Globally optimal solutions are reached by maximizing the flow through the graph.

An 8-pixel neighbourhood is used for pixel connectivity in images. User-marked pixels are used to model foreground and background information. Region weights are set using GMMs based on colour and texture of the image. Soft constraints on all pixels are imposed using probability maps derived from GMMs. Boundary weights are based on the evidence of a pixel being an edge. Conventional methods like Canny edge detectors are tested. A novel edge model based on a dual-component GMM is proposed.

Segmentations are compared to ground truth using performance measures like precision, recall, accuracy and F-score. Different segmentations are analyzed and the reasons for methods working well are studied. A good segmentation is achieved quickly (0.2 seconds on average), with minimal user interaction using 1 iteration of the graph cut algorithm.

Previous work done on image segmentation using graph cuts and shape priors is reviewed. Learning the shape to be segmented and aligning the shape to the image are investigated.

The shape prior is aligned with the image data using Powell's method. The use of shape priors increases the accuracy of the segmentation. Accurate segmentations using shape priors can be used in medical imaging, specifically on the problem of segmenting specific shapes from x-ray and MRI images. The shape to be segmented can be estimated from training images and the alignment can be handled using the information given by the user. The applications of such methods to medical databases can be explored further.

Video segmentation techniques using graph cuts are studied. Various drawbacks, like too much user interference, extensive preprocessing and post-processing, and the algorithm being slow are discussed. The methods used for video segmentation are based on previous work done on image and video segmentation. Videos are viewed as 3D objects and inter-frame and intra-frame connections are used to extract and utilize more information from the video.

Graph cuts with shape priors are used to improve video segmentation. A circular shape prior, defined by its center and radius, is used to segment faces from video sequences. Powell's method of minimization is used to align the shape with each frame to minimize the cost. A proximity term is added to the cost function to maintain the continuity of the object being segmented. The shape prior is used in addition to the colour GMMs and edge models. Results of segmentations with and without shape priors are compared. It is observed that shape priors result in more accurate segmentation than only using GMMs and edge detection methods. Segmentation using shape priors is tested in cases where the background is similar to the foreground in colour, where there is motion in the background and where the object is moving. Powell's method is very effective in accurately aligning the shape to the object in all of these cases.

6.2 Future research suggestions

The graph cut algorithm is powerful and fast. Future research on graph cuts and other methods related to it can be as follows:

- Bi-directional graphs were used in this thesis to set the edge weights between nodes. Directed graphs with symmetric weights can be studied and can assign an orientation to edges. Also, different implementations of the graph cut algorithm can be tested and the best one chosen. Dynamic graphs cuts [4] were researched for this thesis, but other ideas to make small changes in each iteration of the algorithm can be explored. This can reduce the running time of the segmentation, but because graph cuts are fast this aspect was not studied in detail in this thesis.

- Belief Propagation algorithms can be used to perform the same tasks as graph cut algorithms. Belief propagation is a message passing algorithm for inference on graphical models, such as Bayesian networks or Markov random fields. A comparative study of belief propagation and graph cuts will be interesting. Belief propagation can be a powerful tool for video segmentation as the number of nodes in the graph increases rapidly.

In image segmentation, future research can be concentrated in the following areas:

- The foreground and background data can be modeled in different ways. Histograms and k-means clustering can be used to model data. Some work was done with k-means clustering, by clustering data before performing the graph cut. Clustering or weighting GMMs can help in classifying pixels better. For example, if foreground consists of mostly red and yellow pixels, clustering will help to classify other pixels accurately.

- Finding the features that distinguish foreground and background is as important as modeling the data. A collection of colour and texture features is used in this thesis and the performance is evaluated. Feature selection can be researched further to find the important features for specific images. If this is done, a feature selection phase can be added to the algorithm, where the features that best separate foreground and background chosen. This collection of features can be used to do the segmentation for that specific image. In theory, this approach should be more accurate as it selects the best features.

