Global optimisation techniques for image segmentation with higher order models

Sara Alexandra Gomes Vicente

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of the

University College London.

Department of Computer Science

University College London

November 14, 2011

I, Sara Alexandra Gomes Vicente confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

_____

Abstract

Energy minimisation methods are one of the most successful approaches to image segmentation. Typically used energy functions are limited to pairwise interactions due to the increased complexity when working with higher-order functions. However, some important assumptions about objects are not translatable to pairwise interactions. The goal of this thesis is to explore higher order models for segmentation that are applicable to a wide range of objects. We consider: (1) a connectivity constraint, (2) a joint model over the segmentation and the appearance, and (3) a model for segmenting the same object in multiple images.

We start by investigating a connectivity prior, which is a natural assumption about objects. We show how this prior can be formulated in the energy minimisation framework and explore the complexity of the underlying optimisation problem, introducing two different algorithms for optimisation. This connectivity prior is useful to overcome the "shrinking bias" of the pairwise model, in particular in interactive segmentation systems.

Secondly, we consider an existing model that treats the appearance of the image segments as variables. We show how to globally optimise this model using a Dual Decomposition technique and show that this optimisation method outperforms existing ones.

Finally, we explore the current limits of the energy minimisation framework. We consider the cosegmentation task and show that a preference for object-like segmentations is an important addition to cosegmentation. This preference is, however, not easily encoded in the energy minimisation framework. Instead, we use a practical proposal generation approach that allows not only the inclusion of a preference for object-like segmentations, but also to learn the similarity measure needed to define the cosegmentation task.

We conclude that higher order models are useful for different object segmentation tasks. We show how some of these models can be formulated in the energy minimisation framework. Furthermore, we introduce global optimisation methods for these energies and make extensive use of the Dual Decomposition optimisation approach that proves to be suitable for this type of models.

Acknowledgements

I would like to thank my supervisors: Vladimir Kolmogorov for sharing his extensive knowledge of computer vision and combinatorial optimisation, and Carsten Rother for believing and being enthusiastic about my research. I was very lucky to have the opportunity to work with both of them and their guidance and support was essential for completing this thesis.

I am grateful to Simon Prince and Gabriel Brostow for maintaining a dynamic computer vision research group at UCL and for encouraging me to be part of it. Also, I am grateful to several students, particularly Michael Firman, Oisin Mac Aodha, Maciej Gryka and Malcolm Reynolds, for helpful discussions and ramblings, and for proof-reading this thesis.

I would like to thank Dr. Andrew Fitzgibbon and Professor Simon Arridge for accepting to be examiners of this thesis and for helpful comments that made it accessible to a more general audience.

I would like to thank the people that were part of the UCL Adastral Campus in Ipswich, the administrative, research and technical staff, for making it a more enjoyable place to work in.

My stay in Ipswich, Cambridge and London was made more pleasant by several people: Giacomo Cancelli, Peng Wu, Antti Larjo, Dheeraj Singaraju, Martina Campanella, Gary Zhang and Teresa Correia.

I would like to thank my friends back in Portugal. ”Os amigos que são a família que eu escolhi”: Ana, Marco, Nuno, Valter and João.

I want to thank my family: my parents and sisters for their unconditional support, and Rafael, for making life look simple. Finally, I am grateful to Hugo for always being there.

My PhD was funded by Microsoft Research Cambridge through its European PhD Scholarship Programme and by Fundação para a Ciência e a Tecnologia.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer vision aims to extract useful information about the real world from images. One of the tasks extensively studied in the computer vision field is segmentation, i.e. the task of separating the image into coherent regions. In particular, we are interested in the more specific problem of *object segmentation* where the goal is to separate an object of interest from the rest of the image.

The information provided by a pixel-accurate segmentation is useful in a range of applications. Take the first example in Fig. 1.1. A human can easily recognise the object present in (a), by simply looking at the binary segmentation in (b), since the shape of the object is very discriminative and fully captured by the binary segmentation. This motivates object recognition algorithms that use segmentation as a pre-processing stage, using these segments as an alternative to sliding windows, e.g. [94, 75, 70].

For the cases where shape is not discriminative enough, segmentation is still useful to isolate the object of interest from a cluttered background. For example, the foreground pixels in Fig. 1.1 (d) contain the full extent of the train and could be used in several tasks: as input to an object recognition system, thus reducing the number of pixels that need to be processed, or to create an image composite in a typical photo editing task, by combining this foreground with a different background.

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

Figure 1.1: Examples of object segmentation. For some images (a), a pixel wise segmentation of the object (b) provides enough information for a human to recognise it. For other images (c), segmentation is useful to isolate the object of interest from a cluttered background (d).

Although segmentation is usually perceived as a useful task, it is also an ill-defined problem. Some of the criticisms of segmentation are:

- Object segmentation is in general an ill-posed and ambiguous problem, since the image may contain several different objects or objects with different components that can be objects themselves. For example, for the image in Fig. 1.1 (c) the object of interest may be a specific container or only the locomotive, as opposed to the segmentation presented in (d), which includes the full train.

- Even if the image contains a single object and the ambiguity inherent to segmentation is reduced, it is still not always possible to segment it correctly based only on low-level features. Low-level segmentation relies on the assumptions that the object has distinct properties from the background (like texture or colour) and that there are strong edges separating the two segments. However, these assumptions are not always valid or sufficient to correctly segment an image.

The first criticism is overcome in some applications of segmentation, such as interactive image segmentation where a user provides extra cues. When several images of the same object are available, the *cosegmentation* task (loosely defined as the joint segmentation of the same object in multiple images) can also be useful to address the ambiguities of single image segmentation, since the use of multiple images can help select and locate the object of interest. We will consider both interactive segmentation and cosegmentation tasks in this thesis.

The second criticism can be addressed by including other types of low-level information, such as motion information obtained from video sequences, or by incorporating extra top-down information that helps further constraining the problem, such as knowledge about the shape of the object.

Despite these criticisms, segmentation has proven to be a useful tool in some specific tasks and applications, such as medical imaging [69], photo editing [91] and object recognition [94, 70].

1.1 Models for segmentation

In this thesis we treat segmentation as a binary labelling problem, where each pixel is assigned a label (object or background) and formulate the task as a discrete energy minimisation problem. We refer to this approach to segmentation as the "energy minimisation framework". Energy minimisation techniques have been extensively used in computer vision. They are derived from principled probabilistic formulations and have proved to be useful not only for segmentation

(a) Smoothness prior (b) Class specific prior

Figure 1.2: Models for segmentation. Existing priors for segmentation range from generic smoothness priors (a) to models specific for a certain object class (b). Images reproduced from [85, 62].

[14] but for other vision tasks like stereo matching [18] and denoising [90].

Existing segmentation models defined in the energy minimisation framework encode assumptions ranging from generic smoothness priors to complex priors for specific object classes.

A smoothness prior has a preference towards assigning the same label to neighbouring pixels. This is an intuitive assumption for segmentation, since pixels belonging to most objects tend to form a compact set as opposed to being dispersed in the image. Fig. 1.2 (a) shows samples from a probabilistic prior, the Ising prior, which encodes smoothness. A prior of this form can be formulated as an energy function with pairwise potentials. This prior is applicable to most objects; some exceptions are objects with long boundaries, such as plants or fences.

At the other end of the spectrum are priors that are specific to a certain object class, for example by imposing specific shapes. Fig. 1.2 (b) shows samples from a shape prior specific for cows, introduced in [62]. Priors of this form require both training examples for learning, and knowledge at test time about the class of the object present in the image. Besides shape, they can also incorporate knowledge about the appearance of the object of interest, like texture and colour.

Both types of models have been successfully used for segmentation. However, they have some limitations. Models for specific object classes generally provide high-quality segmentations, but are very restrictive since they are only applicable in very specific scenarios. On the other hand, smoothness priors are widely applicable, but less reliable. Recently, there has been some interest in models that lie in the middle ground between these extremes [53, 107, 23, 48]. This is also the type of model we analyse in this thesis.

In summary, we are interested in models that:

- Do not require information of the object class and are applicable to a wide range of objects.

- Encode assumptions that go beyond the traditional smoothness assumption and that help to overcome the limitations of using only low-level information.

Furthermore, we give particular attention to models that:

- Can be formulated in the energy minimisation framework. This framework can have a principled probabilistic interpretation and it has been successfully used for segmentation, including in commercial products [92].

- Can be globally optimised. Good optimisation methods are an essential part of the energy minimisation framework. They are important to find good solutions, if the energy used is well suited for the task, and to reveal the inaccuracy of the energy formulation, if the global optimum of the energy is a poor solution.

An example of a model that fits all these requirements is a connectivity prior. Connectivity is an intuitive constraint for a wide range of object classes that goes beyond traditional smoothness priors. It can be included in the energy minimisation framework and it can be globally optimised. A connectivity prior is the subject of chapter 3.

In chapter 5 we discuss some properties that, although useful for the task of cosegmentation, cannot be easily incorporated in the energy minimisation framework.

1.2 Summary of contributions

In this thesis, we address the problem of object segmentation by investigating different models that are generic and applicable to a variety of objects without making strong assumptions, for example about the object class.

We build on existing energy minimisation techniques for segmentation. Commonly used energy functions are restricted to pairwise models. However, the properties we are interested in cannot be formulated as pairwise functions. Instead, we use *higher-order models*, i.e. energy functions that contain potential functions which are dependent on the labels of more than two pixels. Since existing optimisation methods are not suitable for energy functions of this form, we also develop powerful *global optimisation methods* for the models presented.

The main contributions of the thesis are:

- An energy based method to impose connectivity constraints in the segmentation. We develop both a higher order model for this purpose and two associated optimisation algorithms. We demonstrate that a prior of this form is helpful for interactive segmentation, in particular to segment objects with thin structures.

- A new optimisation method for a powerful model that jointly considers the inference over the segmentation and the appearance models of each segment. Models of this form

have been extensively used for segmentation, however they are usually optimised with coordinate descent techniques that converge to a local minimum. We start by rewriting the model as a function of the segmentation only, by eliminating the appearance variables. This new formulation allows the use of a new optimisation method based on Dual Decomposition that outperforms existing approaches.

- A new optimisation method for energy minimisation models with applications in cosegmentation. We review existing energy minimisation models for cosegmentation and propose a new optimisation method for these models based on Dual Decomposition, which outperforms currently used optimisation techniques.

- A cosegmentation model that explicitly prefers object-like segmentations. The inclusion of this assumption leads to a method that outperforms the existing state of the art in cosegmentation. We rely on a proposal generation mechanism that extracts plausible, object-like, binary segmentations for each image and learn a similarity measure between the proposals, to select the best proposal for each image.

1.3 Structure of the thesis

Chapter 2 contains background on energy minimisation, image segmentation and optimisation methods for energy functions. Chapter 3 describes the new method to include connectivity constraints in the segmentation and its usefulness in interactive systems. Chapter 4 describes the new optimisation method for the joint model. In chapter 5 we introduce the task of cosegmentation, discuss common energy minimisation approaches for this task and present a new method based on object-like segmentations. In chapter 6 we discuss our conclusions and future directions to extend this work. Finally, there are two appendices. Appendix A provides proofs of several theorems included in the thesis and Appendix B provides an illustration of the DijkstraGC algorithm.

1.4 Publications

Some of the work presented in this thesis has been published in the following conference papers:

Chapter 3

Sara Vicente, Vladimir Kolmogorov and Carsten Rother. “Graph cut based image segmentation with connectivity priors”. In *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), June 2008.

Chapter 4

Sara Vicente, Vladimir Kolmogorov and Carsten Rother. "Joint optimisation of segmentation and appearance models". In *IEEE International Conference on Computer Vision* (ICCV), October 2009.

Chapter 5

Sara Vicente, Vladimir Kolmogorov and Carsten Rother. "Cosegmentation revisited: models and optimisation". In *European Conference on Computer Vision* (ECCV), September 2010.

Sara Vicente, Carsten Rother and Vladimir Kolmogorov. "Object cosegmentation". In *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), June 2011.

Chapter 2

Background

In this chapter we introduce the optimisation approach to vision. We start by reviewing labelling approaches based on energy formulations (section 2.2) and in particular the task of image segmentation, when seen as a labelling task (section 2.3). In section 2.3.1 we discuss graph cut methods, which provide efficient optimisation tools for energy minimisation. Extensions of pairwise energy functions to higher-order models with applications in segmentation are introduced in section 2.4. We conclude the chapter by discussing Dual Decomposition, a generic optimisation method suitable for optimisation of higher-order energy functions (section 2.5.1).

2.1 Optimisation approach to vision

One of the goals of computer vision is to extract information about the real world from images. Although for humans this task is performed effortlessly, designing systems that mimic this human behaviour can be very challenging.

A successful approach to many vision tasks is to formulate them as optimisation problems. In this approach, the solution to the problem is defined as the minimum of an objective function that measures the *goodness* of all possible solutions. Two major steps are needed in order to formulate such optimisation problems. In the first stage, the objective function is defined. This objective function has to be chosen carefully so that it correctly represents the problem. In the second stage, an optimisation algorithm is chosen to minimise the objective function.

In most cases both stages are coupled. The choice of objective function is usually influenced by known optimisation methods since there is a preference in defining an objective function that is tractable, but it can also motivate the development of new optimisation techniques.

Both steps of the process are equally important. A poorly defined objective function will misrepresent the properties of the system, while a weak optimisation algorithm will not guarantee that an optimal solution is achieved.

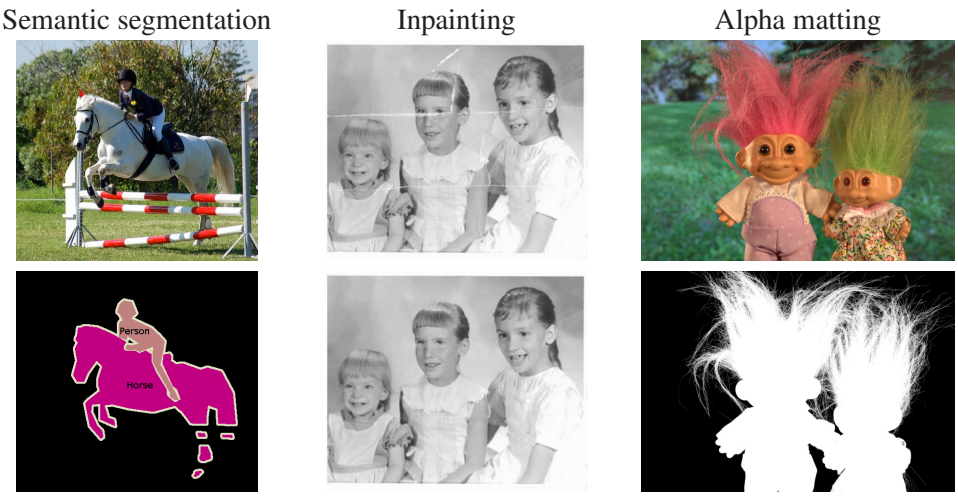


Figure 2.1: Labelling problems in computer vision. The goal of semantic segmentation is to identify the full extent of objects in images. The labels for this task correspond to object classes, e.g. "horse" and "person". Inpainting consists in reconstructing damaged images (images from [90]) and the set of labels for this task ranges from 0 to 255, corresponding to the image gray levels. Alpha matting provides a continuous mask representing different levels of transparency. The set of labels for alpha matting is the continuous interval [0,1].

2.2 Labelling problems and Markov Random Fields

Given a set of sites $\mathcal{V} = \{1, ..., n\}$ and a set of labels $\mathcal{L}$ a labelling problem consists of assigning to each site in $\mathcal{V}$ a label from the set $\mathcal{L}$. The sites usually correspond to pixels in the image, but they can also correspond to other higher-level components, such as superpixels. Fig. 2.1 shows examples of labelling problems previously considered in vision. We will consider problems with a discrete set of labels $\mathcal{L} = \{1, ..., L\}$.

We denote by $\boldsymbol{x} = \{x_p \mid p \in \mathcal{V}\}$ with $x_p \in \mathcal{L}$ a possible labelling. A *clique* c is defined as a subset of sites and $\mathcal{C}$ is a set of cliques.

To formulate a labelling problem as an optimisation problem an objective function is defined in the space of possible labellings. Following common terminology, we use the term *energy function* to refer to this objective function. Commonly used energy functions have the following form:

$$E(\boldsymbol{x}) = \sum_{c \in \mathcal{C}} \phi_c(\boldsymbol{x}_c) \tag{2.1}$$

where $\phi_c(x_c)$ are functions named *clique potentials*, that depend only on $\boldsymbol{x}_c = \{x_p | p \in c\}$, i.e. the labels of the sites included in c.

Markov Random Fields

Energy functions of the form (2.1) were first introduced in the context of *Maximum a Posteriori* estimation of Markov Random Fields (MRF) [38]. MRFs are a probabilistic framework that

capture the spatial consistency present in images. A random field is a set of random variables $\mathcal{X} = \{X_1, ..., X_n\}$ associated with the sites in $\mathcal{V}$ where each random variable X_i takes values in $\mathcal{L}$. Given a neighbourhood system $\mathcal{N} = \{(p,q)|\{p,q\} \subset c, c \in \mathcal{C}\}$ a *Random Field* is a *Markov Random Field* if it satisfies the following properties:

$$\Pr(\boldsymbol{x}) > 0, \forall \boldsymbol{x} \in \mathcal{L}^n \tag{2.2}$$

$$\Pr(x_p \mid x_{\mathcal{V}-\{p\}}) = \Pr(x_p \mid x_{\mathcal{N}_p}) \tag{2.3}$$

where $\Pr(\boldsymbol{x})$ refers to $\Pr(X = \boldsymbol{x})$, $\Pr(x_p)$ refers to $\Pr(X_p = x_p)$, $x_{\mathcal{V}-\{p\}} = \{x_q | q \in \mathcal{V} - \{p\}\}$, and $x_{\mathcal{N}_p} = \{x_q | (p,q) \in \mathcal{N}\}$ are the labels of the neighbors of p.

The first property is assumed for technical reasons and ensures that the joint probability is uniquely determined by its local conditional probabilities. The second property, also called Markov property, states that a site interacts directly only with its neighbour sites, i.e. the global dependency relations can be reduced to the dependency of a small subset.

The Hammersley-Clifford theorem states that the probability distribution of an MRF factorise as a product of *compatibility functions* over cliques or equivalently follows a *Gibbs distribution* and that it can be written in the form:

$$\Pr(\boldsymbol{x}) = \frac{1}{Z} \exp\left(-\sum_{c \in \mathcal{C}} \phi_c(\boldsymbol{x}_c)\right) \tag{2.4}$$

where Z is a normalising constant known as the partition function.

Maximum a Posteriori of Markov Random Fields

MRFs are commonly used to define the prior distribution in a Bayesian approach to labelling problems [38].

Recall that Bayesian inference requires a prior model and a likelihood function given the observed variables $\boldsymbol{y}$. Then, the *Maximum a Posteriori* (MAP) solution $\boldsymbol{x}^*$ is obtained by maximising the posterior probability $\Pr(\boldsymbol{x}|\boldsymbol{y})$ or equivalently, from Bayes theorem:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \Pr(\boldsymbol{y}|\boldsymbol{x}) \times \Pr(\boldsymbol{x}). \tag{2.5}$$

For many labelling problems, and in particular for the case of image segmentation illustrated in Fig. 2.2, the observed variables correspond to RGB colour values for each pixel. We assume that the likelihood function factorises over sites, i.e. $\Pr(\boldsymbol{y}|\boldsymbol{x}) = \prod_{p \in \mathcal{V}} \Pr(y_p|x_p)$, and that it is given by a fixed appearance model for each of the labels.

Assuming that we are using a pairwise MRF as a prior model, i.e. each *clique* c has two

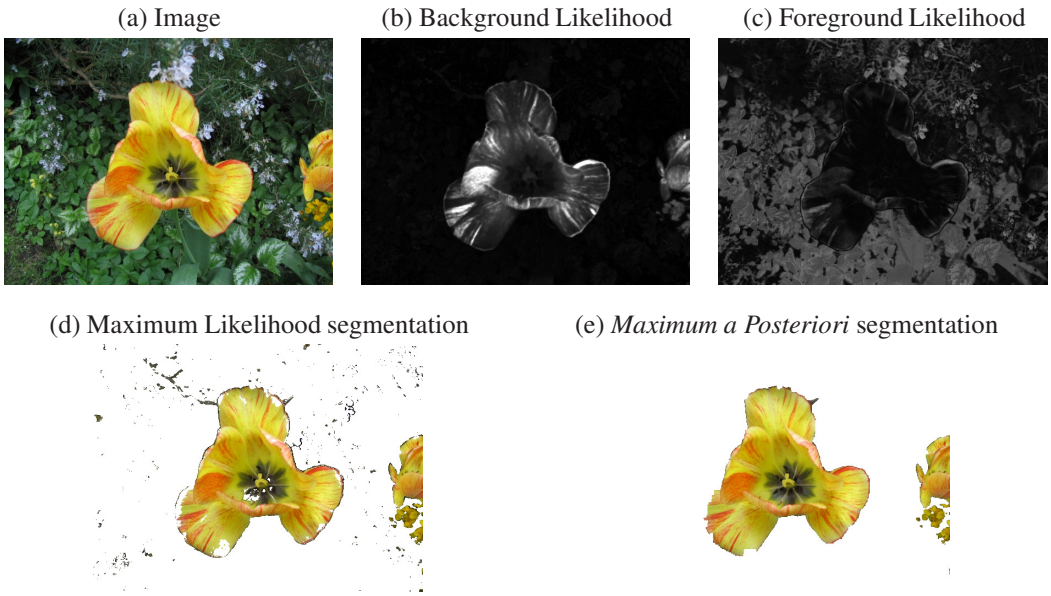


Figure 2.2: Comparison of the Maximum Likelihood and the *Maximum a Posteriori* segmentations. The negative log-likelihood correspondent to each label is shown in (b) and (c). Brighter pixels disagree with the corresponding appearance model. The Maximum Likelihood segmentation (d) suffers from fragmentation, which can be easily overcome by using an MRF prior (e).

elements, finding the MAP solution corresponds to minimising an energy of the form

$$E(\boldsymbol{x}) = \sum_{p \in \mathcal{V}} \phi_p(x_p) + \sum_{(p,q) \in \mathcal{N}} \phi_{pq}(x_p, x_q) \tag{2.6}$$

where $\phi_p(x_p) = -\log(\Pr(y_p|x_p))$ is the likelihood function that depends on the data. We will discuss in more detail the form of the likelihood function for the segmentation task in section 2.3.1.

Fig. 2.2 illustrates the use of Markov Random Fields as priors for the task of segmentation. The negative log-likelihood functions are represented in Fig. 2.2 (b) and (c). Bright pixels correspond to pixels that disagree with the corresponding appearance models. The *Maximum Likelihood segmentation* (d) is obtained by selecting for each pixel independently the label with maximum likelihood. Not surprisingly, this results in a labelling that suffers from fragmentation and lack of spatial coherence. Using a Markov Random Field as a prior results in a smoother and more coherent segmentation.

Conditional Random Fields

Energy functions of the form of equation (2.6) also occur in inference of Conditional Random Fields [63]. A Conditional Random Field models the distribution $\Pr(\boldsymbol{x}|\boldsymbol{y})$ without explicitly modelling the joint distribution $\Pr(\boldsymbol{x}, \boldsymbol{y})$ and it can be seen as a discriminative learned counter-

part to a Markov Random Field, which is a generative model.

In general, when an energy function of the form of equation (2.6) originates from a Conditional Random Field, the potential functions ϕ_p and ϕ_{pq} are dependent on the observed data $\boldsymbol{y}$.

Beyond Pairwise Models

Pairwise energies of the form of equation (2.6) have been extensively used in vision. They include a data term preferring agreement with the observed data and a smoothness term that prefers neighbour sites to have the same label. This incorporation of priors regarding the smoothness of the labelling helps to overcome noise and uncertainty in the available data.

Recently, there has been an increased interest in models that use higher-order energy functions, i.e. the clique potentials depending on the labels of more than two sites. They have a greater expressive power and have been shown to outperform previous existing pairwise models: they better capture the statistics of natural scenes thus improve results on denoising and inpainting tasks [90]; they can encode intuitive constraints like label agreement of all pixels belonging to a superpixel [53]; and they allow for more realistic modelling of 3D surfaces with applications in stereo[114].

Learning in Random Fields

The exact form of the potential functions ϕ_c can be learned when training data is available. Methods for learning these potential functions include: *probabilistic parameter learning*, such as maximum likelihood estimation usually used for learning Markov Random Fields, and *margin-based parameter learning*, including Structured Support Vector Machines, which are used for discriminative learning of Conditional Random fields. A detailed review of these different methods can be found in [83].

In this thesis we do not address the task of learning potential functions from training data, as our focus is on optimisation algorithms for inference. We use potential functions which have been successfully used for segmentation or constraints that do not require learning, such as the connectivity constraints in chapter 3.

Optimisation methods for energy functions

The success of energy minimisation approaches is greatly due to the existence of efficient methods to optimise functions of the form (2.1).

Initially proposed optimisation methods, such as iterated conditional modes (ICM) [9] and simulated annealing [38] were very inefficient which delayed the general use of these models.

The appearance of new and more efficient optimisation techniques contributed to an increase in their use in the past years. Examples of these techniques are: Loopy Belief Propaga-

Figure 2.3: Examples of different human segmentations for the same image. Reproduced from the Berkeley segmentation dataset and benchmark [76].

tion [115, 34], Graph Cut based methods [18, 58, 46] and methods based on Linear Programming relaxations [111, 110, 60]. A comparison of these methods is provided in [105].

The efficiency of these optimisation methods is highly dependent on the size of the clique potentials used to define the energy function. Many of the methods are only feasible for pairwise or low-order energy functions. We discuss in more detail optimisation methods for binary higher-order energy functions in section 2.5.

2.3 Image segmentation

Image segmentation is a widely studied problem in computer vision. It consists of separating an image into *meaningful coherent regions*, where the exact definition of *meaningful* and *coherent* is application dependent.

One of the most common definitions of segmentation is inspired by perceptual grouping, the tendency of the human visual system to group some components of an image together and to perceive them together. Some of the most successful segmentation algorithms are designed to mimic this human behaviour [101, 35].

This definition is, however, still ambiguous. An experiment reported in [76] showed that different individuals gave different answers when presented with the same segmentation task. Fig. 2.3 shows examples of different segmentations performed by different individuals for the same image. This experiment also showed that despite the differences between the segmentations provided by the different human subjects, they are in general consistent since they can be organised in a hierarchical segmentation tree. For example, the two different segmentations for the first image in Fig. 2.3 can be seen as providing different levels of refinement for the segmentation task.

Some of these different levels of refinement correspond to well-known problems in the segmentation field, such as *superpixelization* and *multi-region segmentation*. The goal of superpixelization is to extract a segmentation where each region is consistent in terms of colour and texture. Superpixelization methods are usually used to reduce the computational burden of working at the pixel level, for example, by reducing the number of sites considered in a labelling approach. In multi-region segmentation, each region corresponds to the full extent of

Superpixelization Multi-region segmentation

Figure 2.4: Examples of different segmentation tasks. They differ in the level of refinement. While a superpixelization segments the image into small homogeneous regions, the goal of multi-region segmentation is that each segment covers an object.

a single object. This task may require a semantic understanding of the image since an object can contain different colours and textures. Multi-region segmentation has been used as an important pre-processing step for object recognition systems [94]. Fig. 2.4 shows examples of the two tasks.