- Evidence for boundaries can be researched further. The edge models proposed in this thesis are useful to detect relevant edges, but more exploration can be done in this context. Edge detection methods are used in many algorithms other than graph cuts and research can prove very important. Detection of specific edges is more important than detecting all edges.

- The problem of binary segmentation, i.e. segmentation into 'foreground' and 'background' classes, is explored in this thesis. However, graph cuts can be used to segment images into many more regions. In some cases, it is helpful to segment images in more than two classes. Methods to do this have been previously researched but are out of the scope of this thesis.

- Shape priors can be combined with graph cuts to give accurate segmentations. Instead of imposing tight and strict shape models, certain deformable models can be studied. Active Shape Models (ASMs) and Constrained Localized Models (CLMs)

were investigated, but through testing and evaluation of these methods needs to be done.

- The graph cut algorithm is powerful and fast. An average image segmentation takes 0.2 seconds. Instead of improving the algorithm, research can be done into its application. Graph cuts can be used for different applications. Medical imaging and image retrieval applications have been researched, but image segmentation can be used in other applications. The algorithm is fast and interactive, making it a quick and useful tool.

Video segmentation can be extended further in the following ways:

- Motion estimation was worked on using frame subtraction. As with boundary properties for images, motion estimation can be explored further. Tracking moving objects can make segmentation easy. These methods play an important role in many segmentation techniques, not just the ones using graph cuts.
- In this thesis, a circular shape prior is used for segmentation. In future, a complex shape prior can be used to segment objects. A shape prior can be chosen based on the object to be segmented.