In this thesis we will focus on *object segmentation* where the goal is to separate the image into only two distinct regions: *background* and *object* (alternatively referred to as *foreground*). In the rest of this thesis, *segmentation* refers to this case, unless otherwise stated.

Similarly to multiple region segmentation, object segmentation is also ill-defined. For example, if the task is posed as "segment the object in the first image of Fig 2.3", *object* can refer to different parts of the image, e.g. any of the two persons, the two persons simultaneously or to one of the helmets.

In order to address this ambiguity, some existing approaches restrict their attention to specific application scenarios, incorporating extra assumptions that constrain the problem. Some examples of the different application scenarios previously addressed include:

Interactive segmentation Assumes the existence of a user that provides information regarding the location and properties of the object of interest. Interactive techniques are commonly used in medical imaging and in commercial photo editing tools. Successful approaches to interactive image segmentation include: active contours [51], intelligent scissors [77] and graph cuts [14].

Segmentation with shape constraints In some scenarios there is information about the shape of the object of interest, which can be used as a prior. This is the case for many medical imaging applications, e.g. segmentation of the corpus callosum [69].

Class segmentation The goal is to segment objects of a certain class known *a priori*. For example, this task can be posed as "segment the horse". The class models are typically

learned from other images [12] or videos [62] containing the same object class.

2.3.1 Graph cuts

Similar to other problems in vision, segmentation can be cast as a labelling problem where the set of sites $\mathcal{V}$ is the set of pixels in the image and the set of labels is defined as $\mathcal{L} = \{0, 1\}$ where the label 0 corresponds to the *background* region and label 1 corresponds to the *object* region.

MRF models form the basis for many successful approaches to segmentation [40, 14, 91]. Recall the generic form of the energy function corresponding to a pairwise MRF:

$$E(\boldsymbol{x}) = \sum_{p \in \mathcal{V}} \phi_p(x_p) + \sum_{(p,q) \in \mathcal{N}} \phi_{pq}(x_p, x_q) \tag{2.7}$$

where x_p takes values in the set of labels $\mathcal{L} = \{0, 1\}$.

The popularity of these models is related with the existence of efficient optimisation methods for energies of the form (2.7). This energy function can be globally minimised, if the following *submodularity* condition is satisfied:

$$\phi_{pq}(0,0) + \phi_{pq}(1,1) \leq \phi_{pq}(0,1) + \phi_{pq}(1,0) \tag{2.8}$$

for all pairwise potential functions ϕ_{pq}.

If the submodularity condition (2.8) is satisfied, minimising energy in (2.7) reduces to finding an *s-t minimum cut* in a specially constructed graph [40]. We now review this construction and some related concepts.

The s-t minimum cut problem

A weighted graph $(\mathcal{V}, \mathcal{E}, \mathcal{W})$ is defined by a set of nodes $\mathcal{V}$, a set of edges $\mathcal{E}$ and an edge cost function $\mathcal{W}$ that associates to each edge (p, q)[1] a non-negative number w_{pq}. We also consider two special nodes, s and t, called terminal nodes.

An s-t cut induces a partition of the nodes into two disjoint sets, $\mathcal{S}$ and $\mathcal{T}$, such that $s \in \mathcal{S}$ and $t \in \mathcal{T}$. The cut is defined by the subset of edges $C \subset \mathcal{E}$ that connect both sets, i.e. edges (p, q) with $p \in \mathcal{S}$ and $q \in \mathcal{T}$. The cost of the cut is the sum of the weights of the edges included in the cut:

$$|C| = \sum_{(p,q) \in C} w_{pq}. \tag{2.9}$$

The *s-t minimum cut problem* consists of finding the s-t cut with the minimum cost. By the Ford-Fulkerson theorem, finding a s-t minimum cut is equivalent to computing a maximum

[1] Note that, edge (p, q) is equivalent to (q, p) since we consider an undirected graph.

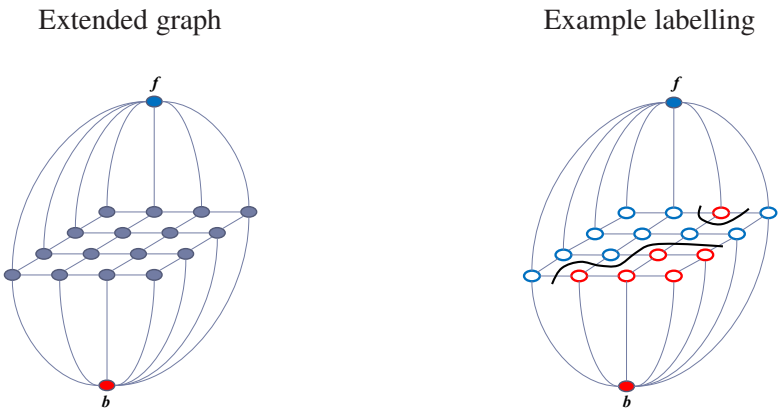


Figure 2.5: Illustration of the extended graph construction for graph cut methods and of a possible labelling assignment. Note that not all edges are represented.

flow from s to t [36], and there exist polynomial time algorithms to compute this.

Graph Construction

In order to review the graph construction for minimising the energy function (2.7) we restrict the form of the pairwise potentials to $\phi_{pq}(x_p, x_q) = a_{pq}|x_p - x_q|$, where a_{pq} is a non-negative constant. This restriction is only considered for simplicity of presentation and a similar graph construction exists for any submodular energy [58].

The set of nodes is defined as $\bar{\mathcal{V}} = \mathcal{V} \cup \{f, b\}$, containing a node per site in the image and the two terminal nodes, f and b , associated with label 1 (foreground) and 0 (background) respectively[2]. The set of edges, $\mathcal{E}$, contains two types of edges: *n-links* (neighbourhood links) connecting neighbouring sites and *t-links* (terminal links) of the form (p, f) and (p, b) connecting each node $p \in \mathcal{V}$ with the two terminal nodes, i.e. $\mathcal{E} = \mathcal{N} \cup \{(p, q)|p \in V\, q \in \{f, b\}\}$. For each edge, its weight is defined as follows:

$$w_{pq} = a_{pq} \text{ if } p, q \in \mathcal{V}; \tag{2.10a}$$

$$w_{pf} = \phi_p(0); \tag{2.10b}$$

$$w_{pb} = \phi_p(1). \tag{2.10c}$$

Figure 2.5 shows an illustration of the graph construction.

The main property of this graph is that the cost of the minimum cut is equal to the minimum of the energy function. The corresponding optimal labelling can be recovered by observing that the minimum cut contains exactly one t-link for each node $p \in \mathcal{V}$. Suppose that for node p the t-link that belongs to the minimum cut is (p, f). In this case, the optimal label assigned to node

[2]We refer to the terminal nodes as f and b as opposed to s and t to emphasise that they are associated with the labels *foreground* and *background*.

p is 0. Similarly, if the minimum cut contains the t-link (p, b) the node p is assigned the label 1.

The success of MRF based methods for binary segmentation is related with the existence of efficient optimisation techniques based on graphs, therefore, the MRF based approaches for binary segmentation are commonly referred to as *graph cut* methods.

The use of graph cut methods for energy minimisation in vision problems motivated specially designed maxflow algorithms that take advantages of the specific properties of graphs arising in computer vision (grid graphs, where each node has a small number of neighbours) [16] and new dynamic algorithms for sequential computation of maximum flows [55].

Energy modelling

It remains to describe the exact form of the potentials used in graph cut methods.

As with previously described MAP-MRF models, the unary potentials measure the agreement between the data y_p and a probabilistic model associated with the label assigned to p and are typically called data costs. $\phi_p(x_p)$ corresponds to a likelihood term derived from the appearance models. Given the probabilistic appearance models θ^1 and θ^0 for foreground and background respectively, the unary potentials are defined as follows:

$$\phi_p(0) = -\log(\Pr(y_p|\theta^0)) \qquad \phi_p(1) = -\log(\Pr(y_p|\theta^1)) \tag{2.11}$$

where y_p is the observed data for site p.

When the observed data y_p consists of the grey value or RGB colour of pixel p, empirical histograms or Gaussian Mixture Models (GMMs) are commonly used as appearance models [14, 11, 91]. These models can be either learned from similar training data or from user provided scribbles [14], in the case of interactive segmentation.

The pairwise potentials $\phi_{pq}(x_p, x_q)$ encode the prior assumption of labelling smoothness and they have been previously defined as contrast sensitive terms of the form [14]:

$$\phi_{pq}(x_p, x_q) = w_{pq}|x_p - x_q| \;\text{ with }\; w_{pq} = \frac{1}{\text{dist}\,(p,q)}\left(\lambda_1 + \lambda_2 \exp -\beta \left\|y_p - y_q\right\|^2\right) \tag{2.12}$$

where λ_1 and λ_2 are positive weights for the different terms, dist (p, q) is the Euclidean distance between nodes p and q, and $\beta = \left(2\left\langle (y_p - y_q)^2 \right\rangle\right)^{-1}$, where $\langle\cdot\rangle$ denotes expectation over the image. Although in a MRF model the pairwise terms do not depend on the data, since they correspond to the prior distribution, this dependency can be justified in the Conditional Random Field framework [63].

Note that the definition of the pairwise term in equation (2.12) is an ad-hoc function that has been successfully and extensively used in segmentation, e.g. [14, 91]. A pairwise term of

this form penalises discontinuities between pixels with similar colour, where the parameter β defines a threshold for the similarity. Furthermore, the choice of weights λ_1 and λ_2 is crucial to balance the importance of the different parts of the model. If both these terms are set to 0, the model reduces to a per-pixel labelling without spatial coherency. Increasing these weights favours smoother segmentations and in the extreme scenario where the weights are set to infinity, the segmentation that minimises the energy is constant, i.e. all the pixels are assigned the same label.

Throughout this thesis we fix the values of these terms to $\lambda_1 = 2.5$ and $\lambda_2 = 47.5$ when using GMMs as colour models and $\lambda_1 = 1$ and $\lambda_2 = 10$ when using histograms[3]. These values were hand picked by visually inspecting the results achieved for different test values. For a more principled choice of values we could resort to the learning techniques briefly discussed in the previous section.

Alternative interpretations of the graph cut model

Interestingly, the pairwise potentials can also be interpreted as measuring the length of the implicit contour defined by the segmentation. These potentials can be defined in order to approximate the length related with any Riemannian metric and this approximation can be made arbitrarily accurate by increasing the local neighbourhood size [15]. Therefore, graph cut methods provide exact optimisation of a discrete version of continuous functionals for length regularisation.

A different interpretation of the graph cut model was provided in [103]. The graph cut model fits into the more generic class of energy minimisation problems, with an energy function of the form:

$$E(\boldsymbol{x}) = \sum_{(p,q)\in\mathcal{E}} (w_{pq}|x_p - x_q|)^i \tag{2.13}$$

where $x_p \in [0, 1]$. For $i = 1$ this energy reduces to the graph cut energy. Notably, although the formulation allows for continuous labels, for $i = 1$ this energy has a binary minimiser [84]. For other values of i, this model corresponds to other segmentation algorithms: random walker for $i = 2$ [39] and geodesic distance for $i = \infty$ [103]. It is less clear for these models how to select a binary segmentation from the continuous solution obtained from minimising the energy.

Changing the value of i affects the degree of *shrinking bias* in the final solution. The shrinking bias consists of a preference towards shorter boundaries and is more evident for $i = 1$, while the random walker algorithm is less affected.

[3]The difference is justified by the properties of the unary term $\phi_p(x_p)$ in both scenarios. While for histograms we have $\phi_p(x_p) \geq 0$, since $\Pr(y_p|\theta^{x_p}) \leq 1$ this may not be the case for the GMM model since a probability density function is used instead.

2.3.2 Continuous formulation

Discrete models for segmentation, like MRF based models, are justified by the discrete nature of digital images. However, the world captured by the images is not spatially discrete. This motivated methods that are based on a continuous representation of the image, where $I : \Omega \to \mathbb{R}$ with $\Omega \subset \mathbb{R}^2$ being the representation of a grey scale input image.

Contour based methods

Contour based methods for segmentation are based on the assumption that an object boundary usually aligns with strong intensity gradients in the image. The task of segmentation is then formulated as an energy minimisation problem in the space of all possible contours, where the cost of a contour, C, depends both on internal and external properties [51]. The internal properties measure the smoothness of the contour while the external properties attract the contour to edges in the image:

$$E(C) = \int_C \underbrace{\alpha|C_v(v)|^2 + \beta|C_{vv}(v)|^2}_{\text{Internal properties}} \underbrace{-|\nabla I(C(v))|}_{\text{External properties}} \, dv \tag{2.14}$$

where C_v and C_{vv} are the first and second derivatives of C with respect to contour parameter v and α, β are weights of the different parts of the model.

Interestingly, when $\beta = 0$, this formulation is equivalent to computing *geodesics* (curves of minimum length) in a Riemannian space induced by the image [24].

These models were first minimised by gradient descent [51], which do not allow for topological changes of the initial contour, and later by level set methods [24]. Both minimisation techniques are local methods that require initialisation and do not guarantee global optimality. Furthermore, most energy functions based on contours have trivial global optima (an infinitesimally small curve), making the use of local methods a necessity.

Region based methods

Region based methods aim at identifying regions of smooth (or homogeneous) intensity. The Chan-Vese functional for segmentation is an example of a region based method [25]. It assumes that the image is formed by two regions with approximately constant intensities and that the average intensity for each region have distinct values c_1 and c_2. It can be interpreted as a restriction of the Mumford-Shah functional [79] to two regions with constant intensities.

The goal is to jointly estimate the segmentation (represented by its contour C) and the average intensity values c_1 and c_2 of each region and it is formulated with an energy function

of the form:

$$\begin{aligned} E(C, c_1, c_2) =& \lambda_L \times \text{Length}(C) + \lambda_A \times \text{Area}(inside(C)) \\ &+ \lambda_I \int_{inside(C)} |y_p - c_1|^2 \, dp + \lambda_O \int_{outside(C)} |y_p - c_2|^2 \, dp \end{aligned} \tag{2.15}$$

where λ_L, λ_A, λ_I and λ_O are weights for the different components of the model and y_p is the intensity for site p.

This function is minimised by iteratively alternating between two steps. In the first step, the contour is fixed and c_1 and c_2 are computed as a function of the contour: c_1 is the average intensity of pixels p inside the contour and c_2 is the average intensity of pixels outside the contour. In the second step, both c_1 and c_2 are fixed and (2.15) is minimised with respect to the contour C. This is achieved using level sets.

Global optimisation by convex relaxation

One of the drawbacks of classical continuous approaches is the use of local minimisation methods [51, 24, 79, 25].

Recently, it has been shown that energy functions that combine both contour and region properties can be globally optimised by solving a convex relaxation of the problem (see [27] for a review of optimisation techniques).

An important result is that, for the case of binary segmentation, solving the convex relaxation and thresholding the continuous solution provides a global solution for the original non-convex labelling problem.

2.3.3 Interactive segmentation

As previously mentioned, segmentation is a well-defined problem in an interactive scenario, since there is a user specifying what is the object of interest. The goal of interactive segmentation systems is to assist the user in extracting the desired object, while minimising the effort required to perform the task.

Different interactive segmentation models have been proposed and they require different user input. The input can be of the form of an initial contour (Fig. 2.6 (a)) as in the case of active contours methods [51, 24]. Boundary seeds (Fig. 2.6 (b)) are another popular form of user input for methods that trace the boundary of the object [77]. In graph based methods [14, 39, 86] the user provides foreground and background seeds by brushing some pixels (Fig. 2.6 (c)). Throughout the thesis, we alternatively refer to region seeds as *scribbles* or *brush strokes*. Alternatively, some methods require only one type of region seed in the form of a bounding box surrounding the object [91] (Fig. 2.6 (d)).

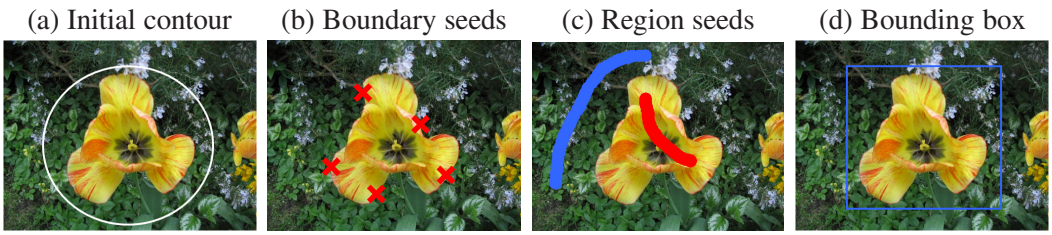


Figure 2.6: Illustration of different types of user interaction

The different types of user input have different properties. Regions seeds require less precision from the user, while boundary seeds are not appropriate for objects with very complicated boundaries, like trees, since they require more seeds. A bounding box surrounding the object is an intuitive and minimal form of user input. However, it may have to be complemented with other forms of user input, such as region seeds, when the method provides an incorrect initial segmentation [91].

Recently, there has been some interest in methods that combine different types of user input giving more flexibility to the user [71, 72] and in comparing methods taking into account the amount of user interaction needed to produce similar results [80].

2.4 Higher-order models for segmentation

Models that include pixel-based costs and local consistency, in particular graph cut based models, have been successfully applied to segmentation. However, they have some limitations: they require apriori known appearance models or foreground and background seeds in order to estimate them; they do not encode higher-order properties of the boundary, like curvature and boundary continuity; they do not incorporate higher-order properties of the segmentation region, like class specific shape priors or topological constraints.

Different higher-order models were proposed to overcome some of these limitations. In this thesis, *higher-order model* refers to any model that encodes properties of the segmentation beyond the traditional assumptions: data agreement with a fixed appearance model and labelling smoothness. In the energy minimisation framework, higher-order models usually are defined using higher-order potentials (i.e. potentials depending on more than two variables) or using extra (possibly multilabel) auxiliary variables.

In this section we discuss some of the previously proposed higher-order models for segmentation, focusing on models formulated as minimisation of discrete energy functions.

2.4.1 Appearance models as variables

The Chan-Vese functional discussed in section 2.3.2 is an example of a model that includes the appearance of each region as a variable. Recall that the goal is to jointly infer the segmentation

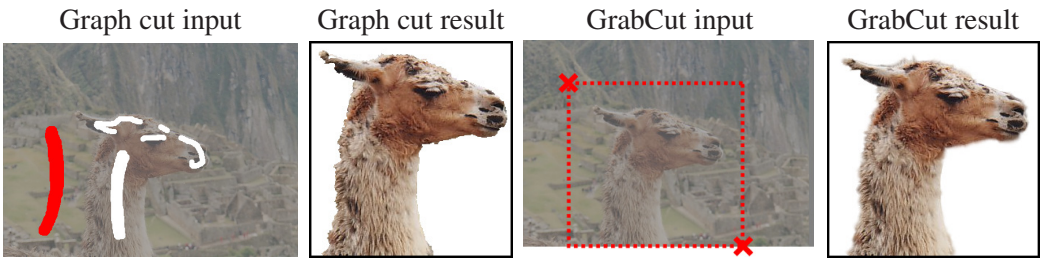


Figure 2.7: Comparison of traditional graph cut methods [14] with GrabCut [91]. Both methods give comparable results, but the GrabCut model requires less user interaction. Images reproduced from [91].

(C) and the average intensity value of each piecewise constant segment (c_1 and c_2) and that the optimisation is performed by alternating between estimating the segmentation and c_1 and c_2.

Although this is a popular approach it is also limited, since it can only model piecewise constant images. Such approach can be extended in a probabilistic framework, by replacing the last two terms of (2.15) with

$$\lambda_I \int_{inside(C)} -\log(\Pr(y_p|\theta^1))\, dp + \lambda_O \int_{outside(C)} -\log(\Pr(y_p|\theta^0))\, dp \tag{2.16}$$

where θ^1 and θ^0 are probabilistic models for the appearance of the two segments [28]. Note that this expression is similar to (2.11), however in the formulation considered in section 2.3.1 the appearance models were fixed, i.e. they were not a variable in the model.

The joint optimisation of probabilistic appearance models together with the segmentation in a discrete setting forms the basis of the popular GrabCut approach [91]. The appearance models considered in [91] were GMMs over RGB colour, which allow a rich representation of image colour. The optimisation was performed in a similar iterative way, alternating between estimating the colour models and estimating the segmentation using graph cuts.

In the context of interactive image segmentation, these models have the advantage of coping with incomplete user input, in contrast with the graph cut model discussed in section 2.3.1, which requires region seeds for both segments in order to compute the appearance models (θ^0, θ^1). A model like GrabCut allows for alternative forms of user input (e.g. a bounding box surrounding the object) and requires less user interaction [91]. Fig. 2.7 compares traditional graph cut methods and GrabCut in terms of the input required and the result obtained.

The fact that GrabCut can cope with user input in the form of a bounding box is relevant for applications other than interactive segmentation. Methods for class specific object detection provide as output a bounding box surrounding the object and this bounding box can similarly be used as input to GrabCut (see e.g. [2]).

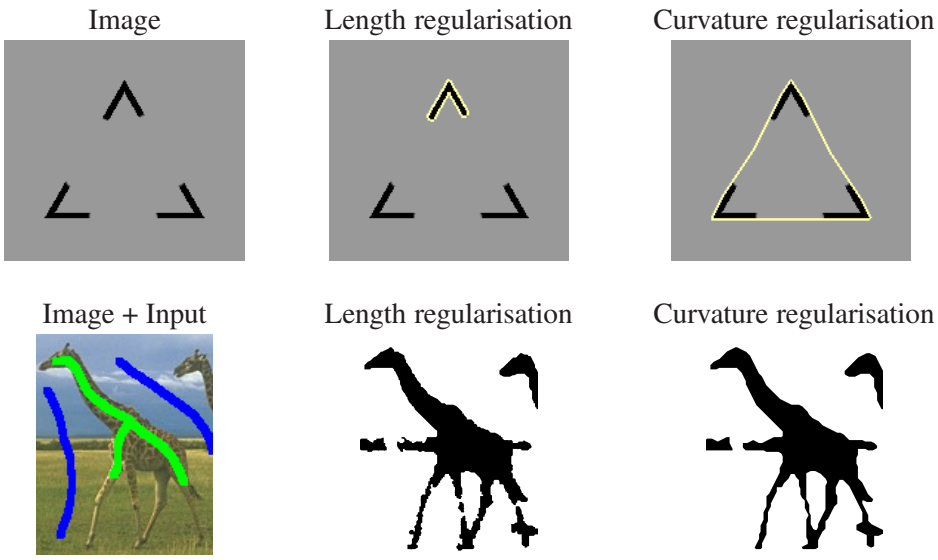


Figure 2.8: Curvature regularisation. Curvature better mimics human perception in the task of segmenting illusory contours and improves boundary continuity, leading to a solution that is less fragmented. Images reproduced from [99, 100].

A global optimisation method for a model that jointly optimises over the segmentation and appearance models will be presented in chapter 4.

2.4.2 Boundary properties

The inclusion of length regularisation is common to both continuous and discrete approaches to segmentation [51, 24, 15]. This regularisation is important to extract smooth contours but it has an undesirable bias towards short boundaries, known as the *shrinking bias*.

An interesting alternative to length regularisation is to combine different functionals defined either along or inside the contour, by minimising their ratio [49]. In some cases, this can be done efficiently by finding cycles of negative weight in a graph [49] or by using parametric maxflow [56]. [49] proposes to minimise the ratio between the flux of a vector field over the boundary length. This ratio has no bias towards a particular shape and it is scale independent, overcoming the *shrinking bias* of length based regularisation.

Curvature regularisation has also been introduced in the context of ratio minimisation [99]. Curvature is known to better mimic human perception in the task of segmenting illusory contours and improves boundary continuity in the presence of noise or missing data. Examples are shown in Fig. 2.8. Posteriorly, curvature regularisation was used in a region based approach [100] and optimised using linear programming relaxation. In [104] the authors show that curvature regularisation can be expressed by an energy function with cliques of size four.

A model that explicitly favours long homogeneous boundaries was introduced in [48]. The method starts by extracting long homogenous chains that are plausible boundaries of the object.

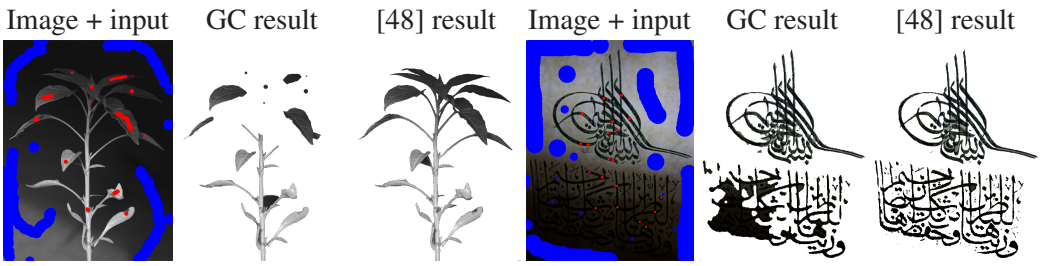


Figure 2.9: Extraction of long homogeneous boundaries. Graph cut methods (GC) suffer from a *shrinking bias* since they include a length regularisation. This bias can be overcome by discounting the cost associated to long homogeneous boundaries. Images reproduced from [48].

The main observation is that the additive pairwise cost assigned to these boundary chains can be replaced by a submodular function, i.e. each edge in the chain added as a boundary edge would not directly be added as a cost, and instead each chain contributes with a diminishing joint cost. This formulation favours adding edges of a chain to the boundary, if other edges of the chain are already included in the boundary. In practice, it helps overcoming the *shrinking bias* of length based methods when the object has well defined, sharp boundaries. Results comparing the traditional length regularisation of graph cut methods and this new model are shown in Fig. 2.9.

Note that, the work described in chapter 3 pre-dates and partially motivates some of these methods ([100] and [48]).

2.4.3 Shape priors

Commonly used energy minimisation methods include a region term in the form of a pixel-based cost. A term of that form is useful to encode pixel-based preferences for one of the labels. However, it can be limited when there is ambiguity between the appearances of both segments, even when combined with regularisation in the form of a smoothness prior.

To overcome this limitation, one possibility is to further constrain the problem, by only allowing segmentations that follow a predefined shape. This type of constraint is called a *shape prior* and requires *apriori* information regarding the object of interest. Successful methods that incorporate shape priors have been used in medical imaging [69] and segmentation of objects from a predefined class, e.g. [12, 62].

The first step of these methods is to find a convenient probabilistic representation of allowed shapes that captures their properties and diversity, usually by using training examples. This shape representation can then be used in an iterative framework that alternates between better adapting the shape to the current segmentation and updating the segmentation based on

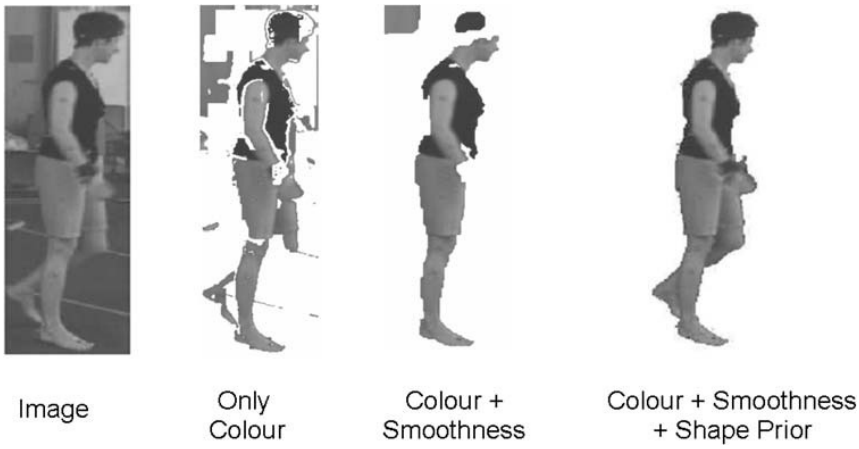


Figure 2.10: Class specific shape priors for segmentation. Comparison of results for sequentially more complex models that include appearance, smoothness and class specific shape priors. Image reproduced from [54].

the shape. Commonly used energy functions have the form [62, 54]:

$$E(\boldsymbol{x}, \Theta) = \sum_{p \in \mathcal{V}} [\phi_p(x_p) + \varphi_p(x_p, \Theta)] + \sum_{(p,q) \in \mathcal{N}} \phi_{pq}(x_p, x_q) \tag{2.17}$$

where Θ is a continuous variable that represents the location and pose of the shape prior.

Although shape models have been shown to significantly improve the quality of the segmentation (see Fig. 2.10 for an example), they are limited to applications where *apriori* knowledge and exemplar shapes for learning are available. This limitation motivated generic shape priors that are not limited to a specific object class.

An example of a generic shape prior is the *star shape prior* introduced in [107]. A star shape is defined with respect to a centre point c. A segmentation follows this prior if for any point p in the foreground, all points in the straight line connecting c and p are also foreground. Convex shapes are a special case of star shapes, since any point can be chosen as the central point. Also, star shapes are a special case of connected shapes. Fig. 2.11 illustrates these different properties of shapes.

Although the star shape prior encodes higher order properties of the segmentation, it can be imposed using a pairwise energy function of the form [107]:

$$E(\boldsymbol{x}, c) = \sum_{p \in \mathcal{V}} \phi_p(x_p) + \sum_{(p,q) \in \mathcal{N}} \phi_{pq}(x_p, x_q) + \sum_{(p,q) \in \mathcal{N}} S_{pq}(x_p, x_q) \tag{2.18}$$

where the last term assumes that p and q are neighbour pixels in a line passing through the

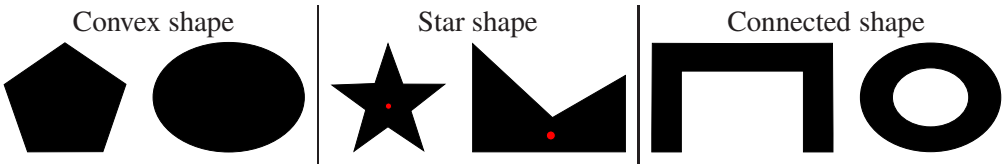


Figure 2.11: Examples of different types of shape.

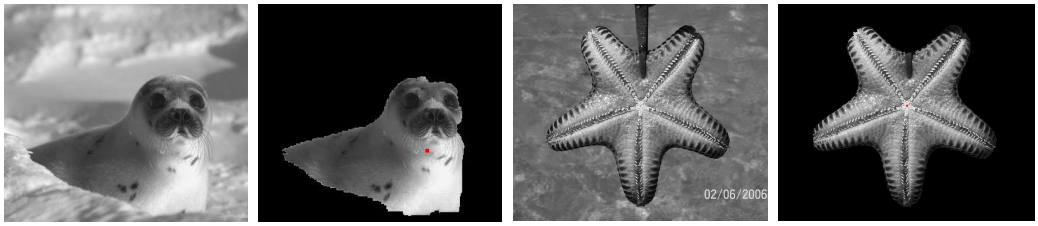


Figure 2.12: Results obtained using the star shape prior. Images reproduced from [107].

centre pixel c, q is in between p and c, and the pairwise term is defined as:

$$S_{pq}(x_p, x_q) = \begin{cases} 0 & \text{if } x_p = x_q \\ \infty & \text{if } x_p = 1 \text{ and } x_q = 0 \\ \beta & \text{if } x_p = 0 \text{ and } x_q = 1 \end{cases} \tag{2.19}$$

where β is a constant that controls the size of the segmentation. A pairwise term of this form ensures that if p belongs to the segmentation, all the pixels contained in the line segment connecting p and c also belong to the segmentation. Since this energy function is submodular, the model can be globally optimised using graph cuts.

Fig. 2.12 shows results obtained using this prior. In [42] the star shape prior was extended to multiple stars and geodesic paths, as opposed to straight lines, in the context of interactive image segmentation.

Since a star shape is a special case of a connected shape, a natural next step is to consider connectivity priors. In contrast with the star shape prior, a connectivity prior leads to an NP-hard optimisation problem. Connectivity priors will be discussed in detail in chapter 3.

So far, the shape models discussed correspond to prior models that do not depend on image information. A different type of region based higher-order model was proposed in [52] and extended in [53]. The motivation for those approaches is the excessive smoothness effect present in pairwise models. Although smoothness improves results over likelihood only models (see Fig. 2.2), it has a known *shrinking bias* and it can over smooth complex boundaries. In contrast, some unsupervised superpixelization methods are able to extract small segments that closely follow object boundaries. Ideally, we would then use superpixelization methods as a

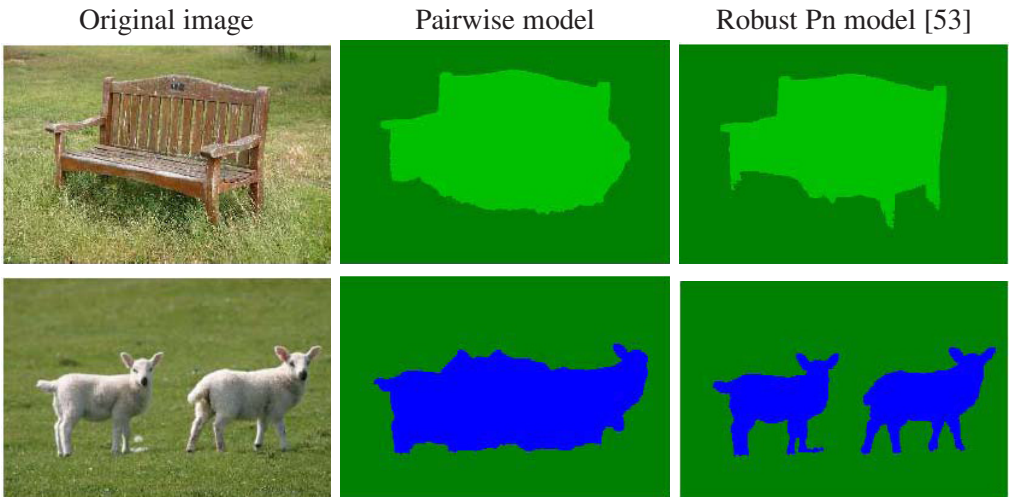


Figure 2.13: Using superpixels as a soft constraint helps overcoming the oversmoothing effect of the pairwise model. Images reproduced from [53].

pre-processing step and formulate the labelling problems in the superpixel level, i.e. assign a label to each superpixel.

Fully replacing pixels by superpixels in the inference process has, however, some drawbacks. In particular, superpixels do not always respect object boundaries. To alleviate this problem, some authors propose to use multiple superpixelizations [94, 75]. Alternatively, the methods proposed in [52, 53] impose superpixels as soft constraints and do not discard the pixel level, combining the best of both approaches: superpixels help overcoming oversmoothing and maintaining the pixel level can help to recover from an incorrect superpixelization.

For binary segmentation, the problem formulated in [52, 53] uses a higher-order energy function of the form:

$$E(\boldsymbol{x}) = \sum_{p \in \mathcal{V}} \phi_p(x_p) + \sum_{(p,q) \in \mathcal{N}} \phi_{pq}(x_p, x_q) + \sum_{c \in S} \phi_c(x_c) \tag{2.20}$$

where S is the set of all superpixels and the clique potential for each superpixel is defined as:

$$\phi_c(x_c) = g\left(\sum_{p \in c} x_p\right) \tag{2.21}$$

with $g(.)$ a concave function. By choosing an appropriate function $g(.)$, a clique potential of this can be used to encourage all the pixels in a superpixel to take the same label. Interestingly, this higher-order function can be converted into a pairwise function by adding extra binary variables, leading to a submodular pairwise function that can be optimised using graph cuts.

Energy functions of this form have been successfully used for semantic image labelling

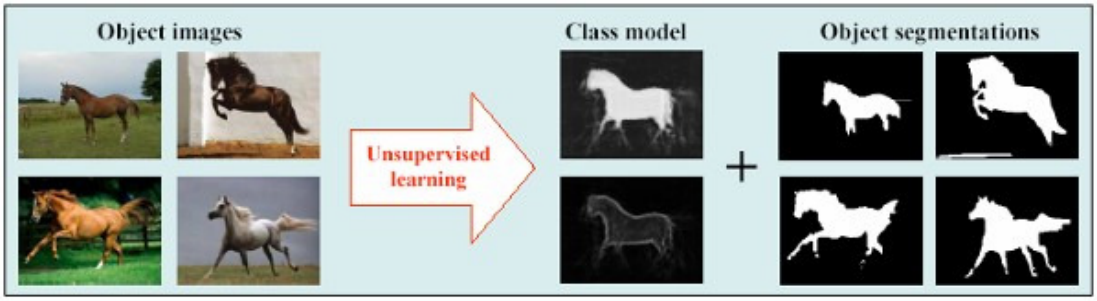


Figure 2.14: LOCUS model for segmenting multiple images. LOCUS builds a class model from multiple images of the same object and uses that model to segment the individual images. Image reproduced from [113].

in [53] and shown to be a good counterbalance to the oversmoothing effect of pairwise terms (Fig. 2.13).

2.4.4 Segmentation of multiple images

The models discussed so far are designed to improve the segmentation accuracy of a single image, by incorporating assumptions and constraints that are usually observed for objects.

A different type of higher-order model arises when the goal is to segment multiple images jointly. In this scenario, the model should identify and make use of the information that is common to multiple images, in order to improve the segmentation of each individual image.

The LOCUS model [113] is an example of such a higher-order model. It is applicable to images containing objects of the same class in a similar pose, e.g. left facing horses. It favours segmentations with similar shape across the different images, allowing for specific object appearance in each individual image. Fig. 2.14 illustrates this approach.

A different approach is followed by cosegmentation methods [93, 78, 45]. The goal is to find segmentations that match in terms of appearance, favouring foreground segments with the same appearance histogram. This is applicable to images where the object has a similar appearance but considerable variation in terms of pose.

2.5 Optimisation methods and Dual Decomposition

In the following chapters, we will discuss higher-order models that use an energy formulation that has the following generic form:

$$E(\boldsymbol{x}) = \sum_{c \in \mathcal{C}} \phi_c(\boldsymbol{x}_c) \tag{2.22}$$

with $x_p \in \{0, 1\}$.

As discussed in section 2.3.1, if c contains at most two pixels (pairwise energy function) and all terms are submodular, then the energy can be globally minimised using graph cuts. If these assumptions are relaxed, the problem is in general NP-hard. In this section, we discuss optimisation methods useful for such energy functions. We are particularly interested in global optimisation methods, i.e. methods capable of producing a global minimum (for some instances) and that provide a certificate of optimality. We will discuss in detail one of these methods: Dual Decomposition.

QPBO for non-submodular energy functions

Pairwise energy functions that do not satisfy the submodularity condition can be optimised using QPBO, a graph cut algorithm in a specially constructed graph (see [57] for a review).

Since the problem becomes NP-hard, there is no guarantee that the global optimum will be achieved. Instead, QPBO finds an optimal solution of a linear relaxation of the original problem, that allows $x_p \in [0, 1]$. The solution, $\boldsymbol{x}$, of the relaxed problem has some important properties:

1. $x_p \in \{0, \frac{1}{2}, 1\}$;
2. if x_p is integer for all p, then this solution is the global solution of the original problem;
3. there exists a global minimum of the original problem, $\boldsymbol{x}^*$, such that $x_p^* = x_p$ for all nodes p with $x_p \in \{0, 1\}$.

A node p is called unlabelled if $x_p = \frac{1}{2}$. From these properties, it follows that the QPBO method provides a partial solution for the problem and its efficacy is measured by the number of nodes left unlabelled.

Methods for higher-order functions

Several methods for higher-order energies reduce the energy function to a pairwise function [58, 53, 47]. The construction in [47] is generic and applicable to any higher-order function. However, in a worst case scenario, it introduces an exponential number of auxiliary variables. Furthermore, it leads in general to non-submodular pairwise energies. In practice, this construction is only useful for energy functions with small clique size (in [47] the cliques used have maximum size four).

Other interesting reductions, such as [52, 53], introduce a limited number of auxiliary variables and lead to submodular pairwise functions. However, they are only applicable to potentials of a special form.

Alternative optimisation algorithms based on efficient belief propagation and message passing techniques [64, 59] have also been proposed. However, they are also limited to cliques of small size or higher-order cliques of a special form [106].

2.5.1 Dual Decomposition

In the following chapters, we will discuss higher-order energy functions that have cliques with a large size. In some cases, the cliques considered include all the pixels in the image. We use the term *global potential* to emphasise this property, in contrast with the alternative *higher-order potential* that refers to any potential function depending on more than two variables.

These global potentials prevent the use of the minimisation techniques discussed previously, since they are restricted to cliques of small size. Therefore, we use Dual Decomposition, a standard technique for solving combinatorial optimisation problems [8].

The main idea of *Dual Decomposition* (also named *Lagrangian Decomposition*) is to decompose the original problem into several "easier" subproblems. Combining the minima of different subproblems gives a lower bound on the original energy.

The original minimisation problem is given by:

$$\min_{\boldsymbol{x}} E(\boldsymbol{x}) = \sum_{c \in \mathcal{C}} \phi_c(\boldsymbol{x}_c). \tag{2.23}$$

To use Dual Decomposition it is first necessary to identify a split of the energy function into components that are easier to optimise separately. We consider the simplest example of a split into only two subproblems. We assume the set of all cliques, $\mathcal{C}$, can be separated accordingly into two disjoint sets $\mathcal{C}_1$ and $\mathcal{C}_2$, such that $\mathcal{C}_1 \bigcup \mathcal{C}_2 = \mathcal{C}$. Then, the optimisation problem (2.23) can be equivalently written as:

$$\min_{\boldsymbol{x}_1, \boldsymbol{x}_2} \sum_{c \in \mathcal{C}_1} \phi_c(\boldsymbol{x}_{1c}) + \sum_{c \in \mathcal{C}_2} \phi_c(\boldsymbol{x}_{2c}) \tag{2.24a}$$

$$\text{s.t.} \boldsymbol{x}_1 = \boldsymbol{x}_2 \tag{2.24b}$$

where the variables were duplicated and a consistency constraint (2.24b) was added to make the problem equivalent with the original problem (2.23).

Since we assume the optimisation of the two subproblems is easier if done separately, the constraint (2.24b) can be seen as a "complicating" constraint that connects otherwise separate subproblems.

Dual Decomposition is equivalent to Lagrangian relaxation of those "complicating" constraints. We form the Lagrangian function by relaxing the constraints (2.24b) and introducing

Lagrangian multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{\mathcal{V}}$:

$$L(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{\lambda}) = \sum_{c \in \mathcal{C}_1} \phi_c(\boldsymbol{x}_{1c}) + \sum_{c \in \mathcal{C}_2} \phi_c(\boldsymbol{x}_{2c}) + \langle \boldsymbol{\lambda}, \boldsymbol{x}_1 - \boldsymbol{x}_2 \rangle \tag{2.25}$$

where $\langle \boldsymbol{\lambda}, \boldsymbol{x} \rangle = \sum_p \lambda_p x_p$.

Minimising the Lagrangian over $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ gives the dual function $\Phi(\boldsymbol{\lambda})$, a lower bound on the original problem:

$$\Phi(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}_1, \boldsymbol{x}_2} L(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{\lambda}) \tag{2.26a}$$

$$= \min_{\boldsymbol{x}_1} \left[\sum_{c \in \mathcal{C}_1} \phi_c(\boldsymbol{x}_{1c}) + \langle \boldsymbol{\lambda}, \boldsymbol{x}_1 \rangle \right] + \min_{\boldsymbol{x}_2} \left[\sum_{c \in \mathcal{C}_2} \phi_c(\boldsymbol{x}_{2c}) - \langle \boldsymbol{\lambda}, \boldsymbol{x}_2 \rangle \right] \tag{2.26b}$$

$$\Phi(\boldsymbol{\lambda}) \leq E(\boldsymbol{x}) \tag{2.26c}$$

The *Dual problem* is to find the tightest possible bound by solving $\max_{\boldsymbol{\lambda}} \Phi(\boldsymbol{\lambda})$. Since the function $\Phi(\boldsymbol{\lambda})$ is concave and for a fixed value of $\boldsymbol{\lambda}$ it can be efficiently evaluated by minimising the two subproblems separately, we use the subgradient method to solve the dual problem. In general, we obtain the solution to the original problem by selecting one of the solutions of the individual subproblems.

One of the main benefits of the Dual Decomposition method is that it provides a lower bound. This lower bound allows to assess the optimality of the solution in a per-instance basis.

The Dual Decomposition approach has been previously used in vision, most notably for inference in multilabel pairwise MRF models [110, 60]. In these approaches the subproblems considered are inference on trees which can be solved efficiently. The methods used to solve the dual problem were message passing techniques [110], which do not necessarily find the best lower bound, and the subgradient method [96, 97, 60].

Subgradient method

The subgradient method is an iterative method for minimising convex, typically non-differentiable, functions (or equivalently maximise concave functions). Given the convex problem $min_x f(x)$, the subgradient method uses the following iteration to minimise f:

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)} \tag{2.27}$$

where $g^{(k)}$ is a subgradient of f at $x^{(k)}$ and α_k is the step size. If f is differentiable, the only possible choice for $g^{(k)}$ is the gradient vector $\nabla f(x^{(k)})$.

Since the subgradient method is not necessarily a descent method[4], we keep track of the lowest value of f found so far, f_{best}, and the corresponding solution, x_{best}.

When the subgradient method is applied in the context of Dual Decomposition, a subgradient direction is given by combining the solution of the two subproblems: $g^{(k)} = \boldsymbol{x}_1 - \boldsymbol{x}_2$, where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are solutions of the two subproblems for the current value of $\boldsymbol{\lambda}$.

It remains to specify how to choose the step size. Different rules can be applied, in particular if the step size follows a nonsummable diminishing rule:

$$\lim_{k\to\infty} \alpha_k = 0, \quad \sum_{k=1}^{\infty} \alpha_k = \infty \tag{2.28}$$

the algorithm is guaranteed to converge to the optimal value [8]. In practice, more elaborate step size rules may achieve better performance.

We will use an adaptive technique mentioned in [8]. We set $\alpha_k = (f^{best} + \delta - f(x^{(k)}))/||g^{(k)}||^2$ where δ is a positive number which is updated as follows: if the last iteration improved the best lower bound f_{best} then δ is increased by a certain factor (2 in our experiments), otherwise it is decreased by a certain factor (0.95).

Subgradient methods have some advantages that make them appropriate to use for solving the *Dual problem*: they are guaranteed to converge when using an appropriate step size rule and their efficiency relies on efficient optimisation techniques for each of the subproblems.

2.6 Conclusion

In this chapter we reviewed energy based methods for image segmentation. We started by describing the generic MAP-MRF framework for labelling problems. These methods have been successfully applied to image segmentation and their success is highly related with the existence of efficient optimisation algorithms based on graph cuts.

Recently, there has been a grown interest in models that go beyond the traditional assumptions, such as labelling smoothness, and that incorporate useful object properties, like shape and boundary continuity. However, these properties cannot usually be encoded using pairwise energies and in many cases correspond to NP-hard problems. This implies that there is no guarantee that an optimisation method will reach the best solution. Furthermore, the optimisation methods that are traditionally used for vision problems are not always applicable and more generic optimisation techniques, like Linear Programming, have been successfully used. We reviewed one of those techniques, Dual Decomposition, that will be extensively used in later chapters.

[4]For a convex differentiable function the subgradient corresponds to the gradient, which is always a descent direction, but if the step length α is too large the function value may increase.

Chapter 3

Connectivity of segmentation

3.1 Introduction

Generic higher-order region priors that can be applicable to different types of objects without *apriori* knowledge about their shape or class are desirable to disambiguate and better constrain the task of segmentation.

One of such priors is *connectivity*. Connectivity is a natural constraint for segmentation since 3D objects are typically connected. Although this does not necessarily translate to connectivity in the 2D image domain, in practice, many 2D representations of objects also have this property.

In this chapter, we analyse the problem of minimising a pairwise energy function, common to graph cut based methods, subject to certain connectivity constraints. Although the most natural constraint is to impose that a segmentation is fully connected (Constraint **C0**), we will focus on a different constraint: we enforce connectivity only between two special nodes, which we refer to as *terminal nodes*, disregarding the rest of the segmentation (Constraint **C1**). The difference between these two constraints is illustrated in Fig. 3.1 and they are formally defined in section 3.2.

Our choice of constraint **C1**, instead of constraint **C0**, is justified by the relative simplicity of designing efficient heuristic algorithms for **C1**. Although, constraint **C1** is a less intuitive

Constraint **C0** Constraint **C1**

Figure 3.1: Connectivity constraints. Constraint **C0** corresponds to a full connected segmentation, while constraint **C1** only enforces connectivity between the two terminal nodes (highlighted). We will focus on constraint **C1**.

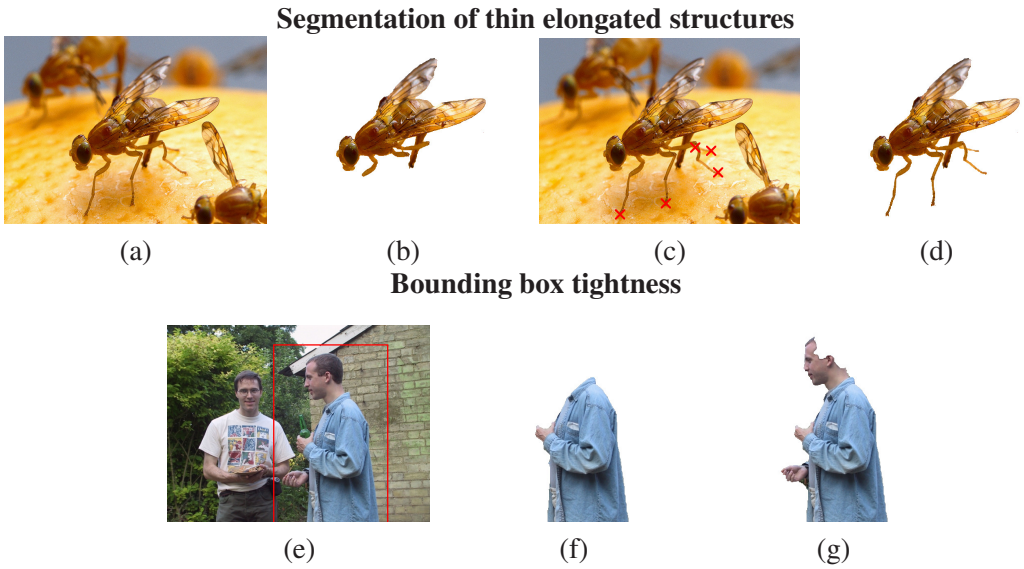


Figure 3.2: Different tasks in interactive segmentation benefit from using the connectivity constraint discussed in this chapter. ***Segmentation of thin elongated structures.*** To correct the initial segmentation obtained with graph cuts (b) the user only needs to provide an extra click (c), per thin structure, in order to extract the structures that were incorrectly excluded from the initial segmentation (d). ***Bounding box tightness.*** A natural assumption in interactive segmentation is that the user provided bounding box is drawn tightly enclosing the object of interest [68], like in (e). Traditional graph cut methods can produce results that do not follow this assumption (f), while the connectivity constraint **C1** can be used to overcome this problem, producing result (g). The results shown in images (d) and (g) were obtained using the algorithm DijkstraGC that will be described in section 3.3.1.

higher-order constraint then **C0**, we will show it can be useful to overcome some of the limitations of pairwise models, specially in an interactive scenario, motivating new forms of user input. Fig. 3.2 shows examples of tasks that benefit from the connectivity constraint **C1**.

This chapter is organised as follows. We start by discussing previous related work in section 3.1.1. We then describe our formulation in section 3.2 and introduce two optimisation algorithms in section 3.3. We discuss applications of the connectivity constraint in an interactive scenario for image segmentation in section 3.4 and report experimental results in section 3.5. Finally, we discuss limitations in section 3.6 and conclude in section 3.7.

3.1.1 Related work

Connectivity in graph cut methods is discussed in [17] for energy functions of a restricted form. For example, if the unary potentials satisfy the following conditions:

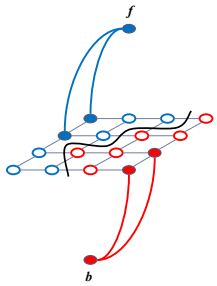


Figure 3.3: Illustration of the connectivity in graph cut based methods with a restricted energy function. The coloured bold edges correspond to the hard constraints imposed by setting the unary potentials to ∞.

$$\phi_p(0) = \infty, \qquad \phi_p(1) = 0 \qquad \text{if } p \in S_f \tag{3.1a}$$

$$\phi_p(0) = 0, \qquad \phi_p(1) = \infty \qquad \text{if } p \in S_b \tag{3.1b}$$

$$\phi_p(0) = 0, \qquad \phi_p(1) = 0 \qquad \text{otherwise} \tag{3.1c}$$

where S_f and S_b are disjoint subsets of connected nodes and the pairwise terms are of the form: $\phi_{pq}(x_p, x_q) = w_{pq}|x_p - x_q|$, then the resulting optimal segmentation is connected. This restriction implies that only the nodes in S_f have *t-links* connecting with f and, correspondingly, only the nodes in S_b have *t-links* connecting with b. Nodes in S_f and S_b can be seen as *hard constraints* and can, for example, correspond to user provided region seeds in an interactive system. This construction is illustrated in Fig. 3.3.

Similar restrictions have been used in other graph based segmentation techniques, like random walker [39] and geodesic distance [86]. Other interesting topological constraints can also be achieved by manipulating the weight of *n-links* [107].

A connectivity constraint for unrestricted energy functions was considered in [116] and [82]. After posing the problem the authors of [116] proved it to be NP-hard and proposed to modify the maxflow algorithm in [16] so that the topology of the segmentation is preserved with respect to a user provided initial segmentation. From our experiments using the author's implementation, we observed that this method has several drawbacks: the results change considerably for different initial segmentations with the same topology and do not always conform with the property stated in Theorem 2 (introduced in the next section).

The work described in the rest of this chapter precedes [82] where an LP relaxation approach to minimise the energy under constraint **C0** is presented. Their method differs from our approach in the use of superpixels instead of individual pixels, due to the large complexity of the

optimisation problem. Interestingly, working on superpixels could potentially be an advantage since degenerate solutions which are one pixel wide (details later), are prohibited.

Connectivity is automatically enforced in the classical "snakes" approach [51], since the segmentation is represented by a simple closed contour. A topology preserving level set method which allows to specify more general topologies was proposed in [44]. A disadvantage of both techniques is that the objective is optimised via gradient descent, which can easily get stuck in a local minimum.

3.2 Problem formulation

Recall the standard form of the energy function used in graph cut based image segmentation approaches

$$E(\boldsymbol{x}) = \sum_{p \in \mathcal{V}} \phi_p(x_p) + \sum_{(p,q) \in \mathcal{N}} \phi_{pq}(x_p, x_q) \tag{3.2}$$

where $(\mathcal{V}, \mathcal{N})$ is an undirected graph whose nodes correspond to pixels. $x_p \in \{0, 1\}$ is the segmentation label of pixel p, where 0 and 1 correspond to the background and the foreground, respectively. The pairwise terms ϕ_{pq} considered are submodular.