Appendix A

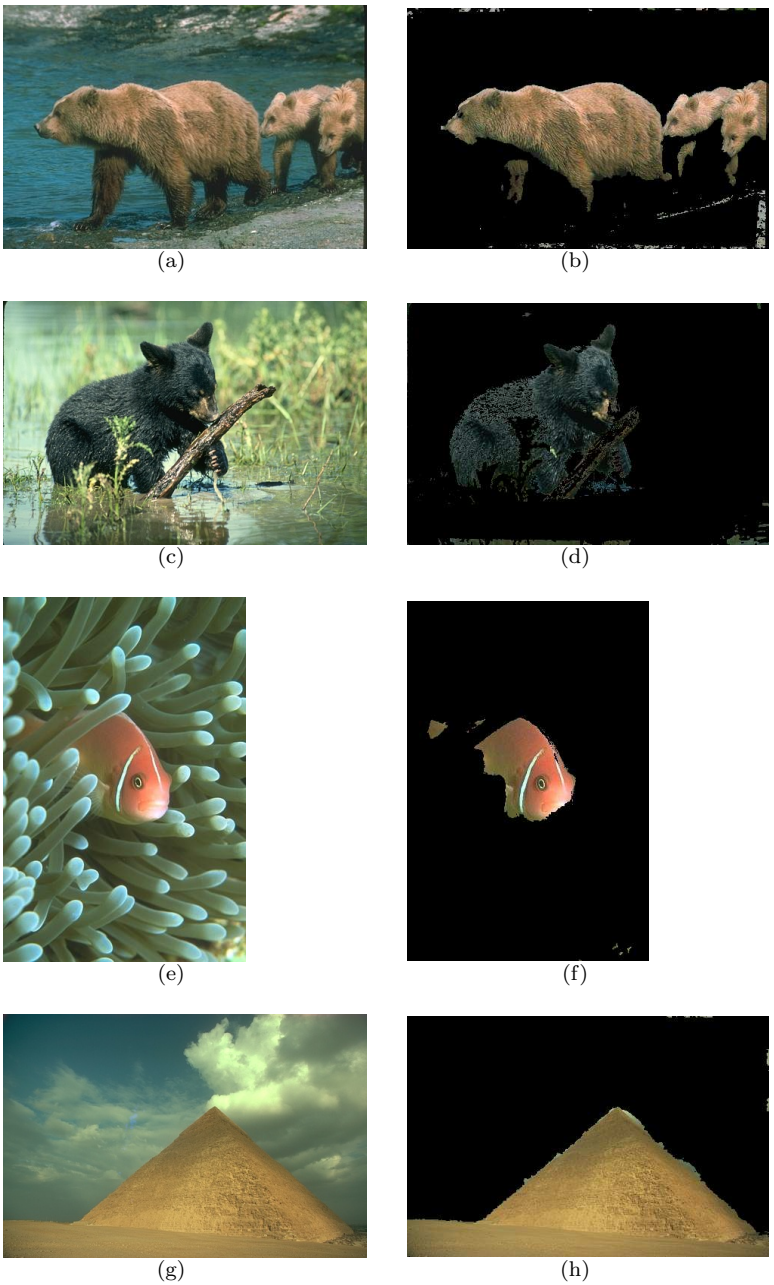
Segmentation results and performance

This appendix includes the results of segmentation using graph cuts for other images in the Berkeley dataset [1]. All these results were recorded after 1 iteration of the algorithm. Figures A.1, A.2 and A.3 show segmentations of images using graph cuts.

Table A.1 shows the precision (p) and recall (r) for segmentations of colour images and Table A.2 shows the F-score (F) and Accuracy (A).

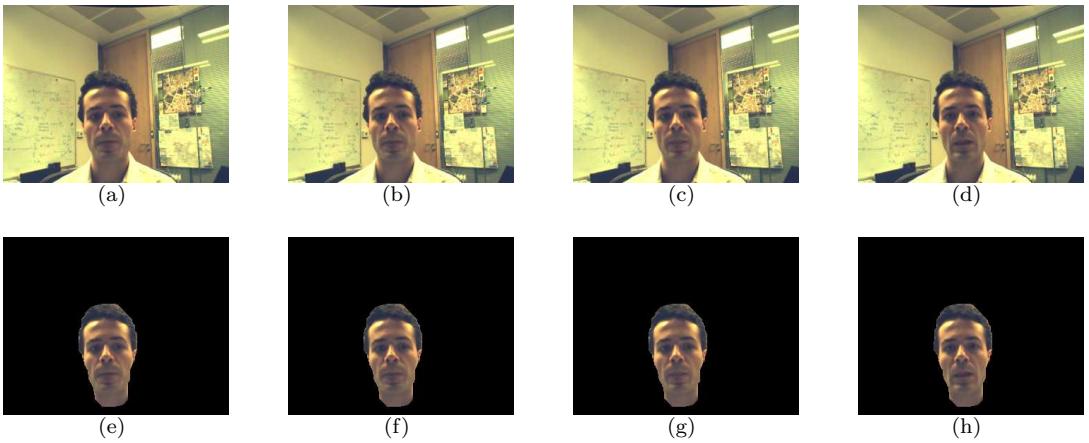
TABLE A.1: A table showing precision (p) and recall (r) of the segmentations for colour images.

	'birds'	**'plane'**	**'flowers'**	**'grass'**	**'eagle'**
R, G, B	p = 0.914 r = 0.979	p = 0.952 r = 0.939	p = 0.929 r = 0.997	p = 0.916 r = 0.995	p = 0.942 r = 0.891
G, (G-R), (G-B)	p = 0.900 r = 0.983	p = 0.952 r = 0.948	p = 0.795 r = 0.940	p = 0.935 r = 0.995	p = 0.940 r = 0.892
L, u, v	p = 0.857 r = 0.989	p = 0.943 r = 0.961	p = 0.912 r = 0.998	p = 0.966 r = 0.995	p = 0.985 r = 0.888
G, (G-R), (G-B), L, u, v, MR8	p = 0.629 r = 0.999	p = 0.907 r = 0.922	p = 0.957 r = 0.986	p = 0.973 r = 0.995	p = 0.960 r = 0.888
R, G, B, L, u, v, MR8	p = 0.629 r = 0.997	p = 0.948 r = 0.905	p = 0.953 r = 0.992	p = 0.941 r = 0.996	p = 0.970 r = 0.879
G, (G-R), (G-B), L, u, v	p = 0.936 r = 0.966	p = 0.954 r = 0.899	p = 0.953 r = 0.985	p = 0.951 r = 0.995	p = 0.948 r = 0.896
L, u, v, MR8	p = 0.496 r = 1.000	p = 0.724 r = 0.949	p = 0.910 r = 0.998	p = 0.941 r = 0.973	p = 0.912 r = 0.929



(a) (b) (c) (d) (e) (f) (g) (h)

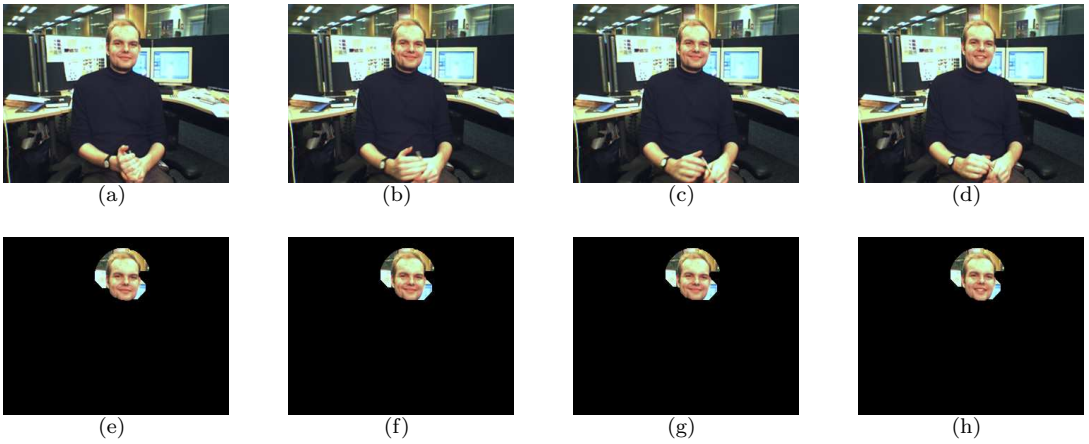
FIGURE A.1: Original images and their segmentations using shape priors with colour GMMs to assign region costs and gradient-based method to assign boundary costs.



(a) (b) (c) (d)

(e) (f) (g) (h)

FIGURE A.2: Original images and their segmentations using shape priors with colour GMMs to assign region costs and gradient-based method to assign boundary costs.



(a) (b) (c) (d)

(e) (f) (g) (h)

FIGURE A.3: Original images and their segmentations using shape priors with colour GMMs to assign region costs and gradient-based method to assign boundary costs.

TABLE A.2: A table showing F-score (F) and Accuracy (A) of the segmentations for colour images.

	'birds'	**'plane'**	**'flowers'**	**'grass'**	**'eagle'**
R, G, B	F = 0.945 A = 0.994	F = 0.946 A = 0.994	F = 0.962 A = 0.966	F = 0.954 A = 0.985	F = 0.916 A = 0.966
G, (G-R), (G-B)	F = 0.940 A = 0.994	F = 0.949 A = 0.994	F = 0.862 A = 0.873	F = 0.964 A = 0.988	F = 0.915 A = 0.966
L, u, v	F = 0.918 A = 0.991	F = 0.952 A = 0.995	F = 0.953 A = 0.959	F = 0.980 A = 0.994	F = 0.934 A = 0.974
G, (G-R), (G-B), L, u, v, MR8	F = 0.772 A = 0.971	F = 0.915 A = 0.990	F = 0.971 A = 0.975	F = 0.984 A = 0.995	F = 0.923 A = 0.969
R, G, B, L, u, v, MR8	F = 0.771 A = 0.971	F = 0.926 A = 0.992	F = 0.972 A = 0.976	F = 0.968 A = 0.989	F = 0.922 A = 0.969
G, (G-R), (G-B), L, u, v	F = 0.951 A = 0.995	F = 0.926 A = 0.992	F = 0.969 A = 0.973	F = 0.972 A = 0.991	F = 0.922 A = 0.968
L, u, v, MR8	F = 0.663 A = 0.950	F = 0.822 A = 0.977	F = 0.952 A = 0.958	F = 0.956 A = 0.986	F = 0.921 A = 0.967