As stated in the introduction, the goal is to minimise function $E(\boldsymbol{x})$ under certain connectivity constraints on the segmentation $\boldsymbol{x}$. Three possible constraints are formulated below. In all of them it is assumed that an undirected graph $(\mathcal{V}, \mathcal{F})$ defining the "connectivity" relations between nodes in $\mathcal{V}$ is given. This graph can be different from the graph $(\mathcal{V}, \mathcal{N})$ defining the structure of function $E(\boldsymbol{x})$ in (3.2). In the experiments $(\mathcal{V}, \mathcal{N})$ is an 8-connected 2D grid graph and $(\mathcal{V}, \mathcal{F})$ a 4-connected.

The most natural connectivity constraint is the following:

C0 *The set $[\boldsymbol{x}]$ corresponding to segmentation $\boldsymbol{x}$ must form a single connected component in the graph $(\mathcal{V}, \mathcal{F})$.*

$[\boldsymbol{x}]$ denotes the set of nodes with label 1, i.e. $[\boldsymbol{x}] = \{p \in \mathcal{V} \mid x_p = 1\}$. Although, this is the most intuitive connectivity constraint, minimising function (3.2) under the **C0** can be shown to be NP-hard even if function (3.2) has only unary terms (see below).

The focus of this chapter will be on different constraints **C1** and **C2**. It is assumed that there are two special nodes $s, t \in \mathcal{V}$. Constraint **C1** is then formulated as follows:

C1 *Nodes s, t must be connected in the segmentation set $[\boldsymbol{x}]$, i.e. there must exist a path in the graph $(\mathcal{V}, \mathcal{F})$ from s to t such that all nodes p in the path belong to the segmentation, i.e. $x_p = 1$.*

Unfortunately, minimising (3.2) under **C1** is an NP-hard problem as well (see below). However, it appears that it is easier to design good heuristic algorithms for **C1** than for **C0**. In particular, if function $E(\boldsymbol{x})$ has only unary terms, optimising it under constraint **C1** can be reduced to a shortest path computation between the two terminal nodes and thus can be solved in polynomial time (see section 3.3.1).

Enforcing constraint **C1** may result in a segmentation which has a "width" of one pixel in certain places, which may be undesirable (see Fig. 3.10) . One way to fix this problem is to introduce a parameter δ which controls the minimum "width" of the segmentation. Formally, assume that for each node $p \in \mathcal{V}$ there is a subset $\mathcal{Q}_p \subseteq \mathcal{V}$. (This subset would depend on δ; for example, for a grid graph $\mathcal{Q}_p$ could be the set of all pixels q such that the distance from p to q does not exceed δ.) Using these subsets, the following connectivity constraint is defined:

C2 *There must exist a path in the graph $(\mathcal{V}, \mathcal{F})$ from s to t such that for all nodes p in the path the subset $\mathcal{Q}_p$ belongs to $[\boldsymbol{x}]$, i.e. $x_q = 1$ for $q \in \mathcal{Q}_p$.*

Clearly, **C1** is a special case of **C2** if $\mathcal{Q}_p = \{p\}$ for all nodes p.

Throughout the chapter, **P0**, **P1**, **P2** denote the problems of minimising function (3.2) under constraints **C0**, **C1**, **C2**, respectively. The theorem below shows the difficulty of the problems and its proof is given in Appendix A.1.

Theorem 1. *Problems* **P0**, **P1**, **P2** *are NP-hard.* **P0** *and* **P2** *remain NP-hard even if the set $\mathcal{N}$ is empty, i.e. function* (3.2) *does not have pairwise terms.*

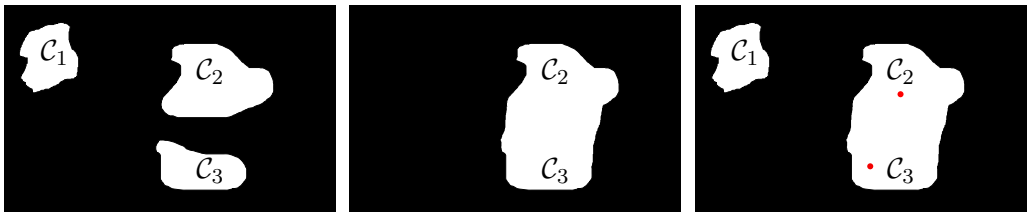
Note, it was also shown in [116] that the following problem is NP-hard: minimise function (3.2) on a planar 2D grid so that the foreground is 4-connected and the background is 8-connected. It is straightforward to modify the argument in [116] to show that the problem is NP-hard if only the 4-connectedness of the foreground is imposed (in other words, **P0** is NP-hard even for planar 2D grids).

To conclude this section, some simple facts about the relationship of problems **P0**-**P2** and the problem of minimising function $E(\boldsymbol{x})$ without any constraints are presented. A proof is given in Appendix A.2 and Fig. 3.4 illustrates these properties.

Theorem 2. *Suppose that $\boldsymbol{x}$ is a global minimum of function* (3.2) *without any constraints.*

(a) There exists an optimal solution $\boldsymbol{x}^*$ of **P2** *which includes $\boldsymbol{x}$, i.e. $[\boldsymbol{x}] \subseteq [\boldsymbol{x}^*]$. The same holds for the problem* **P1** *since the latter is a special case.*

(b) Suppose that $\mathcal{N} \subseteq \mathcal{F}$. Let $\mathcal{C}_1, \ldots, \mathcal{C}_k \subseteq V$ be the connected components of the set $[\boldsymbol{x}]$ in the graph $(\mathcal{V}, \mathcal{F})$. Then there exists an optimal solution $\boldsymbol{x}^*$ of **P0** *such that each*



No constraint Constraint **C0** Constraint **C1**

Figure 3.4: Illustration of Theorem 2. The optimal solutions of both problems **P0** and **P1** relate with the minimiser of the energy without connectivity constraints: the solution of **P1** *includes* it, while the solution of **P0** *fully includes* ($\mathcal{C}_2$ and $\mathcal{C}_3$) or *fully excludes* ($\mathcal{C}_1$) each of its connected components. The red dots for constraint **C1** represent the special terminal nodes.

component $\mathcal{C}_i$ is either entirely included in $[\boldsymbol{x}^*]$ or entirely excluded. In other words, if $\mathcal{C}_i$ and $[\boldsymbol{x}^*]$ intersect then $\mathcal{C}_i \subseteq [\boldsymbol{x}^*]$.

3.3 Algorithms

After formulating the connectivity constraints it remains to discuss how to solve problems **P1** and **P2**. The effectiveness of the optimisation algorithms used is crucial for a successful application of the model. Since both problems are NP-hard we cannot expect to solve them exactly. However, it is still important to evaluate the optimality of the algorithms used. We have developed two different algorithms for optimising energy (3.2) under connectivity constraints **C1** and **C2**.

The first method, which we call *DijkstraGC*, is a practical heuristic technique that can be seen as a fusion between the Dijkstra algorithm and graph cut optimisation. *DijkstraGC* is presented in section 3.3.1.

Then in section 3.3.2 we propose an alternative method for a special case of problem **P1** based on the idea of *Dual Decomposition*. The main feature of the second technique is that it provides a lower bound on the optimal value of **P1**.

We will use the second method for assessing the performance of DijkstraGC: in the experimental section it will help us to verify that for some instances DijkstraGC gives an optimal solution.

3.3.1 DijkstraGC: merging Dijkstra and graph cuts

Our first method is motivated by two observations previously stated. First, problem **P1** without pairwise terms can be solved exactly with a shortest path algorithm. Second, the solution of **P2** (and **P1**) includes the solution of the unconstrained problem. A solution to **P1** can be obtained from the solution to the unconstrained problem by selecting a path connecting the two terminal nodes and forcing the label of all the nodes in that path to be 1, i.e. "adding" this path to the

initialise: $\mathcal{S}=\varnothing$, $PARENT(p)=NULL$ for all nodes p,
$d(s) = \min\{E(\boldsymbol{x}) \mid \mathcal{Q}_s \subseteq [\boldsymbol{x}]\}$,
$d(p) = +\infty$ for $p \in \mathcal{V} - \{s\}$

while $t \notin \mathcal{S}$ and $\mathcal{V} - \mathcal{S}$ contains nodes p with $d(p) < +\infty$

- find node $p \in \mathcal{V} - \mathcal{S}$ with the smallest distance $d(p)$
- add p to $\mathcal{S}$
- **for** all nodes $q \in \mathcal{V} - \mathcal{S}$ which are neighbours of p (i.e. $(p, q) \in \mathcal{F}$) **do**
  - using $PARENT$ pointers, get path $\mathcal{P}$ from s to q through p; compute corresponding set $\bar{\mathcal{P}} = \cup_{r \in \mathcal{P}} \mathcal{Q}_r$
  - compute a minimum $\boldsymbol{x}$ of function (3.2) under the constraint $\bar{\mathcal{P}} \subseteq [\boldsymbol{x}]$
  - if $d(q) > E(\boldsymbol{x})$ set $d(q) := E(\boldsymbol{x})$, $PARENT(q) := p$

Figure 3.5: DijkstraGC algorithm.

unconstrained solution. Different choices of paths will have different energy costs and the main goal is to select a path that it is not too "expensive".

This is achieved by the DijkstraGC algorithm, a combination of the Dijkstra algorithm with graph cuts. Recall that the Dijkstra algorithm computes shortest distances $d(p)$ in a directed graph with non-negative weights from a specified "source" node s to all other nodes p.

Similar to the Dijkstra method, DijkstraGC computes solutions to the problem **P2** for a fixed node s and all nodes $p \in \mathcal{V}$ (only now these solutions will not necessarily be global minima). The "distance" $d(p)$ will now indicate the cost of the computed solution for the pair of nodes $\{s, p\}$.

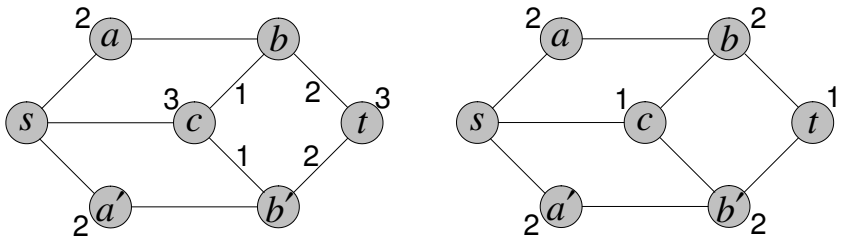
The algorithm is shown in Fig. 3.5 and an illustration is provided in the Appendix B. During the algorithm, the current solution $\boldsymbol{x}^p$ for node p with $d(p) < +\infty$ can be obtained as follows: using $PARENT$ pointers get path $\mathcal{P}$ and corresponding set $\bar{\mathcal{P}} = \cup_{r \in \mathcal{P}} \mathcal{Q}_r$, and then compute a minimum of function (3.2) under the constraint $\bar{\mathcal{P}} \subseteq [\boldsymbol{x}]$, by enforcing $x_r = 1$ for all $r \in \bar{\mathcal{P}}$. Clearly, the obtained solution $\boldsymbol{x}^p$ satisfies the hard constraint **C2** for the pair of nodes $\{s, p\}$.

The set $\mathcal{S}$ contains "permanently labelled" nodes: once a node p has been added to $\mathcal{S}$, its cost $d(p)$ and the corresponding solution will not change anymore.

Let us list some of the invariants that are maintained during DijkstraGC (they follow directly from the description):

I1 If $d(p) = +\infty$ then $p \neq s$ and $PARENT(p) = NULL$.

I2 If $d(p) < +\infty$ then $PARENT$ pointers give the unique path $\mathcal{P}$ from s to p, and $d(p) = \min\{E(\boldsymbol{x}) \mid \bar{\mathcal{P}} \subseteq [\boldsymbol{x}]\}$ where $\bar{\mathcal{P}} = \cup_{r \in \mathcal{P}} \mathcal{Q}_r$.



$\mathcal{Q}_c = \{c, b, b'\}$, $\mathcal{Q}_t = \{t, b, b'\}$
$\mathcal{Q}_p = \{p\}$ for all other nodes p

(a) Problem **P1** (b) Problem **P2**, no pairwise terms

Figure 3.6: Suboptimality of DijkstraGC. Examples of problems on which DijkstraGC gives suboptimal results. Graphs shown in the images are the connectivity graphs $(\mathcal{V}, \mathcal{F})$. Number w_p at node p gives the unary term $w_p \cdot x_p$, number w_{pq} at edge (p, q) gives the pairwise term $w_{pq}|x_q - x_p|$. Both in (a) and (b) DijkstraGC will output solution $\{s,a,b,b',t\}$ or $\{s,a',b,b',t\}$ with cost 7, while the optimal solution $\{s,c,b,b',t\}$ has cost 6.

I3 If $PARENT(q) = p$ then $d(p) \leq d(q) < +\infty$.

I4 $d(p) < +\infty$ for nodes $p \in \mathcal{S}$.

Theorem 3. *If function $E(\boldsymbol{x})$ does not have pairwise terms and $\mathcal{Q}_p = \{p\}$ for all nodes p (i.e it is an instance of* **P1***) then the algorithm in Fig. 3.5 produces an optimal solution.*

A proof is given in Appendix A.3.

If conditions of the theorem are relaxed then the problem may become NP-hard, as theorem 1 states. Not surprisingly, DijkstraGC may then produce a suboptimal solution. Two examples are shown in Fig. 3.6. Note that in these examples the "direction" of DijkstraGC matters: running DijkstraGC from s to t gives a suboptimal solution, but running it from t to s will give an optimal segmentation.

We now turn to the question of efficient implementation. One computational component of the algorithm is to find a node $p \in \mathcal{V} - \mathcal{S}$ with the smallest value of $d(p)$ (same as in the Dijkstra algorithm). We use a binary heap structure for implementing the priority queue which stores nodes $p \in \mathcal{V} - \mathcal{S}$ with $d(p) < +\infty$. The bottleneck, however, is maxflow computations: DijkstraGC requires many calls to the maxflow algorithm for minimising function (3.2) under the constraints $x_r = 1$ for nodes $r \in \bar{\mathcal{P}}$. These computations are considered in the remainder of this section.

Optimised DijkstraGC

We now describe a technique which allows to reduce the number of calls to maxflow. Consider the step that adds node p to the set of permanently labelled nodes $\mathcal{S}$. Denote $\mathcal{P}$ to be the path from s to p given by $PARENT$ pointers, and let $\bar{\mathcal{P}} = \cup_{r \in \mathcal{P}} \mathcal{Q}_r$. Let us fix nodes in $\bar{\mathcal{P}}$ to 1

initialise: $\mathcal{S}=\varnothing$, $PARENT(p)=NULL$ for all nodes p,
$d(s) = \min\{E(\boldsymbol{x}) \mid \mathcal{Q}_s \subseteq [\boldsymbol{x}]\}$,
$d(p) = +\infty$ for $p \in \mathcal{V} - \{s\}$

while $t \notin \mathcal{S}$ and $\mathcal{V} - \mathcal{S}$ contains nodes p with $d(p) < +\infty$

- find node $p \in \mathcal{V} - \mathcal{S}$ with the smallest distance $d(p)$
- using $PARENT$ pointers, get path $\mathcal{P}$ from s to p; compute corresponding set $\bar{\mathcal{P}} = \cup_{r\in\mathcal{P}} \mathcal{Q}_r$
- compute a minimum $\boldsymbol{x}$ of function (3.2) under the constraint $\bar{\mathcal{P}} \subseteq [\boldsymbol{x}]$
- add p to $\mathcal{S}$, set $\mathcal{A} = \{p\}$, mark p as "unprocessed"
- **while** $\mathcal{A}$ has unprocessed nodes
  - pick unprocessed node $p' \in \mathcal{A}$
  - **for** all edges $(p', q) \in \mathcal{F}$ with $q \in \mathcal{V} - \mathcal{S}$ **do**
    - ◇ if $\mathcal{Q}_q \subseteq [\boldsymbol{x}]$ set $d(q) := E(\boldsymbol{x})$, $PARENT(q) := p'$, add q to $\mathcal{S}$ and to $\mathcal{A}$ as an unprocessed node
  - mark p' as "processed"
- **for** all nodes $q \in \mathcal{V} - \mathcal{S}$ which are neighbours of $\mathcal{A}$ (i.e. $(p', q) \in \mathcal{F}$ for some node $p' \in \mathcal{A}$) **do**
  - pick node $p' \in \mathcal{A}$ with $(p', q) \in \mathcal{F}$
  - using $PARENT$ pointers, get path $\mathcal{P}$ from s to q through p'; compute corresponding set $\bar{\mathcal{P}} = \cup_{r\in\mathcal{P}} \mathcal{Q}_r$
  - compute a minimum $\boldsymbol{x}$ of function (3.2) under the constraint $\bar{\mathcal{P}} \subseteq [\boldsymbol{x}]$
  - if $d(q) > E(\boldsymbol{x})$ set $d(q) := E(\boldsymbol{x})$, $PARENT(q) := p'$

Figure 3.7: Optimised version of the DijkstraGC algorithm.

and compute a minimum $\boldsymbol{x}$ of function (3.2) under these constraints. The segmentation set $[\boldsymbol{x}]$ will contain $\bar{\mathcal{P}}$, but it may include many other nodes as well. Then it might be possible to add several nodes to $\mathcal{S}$ using this single computation. Indeed, suppose p has a neighbour $q \in \mathcal{V} - \mathcal{S}$, $(p, q) \in \mathcal{F}$, such that $\mathcal{Q}_q \subseteq [\boldsymbol{x}]$. The algorithm in Fig. 3.5 would set $d(q) = d(p) = E(\boldsymbol{x})$ while exploring neighbours of p. This would make the distance $d(q)$ to be the smallest among nodes in $\mathcal{V} - \mathcal{S}$, so the node q could be the next node to be added to $\mathcal{S}$. Therefore, we can add q to $\mathcal{S}$ immediately.

An algorithm which implements this idea is shown in Fig. 3.7. Before exploring neighbours of q, we check which nodes can be added to $\mathcal{S}$ for "free". The set of these nodes is denoted as $\mathcal{A}$; clearly, it includes p. After adding nodes in $\mathcal{A}$ to $\mathcal{S}$, we explore neighbours of $\mathcal{A}$ which are still in $\mathcal{V} - \mathcal{S}$.

Note that there is a certain freedom in implementing the DijkstraGC algorithm: it does not specify which node $p \in \mathcal{V} - \mathcal{S}$ with the minimum distance to choose if there are several such nodes. It is not difficult to see that under a certain selection rule DijkstraGC becomes equivalent

to the algorithm in Fig. 3.7.

3.3.2 Dual Decomposition

In this section we propose a different technique for a special case of problem **P1** based on *Dual Decomposition*. This technique will be used for assessing the performance of DijkstraGC.

Recall that Dual Decomposition relies on the splitting of the original problem into easier subproblems. To get tractable subproblems, we impose the following simplifying assumptions. First, we assume that the graph $(\mathcal{V}, \mathcal{F})$ is planar, and $\mathcal{N} = \mathcal{F}$. Second, we assume that pixels on the image boundary are constrained to be background, i.e. their label is 0. These assumptions are illustrated in Fig. 3.8. We argue that they represent an important practical subclass of the image segmentation task, and thus can be used for assessing the performance of DijkstraGC for real problems. Note that the second assumption encodes the prior knowledge that the object lies entirely inside the image, which is very often the case in practice.

We denote $C(\boldsymbol{x})$ to be the hard constraint term which is 0 if the segmentation $\boldsymbol{x}$ satisfies the connectivity constraint **C1** and the background boundary condition described above, and otherwise $C(\boldsymbol{x})$ is $+\infty$. Some of these hard constraints will also be included in function $E(\boldsymbol{x})$ as unary terms, namely the background boundary constraints and foreground constraints $x_s = x_t = 1$, which follow from **C1**.

Our dual vector $\boldsymbol{\lambda}$ will have two parts: $\boldsymbol{\lambda} = (\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2)$ where vectors $\boldsymbol{\lambda}^1$ and $\boldsymbol{\lambda}^2$ correspond to relaxing consistency constraints obtained from duplicating the nodes and edges of the graph $(\mathcal{V}, \mathcal{N})$, respectively ($\boldsymbol{\lambda}^1 \in \mathbb{R}^{\mathcal{V}}$, $\boldsymbol{\lambda}^2 \in \mathbb{R}^{\mathcal{N}}$). Given labelling $\boldsymbol{x}$, let $\varphi(\boldsymbol{x}) \in \{0,1\}^{\mathcal{N}}$ be the vector of indicator variables showing discontinuities of $\boldsymbol{x}$, i.e. $\varphi_{pq}(\boldsymbol{x}) = |x_q - x_p|$ for an edge $(p,q) \in \mathcal{N}$.

We will use the following lower bound, based on Dual Decomposition:

$$
\begin{aligned}
\Phi(\boldsymbol{\lambda}) \leq & E(\boldsymbol{x}) + C(\boldsymbol{x}) \\
\Phi(\boldsymbol{\lambda}) = & \min_{\boldsymbol{x}_0} \left[E(\boldsymbol{x}_0) - \langle \boldsymbol{\lambda}^1, \boldsymbol{x}_0 \rangle - \langle \boldsymbol{\lambda}^2, \varphi(\boldsymbol{x}_0) \rangle \right] && \text{(Subproblem 0)} \\
& + \min_{\boldsymbol{x}_1} \left[C(\boldsymbol{x}_1) + \langle \boldsymbol{\lambda}^1, \boldsymbol{x}_1 \rangle \right] && \text{(Subproblem 1)} \\
& + \min_{\boldsymbol{x}_2} \left[C(\boldsymbol{x}_2) + \langle \boldsymbol{\lambda}^2, \varphi(\boldsymbol{x}_2) \rangle \right] && \text{(Subproblem 2)}
\end{aligned}
$$

We now discuss how to minimise each subproblem in more detail.

Subproblem 0

This subproblem consists in minimising a function with unary and pairwise terms. We will require this function to be submodular; this is equivalent to specify upper bounds on components

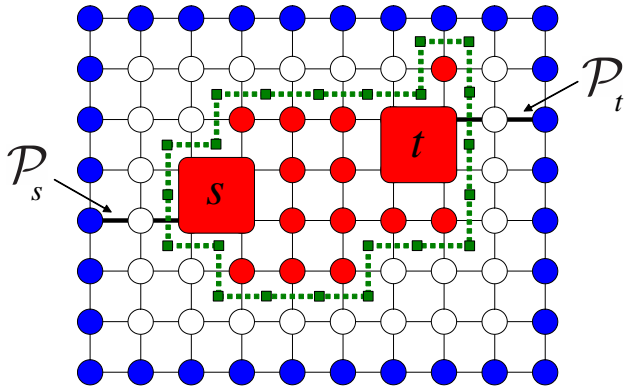


Figure 3.8: **Solving P1 via problem decomposition.** Blue pixels at the image border have hard background constraints, nodes s, t have hard foreground constraints. Note that s, t cover several pixels since before starting the algorithm we compute a minimum of function (3.2) without the connectivity constraint and contract pixels connected to s and to t to single nodes. (This is justified by theorem 2.) A possible segmentation satisfying all hard constraints is shown in red. Its boundary in the dual graph $(\mathcal{V}^*, \mathcal{N}^*)$ is a simple closed contour (shown in green) passing through faces in $\mathcal{V}^*$.

$\boldsymbol{\lambda}^2$. Since there are no connectivity constraints, we can compute the global minimum using a maxflow algorithm.

Subproblem 1

A global minimum can be computed using DijkstraGC algorithm, since the function has only unary terms and the connectivity constraint **C1**. Note, in this case DijkstraGC reduces to the Dijkstra algorithm.

Subproblem 2

We require vector $\boldsymbol{\lambda}^2$ to be non-negative. Instead of attempting to get the global minimum, we compute a lower bound on function $E^2(\boldsymbol{x}|\boldsymbol{\lambda}^2) = C(\boldsymbol{x}) + \langle \boldsymbol{\lambda}^2, \varphi(\boldsymbol{x}) \rangle$, using a very fast technique that we now describe in detail.

The graph $\mathcal{G} = (\mathcal{V}, \mathcal{N})$ is planar; thus, we can construct the dual graph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{N}^*)$ whose nodes are the faces of $(\mathcal{V}, \mathcal{N})$. Graph $\mathcal{G}^*$ will be weighted: for each edge $(p, q) \in \mathcal{N}$ in the original graph there will be an edge $(i, j) \in \mathcal{N}^*$ with weight $c_{ij} = \lambda^2_{pq}$ where $i, j \in \mathcal{V}^*$ are the two faces that border the edge (p, q).

We can assume without loss of generality that an optimal segmentation is connected in $\mathcal{G}$. (If not, we could remove all connected components except for the one containing s and t; the hard constraints would still be satisfied, and the cost would not increase.) Any connected segmentation $\boldsymbol{x}$ satisfying the hard constraints, i.e. $C(\boldsymbol{x}) = 0$, defines an edge-disjoint closed contour in $\mathcal{G}^*$ whose interior contains s and t (Fig. 3.8). Furthermore, the cost of edges in the contour equals $E^2(\boldsymbol{x})$. Note that the contour cannot cross the image border, therefore we can

remove the outer face and incident edges.

Let $\mathcal{P}_s$ and $\mathcal{P}_t$ be paths from s and t, respectively, to the image border (Fig. 3.8). $\mathcal{P}_s$ and $\mathcal{P}_t$ will be viewed as subsets of edges of $\mathcal{N}$. Clearly, the contour corresponding to $\boldsymbol{x}$ intersects both $\mathcal{P}_s$ and $\mathcal{P}_t$ at least once. Thus, the contour passes through one of the nodes in $\mathcal{P}_s^*$ and through one of the nodes in $\mathcal{P}_t^*$, where $\mathcal{P}_s^*$ and $\mathcal{P}_t^*$ are respectively the subsets of faces in $\mathcal{V}^*$ that border edges in P_s and P_t on a particular side, say left. Thus, we can obtain a lower bound on $E^2(\boldsymbol{x}|\boldsymbol{\lambda})$ by computing the minimum cost of two edge-disjoint paths from $\mathcal{P}_s^*$ to $\mathcal{P}_t^*$. Note that, this edge disjoint paths do not necessarily correspond to a simple contour since they can intersect or have different starting points.

To solve the latter problem, we use a standard reduction to the minimum cost network flow problem [1]. We construct a graph with nodes $\mathcal{V}^* \cup \{s, t\}$ where s^*, t^* are two new nodes. We add directed arcs from s^* to the nodes in $\mathcal{P}_s^*$ with capacity 2 and cost 0, and arcs from the nodes in $\mathcal{P}_t^*$ to t^* with the same capacity and cost. For each edge $(i, j) \in \mathcal{N}^*$ we add two directed arcs $(i \to j)$, $(j \to i)$ with capacity 1 and cost c_{ij}. Finally, we set the flow excess of s^* and t^* to be +2 and -2, respectively. Clearly, any integer flow that sends two units from s to t defines two paths from $\mathcal{P}_s^*$ to $\mathcal{P}_t^*$ (an edge belongs to one of the paths iff it carries some flow).

To compute a minimum cost flow, we used the successive shortest path algorithm [1]. It works by iteratively running the Dijkstra algorithm in a certain graph. Each iteration sends one unit of flow, therefore there will be two Dijkstra computations.

Maximising the lower bound

The lower bound $\Phi(\boldsymbol{\lambda})$ is maximised using subgradient method described in section 2.5.1.