Bibliography

[1] 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 *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[2] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. volume 1, pages 105–112, July 2001.

[3] Microsoft Research. Microsoft i2i dataset, April 2010. URL http://www.research.microsoft.com/vision/cambridge/i2i.

[4] Pushmeet Kohli and Philip H. S. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2079–2088, 2007.

[5] S. Teukolsky W. Press, B. Flannery and W. Vetterling. *Numerical recipes in C*. Cambridge: Cambridge University Press, 1988.

[6] Pushmeet Kohli, Jonathan Rihan, Matthieu Bray, and Philip H. S. Torr. Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *International Journal of Computer Vision*, 79(3):285–298, 2008.

[7] R. Rivest T. Cormen, C. Leiserson and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 2007.

[8] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[9] Neill D. F. Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Automatic 3D object segmentation in multiple views using volumetric graph-cuts. *Image Vision Comput.*, 28(1):14–25, 2010.

[10] Carlos Hernández and George Vogiatzis. Shape from photographs: A multi-view stereo pipeline. In *Computer Vision: Detection, Recognition and Reconstruction*, pages 281–311. 2010.

[11] George Vogiatzis, Carlos Hernández Esteban, Philip H. S. Torr, and Roberto Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2241–2246, 2007.

[12] George Vogiatzis, Philip H. S. Torr, and Roberto Cipolla. Multi-view stereo via volumetric graph-cuts. In *CVPR (2)*, pages 391–398, 2005.

[13] Neill D. F. Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV (1)*, pages 766–779, 2008.

[14] Sampling Greig, Seheult Revisited, C. Fox, and G. K. Nicholls. Exact map states and expectations from perfect sampling: Greig, porteous and seheult revisited, 2001.

[15] L.R. Ford and D.R. Fulkenson. *Flows in Networks.* Princeton University Press, Princeton, NJ, 1962.

[16] Dorin Comaniciu and Peter Meer. Robust analysis of feature spaces: Color image segmentation. *IEEE Conf. Computer Vision and Pattern Recognition (CVPR'97)*, pages 750–755, 1997. San Juan, Puerto Rico.

[17] H. Permuter, J. Francos, and I. Jermyn. Gaussian mixture models of texture and colour for image database. In *ICASSP*, pages 25–88, 2003.

[18] H. Permuter, J. Francos, and I. Jermyn. A study of gaussian mixture models of color and texture features for image classification and segmentation. In *Pattern Recognition*, volume 39 of *4*, pages 695–706, New York, USA, April 2006. Elsevier Science Inc.

[19] Stephen Haddad. Texture measures for segmentation. Master's thesis, University of Cape Town, April 2007.

[20] Jitendra Malik, Serge Belongie, Thomas K. Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[21] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997.

[22] David R. Martin, Charless Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004.

[23] David R. Martin, Charless Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using brightness and texture. In *NIPS*, pages 1255–1262, 2002.

[24] Francisco J. Estrada and Allan D. Jepson. Quantitative evaluation of a novel image segmentation algorithm. In *CVPR (2)*, pages 1132–1139, 2005.

[25] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24: 1026–1038, 1999.

[26] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms using different performance metrics. In *In Proc. 23 rd Intl. Conf. Machine learning (ICML06*, pages 161–168, 2005.