Note that the subgradient method solves a problem dual to the original problem and it does not provide directly a solution to the original primal problem. This solution is obtained as follows: from the solutions of **subproblem 1** we select the one with the smallest original energy. This solution is guaranteed to satisfy the connectivity constraint **C1**.

3.3.3 Comparison with the LP relaxation of Nowozin and Lampert [82]

The Linear Programming relaxation presented in [82] for energy minimisation under constraint **C0** is related with the Dual Decomposition approach described in the previous section. It also solves a relaxation of the problem, providing a lower bound.

Although the LP relaxation in [82] was introduced for constraint **C0**, it can be easily adapted to the constraint **C1**. We will now review this approach and show connections with our Dual Decomposition.

We start by rewriting the unconstrained energy minimisation problem as an Integer Program [95]. Let $\boldsymbol{\mu}_p = \{\mu_p(l)|l = 0, 1\}$ be the vector of indicator variables for node p such

that $\mu_p(l) = 1 \Leftrightarrow x_p = l$. Similarly, let $\boldsymbol{\mu}_{pq} = \{\mu_{pq}(l,l')|l,l' = 0,1\}$ be the vector of indicator variables for edge (p,q) such that $\mu_{pq}(l,l') = 1 \Leftrightarrow x_p = l, x_q = l'$. Finally, let $\boldsymbol{\mu} = \{\{\boldsymbol{\mu}_p\},\{\boldsymbol{\mu}_{pq}\}\}$ be the vector of all indicator variables.

Vector $\boldsymbol{\phi} = \{\{\boldsymbol{\phi}_p\}\{\boldsymbol{\phi}_{pq}\}\}$ contains all MRF-parameters of the form $\boldsymbol{\phi}_p = \{\phi_p(l)|l = 0,1\}$ and $\boldsymbol{\phi}_{pq} = \{\phi_{pq}(l,l')|l,l' = 0,1\}$.

The unconstrained energy minimisation problem is equivalent to the following Integer Program:

$$\min_{\boldsymbol{\mu}} \quad \boldsymbol{\phi}\cdot\boldsymbol{\mu} = \sum_{p\in\mathcal{V}} \boldsymbol{\phi}_p\cdot\boldsymbol{\mu}_p + \sum_{(p,q)\in\mathcal{N}} \boldsymbol{\phi}_{pq}\cdot\boldsymbol{\mu}_{pq} \tag{3.3a}$$

$$s.t. \sum_{l=0,1} \mu_p(l) = 1 \qquad \forall p\in\mathcal{V} \tag{3.3b}$$

$$\sum_{l=0,1} \mu_{pq}(l,l') = \mu_q(l') \qquad \forall (p,q)\in\mathcal{N} \tag{3.3c}$$

$$\mu_p(l), \mu_{pq}(l,l') \in \{0,1\} \tag{3.3d}$$

Following [82], to ensure that the connectivity constraint **C1** is satisfied, the following constraints are added to the Integer Program:

$$\sum_{p\in S} \mu_p(1) \geq 1 \quad \forall S\in\mathcal{S} \tag{3.4}$$

where $\mathcal{S}$ is the set of all separating sets S. A separating set S is a set of nodes whose removal disconnects the terminal nodes s and t. Constraints (3.4) ensure that all separating sets S have at least one node that is labelled 1.

The authors of [82] solve an LP relaxation of the Integer Program (3.3) with extra connectivity constraints (3.4), obtaining a real-valued solution and a lower bound to the original problem by replacing the constraints $\mu\in\{0,1\}$ with $\mu\in[0,1]$.

Alternatively, instead of relaxing the integer constraints (3.3d), we can obtain a lower bound on the full Integer Program by introducing duplicated variables (ν_p) and relaxing the consistency constraints, i.e. by using Dual Decomposition. We start by writing an equivalent

problem to the full Integer Program:

$$\min_{\boldsymbol{\mu},\boldsymbol{\nu}} \quad \sum_{p\in\mathcal{V}} \boldsymbol{\phi}_p \cdot \boldsymbol{\mu}_p + \sum_{(p,q)\in\mathcal{N}} \boldsymbol{\phi}_{pq} \cdot \boldsymbol{\mu}_{pq} \tag{3.5a}$$

$$s.t. \quad \sum_{l=0,1} \mu_p(l) = 1, \sum_{l=0,1} \nu_p(l) = 1 \qquad \forall p \in \mathcal{V} \tag{3.5b}$$

$$\sum_{l=0,1} \mu_{pq}(l,l') = \mu_q(l') \qquad \forall (p,q) \in \mathcal{N} \tag{3.5c}$$

$$\sum_{p\in S} \nu_p(1) \geq 1 \qquad \forall S \in \mathcal{S} \tag{3.5d}$$

$$\mu_p(l) = \nu_p(l) \qquad \forall p \in \mathcal{V}, l \in 0,1 \tag{3.5e}$$

$$\mu_p(l), \nu_p(l), \mu_{pq}(l,l') \in \{0,1\} \tag{3.5f}$$

By relaxing the consistency constraints (3.5e) and introducing Lagrangian multipliers $\boldsymbol{\lambda}$, we obtain a Dual Decomposition relaxation with two subproblems:

Subproblem 0

$$\min_{\boldsymbol{\mu}} \quad \sum_{p\in\mathcal{V}} (\boldsymbol{\phi}_p + \boldsymbol{\lambda}_p) \cdot \boldsymbol{\mu}_p + \sum_{(p,q)\in\mathcal{N}} \boldsymbol{\phi}_{pq} \cdot \boldsymbol{\mu}_{pq}$$

$$s.t. \quad \sum_{l=0,1} \mu_p(l) = 1 \;\; \forall p \in \mathcal{V}$$

$$\sum_{l=0,1} \mu_{pq}(l,l') = \mu_q(l') \;\; \forall (p,q) \in \mathcal{N}$$

$$\mu_p(l), \mu_{pq}(l,l') \in \{0,1\}$$

Subproblem 1

$$\min_{\boldsymbol{\nu}} \quad \sum_{p\in\mathcal{V}} -\boldsymbol{\lambda}_p \cdot \boldsymbol{\nu}_p$$

$$s.t. \quad \sum_{l=0,1} \nu_p(l) = 1 \qquad \forall p \in \mathcal{V}$$

$$\sum_{p\in S} \nu_p(1) \geq 1 \qquad \forall S \in \mathcal{S}$$

$$\nu_p(l) \in \{0,1\}$$

These two subproblems are the same first two subproblems considered in our Dual Decomposition approach, discussed in section 3.3.2. Since only two subproblems are used, the lower bound previously presented with three subproblems is equal or tighter. Furthermore, the lower bound obtained from these two subproblems is equal or tighter than the LP relaxation of [82]. This relation comes from the following observation: the LP relaxation [82] and the Dual Decomposition are both relaxations of the same Integer Programming formulation, but they are obtained from relaxing different constraints. In [41] it was proved that the lower bound obtained by relaxing the consistency constraints after introducing duplicated variables is equal or tighter than the lower bound from LP relaxation.

In conclusion, we have shown that our Dual Decomposition method provides a lower

bound for problem **P1** which is equal or tighter than the lower bound introduced in [82]. It is, however, important to notice that our Dual Decomposition method is not applicable to problem **P0** for which the LP relaxation was proposed in [82]. The reason is that, in a similar decomposition for problem **P0** the **Subproblem 1** would correspond to the optimisation of an energy function with unary terms under connectivity constraint **C0**, which is an NP-hard problem (see Appendix A.1).

3.4 Applications in interactive image segmentation

In the previous sections we discussed the formulation of different connectivity constraints and new optimisation algorithms for energy minimisation under those constraints. In this section we will discuss the usefulness of the connectivity constraints **C1** and **C2** to solve tasks arising in interactive image segmentation.

The applications differ in the way the terminal nodes are chosen. They can be directly selected by the user, automatically placed, or indirectly obtained from other forms of user interaction. Moreover, some of the applications require solving multiple instances of problem **P1**.

Recall that the goal of interactive image segmentation is to extract a high-quality segmentation with minimal user input. We will show that the connectivity constraints enable novel forms of user interaction that reduce the amount of user input required.

3.4.1 Overcoming shrinking bias/ Extraction of elongated structures

The pairwise MRF model for image segmentation can be interpreted as a length minimisation method in a Riemannian space. This length minimisation property leads to a known "shrinking bias" of graph cut based methods, i.e. a preference towards short boundaries "cutting" some of the elongated structures of the object (Fig. 3.9 a)).

In an interactive scenario, the "shrinking bias" can possibly be overcome by correcting a segmentation where elongated structures were cut off. However, it can be cumbersome for the user to manually brush a very thin structure that was wrongly cut out. Instead, the connectivity constraint **C1** suggests a novel form of user interaction: clicking the two endpoints of the elongated structure, selecting them as the terminal nodes of constraint **C1** and recomputing the segmentation under this constraint (Fig. 3.9 b)).

To further reduce the user input, we can assume that one of the terminal nodes is always in the biggest connected component of the current segmentation. This is a natural assumption in interactive scenarios, where we can also make use of the user provided region seeds to select automatically one of the terminal nodes.

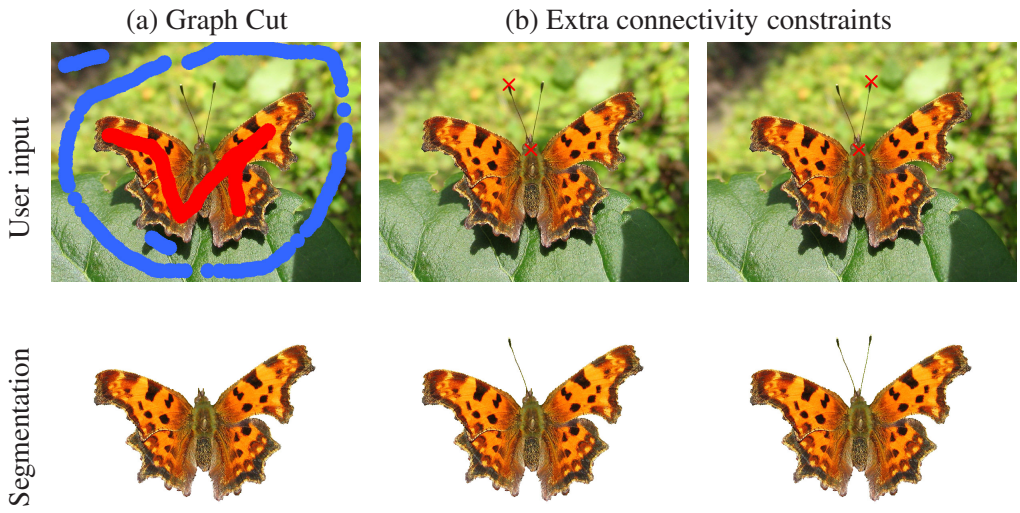


Figure 3.9: Connectivity constraint for extraction of thin elongated structures. Imposing connectivity constraint **C1** based on user input is useful to extract thin elongated structures, like the butterfly antennas, that were cut of by graph cut segmentation without constraints. Only two clicks (red crosses) were necessary to extract each of the butterfly antennas.

The width parameter δ

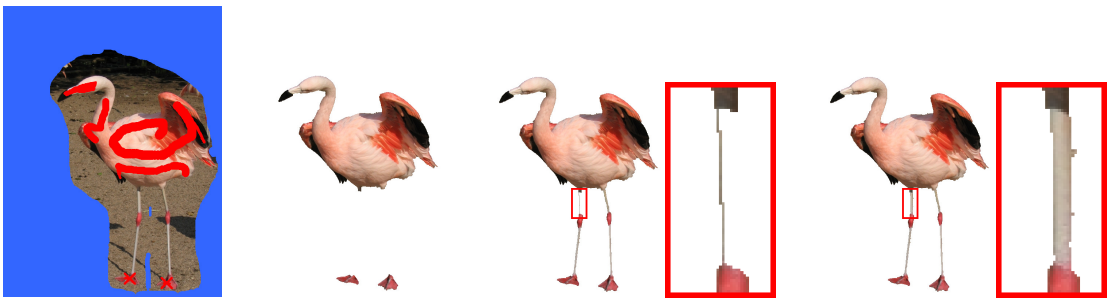
The connectivity constraint **C1** can lead to degenerate solutions that satisfy the constraint by imposing a 1-pixel wide path (Fig. 3.10 (c)). Those solutions can be avoided by using constraint **C2** to impose a minimum width, δ, for the connection between the terminal nodes. This *minimum width* is not included directly in the formulation of constraint **C2**, but it can be achieved by defining for all nodes p a set $\mathcal{Q}_p$ depending on δ. For $\delta = 1$, $\mathcal{Q}_p = \{p\}$; for $\delta = 2$, $\mathcal{Q}_p$ is the set of 4 nodes in a 2×2 square that includes node p and for $\delta = 3$, $\mathcal{Q}_p$ contains p and its neighbours in a 4-connected grid.

Note that in general δ does not have to be the exact width of the structure we want to segment. In fig. 3.10 setting the width parameter to $\delta = 2$ was sufficient to recover the thin leg which is more than 5 pixels wide. In an interactive system, the user could possibly select this parameter depending on the image to be segmented.

3.4.2 Fully connected segmentation using constraint C1

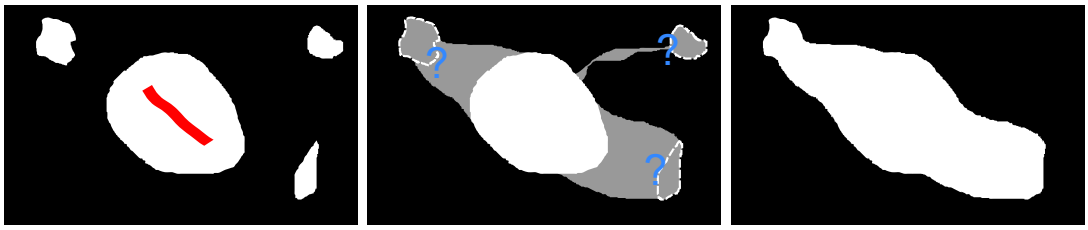
As discussed before, the result of minimising the energy under connectivity constraints **C1** and **C2** is not necessarily a fully connected segmentation, since they are both defined with respect to two special nodes and connectivity is only imposed between those two nodes.

These constraints and corresponding optimisation algorithms can, however, be useful to produce a fully connected segmentation. A technique of this form was proposed in [88]. This heuristic approach is motivated by Theorem 2 and it constructs a solution $\boldsymbol{x}$ to problem **P0** by solving a sequence of problems imposing connectivity constraint **C1**. We now describe how the



(a) User input (b) Graph Cut (c) DijkstraGC $\delta = 1$ (d) DijkstraGC $\delta = 2$

Figure 3.10: Width parameter δ. The result (c) obtained using DijkstraGC satisfies the connectivity constraint **C1** but it is not visually correct. Instead, using constraint **C2** to specify a minimum width results in a better segmentation (d) of the thin structure that was initially cut off from graph cut segmentation (b).



(a) Graph cut (b) Connected components (c) Final segmentation

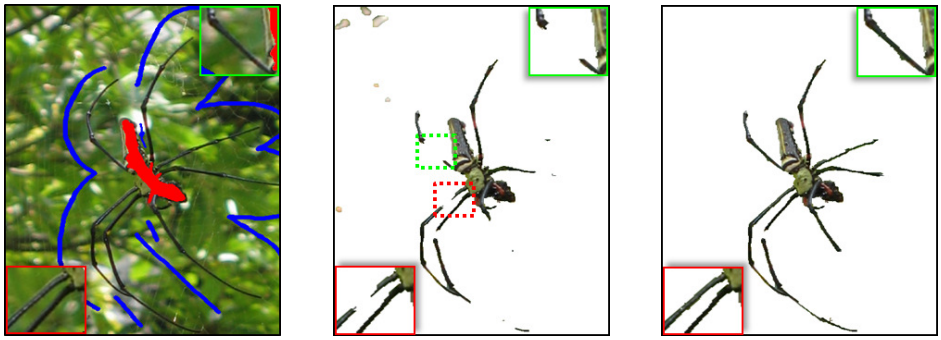
Figure 3.11: Obtaining a fully connected segmentation using constraint **C1**. Given a solution for the unconstrained problem where one of the connected components contains region seeds (a), a solution to problem **P0** shown in (c) is constructed by sequentially choosing which connected components should be retained or excluded (b).

solution $\boldsymbol{x}$ is constructed (an illustration is in Fig. 3.11).

Let $\mathcal{C}_1, \ldots, \mathcal{C}_k \subseteq V$ be the connected components of the segmentation $\boldsymbol{y}$, where $\boldsymbol{y}$ is the global optimum of the energy without connectivity constraints. Also, assume that one of these connected components ($\mathcal{C}_1$) is known to belong to $\boldsymbol{x}$, e.g. it contains region seeds provided by the user (Fig. 3.11 (a)). We initialise $\boldsymbol{x}$ with $\mathcal{C}_1$, i.e. $x_p = 1$ if and only if $p \in \mathcal{C}_1$.

For each of the remaining connected components, $\mathcal{C}_i, i \neq 1$, the algorithm individually decides if the component is part of the final solution $\boldsymbol{x}$ or not. This decision is made in a greedy fashion by comparing two possible solutions, $\boldsymbol{x}^i$ and $\bar{\boldsymbol{x}}^i$, for each component $\mathcal{C}_i$. Solution $\boldsymbol{x}^i$ is obtained by minimising the energy under connectivity constraint **C1** where the terminal nodes are $s \in \mathcal{C}_1$ and $t \in \mathcal{C}_i$. Solution $\bar{\boldsymbol{x}}^i$ is constructed from $\boldsymbol{y}$ by removing the connected component $\mathcal{C}_i$, i.e. by setting $\bar{x}^i_p = 0$ if $p \in \mathcal{C}_i$ and $\bar{x}^i_p = y_p$ otherwise. Finally, solution $\boldsymbol{x}$ is updated as follows: if $E(\bar{\boldsymbol{x}}^i) \leq E(\boldsymbol{x}^i)$ then $\mathcal{C}_i$ is not included in $\boldsymbol{x}$. Otherwise, $\mathcal{C}_i$ and the corresponding connection path obtained from solving **P1** are included in $\boldsymbol{x}$. The algorithm guarantees that by construction the final solution is fully connected.

The authors of [88] also propose a different heuristic for solving problems **P1** and **P2**.



(a) User input (b) Graph cut (c) Fully connected segmentation

Figure 3.12: Example of a fully connected segmentation obtained using **C1**. The connected segmentation (c) is obtained from the graph cut solution (b) by independently selecting which connected components are kept or excluded. It correctly segments most of the spider legs and it removes background regions that were incorrectly included in (b). Image reproduced from [88].

Their algorithm is also based on *shortest paths* and it is shown to perform similarly to *DijkstraGC* with the advantage of being faster. Fig. 3.12 shows an example of using this technique for enforcing constraint **C0**.

A similar method has been recently presented in [26]. The method is extended to more general topological configurations, allowing for any number of connected components and holes. Similarly to the procedure described, the algorithm chooses to remove or merge a connected component based on the cost of each operation.

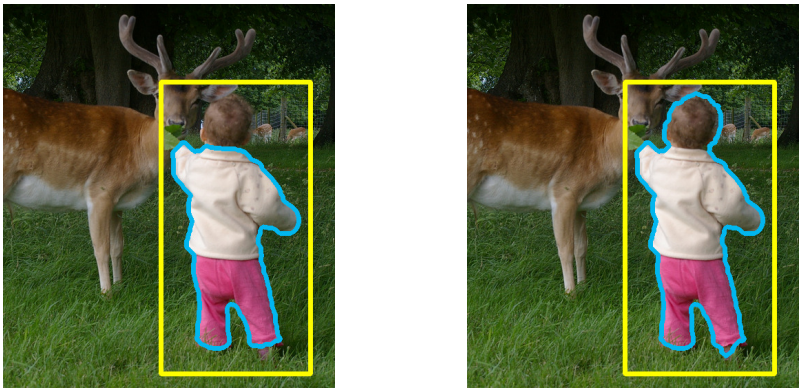
3.4.3 Bounding box tightness constraint

In interactive image segmentation, one of the most popular user provided inputs is in the form of a bounding box surrounding the object of interest. This bounding box is usually used to reduce the size of the region of interest, since all its exterior is background.

In [68] it was suggested to make an additional use of the bounding box. The authors start by observing that although users tend to place the bounding box close to the object of interest, the solution provided by graph cut methods does not always agree with this intuition, like the example in Fig. 3.13 (a).

They overcome this limitation by enforcing the segmentation to be tight to the bounding box. The authors formulated the problem as an integer programming problem and solve a linear programming relaxation. They also propose a heuristic algorithm called *pinpointing* to obtain a solution to the original problem from the relaxed solution.

This tightness prior relates to the connectivity constraint described in this chapter. In particular, it can be enforced by incorporating two connectivity constraints. Given a bounding



(a) (b)

Figure 3.13: Imposing bounding box tightness in segmentation. The segmentation provided by graph cut methods (a) is inconsistent with user input, since it is too loose for this bounding box. By imposing a tightness constraint the method of [68] obtains a better segmentation (b). Image reproduced from [68].

box surrounding the object, the tightness constraint is defined with respect to a *inner box* $\mathcal{B}$, i.e. a rectangle inside the bounding box[1]. A segmentation satisfies the *strong tightness* constraint if one of its connected components "touches" all of the four sides of $\mathcal{B}$. Let $\mathcal{B}_t, \mathcal{B}_b, \mathcal{B}_l, \mathcal{B}_r \subset \mathcal{V}$ be the top, bottom, left and right sides of $\mathcal{B}$ respectively. The tightness constraint can be equivalently defined as follows: there are two paths, P_V (vertical) and P_H (horizontal), such that P_V connects $\mathcal{B}_t$ and $\mathcal{B}_b$ and P_H connects $\mathcal{B}_l$ and $\mathcal{B}_r$ and $x_p = 1$ for all nodes p in both paths. This constraint is illustrated in Fig. 3.14 (a).

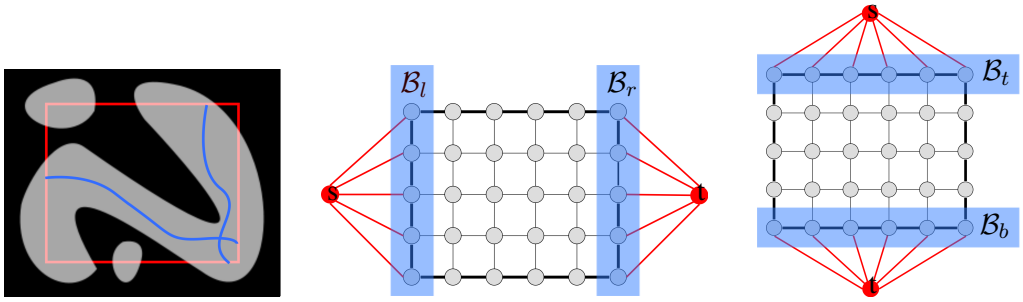
In order to use the connectivity constraint **C1** to impose the bounding box tightness, we need to construct two extended graphs and solve two instances of problem **P1**. The extended graphs are illustrated in Fig. 3.14 (b) and (c). For the first problem, the terminal nodes s and t are auxiliary nodes that connect to all nodes in $\mathcal{B}_l$ and $\mathcal{B}_r$ respectively. Similarly, for the second problem the terminal nodes connect to all nodes in $\mathcal{B}_t$ and $\mathcal{B}_b$. Combining the solution for both problems, by taking the union of both solutions, gives a solution that satisfies the tightness constraint.

3.5 Experimental results

In the previous section we discussed the usefulness of using constraints **C1** and **C2** in different interactive segmentation tasks: extraction of thin elongated structures, imposing full connectivity automatically and enforcing bounding box tightness.

In this section, we evaluate and discuss the properties of the two algorithms proposed for energy minimisation under those constraint. We will focus on the *DijkstraGC* algorithm, since

[1] In [68] two different tightness constraints were introduced and we will focus on *strong tightness*.



(a) (b) (c)

Figure 3.14: Bounding box tightness. (a) illustrates a segmentation that satisfies the tightness constraint. The connectivity constraint **C1** can be used to enforce the bounding box tightness, by imposing it in two extended graphs. The rectangle in red corresponds to the *inner box* used to define the constraint. The constraint illustrated in (b) ensures that there is an *horizontal* path connecting the left and right side of the inner box, while (c) ensures a *vertical* path connecting the top and the bottom sides of the inner box.

this algorithm is applicable to both problems **P1** and **P2** without additional restrictions and it is more suitable for interactive segmentation since it is faster.

We use an energy function similar to other graph cut methods, discussed in section 2.3.1: a unary term defined as the negative log-likelihood with respect to foreground and background GMM colour models and a pairwise term incorporating both an Ising prior and a contrast dependent component and defined in a 8-connected grid graph. The graph defining connectivity, $(\mathcal{V}, \mathcal{F})$, is a 4-connected grid graph.

3.5.1 DijkstraGC for extraction of thin elongated structures

We have tested DijkstraGC on 15 images with a total of 40 connectivity problems. Fig. 3.9, 3.10 and 3.15 show some results, where we compare graph cut, using scribbles only, with DijkstraGC, where the user set additional clicks after obtaining the graph cut result. These results show the potential of using a connectivity constraint and DijkstraGC to minimise the user effort in extracting elongated structures that are typically cut off due to the "shrinking bias" of graph cut methods. To obtain a satisfying result with DijkstraGC the user only needs some additional clicks and the selection of a width parameter δ, which is a considerable reduction in the amount of user interaction needed. For the last example in Fig. 3.15 the number of clicks necessary to extract the segmentation was 11 since the thin structures we want to segment (the legs of the spider) intersect each other and the path that DijkstraGC computes goes throw the already segmented leg.

The running time presented in the last column of Fig. 3.15 is the combined time for processing all the clicks in the image, and it is, as to be expected, related to the number of clicks and image size. The optimised version of DijkstraGC (Fig. 3.7) improved the runtime over the

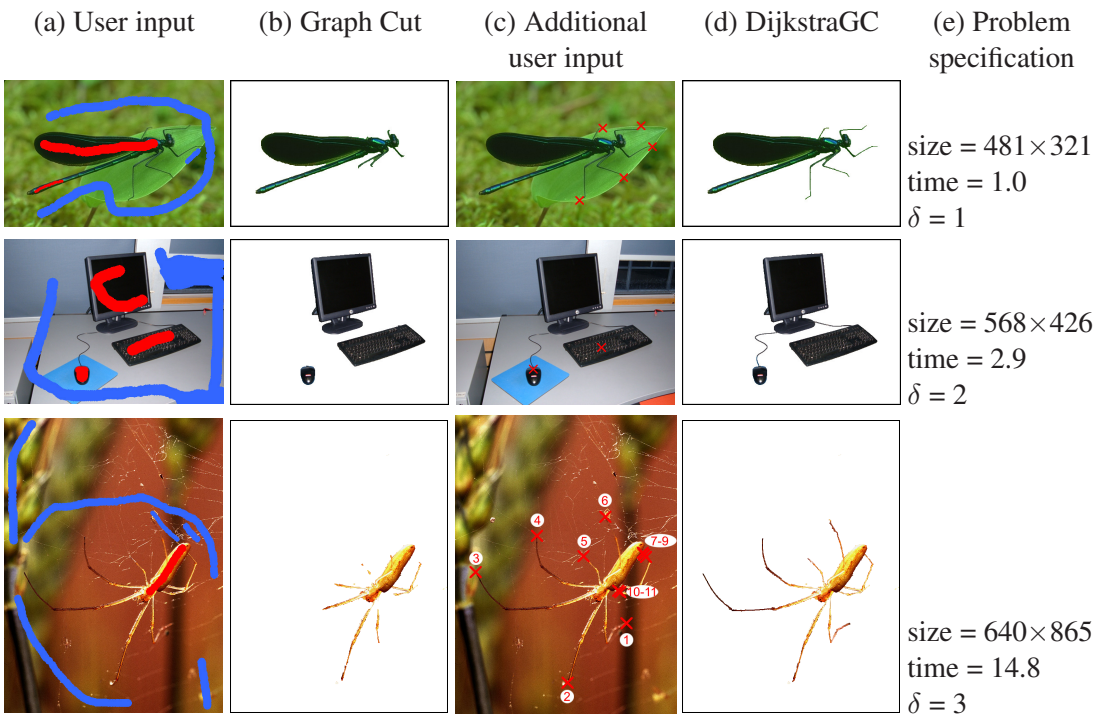


Figure 3.15: Results of the DijkstraGC algorithm. (a) original images with user scribbles; (b) Graph Cut results; (c) Selection of sites for connectivity, where numbers present the input order; (d) DijkstraGC results; (e) Problem specification: image size, running time for DijkstraGC (on 2.16 GHz CPU with 2GB RAM), and minimum width specified by the user.

simple version (Fig. 3.5) from, e.g. 28.4 to 14.8 seconds for the last image in Fig. 3.15.

Direction of DijkstraGC

Swapping the nodes s and t, i.e. changing the direction of DijkstraGC, may lead to two different segmentations as seen in the example of fig. 3.6. However the two segmentations usually differ only by a small number of pixels (on average less than 1% of the number of pixels in set $[\boldsymbol{x}]$) and the difference is often not visually significant.

In contrast, the difference in speed can be substantial. In the examples the run time was on average reduced by half if the "source" node s was in the smaller component (out of the two components that should be connected). Accordingly, this was chosen as the default option and used for the results presented.

3.5.2 Optimality of DijkstraGC

The Dual Decomposition algorithm, described in section 3.3.2, gives both a solution for a special case of **P1** and a lower bound on the optimal value of **P1**. Although this technique is not useful for a practical system, since the running time is on average 3 hours, it can be used to assess the optimality of DijkstraGC.

We considered 40 connectivity problems (i.e. user clicks) where the Dual Decomposition

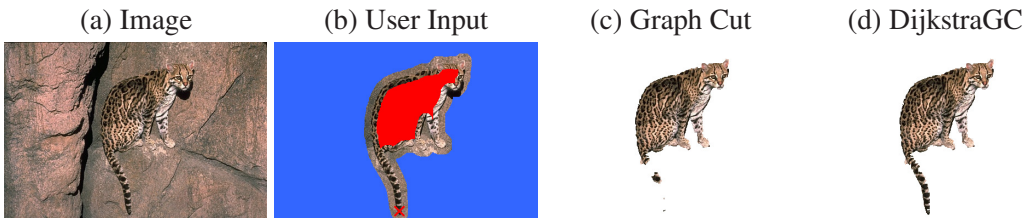


Figure 3.16: Optimality of DijkstraGC. An example of a problem for which both DijkstraGC and the decomposition method give the optimal result.

approach is applicable, i.e. all pixels at the image boundary are background. Another restriction for this approach is that we have to use a planar graph (4-connected 2D grid) for maxflow computations. For 12 out of the 40 problems the Dual Decomposition algorithm gave the global optimum. It is a positive result that for all these 12 cases also DijkstraGC returned the global optimum. One of such example is shown in Fig. 3.16.

For all the other problems we observed that the result provided by DijkstraGC was always better in terms of energy value than the result of the Dual Decomposition method.

3.5.3 DijkstraGC for the bounding box tightness constraint

In this section, we compare the DijkstraGC algorithm with the pinpointing algorithm originally proposed in [68] for energy minimisation under the tightness constraint. The pinpointing algorithm was introduced as a rounding scheme for the solution of the Linear Programming relaxation and it was later used as a heuristic technique based on priority maps. The high-level idea is to sequentially add points to the segmentation until the tightness constraint is satisfied. The order in which the points are added is given by the priority map.

We report results in the 50 images of the GrabCut dataset [91], similarly to [68]. We use the author's implementation for the different algorithms proposed in [68] and for defining the unaries and pairwise terms of the energy function. These are defined in a similar way to other graph cut approaches [91], with small changes in weighting and in initialisation of the GMM colour models.

We compare the algorithms both in terms of energy minimisation and segmentation accuracy. In our experiments, the graph cut result already satisfies the tightness constraint for 28 images. For the other 22 images we impose the tightness constraint using DijkstraGC and pinpointing using three different priority maps: the LP solution[2], the unary potentials and Min-marginals (see [68] for more details). We report results for the best, in terms of energy, of these three solutions.

Table 3.1 shows the results for this experiment. In the first column, we report the error

[2]We were not able to compute the solution for LP relaxation for 8 images due to memory constraints.

Method	Error - 50	Error - 22	Best energy (including ties)
Graph Cut	7.0	9.3	-
DijkstraGC	5.5	6.0	15 (20)
Pinpointing	5.3	5.6	2 (7)

Table 3.1: Comparing DijkstraGC with pinpointing for the bounding box constraint. We compare DijkstraGC and Pinpointing [68] for energy minimisation under the bounding box tightness constraint. Imposing this constraint reduces the error rate when compared with graph cut methods. The first two columns report error rate for two different sets: (Error - 50) for the full GrabCut dataset and (Error - 22) for the subset of images where the bounding box constraint is not immediately satisfied. The last column shows the number of images for which each algorithm performs the best, with the number in parenthesis including images where they performed the same.

rate (percentage of mislabelled pixels inside the bounding box) for all the images in the GrabCut dataset. In the second column, we only consider the images for which the bounding box tightness constraint is not satisfied. The last column shows which of the methods, DijkstraGC or Pinpointing, gives lower energy. DijkstraGC performs better in terms of energy for 15 out of 22 images, the same for 5 images and worse for 2 images.

We show results of using the DijkstraGC algorithm for this constraint in Fig. 3.17. For these images, the bounding box constraint forces the solution to better agree with the user input, considerably improving over graph cut methods.

The original paper where the bounding box constraint was introduced [68] suggests a further use of the constraint in an iterative framework similar to GrabCut [91]. Since our goal was to compare the algorithms' performance in terms of energy minimisation, we do not follow this approach, since the energy would change in each iteration and it would stop being comparable[3].

3.6 Discussion and limitations

In this chapter, we presented different applications of the connectivity constraints and showed how it can be useful to overcome some of the limitations of graph cut methods, in particular the "shrinking bias" and its effects when segmenting objects with thin elongated structures.

We have also discussed that minimising the energy under the connectivity constraint **C1** does not always produce a visually correct segmentation and it can lead to 1-pixel wide segmentations. We call this limitation a "1-pixel width bias" and further discuss its properties and ways to overcome it in this section.

We start by discussing the reasons that lead to the "1-pixel width bias". Recall that the form of the energy (3.2) is fixed and that problem **P1** only differs from traditional graph cuts in

[3]The randomness associated with fitting the Gaussian Mixture Models makes it impossible to compare the energy obtained with the different methods after updating the colour models.

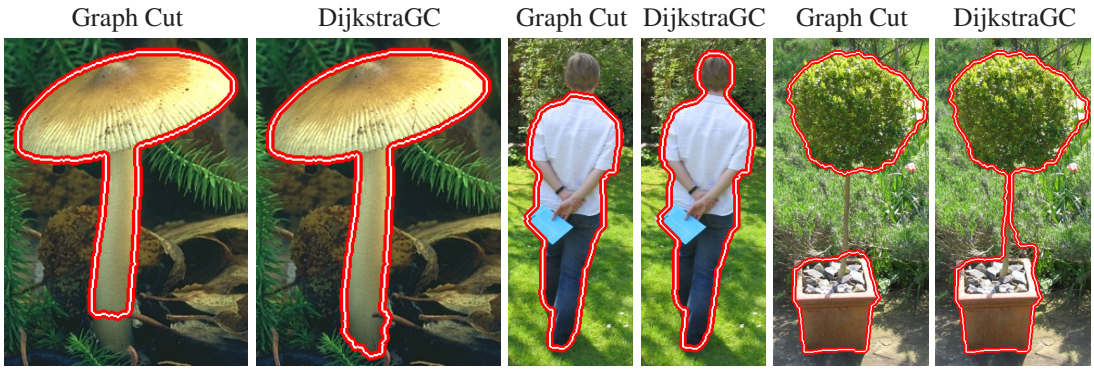
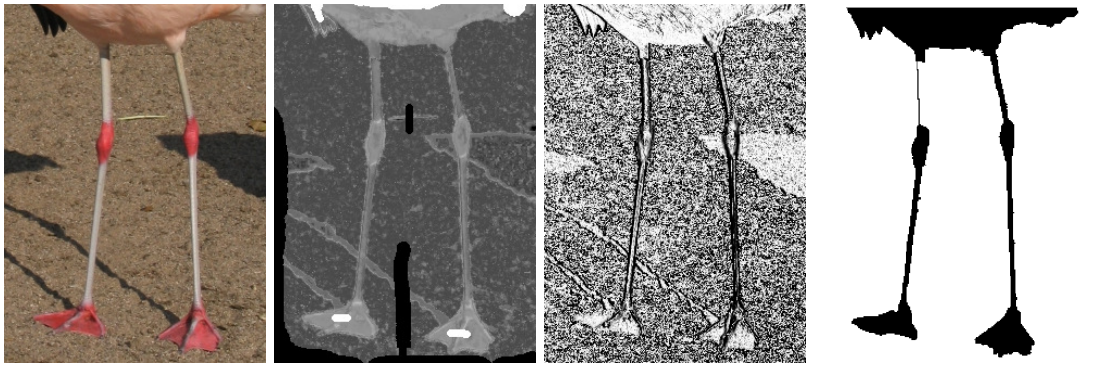


Figure 3.17: Results for the bounding box tightness constraint. We compare the results obtained with graph cuts and with DijkstraGC algorithm. The images were cropped to the size of the bounding box. The tightness constraint helps providing a result that better agrees with the user input and it can also prevent incorrect disconnected segmentations, e.g. the last image. For these examples, the result obtained with pinpointing was similar to the DijkstraGC algorithm.

the addition of the connectivity constraint. Assume that this connectivity constraint is not satisfied by the optimal solution of the problem without constraints. By enforcing this constraint, the label of some pixels that were initially assigned to background becomes foreground. This addition will cause an increase of the energy associated with a feasible solution. Since the goal is to choose a feasible solution with minimum energy, intuitively, this feasible solution should differ as little as possible from the original optimal solution of the unconstrained problem, i.e. the number of pixels that change the label to foreground should be limited. Since a path connecting the two special nodes is enough to ensure the solution is feasible and due to the preference of adding to foreground a small number of pixels, the optimal solution of the connectivity problem tends to only differ from the unconstrained solution by this path.

This "1-pixel width bias" is partially overcome by the addition of the pairwise term, that prefers an alignment between the boundaries of the segmentation and the image edges. This term may be, however, insufficient as shown in Fig. 3.10. We give an intuitive explanation for this case in Fig. 3.18. As can be seen in Fig. 3.18 (c) the pairwise term is lower close to image edges. However, the image edges span at least 2 pixels, making it possible to find a segmentation which is 1-pixel wide in some areas and still aligned with image edges, as seen in Fig. 3.18 (d).

As discussed in section 3.4.1, this limitation can be overcome by imposing a path with minimum width, using constraint **C2**. This minimum width ensures that the segmentation will snap to the correct image edge. It is also clearer that the minimum width does not need to be the exact width of the elongated structure that we want to segment, it only needs to be large enough to ensure that the segmentation does not align with the same image boundary twice.



(a) Image (b) Unary term (c) Pairwise term (d) Connected segmentation

Figure 3.18: The "1-pixel width bias" explained. This image shows a zoomed version of the unary terms (b) and horizontal pairwise terms (c) corresponding to the example in Fig. 3.10. Since a 1-pixel wide path is enough to ensure constraint **C1** is satisfied, the segmentation (d) presents that behaviour in some parts. The pairwise term is not enough to overcome this behaviour since an image edge usually spans more than a single graph edge. This can be seen in (c) and (d) where the pairwise term is less strong in both sides of the 1-pixel wide segmentation.

The "1-pixel width bias" can also be overcome by using superpixels instead of pixels, as done in [82]. In practice, the bias is still present, but since a node in the graph now encloses many pixels, it is less visible. This workaround can however by undesirable in the scenario of segmenting thin structures, since those could be lost in an incorrect superpixelization.

The effects of the "1-pixel width bias" are harder to overcome in the case of the bounding box tightness constraint. Fig. 3.19 shows examples of images where the graph cut segmentation is not tight with respect to the bounding box. Imposing the tightness constraint using DijkstraGC gives segmentations which are visually incorrect but have smaller energy than the ones obtained using Pinpointing. This observation is counter intuitive, since the results obtained with Pinpointing are visually better. Moreover, the results obtained with DijkstraGC do not only suffer from the "1-pixel width bias", but also satisfy the constraint by imposing an incorrect path.

Assuming that the solution provided by DijkstraGC is visually closer to the global optimum solution[4], we can conclude that this energy formulation is not always adequate to this particular problem. Furthermore, the Pinpointing algorithm tends to hide the properties of the energy function, since it outputs a solution with smaller error rate. Only the use of DijkstraGC to solve the same problem reveals that imposing the tightness constraint is not sufficient to ensure that a visually plausible segmentation is obtained.

Note that, the authors of [68] suggest to update the colour models in a GrabCut fashion

[4]We do not have any guarantee regarding the optimality of the DijkstraGC solution since using the lower bound provided by the LP relaxation [68] it was not possible to attest its optimality.

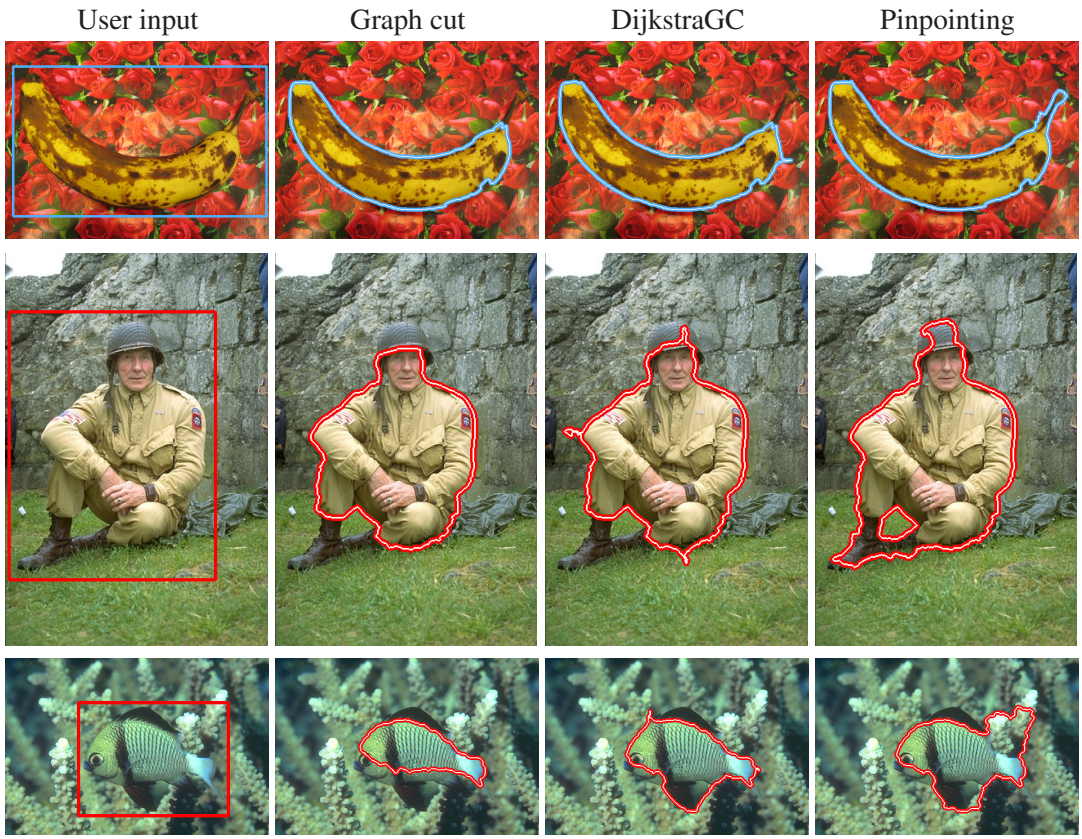


Figure 3.19: Failure cases of the bounding box tightness constraint. The segmentations obtained using DijkstraGC suffer from the "1-pixel width bias", however, they have smaller energy than the segmentations obtained with Pinpointing.

[91], i.e. alternating between two steps: (1) updating the colour models and (2) recomputing the segmentation using Pinpointing. This iterative approach may help mitigating the limitations discussed. However, replacing Pinpointing by DijkstraGC in step (2) would deteriorate the performance of the system, since updating the colour models would not help recovering from the incorrect paths obtained in the examples shown in Fig. 3.19.

3.7 Conclusion

In this chapter, we discussed the advantages of including connectivity constraints when formulating different tasks in interactive segmentation. These constraints considerably help reducing the amount of user interaction necessary to segment thin structures and can also be used to impose an intuitive bounding box tightness constraint.

We also presented a new algorithm, DijkstraGC, that computes a segmentation satisfying those constraints. Although in general DijkstraGC is not guaranteed to compute the global minimum of our NP-hard optimisation problem, we believe that in practice it performs well. This claim is supported by two facts: (i) running DijkstraGC in different directions gives almost

the same result, and (ii) DijkstraGC computes the optimal solution for some particular instances (see sec. 3.5.2).

The connectivity constraints **C1** and the DijkstraGC algorithm were initially proposed for the task of extracting thin elongated structures. We have shown that they not only succeed in this task but also can be used in other tasks. In particular, DijkstraGC outperforms the pinpointing algorithm for energy minimisation under the bounding box tightness constraint and reveals the limitations of this formulation.

Chapter 4

Joint optimisation of segmentation and appearance

4.1 Introduction

In the previous chapter we discussed a prior model for segmentation that focused on its shape properties. We started by translating the connectivity prior into a higher-order model and discussed adequate optimisation algorithms. In this chapter we will discuss a different type of higher-order model that focuses on the appearance properties of the segmentation.

A common criteria for segmentation is to find regions that have *consistent appearance* and that differ from the remaining regions, i.e. have high intra-region similarity and low inter-region similarity. This is an intuitive assumption for object segmentation since objects are usually represented by compact appearance models and are distinct from the surrounding background. Furthermore, this assumption forms the motivation for many successful energy formulation approaches to both binary and multi-region segmentation. The Mumford-Sha functional [79], the Chan-Vese functional [25] and the GrabCut functional [91] are some notable examples.

Although some of these approaches are motivated by different principles, e.g. a Bayesian justification [79] or a Minimum Description Length model [117], they use similar energy functions that can be included in a single framework that we now discuss.

Given a labelling $\boldsymbol{x}$, with $x_p \in \{1, .., L\}$, we define regions $R_1, ..., R_L$ as $R_l = \{p \mid x_p = l\}$, i.e. region R_l contains all the pixels with label l.

The energy function has the form:

$$E(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{l=1,..,L} \left[\sum_{p \in R_l} F(y_p, \theta^l) \right] + Length(\boldsymbol{x}_{contour}) \tag{4.1}$$

where $\boldsymbol{y}$ corresponds to the observed image measurements, e.g. colour or texture, $\boldsymbol{x}_{contour}$

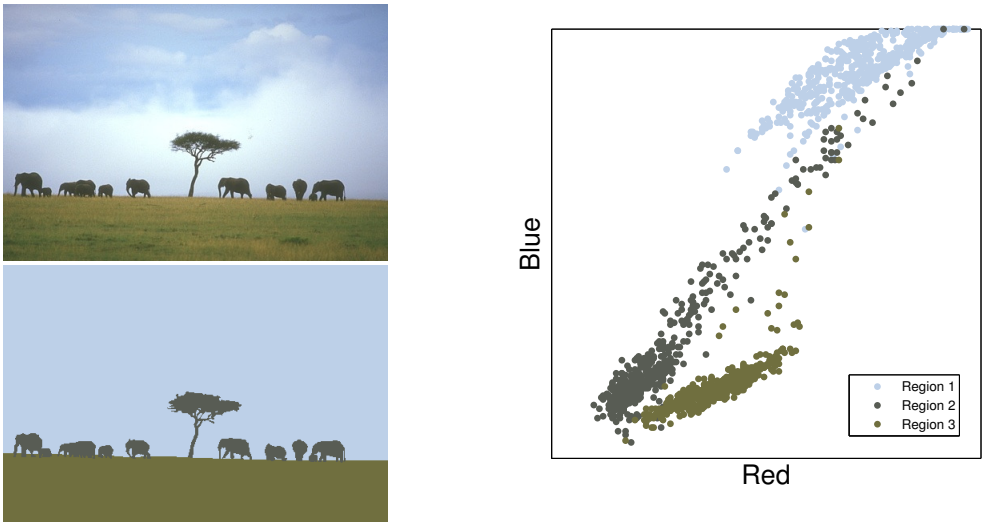


Figure 4.1: Illustration of the joint model. The goal is to jointly infer the segmentation and an appearance model for the colour of the pixels in each region. The left side shows an image and a good segmentation for that image. On the right side, we show the colour space spanned by each of the regions, where each point represents a pixel. An appearance model is inferred for each region and the methods discussed rely on a good separation between these models.

Definition of $F(y_p, \theta^l)$	**Definition of** θ^l	**Used in**
$F(y_p, \theta^l) = (y_p - \theta^l(p))^2$	Smooth function defined in R_l	Mumford-Shah model [79]
$F(y_p, \theta^l) = (y_p - \theta^l)^2$	Estimated intensity: integer number in [0, 255]	Chan-Vese model [25]
$F(y_p, \theta^l) = -\log(\Pr(y_p \vert \theta^l))$	Probability distribution (e.g. single Gaussian or GMM)	GrabCut model [91], [117, 28, 31]

Table 4.1: Summary of models using an energy function of the form (4.1).

corresponds to the contour of the segmentation, θ^l are appearance models for each region and $F(.)$ measures the agreement between the appearance models and the observed variables in that region. An illustration of this model is given in Fig. 4.1.

Previous methods differ in the way the function $F(.)$ and the appearance models θ^l are defined. Models for multi-region segmentation can also include an extra term in function (4.1) that penalises the number of regions (or equivalently the number of labels), encouraging a small number of regions [117, 31].

Table 4.1 summarises some of the special cases of energy function (4.1) previously proposed, detailing the definition of $F(.)$ and θ^l.

Although this type of model has been extensively and successfully used for both multi-region [79, 117, 31] and binary segmentation [25, 91], the optimisation is typically performed

(a) User input (b) EM-style solution (c) Our solution

Figure 4.2: Overcoming the limitations of the EM-style optimisation. Given the image and bounding box in (a) the EM-style method produces result in (b). Our new algorithm gives the segmentation in (c) which is not only visually better but also has a lower energy (for the first image it is the global optimum of our energy formulation).

in a coordinate descent fashion without guarantees of optimality. Most common algorithms alternate between two steps: (1) fixing the models θ^l and updating the segmentation $\boldsymbol{x}$ and (2) fixing the segmentation and updating the models θ^l. These algorithms are usually termed EM-style optimisation techniques.

In this chapter we discuss a global optimisation strategy for a special case of this model. For simplicity we consider colour as the only appearance feature. However, other features could be included in a similar way. Furthermore, we use this model for the task of interactive image segmentation, similarly to the GrabCut model [91], and use a simple rectangle containing the object as user input. We show that our new optimisation outperforms for most images the iterative EM-style approach previously used. Two examples are shown in Fig. 4.2.

This chapter is structured as follows. Section 4.2 introduces the problem formulation and further discusses related work. In section 4.3 the model is rewritten using an energy with higher-order cliques, in a way that the segmentation is the only unknown variable. The new formulation reveals an interesting bias of the model towards balanced segmentations, i.e. the preference of fore- and background segments to have similar area. Then in section 4.4 we discuss the new optimisation method for this higher-order energy. The method presented relies on the parametric maxflow algorithm and can improve over the local minima of an EM-style algorithm. It also provides a lower bound and we show that in practice the bound is often tight. The experimental section 4.5 investigates our approach on a large dataset. We further discuss properties and limitations of the optimisation and of the model in section 4.6 and conclude in section 4.7.

4.2 Problem formulation

We will use a discrete energy function of the form of equation (4.1). Since we will focus on interactive image segmentation, we consider only two labels. Furthermore, θ^l represents a probability distribution. The energy function used is most related with the GrabCut functional [91], having the final form:

$$E(\boldsymbol{x}, \theta^0, \theta^1) = \underbrace{\sum_{p \in \mathcal{V}} -\log \Pr(y_p|\theta^{x_p})}_{U(\boldsymbol{x}, \theta^0, \theta^1)} + \underbrace{\sum_{(p,q) \in \mathcal{N}} w_{pq}|x_p - x_q|}_{P(\boldsymbol{x})} \tag{4.2}$$

where $\mathcal{V}$ is the set of pixels, $\mathcal{N}$ is the set of neighbouring pixels, and $x_p \in \{0, 1\}$ is the segmentation label of pixel p. The second term ($P(\boldsymbol{x})$) is the previously mentioned contrast sensitive edge term. This term can be seen as a discrete version of a measure of the contour length [15]. We refer to the energy function (4.2) as the *joint model* to emphasise the fact that the appearance models are also variables that are jointly optimised with the segmentation.

Probabilistic appearance models for colour

Many different probabilistic distributions have been previously used as colour models for segmentation. Two popular ones are histograms [14] and Gaussian Mixture Models (GMMs) [11, 91]. Simpler models, like a single Gaussian, are also common, but more suited to the multi-region problem [117].

We will use histograms for colour modelling and their use will be essential for our approach. Note, it is well-known that Maximum Likelihood estimation of a GMM model is strictly speaking an ill-posed problem since by fitting a Gaussian to the colour of a single pixel we may get an infinite likelihood[1].

We assume that the histogram has B bins indexed by $b = 1, \ldots, B$. Each pixel p in the image is assigned to a single bin and the bin in which the pixel falls is denoted as b_p. $\mathcal{V}_b \subseteq \mathcal{V}$ denotes the set of pixels assigned to bin b. θ^0 and θ^1 are vectors in $[0, 1]^B$ representing the distribution over fore- and background, respectively, and sum to 1, i.e. $\sum_b \theta_b^0 = \sum_b \theta_b^1 = 1$. The likelihood model is then given by

$$U(\boldsymbol{x}, \theta^0, \theta^1) = \sum_p -\log \theta_{b_p}^{x_p} . \tag{4.3}$$

[1]For more details see [10], section 9.2.1. This problem can be overcome by considering a prior and MAP estimation of the GMM.

Optimisation

The main goal of this chapter is to study the problem of minimising energy function (4.2). As we show in Appendix A.4 the problem is NP-hard.

As previously discussed, this type of model is typically optimised using an EM-style algorithm alternating between the following steps: (1) Fix colour models (θ^0, θ^1) and minimise energy (4.2) over segmentation $\boldsymbol{x}$. (2) Fix segmentation $\boldsymbol{x}$ and minimise energy (4.2) over colour models (θ^0, θ^1).