[27] Yuri Boykov and Marie-Pierre Jolly. Interactive organ segmentation using graph cuts. In *In Medical Image Computing and Computer-Assisted Intervention*, pages 276–286, 2000.

[28] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004. ISSN 0730-0301. doi: http://doi.acm.org/10.1145/1015706.1015720.

[29] Matthew Marsh, Shaun Bangay, and Adele Lobb. Implementing the "GrabCut" segmentation technique as a plugin for the GIMP. In *Afrigraph '06: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 171–175, New York, NY, USA, January 2006. ACM Press.

[30] Y. Tian, T. Guan, C. Wang, L. Li, and W. Liu. Interactive foreground segmentation method using mean shift and graph cuts. *Sensor Review*, 29(2):157–162, 2009. Emerald Group Publishing Limited.

[31] Ingemar J. Cox and Yu Zhong. Ratio regions: a technique for image segmentation. In *in International Conference on Pattern Recognition*, pages 557–564, 1996.

[32] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[33] Song Wang and Jeffrey Mark Siskind. Image segmentation with ratio cut. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6), 2003.

[34] Song Wang, Toshiro Kubota, Jeffrey Mark Siskind, and Jun Wang. Salient closed boundary extraction with ratio contour. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4):546–561, 2005.

[35] Matthieu Bray, Pushmeet Kohli, and Philip H. S. Torr. Posecut: Simultaneous segmentation and 3D pose estimation of humans using dynamic graph-cuts. In *In ECCV*, pages 642–655, 2006.

[36] Bo Peng and Olga Veksler. Parameter selection for graph cut based image segmentation. In *In BMVC*, 2008.

[37] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *in ECCV*, pages 428–441, 2004.

[38] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. Technical report, 2008.

[39] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. pages 277–284, 2009.

[40] M. Pawan Kumar, Philip H. S. Torr, and A. Zisserman. Obj cut. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 18–25, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: http://dx.doi.org/10.1109/CVPR.2005.249.

[41] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient matching of pictorial structures. In *CVPR*, pages 2066–, 2000.

[42] Pedro F. Felzenszwalb, Daniel P. Huttenlocher, and Jon M. Kleinberg. Fast algorithms for large-state-space hmms with applications to web usage analysis. In *NIPS*, 2003.

[43] Olga Veksler. Star shape prior for graph-cut image segmentation. In *ECCV (3)*, pages 454–467, 2008.

[44] Daniel Freedman and Tao Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 755–762, 2005.

[45] M. Heikkil, M. Pietikinen, and J. Heikkil. A texture-based method for detecting moving objects. In *British Machine Vision Conference*, pages 187–196, 2004.

[46] Shiloh L. Dockstader, Nikita S. Imennov, and A. Murat Tekalp. A robust Bayesian network for articulated motion classification. In *ICIP (3)*, pages 305–308, 2003.

[47] Ahmet Ekin and A. Murat Tekalp. Robust dominant color region detection and color-based applications for sports video. In *ICIP (1)*, pages 21–24, 2003.

[48] Çigdem Eroglu Erdem, Bülent Sankur, and A. Murat Tekalp. Performance measures for video object segmentation and tracking. In *VCIP*, pages 29–40, 2003.

[49] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 53–60, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: http://dx.doi.org/10.1109/CVPR.2006.69.

[50] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bi-layer segmentation of binocular stereo video. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 407–414, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: http://dx.doi.org/10.1109/CVPR.2005.91.

[51] Yin Li, Jian Sun, and Heung yeung Shum. Video object cut and paste. *ACM Transactions on Graphics*, 24:595–600, 2005.

[52] Jue Wang, Pravin Bhat, Alex Colburn, Maneesh Agrawala, and Michael F. Cohen. Interactive video cutout. *ACM Trans. Graph.*, 24(3):585–594, 2005.

[53] Ian Nabney. Netlab: Algorithms for pattern recognition. Springer, 2002.

[54] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/TPAMI.1986.4767851.