For discrete energy functions, like the one we use, the first step can be solved via a maxflow algorithm, similarly to [91]. For continuous formulations, this step is typically solved using level sets [25, 28]. Note that the maxflow algorithm obtains a global minimum for step (1) not achieved using the level set method.

The second step can be solved via standard machine learning techniques for fitting a model to data. Each step is guaranteed not to increase the energy, but the procedure may get stuck in a local minimum. Two examples are shown in Fig. 4.2.

In order to avoid local minima, a branch-and-bound framework was proposed in [67]. They demonstrated that a global minimum can be obtained for 8 bins, when allowed models (θ^0, θ^1) are restricted to a set with 2^{16} elements. Unfortunately, branch-and-bound techniques suffer in general from an exponential explosion, so increasing the number of allowed models would present a problem for the method in [67].

We are not aware of any existing technique which can assess the optimality of the EM-style algorithm when the number of bins is large, or when the space of models (θ^0, θ^1) is unrestricted.

4.3 Rewriting the energy via higher-order cliques

Our new optimisation scheme relies on rewriting the energy (4.2) so that it solely depends on the unknown segmentation $\boldsymbol{x}$. This is achieved by noting that the optimal θ^0 and θ^1 can be written as a function of the segmentation.

Let us denote by n_b^l the number of pixels p that fall into bin b and have label l, i.e. $n_b^l = \sum_{p \in \mathcal{V}_b} \delta(x_p - l)$. All these pixels contribute the same cost, $-\log \theta_b^l$, to the term $U(\boldsymbol{x}, \theta^0, \theta^1)$, therefore we can rewrite it as

$$U(\boldsymbol{x}, \theta^0, \theta^1) = \sum_{l=0,1} \sum_b -n_b^l \log \theta_b^l \,. \tag{4.4}$$

It is well-known that for a given segmentation $\boldsymbol{x}$ distributions θ^0 and θ^1 that minimise $U(\boldsymbol{x}, \theta^0, \theta^1)$ are simply the empirical histograms computed over appropriate segments:

$$\theta_b^l = \frac{n_b^l}{n^l} \tag{4.5}$$

where n^l is the number of pixels with label l: $n^l = \sum_{p \in \mathcal{V}} \delta(x_p - l)$.

Plugging optimal θ^0 and θ^1 into the energy gives the following expression:

$$E(\boldsymbol{x}) = \min_{\theta^0, \theta^1} E(\boldsymbol{x}, \theta^0, \theta^1) = \sum_b g_b(n_b^1) + g(n^1) + P(\boldsymbol{x}) \tag{4.6}$$

$$\text{with} \quad g_b(n_b^1) = -n_b^1 \log(n_b^1) - (n_b - n_b^1) \log(n_b - n_b^1) \tag{4.7}$$

$$g(n^1) = n^1 \log(n^1) + (n - n^1) \log(n - n^1) \tag{4.8}$$

where $n_b = |\mathcal{V}_b|$ is the number of pixels in bin b and $n = |\mathcal{V}|$ is the total number of pixels. Functions $g_b(\cdot)$ in equation (4.7) are concave and symmetric about $n_b/2$ and function $g(\cdot)$ in equation (4.8) is convex and symmetric about $n/2$.

Interactive segmentation

It is easy to see that the energy of $\boldsymbol{x}$ is the same as the energy of $\mathbf{(1 - \boldsymbol{x})}$ which corresponds to flipping the labels. Therefore, there is an ambiguity on the labels of the optimal solution, since the flipped solution is also optimal.

This ambiguity is easily overcome in interactive segmentation, where the user provided region seeds can be seen as hard-constraints. To represent these hard-constraints, we add an extra-term in the energy (4.6), a sum of unary terms of the form:

$$H(\boldsymbol{x}) = \sum_{p \in S_B} m x_p + \sum_{p \in S_F} m(1 - x_p) \tag{4.9}$$

where S_B and S_F are sets of pixels that correspond to the background and foreground region seeds respectively, and m is a sufficiently large constant to ensure that the region seeds are satisfied.

The user input we consider is a bounding box surrounding the object of interest. This type of input provides essential information regarding the location of the object in the image.

Bias of the model

The form of equation (4.6) allows an intuitive interpretation of this model. The first term (sum of concave functions) has a preference towards assigning all pixels in the same bin to the same segment. The convex part prefers *balanced* segmentations, i.e. segmentations in which the background and the foreground have the same number of pixels.

This bias is most pronounced in the extreme case when all pixels are assigned to unique

bins, so $n_b = 1$ for all bins. Then all concave terms $g_b(n_b^1)$ are constants, so the energy consists of the convex part $g(n^1)$ and pairwise terms. Note, however, that the bias disappears in the other extreme case when all pixels are assigned to the same bin ($B = 1$); then concave and convex terms cancel each other. The lemma below gives some intuition about intermediate cases.

Lemma 4. *Let $\mathcal{V}_b$ be the set of pixels that fall in bin b. Suppose that pixels in $\mathcal{V}_b$ are not involved in pairwise terms of the energy, i.e. for any $(p, q) \in \mathcal{N}$ we have $p, q \notin \mathcal{V}_b$. Also suppose that energy* (4.6) *is minimised under user-provided hard constraints that force a certain subset of pixels to the background and another subset to the foreground. Then there exists a global minimiser $\boldsymbol{x}$ in which all unconstrained pixels in $\mathcal{V}_b$ are assigned either completely to the background or completely to the foreground.*

A proof of this lemma is given in Appendix A.5. Note that $g_b(0) = g_b(n_b)$, so if pixels in $\mathcal{V}_b$ are not involved in hard constraints then in the absence of pairwise terms the labelling of $\mathcal{V}_b$ will be determined purely by the convex term $g(n^1)$, i.e. the model will choose the label that leads to a more balanced segmentation.

4.4 Optimisation via Dual Decomposition

The full energy derived in the previous section has the following form:

$$E(\boldsymbol{x}) = \underbrace{\sum_b g_b(n_b^1) + \sum_{(p,q)\in\mathcal{N}} w_{pq}|x_p - x_q| + H(\boldsymbol{x})}_{E^1(\boldsymbol{x})} + \underbrace{g(n^1)}_{E^2(\boldsymbol{x})} \tag{4.10}$$

where $g_b(\cdot)$ are concave functions, $g(\cdot)$ is a convex function and $H(\boldsymbol{x})$ corresponds to the unaries that come from the user hard constraints. Recall that n_b^1 and n^1 are functions of the segmentation: $n_b^1 = \sum_{p\in\mathcal{V}_b} x_p$, $n^1 = \sum_{p\in\mathcal{V}} x_p$. This energy function is composed of a submodular part ($E^1(\boldsymbol{x})$) and a supermodular ($E^2(\boldsymbol{x})$) part.

As we showed, minimising function (4.10) is an NP-hard problem. Instead of the EM-style two step approach, we use a Dual Decomposition method to minimise this function. We define the lower bound as follows:

$$\begin{aligned}\Phi(\boldsymbol{\lambda}) &= \underbrace{\min_{\boldsymbol{x}_1}[E^1(\boldsymbol{x}_1) - \langle\boldsymbol{\lambda}, \boldsymbol{x}_1\rangle]}_{\Phi^1(\boldsymbol{\lambda})} + \underbrace{\min_{\boldsymbol{x}_2}[E^2(\boldsymbol{x}_2) + \langle\boldsymbol{\lambda}, \boldsymbol{x}_2\rangle]}_{\Phi^2(\boldsymbol{\lambda})} \\ &\leq \min_{\boldsymbol{x}} E(\boldsymbol{x})\end{aligned} \tag{4.11}$$

where $\boldsymbol{\lambda}$ is the dual vector in $\mathbb{R}^n$, $n = |\mathcal{V}|$.

Note that both minima can be computed efficiently. In particular, the first term can be optimised via a reduction to a min s-t cut problem [53]. In section 4.4.1 we review this reduction and propose one extension.

To get the tightest possible bound, we need to maximise $\Phi(\boldsymbol{\lambda})$ over $\boldsymbol{\lambda}$. Function $\Phi(\cdot)$ is concave, therefore we could use some standard concave maximisation technique, such as a subgradient method which is guaranteed to converge to an optimal bound. Note, such decomposition was used as an example in [112] for enforcing the area constraint; the bound was optimised via a max-sum diffusion algorithm.

We will show that in our case the tightest bound can be computed in polynomial time using a parametric maxflow technique [37].

Theorem 5. *Suppose that continuous functions $\Phi^1, \Phi^2 : \mathbb{R}^{|\mathcal{V}|} \to \mathbb{R}$ have the following properties:*

(a)

$$\Phi^1(\boldsymbol{\lambda} + \delta \cdot \chi_p) \geq \Phi^1(\boldsymbol{\lambda}) + \min_{x \in \{0,1\}} \{-x\delta\} \tag{4.12}$$

for all vectors $\boldsymbol{\lambda}$ and nodes $p \in \mathcal{V}$, where χ_p is the vector of size $|\mathcal{V}|$ with $(\chi_p)_p = 1$ and all other components equal to zero;

(b)

$$\Phi^2(\boldsymbol{\lambda}) = \min_{\boldsymbol{x} \in \{0,1\}^{|\mathcal{V}|}} E^2(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{x} \rangle \tag{4.13}$$

where $E^2(\boldsymbol{x}) = g(\sum_{p \in \mathcal{V}} x_p)$ and function $g(\cdot)$ is convex on $[0, n]$ where $n = |\mathcal{V}|$, i.e. $2g(k) \leq g(k-1) + g(k+1)$ for $k = 1, \ldots, n-1$.

Under these conditions function $\Phi(\boldsymbol{\lambda}) = \Phi^1(\boldsymbol{\lambda}) + \Phi^2(\boldsymbol{\lambda})$ has maximiser $\boldsymbol{\lambda}$ such that $\lambda_p = \lambda_q$ for any $p, q \in \mathcal{V}$.

A proof is given in the Appendix A.6.

Theorem 5 implies that it suffices to consider vectors $\boldsymbol{\lambda}$ of the form $\boldsymbol{\lambda} = \lambda \mathbf{1}$, where $\mathbf{1}$ is the vector in $\mathbb{R}^n$ with components 1. It is easy to see that we can evaluate $\Phi(\lambda \mathbf{1})$ efficiently for all values of λ. Indeed, we need to minimise functions

$$E^1(\boldsymbol{x}) - \lambda \langle \mathbf{1}, \boldsymbol{x} \rangle \ , \quad E^2(\boldsymbol{x}) + \lambda \langle \mathbf{1}, \boldsymbol{x} \rangle \ .$$

For the first function we need to solve a *parametric maxflow* problem [37]. Recall that the parametric maxflow problem is defined as: *minimise energy functions $E^\lambda(\boldsymbol{x})$ of binary variables for different values of parameter λ where: $E^\lambda(\boldsymbol{x}) = \sum_p (a_p + b_p \lambda) x_p +$*

$\sum_{(p,q)} \phi_{pq}(x_p, x_q)$. In our case a_p is only different from zero if p belongs to a region seed and $b_p = 1$ for all p.

The result is a nested sequence of m solutions $\boldsymbol{x}$, $2 \leq m \leq n+1$ and the corresponding $m-1$ breakpoints of λ. Accordingly, function $\min_{\boldsymbol{x}}[E^1(\boldsymbol{x}) - \lambda\langle \mathbf{1}, \boldsymbol{x}\rangle]$ is a piecewise-linear concave function. All solutions and breakpoints can be computed efficiently by a divide-and-conquer algorithm (see e.g. [56] for a review).

The second function can also be handled efficiently. In fact, if $E^2(\boldsymbol{x}) = g(\sum_{p\in\mathcal{V}} x_p)$ then $\min_{\boldsymbol{x}}[E^2(\boldsymbol{x})+\lambda\langle \mathbf{1}, \boldsymbol{x}\rangle]$ is a piecewise-linear concave function with breakpoints $g(n-1)-g(n)$, $g(n-2)-g(n-1), \ldots, g(0)-g(1)$. This implies that $\Phi(\cdot)$ is a piecewise-linear concave function with at most $2n$ breakpoints. In our implementation we construct this function explicitly; after that computing the tightest lower bound (i.e. the maximum of $\Phi(\cdot)$) becomes trivial. Note, however, that this is not the most efficient scheme: in general, maximising a concave function does not require evaluating all breakpoints.

It remains to specify how to get labelling $\boldsymbol{x}$. From the sequence of solutions obtained using parametric maxflow, we choose the one with minimum energy to be the solution for the original problem.

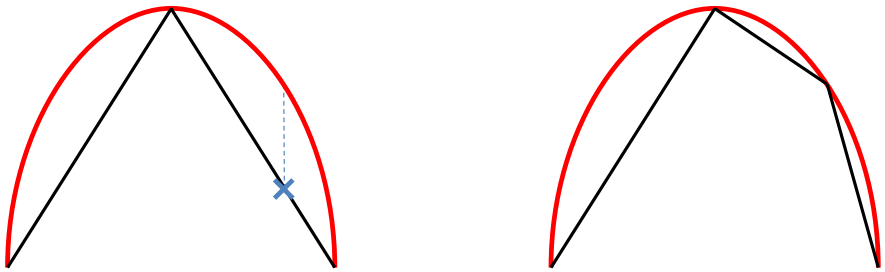
4.4.1 Minimising submodular functions with concave higher order potentials

The decomposition approach described above requires minimising functions of the form

$$f(\boldsymbol{x}) = \sum_p f_p(x_p) + \sum_{(p,q)} f_{pq}(x_p, x_q) + \sum_b g_b(n_b^1) \tag{4.14}$$

where terms $f_{pq}(\cdot,\cdot)$ are submodular and $g_b(\cdot)$ are concave functions. Since each function $g_b(n_b^1)$ is dependent on the labels of all nodes in $\mathcal{V}_b$, these functions correspond to potentials defined over higher order cliques. It is known that the problem of minimising $f(\cdot)$ can be reduced to a min s-t cut problem [53]. Let us review how this reduction works. Consider term g_b defined over subset $\mathcal{V}_b$. $g_b(\cdot)$ needs to be defined only for integer values $0, 1, \ldots, n_b = |\mathcal{V}_b|$ so we can assume without loss of generality that $g_b(\cdot)$ is a piecewise-linear concave function with β_b breakpoints. The method in [53] first represents the function as a sum of β_b piecewise-linear concave functions with *one* breakpoint. For each function we add an auxiliary variable which is connected to the source, to the sink and to all nodes in $\mathcal{V}_b$. Thus, the number of added edges is $O(n_b\beta_b)$. We review this construction in the end of this section.

In our case function $g_b(\cdot)$ is strictly concave, which implies $\beta_b = O(n_b)$. Thus, the method would add $O((n_b)^2)$ edges. This makes it infeasible in practice for large n_b; even keeping edges in memory would be a problem.



(a) (b)

Figure 4.3: Iterative procedure for concave higher-order potentials. We approximate the concave function (in red) with a piecewise linear function with increasing number of breakpoints. In the first iteration (a) we consider an approximation with a single breakpoint and obtain the optimal solution for that approximation, represented by the blue cross. In the following iteration (b) we improve the approximation by adding the breakpoint corresponding to the previous solution.

We use the following iterative technique instead. Let us approximate $g_b(\cdot)$ with a piecewise-linear concave function $\bar{g}_b(\cdot)$ whose set of breakpoints $\mathcal{B}_b$ satisfies $\{0, n_b\} \subseteq \mathcal{B}_b \subseteq \{0, 1, \ldots, n_b\}$. We require $\bar{g}_b(r) = g_b(r)$ for every breakpoint $r \in \mathcal{B}_b$. Using this property, we can uniquely reconstruct function $\bar{g}_b(\cdot)$ from the set $\mathcal{B}_b$. It is not difficult to see that $\bar{g}_b(r) \leq g_b(r)$ for all integer values of r in $[0, n_b]$.

We initialise sets $\mathcal{B}_b$ with a small number of breakpoints, namely $\{0, \lfloor n_b/2 \rfloor, n_b\}$. We then iterate the following procedure: (1) minimise function (4.14) in which terms $g_b(n_b^1)$ are replaced with approximations $\bar{g}_b(n_b^1)$; obtain optimal solution $\boldsymbol{x}$ and corresponding counts n_b^1; (2) for each bin b set $\mathcal{B}_b := \mathcal{B}_b \cup \{n_b^1\}$. We terminate if none of the sets $\mathcal{B}_b$ change in a given iteration. This procedure is illustrated in Fig. 4.3

This technique must terminate since sets $\mathcal{B}_b$ cannot grow indefinitely. Let $\boldsymbol{x}$ be the labelling produced by the last iteration. It is easy to verify that for any labelling $\boldsymbol{x}'$ there holds

$$f(\boldsymbol{x}') \geq \bar{f}(\boldsymbol{x}') \geq \bar{f}(\boldsymbol{x}) = f(\boldsymbol{x})$$

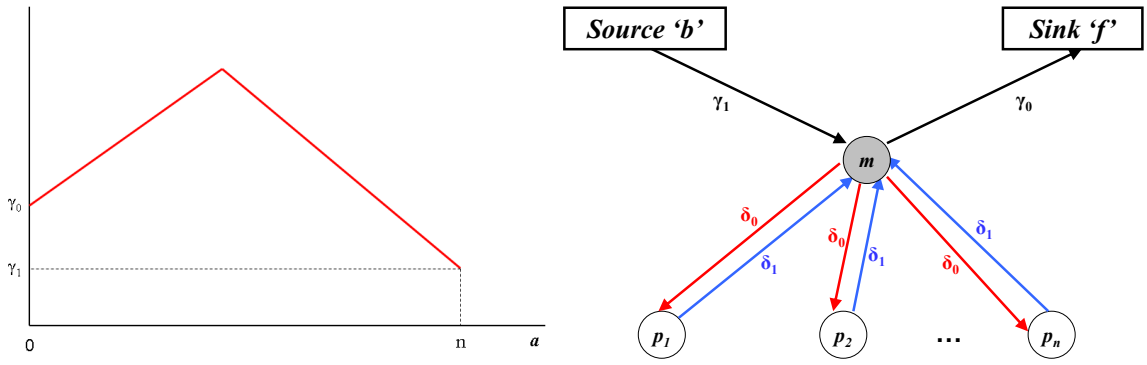
where $\bar{f}(\cdot)$ is the function minimised in the last iteration. Thus, $\boldsymbol{x}$ is a global minimum of function (4.14).

Construction for piecewise-linear concave functions with *one* breakpoint

We now review the reduction to pairwise terms of minimising a potential function of the form:

$$\phi_c(\boldsymbol{x}_c) = \psi_c(a) = \min\{a\delta_0 + \gamma_0, (n-a)\delta_1 + \gamma_1\} \tag{4.15}$$

where, $n = |c|$ is the total number of nodes in the clique, $a = \sum_{p \in c} x_p$ and δ_0, γ_0, δ_1 and γ_1 are positive constants. The form of this function is illustrated in Fig. 4.4 (a).



(a) Function of the form (4.15) (b) Graph construction

Figure 4.4: Higher-order potentials construction. Higher-order potentials of the form (a) can be converted into pairwise terms represented in the graph (b), by adding an extra variable m.

To transform this potential into a pairwise potential, an extra variable m is introduced. Fig. 4.4 (b) shows the additional node and edges added to the graph for this higher order potential. Note that the nodes $p_1, ..., p_n$ are the nodes that belong to the clique.

4.4.2 Semi-global iterative optimisation

In our experiments, we observed that for some images the Dual Decomposition technique performed rather poorly: the number of breakpoints obtained using parametric maxflow was small and none of those breakpoints corresponded to a good solution. In such cases we would probably need to resort to an EM-style iterative technique. In this section we describe how we can use the Dual Decomposition approach for such iterative minimisation.

Suppose that we have a current solution $\bar{\boldsymbol{x}}$. The EM-style approach would compute empirical histograms $(\bar{\theta}^0, \bar{\theta}^1)$ over $\bar{\boldsymbol{x}}$ using formulas (4.5) and then minimise energy $E^{\text{EM}}(\boldsymbol{x}) = E(\boldsymbol{x}, \bar{\theta}^0, \bar{\theta}^1)$. We now generalise this procedure as follows. Consider the energy function

$$\bar{E}(\boldsymbol{x}) = (1-\alpha)E^{\text{EM}}(\boldsymbol{x}) + \alpha E(\boldsymbol{x}) \tag{4.16}$$

where α is a fixed parameter in $[0, 1]$ and $E(\boldsymbol{x})$ is defined by (4.6). Note that $\alpha = 0$ gives the energy used by the EM approach, and $\alpha = 1$ gives the global energy (4.6).

Lemma 6. *Suppose that $\boldsymbol{x}$ is a minimiser of $\bar{E}(\cdot)$ for $\alpha \in (0, 1]$ and $\boldsymbol{x}^{EM}$ is a minimiser of $E^{EM}(\cdot)$. Then $E(\boldsymbol{x}) \leq E(\boldsymbol{x}^{EM})$. Furthermore, if $(\bar{\theta}^0, \bar{\theta}^1)$ is computed from some segmentation $\bar{\boldsymbol{x}}$ then $E(\boldsymbol{x}) \leq E(\boldsymbol{x}^{EM}) \leq E(\bar{\boldsymbol{x}})$.*

Proof. Denote $\beta = 1 - \alpha$. Optimality of $\boldsymbol{x}$ and $\boldsymbol{x}^{\text{EM}}$ imply

$$\begin{aligned} \beta E^{\text{EM}}(\boldsymbol{x}) + \alpha E(\boldsymbol{x}) &\leq \beta E^{\text{EM}}(\boldsymbol{x}^{\text{EM}}) + \alpha E(\boldsymbol{x}^{\text{EM}}) \\ \beta E^{\text{EM}}(\boldsymbol{x}^{\text{EM}}) &\leq \beta E^{\text{EM}}(\boldsymbol{x}) \end{aligned}$$

Original image GMM histogram Original image GMM histogram

Figure 4.5: Illustration of the histograms based on GMM. The 25 different colours shown correspond to the average colour of the pixels assigned to that bin.

Adding these inequalities and cancelling terms gives $\alpha E(\boldsymbol{x}) \leq \alpha E(\boldsymbol{x}^{\text{EM}})$, or $E(\boldsymbol{x}) \leq E(\boldsymbol{x}^{\text{EM}})$ since $\alpha > 0$. It is well-known that both steps of the EM-style method do not increase the energy; this implies the second claim $E(\boldsymbol{x}^{\text{EM}}) \leq E(\bar{\boldsymbol{x}})$. □

The lemma suggests a semi-global iterative optimisation approach in which the next solution is obtained by minimising function $\bar{E}(\cdot)$ for some value of α. Clearly, techniques discussed in section 4.4 are applicable to function (4.16) as well. We can expect that for sufficiently small values of α the Dual Decomposition approach will produce a global minimum of $\bar{E}(\cdot)$; this is certainly true for $\alpha = 0$.

4.5 Experimental results

In this section we evaluate the performance of the new Dual Decomposition approach for optimisation of the joint model in the context of interactive image segmentation

We first give some implementation details. We use a 8-connected grid and define the pairwise potentials as a contrast sensitive term, similarly to other graph cut methods. We consider that the user input is in the form of a bounding box surrounding the object.

We tried two different histograms based on colour. The simplest histogram is obtained by dividing the RGB colour space into 16^3 bins of equal size. The other histogram is obtained by first fitting a GMM with 25 components to the full image and assigning each pixel to one of 25 bins corresponding to the components. We refer to these two cases as *regular histogram* and *GMM histogram*. The most important difference between these two histogram representations is the number of bins used. For the dataset used, the regular histograms have an average of more than 300 bins, while the GMM histograms have at most 25 bins. Fig. 4.5 shows examples of the GMM histograms used for two images.

The EM-style procedure requires initialisation of the colour models θ^0 and θ^1. We used two different approaches to initialise them, following [91] and [68]. For the first approach, θ^0 is initialised as the histogram of the pixels outside the bounding box and θ^1 is the histogram of the pixels inside the bounding box. Recently, it has been suggested by [68] that a different

	Regular Histogram		GMM histogram	
	Best energy	**Global Optimum**	**Best energy**	**Global Optimum**
Dual Decomposition	41	30	32	25
EM-style 1	9	2	21	13
EM-style 2	2	1	23	12

Table 4.2: Comparison between the new Dual Decomposition method and the two EM-style procedures. For each method, we report the number of images for which the method performed the best, in terms of energy, and obtained the global optimum. The total number of images is 49. If the best solution is obtained by two methods, it is counted for both of them.

initialisation provides better results. For the second initialisation considered, we start by computing the empirical histogram of the pixels outside the bounding box. Them, we evaluate the probability of the pixels inside the bounding box under this distribution. Finally, θ^1 is the colour histogram of the 33% pixels with smaller probability under that distribution and θ^0 is the colour histogram of the pixels outside the bounding box together with 33% pixels inside the bounding box that better fitted the background distribution. We call these two variations *EM-style 1* and *EM-style 2* respectively.

We report results for the GrabCut database [91] of 49 images with associated user defined bounding box[2]. The outside of the box is constrained to be background. We downsized each image to a maximum side-length of 250 pixels for efficiency.

Table 4.2 shows the comparison between the three different methods: our Dual Decomposition and the EM-style procedure with two different initialisations. We show results for the two different histograms described previously.

The Dual Decomposition method outperforms both EM-style approaches for the two types of histograms, achieving global optimality for more than half of the images and a better energy in many other cases. Some examples of segmentations obtained with Dual Decomposition that correspond to the global optimum for the regular histograms are shown in Fig. 4.6.

The results also suggest that the Dual Decomposition method benefits from using histograms with a higher number of bins, achieving the global optimum more often for this case, while the EM-style approaches benefit from a smaller number of bins achieving in that case the global optimum for a quarter of the images.

Unfortunately, the improvement in terms of energy minimisation obtained by using Dual Decomposition does not correspond directly to an improvement in terms of average error rate. This can be seen in Table 4.3 where we report the error rate (given by the number of misclas-

[2]We exclude the "cross" image since the bounding box covers the whole image.

Figure 4.6: Global optimum results obtained with Dual Decomposition. The first and third rows show the user input and the second and fourth rows the segmentation obtained using Dual Decomposition. For all these images the solution corresponds to the global optimum of the energy.

sified pixels over the size of the inference region) for the different methods. Table 4.3 also shows the average error rate for the best solution, in terms of energy, given by any of the three methods.

The poor performance of the Dual Decomposition method in terms of error rate is explained by a few failure cases which affect considerably the average. In our experiments, we observed that this method performs poorly, both in terms of energy and error rate, for camouflage images. Figure 4.7 shows some of these failure cases.

To remove the effect of these failure cases, we also show in Table 4.3 the error rate when restricting to the images for which the global optimum was achieved. For these images, the error rate drops significantly to 4% and Dual Decomposition has the smallest error rate of the three methods considered, which shows the advantage of achieving global optimality.

Semi-global method

Motivated by the failure cases of the Dual Decomposition method we proposed the semi-global method (section 4.4.2) that uses Dual Decomposition in an iterative procedure. To show that this method is more powerful than the EM-style procedure we take solution $\bar{\boldsymbol{x}}$ to be an EM fixed point, i.e. the EM procedure cannot further reduce the energy. We then run the semi-global

	Regular Histogram		GMM histogram	
	Full set	**30 images (GO set)**	**Full set**	**25 images (GO set)**
Dual Decomposition	10.5%	4.1%	11.1%	3.8%
EM-style 1	8.1%	4.7%	9.1%	3.8%
EM-style 2	10.4%	8.1%	9.9%	6.3%
Best solution	7.4%	-	8.0%	-
Semi global	7.2%	-	7.6%	-

Table 4.3: Error rate obtained with Dual Decomposition and EM-style methods. The error rate is computed for the full dataset and for the images where the global optimum is achieved.

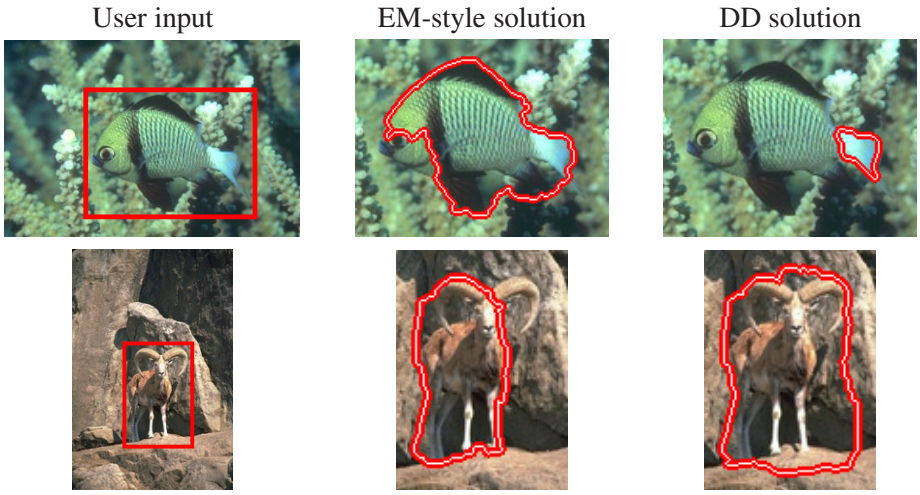


Figure 4.7: Failure cases of Dual Decomposition. The method performs poorly for camouflage images. For both images the energy obtained with the EM-style optimisation is better than with Dual Decomposition.

method and report how often it improves the energy, i.e. escapes from that local minimum. For $\alpha = 0.5$, the semi-global method improved over EM for 73% of the images.

As a final experiment, we consider the images where the global optimum was not achieved. For each image, we choose the best solution obtained with any of the three methods as an initial solution. Then we run sequentially the semi-global method for $\alpha = 0.75, 0.5, 0.25, 0$. Each run is initialised with the lowest energy result from the previous run. This semi-global procedure improved over the best segmentation for 16 out of 19 images using the regular histograms and for 13 out of 24 images using GMM histograms. The error rate from this experiment is also reported in Table 4.3 and is smaller than the error rate of the solutions used to initialise the semi-global method.

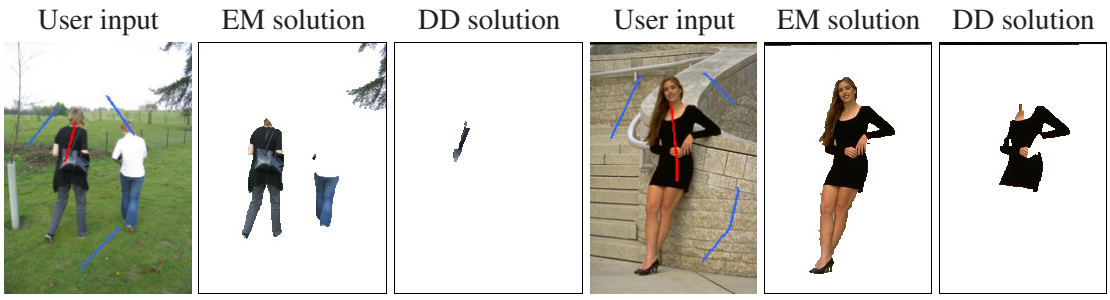


Figure 4.8: Results using a few brush strokes as input. The Dual Decomposition method fails when only few pixels are provided as region seeds. For both images shown the energy for the EM-style solution is lower than for the Dual Decomposition solution. For comparison, the results obtained for these two images using the bounding box as input are shown in Fig. 4.6.

4.6 Discussion and limitations

In the previous section we presented a quantitative comparison of the different optimisation methods for the joint model and showed that Dual Decomposition outperforms EM-style approaches for the majority of the images. Furthermore, Dual Decomposition provides a lower bound which allows to verify that the global optimum is reached for many of the instances. In this section we discuss the limitations of the Dual Decomposition method and some properties of the model.

4.6.1 Limitations of the Dual Decomposition method

The main drawback of the Dual Decomposition method is the running time. While the EM-style optimisation takes at most a few seconds to complete, the Dual Decomposition method takes in the order of minutes, in our unoptimised C++/MATLAB implementation. That makes it infeasible to use in an interactive system, in particular when compared with the EM-style methods.

Furthermore, Dual Decomposition provides unsatisfactory results for camouflage images, as can be seen in Fig. 4.7. For these failure cases, the number of breakpoints of the parametric maxflow procedure is very small affecting negatively the performance of the method.

This limitation can be overcome by using Dual Decomposition in an EM-style optimisation procedure, the semi-global method described in section 4.4.2. We showed that this semi-global method can escape from a local minimum of the EM-style optimisation, obtaining solutions with a smaller energy. However, the semi-global method suffers from the same running time drawback and it looses the main benefit of the Dual Decomposition method, the lower bound that allows to determine if a solution is the global optimum.

In the experimental section, we have shown results for the joint model and corresponding optimisation algorithms in the context of interactive segmentation. We considered that the user

Figure 4.9: Results of the joint model without user constraints. Images reproduced from [98].

interaction was in the form of a bounding box surrounding the object, i.e. only background seeds are provided. More important, a bounding box provides a considerable amount of fixed pixels, as opposed to alternative forms of region seeds, like brush strokes. Fig. 4.8 shows examples of results obtained using Dual Decomposition and EM-style optimisation when the user input consists of a few brush strokes. The Dual Decomposition method does not provide a satisfactory results when few pixels are selected by the user and it is outperformed by the EM-style optimisation.

For the unconstrained case, when no region seeds are provided, none of the methods discussed are applicable. The EM-style optimisation requires region seeds to initialise the colour models. The Dual Decomposition method, when applied to the unconstrained case, would fail since the parametric maxflow procedure would have a single breakpoint with two possible solutions: the empty solution ($x_p = 0 \; \forall p$) or the full solution ($x_p = 1 \; \forall p$).

Very recently, a new Linear Programming relaxation was proposed in [98] for minimising the joint model in the unconstrained scenario. The method builds on our formulation described in section 4.3. Although the model is still well defined in the unconstrained scenario, the assumptions that it encodes (preference for splitting the image into two regions with distinct histograms and of similar size) are not strong enough to obtain a meaningful segmentation into object and background, for some images. This effect can be seen in the results shown in Fig. 4.9. While for the first two images the joint model provides a good segmentation, the last three images suggest that it would need to be combine with stronger priors for more challenging images.

4.6.2 The importance of achieving global optimality

Despite the limitations of the Dual Decomposition previously discussed, this method has a strong advantage when compared with previously existing ones, it provides a lower bound. By comparing the lower bound with the energy of the obtained solution, we can evaluate the quality of that solution and in some cases demonstrate that it is a global optimum.

If the global optimum is achieved and the resulting solution is unsatisfactory, this is caused by a limitation of the model and not by a suboptimal optimisation algorithm. This possibility

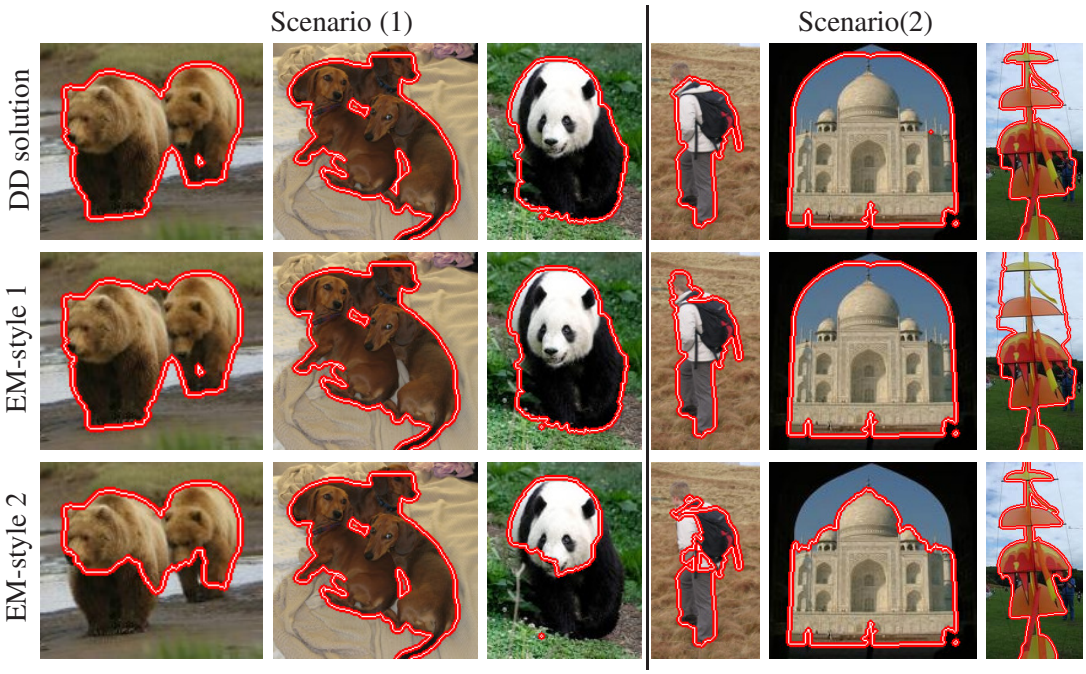


Figure 4.10: Comparison of the different optimisation methods. All the results shown for the Dual Decomposition method correspond to the global optimum of the energy function. The results illustrate two different scenarios: (1) the solution obtained with Dual Decomposition is not only the global optimum solution but also the best solution visually (first three columns); (2) despite being the global optimum, the Dual Decomposition solution is visually worse than the solution obtained with the other method (last three columns) revealing a limitation of the model. For all the results shown, the user input is a bounding box.

of attesting the optimality for some instances, gives the opportunity to evaluate and analyse the model in a way that was not possible before, revealing some of its limitations.

Fig. 4.10 shows results where the solution obtained with the Dual Decomposition is the global optimum. For the first three columns, besides being the global optimum, the solution is also visually better than the solutions provided by the other methods. The results in Fig. 4.10 also reveal common properties of the two competing algorithms. The EM-style 1 algorithm tends to label some of the background pixels as foreground, e.g. the last image of Fig. 4.10. This is due to the initialisation of the colour models. Initialising the foreground model with all the pixels inside the bounding box leads to an initial preference to assign those pixels to foreground. In many cases, this initial bias is overcome in the iterative process in the step where the models are refitted, but in other cases this initial assignment is never corrected. On the other hand, the EM-style 2 algorithm has the opposite tendency, clearly seen in the *bear* and *panda* images of Fig. 4.10. This also shows that the EM-style algorithm is dependent on the initial estimation of the colour models, since EM-style 1 and EM-style 2 only differ in the way the colour models are initialised.

The last three columns of Fig. 4.10 show that the model has some limitations and that

obtaining a non-optimal solution can partly hide those limitations.

Recall that the main properties encoded in the model are: contrast sensitive smoothness, preference towards a balanced segmentation, and a preference that the histograms of both segments are distinct. Not surprisingly, these properties are not always enough to achieve a correct result and they are not always satisfied. For example, in the fourth image of Fig. 4.10, the foreground and background histograms overlap, since the head of the person is very similar in terms of colour with the background. For the Taj Mahal image, the segmentation aligns with stronger edges and in the last image, some of the background clutter is incorrectly labelled foreground.

Some of these limitations can be overcome by including extra information in the model. For example, for segmenting the fourth image, knowing that the object of interest is a person would possibly give enough information to extract the correct segmentation. In other cases, only the use of extra user input would help achieving better results.

4.7 Conclusion

Considering the appearance models as variables is a common theme of different approaches to segmentation. Although, some of these models have been successfully used in different scenarios, the optimisation usually resorts to a less than optimal coordinate descent technique.

In this chapter we have discussed a new Dual Decomposition method that can be used for specific formulations of the joint model. We have showed that it improves over EM-style techniques and that despite its limitations has the major benefit of providing a lower bound and computing the global optimum for many of the instances considered.

We have also shown how to speed up Dual Decomposition involving convex terms of the area, by using parametric maxflow (Theorem 5).

We believe that rewriting the energy purely in terms of the segmentation using higher-order cliques brings a new perspective to the problem and it may motivate new algorithms for this type of energy function.

Chapter 5

Cosegmentation

5.1 Introduction

In the previous chapters we discussed higher-order models that are useful for segmenting a single image, particularly in an interactive scenario. User interaction is available in a variety of applications, like image editing or medical imaging analysis. However, it is not always possible or desirable to have a user in the loop.

In this chapter we will focus on a different type of higher-order model aimed at segmenting several (2 or more) images jointly. We will refer to the task of jointly segmenting multiple images as *cosegmentation* [93].

The task of cosegmentation allows for a wide range of applications, for example to efficiently select all occurrences of an object in multiple images to edit its appearance, e.g. by changing its contrast [5] or as a pre-processing step in 3D reconstruction [21, 61].

Cosegmentation was first introduced in [93] and the main assumption of the task is that the input images have "something in common" and that this common part is the region of interest that should be labelled foreground. This assumption is, however, still too generic and a more precise definition of *common part* is needed. Take the example in Fig. 5.1. If cosegmentation is defined as the task of finding the segments with *similar appearance*, then the foreground would include the green and blue parts in the images, corresponding to grass and sea. On the other hand, if the task of cosegmentation is defined as finding *objects of the same class*, the foreground would correspond to the cows in both images[1]. It is clear from this example that a more precise definition of cosegmentation is needed.

This chapter is organised as follows. We start by reviewing previous work in section 5.2. The methods reviewed differ in the definition of cosegmentation they use. The remaining of the chapter is divided into two parts:

[1]We use the common distinction between objects ("things") and materials ("stuff"). Although the classes grass and sea are represented in both images, they do not correspond to objects.


Figure 5.1: Ambiguity of Cosegmentation. Cosegmentation is broadly defined as the task of segmenting the common part in two images. Depending on the precise definition of *common part* the result for this pair of images can either be the two cows (that are instances of the same class) or the grass and sea (that have the same appearance).

Energy minimisation methods for cosegmentation In section 5.3 we review in detail existing methods based on energy minimisation. We discuss different optimisation strategies for these models and show that a new Dual Decomposition approach outperforms existing ones. Finally, we discuss limitations of the energy minimisation models that motivated a new approach to cosegmentation.

Object cosegmentation We discuss our novel formulation of the cosegmentation task in section 5.4. In particular, we introduce some intuitive assumptions that should be included in the formulation of the cosegmentation task. Although, some of these properties have been previously used for other related tasks, they cannot be easily included in the elegant energy minimisation framework that has been used so far in this thesis. We discuss experimental results in section 5.5. Finally we discuss limitations and possible applications in section 5.6 and present conclusions in section 5.7.

5.2 Related work

Different definitions of *common part* have been used in the past to better constrain and define the cosegmentation task. Furthermore, some of these definitions lead to related tasks and approaches which are typically not referred to as cosegmentation. We will now review some of these approaches.

5.2.1 Histogram based cosegmentation

Previous work on cosegmentation [93, 45, 78], in particular the original work that introduced the task of cosegmentation [93], used the following definition of the task: the goal is to find foreground segments with similar appearance histograms, where colour is typically used as appearance feature. In the previous example illustrated in Fig. 5.1, this would correspond to segment the grass and the sea.

This definition was used for pairs of images and the task of cosegmentation was formulated

as an energy minimisation task with a *global term* that measures the similarity between the foreground histograms of both images. A formulation of this form allows arbitrarily shaped regions, only making the assumption that the size of the segments is similar. We will review these methods in more detail in section 5.3.

A related approach worth mentioning was presented in [50]. They cast the cosegmentation task into a clustering problem with two clusters, where the goal is to group superpixels with similar appearance. The intuition is that all the superpixels belonging to the foreground segments would belong to the same cluster.

5.2.2 Interactive cosegmentation

Since some of the applications of cosegmentation are encountered in an interactive scenario (e.g. the selection of an object in multiple images for appearance editing) several recent papers have considered a simplified cosegmentation task where user interaction is available.

The focus of these approaches is to minimise the user interaction needed to correctly segment all the images. In [30] the authors segment several images of the same object, assuming one of those images is hand-segmented. They model local appearance and edge profiles from the segmented image in order to "transduct" such segmentation into the remaining images. In [6, 5] the user input is in the form of foreground/background scribbles in one or multiple images from the collection. The goal is to guide the user interaction by choosing the image or image regions where additional scribbles should be provided.

For the images in Fig. 5.1, using an interactive approach of this form the user could choose, for example to segment only the grass in both images, by providing region seeds in only one of them.

5.2.3 Unsupervised class segmentation

The goal of unsupervised class segmentation is to segment objects of the same class in a collection of images. There is no information about the class of the objects, only that all the images contain an object of the same class.

Different generative models have been proposed for this task. Examples are the LOCUS model [113] and the use of topic models over image segments [22]. The LOCUS model learns a shape model for the class, while also accounting for differences in appearance for each instance. Topic models [22] assign segments to topics depending on their visual words, i.e. interest point descriptors like SIFT [74]. Both [22, 113] model separately what is common in all images (the shape of the object, sift features) and what is specific to each single image (the appearance of a specific object instance).

Although [113] reports significantly better performance than [22], it uses shape features, assuming a rough alignment of all the objects, given a reference frame. For example, [113] reports results for a subset of the Weizmann horse database [12]. This database contains horses in very similar poses. In [22] there is no modelling of shape information which makes it a more generic model, applicable to other scenarios.

Recently, [2] proposed a method for unsupervised class segmentation inspired by interactive segmentation. Similarly to LOCUS, the method alternates between learning a class model and updating the segmentations and it suffers from the same drawback of requiring rough alignment and similar pose.

This type of methods is typically not suitable for segmenting only two images, like the example in Fig. 5.1, since two images do not provide enough information to build a model for the object class. Nevertheless, these methods are formulated in order to segment objects from the same class and we would expect that the resulting segmentation for the images in Fig. 5.1 would be the cows.

5.2.4 3D reconstruction approaches

Given several images of the same object, the goal of 3D reconstruction is to build a 3D model of the object. Some approaches to solve this reconstruction problem require that the object is segmented in each individual image, e.g. [21]. This formulation of cosegmentation relies on several assumptions that are specific to this task: all images contain exactly the same object instance, the object is rigid and seen from different viewpoints, since the camera is moving. These assumptions translate into additional constraints that are included in the model: a fixation constraint, which requires the object to be more or less in the centre of the image, and a silhouette coherency constraint, which requires that the segmentation contours form a plausible visual hull of a 3D shape.

Methods of this form would not be applicable to the example in Fig. 5.1, since it is not the same object instance depicted in both images.

5.3 Energy minimisation methods for cosegmentation

A few recent methods [93, 45, 78] have used energy minimisation formulations for cosegmentation. They follow in the histogram based category discussed in section 5.2.1. In general, these approaches focus more on finding efficient optimisation methods for minimising the energy function than on a full scale evaluation of the models proposed.

In this section we start by reviewing the exact form of the energy functions used in those approaches. We focus on the case where the number of input images is two. We show that

a Dual Decomposition method, similar to the one used in previous chapters, can improve and complement existing optimisation methods for these energies. Finally, we give an experimental comparison and discuss the limitations of these models.

5.3.1 Models

We start by introducing some notation:

- $x_p \in \{0, 1\}$ is the label for pixel p, where $p \in \mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ and $\mathcal{V}_1$, $\mathcal{V}_2$ are respectively the set of pixels in image 1 and image 2. We use letter $k \in \{1, 2\}$ to denote the image number.

- y_p is the appearance of pixel p (e.g. colour or texture) and such measurement is quantised into a finite number of bins. Variable b ranges over histogram bins ($b \in \{1, ..., B\}$ where B is the total number of bins), and $\mathcal{V}_{kb}$ denotes the set of pixels p in image k whose measurement y_p falls in bin b.

- h_k is the empirical un-normalised histogram of foreground pixels for image k: it is a vector of size B with components $h_{kb} = \sum_{p \in \mathcal{V}_{kb}} x_p$.

The energy based models previously proposed fit into a single framework, where the energy used has the following form:

$$E(\boldsymbol{x}) = \sum_p w_p x_p + \sum_{(p,q)} w_{pq} |x_p - x_q| + \lambda E^{global}(h_1, h_2) \tag{5.1}$$

Jointly, the first two terms form the traditional MRF term for both images, where w_p is a unary term for each pixel and w_{pq} is the contrast sensitive pairwise term. We will refer to these two terms as $E^{MRF}(\boldsymbol{x})$ The last term, E^{global}, is a higher-order term that encodes a similarity measure between the foreground histograms of both images and λ is its weight.

Following [93], the unary term is a ballooning term, constant for every pixel: $w_p = \mu$. This biases the solution to one of the possible labels and it is important to prevent trivial solutions (i.e. both images being labelled totally background or foreground). If the bias is not present (i.e. if $w_p = 0$ and the energy does not have unary terms) such trivial solutions are always a global optimum of the energy.

The models differ in the way the term E^{global} in equation (5.1) is defined.

Model A: L1-norm

This model was first introduced in [93] and it was derived from a generative model. The global term in the energy was defined as follows:

$$E^{global} = \sum_b |h_{1b} - h_{2b}| \tag{5.2}$$

where the L1-norm is used as a similarity measure between foreground histograms.

Model B: L2-norm

This formulation was introduced in [78] and it was defined as follows:

$$E^{global} = \sum_b (h_{1b} - h_{2b})^2 \tag{5.3}$$

It is similar to the previous formulation in equation (5.2), with the difference that the norm used to measure histogram similarity is the L2-norm instead of the L1-norm. The authors motivate this change by arguing that such a model has some interesting properties and allows the use of alternative optimisation methods.

Model C: Reward model

In [45] the authors used the following global term:

$$E^{global} = -\sum_b h_{1b} \cdot h_{2b} \tag{5.4}$$

The motivation behind this global term is to replace the penalisation term with a rewarding term.

Model D: Boykov-Jolly model

We also consider a fourth model, which we call Model D, based on a straightforward extension of the Boykov-Jolly (or graph cut) model for binary image segmentation [14, 93, 91]. The main difference is that instead of having two appearance models for each region (foreground and background), for the cosegmentation task we have *three* appearance models: two separate background models and one common foreground model. This encodes an assumption similar to the other methods, that foreground histograms should be similar. We use an EM-style optimisation for this model and initialise it with the histogram intersection method previously used to initialise TRGC. Models of this type have been previously used for both automatic [21] and interactive [5] cosegmentation.

Both Model A and Model B lead to NP-hard optimisation problems [93], while the Model

C leads to a submodular problem that can be efficiently optimised with graph cuts [45].

5.3.2 Optimisation methods

We now discuss how Dual Decomposition can be used to optimise Models A and B. We start by reviewing the existing optimisation methods for those models.

Trust region graph cut (TRGC)

This method was proposed in [93] for Model A and it can be viewed as a discrete analogue of trust region methods for continuous optimisation. TRGC can be applied to energy functions of the form $E(\boldsymbol{x}) = E_1(\boldsymbol{x}) + E_2(\boldsymbol{x})$ where $E_1(\boldsymbol{x})$ is submodular and $E_2(\boldsymbol{x})$ is arbitrary. It works by iteratively replacing $E_2(\boldsymbol{x})$ with a linear approximation and it produces a sequence of solutions with the guarantee that in each iteration the energy does not go up.

In [93] the authors used TRGC inside an iterative scheme for cosegmentation that alternated between updating the segmentation for each image individually while the foreground histogram of the other image was fixed. This method requires a segmentation for initialisation. In our experiments we observed that its performance is very dependent on that initialisation. We use the implementation of this method from [93]. We also adapted it to Model B, i.e. replaced L1 norm with L2 norm.

Quadratic pseudo boolean optimisation

In [78] the authors observed that Model B is represented by a quadratic pseudo-boolean function. Indeed, histograms h_1 and h_2 depend linearly on $\boldsymbol{x}$: $h_{kb} = \sum_{p \in \mathcal{V}_{kb}} x_p$. Therefore, expanding expression $(h_{1b} - h_{2b})^2$ yields a sum of linear terms and quadratic terms of the form $w_{pq} x_p x_q$, some of which are non-submodular. In [78] a linear programming relaxation of the problem is formulated, which is equivalent to the roof duality relaxation [43, 13] for the quadratic function $E(\boldsymbol{x})$. This relaxation can be solved via the QPBO method discussed in section 2.5, and it yields a partial solution: the nodes are divided into labelled and unlabelled, with the guarantee that the labels of the labelled nodes are optimal.

An important question is how to set the segmentation for unlabelled nodes. In [78] the segmentation obtained by minimising energy $E^{MRF}(\boldsymbol{x})$ was used. In our experiments we use a constant ballooning force ($w_p = \mu$), so this procedure assigns the same label to all unlabelled nodes.

Note that, Model C is also represented by a quadratic function, but unlike the previous case this quadratic function is submodular. Therefore, Model C can be optimised exactly by a single call to a maxflow algorithm [45].

Dual Decomposition

Let us start by writing the energy for Models A and B in an equivalent form:

$$\min_{\boldsymbol{x},\boldsymbol{z}} E^{MRF}(\boldsymbol{x}) + \sum_b g(z_b) \tag{5.5a}$$

$$\text{s.t. } z_b = \sum_{p\in\mathcal{V}_{1b}} x_p - \sum_{p\in\mathcal{V}_{2b}} x_p \equiv \sum_{p\in\mathcal{V}} a_{bp}x_p \qquad b = 1, ..., B \tag{5.5b}$$

where g is a convex function: $g(z) = \lambda|z|$ for Model A and $g(z) = \lambda z^2$ for Model B. Coefficients a_{bp} are defined as follows: $a_{bp} = 1$ if $p \in \mathcal{V}_{1b}$; $a_{bp} = -1$ if $p \in \mathcal{V}_{2b}$ and $a_{bp} = 0$ otherwise.

We form a standard Lagrangian function by relaxing constraints (5.5b) and introducing a Lagrangian multiplier $\boldsymbol{\theta}$:

$$L(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\theta}) = E^{MRF}(\boldsymbol{x}) + \sum_b g(z_b) + \sum_b \theta_b \left(z_b - \sum_p a_{bp}x_p \right) \tag{5.6}$$

Minimising the Lagrangian over $(\boldsymbol{x}, \boldsymbol{z})$ gives a lower bound on the original problem:

$$\Phi(\boldsymbol{\theta}) = \min_{\boldsymbol{x},\boldsymbol{z}} L(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\theta}) \tag{5.7a}$$

$$= \min_{\boldsymbol{x}} \left[E^{MRF}(\boldsymbol{x}) - \sum_{p,b} a_{bp}\theta_b x_p \right] + \sum_b \min_{z_b} \left[g(z_b) + \theta_b z_b \right] \tag{5.7b}$$

$$\Phi(\boldsymbol{\theta}) \leq E(\boldsymbol{x}) \tag{5.7c}$$

In order to obtain the tightest bound we use a subgradient method to maximise $\Phi(\boldsymbol{\theta})$. To compute a subgradient for a given vector $\boldsymbol{\theta}$, we need to solve $1 + B$ minimisation subproblems in (5.7b). The first subproblem requires minimising a submodular energy with pairwise terms, which can be efficiently done using graph cuts. Solving subproblems for bins b is straightforward.

It remains to specify how to choose a primal solution $\boldsymbol{x}$. Let $\boldsymbol{x}^t$ be a minimiser of the first subproblem in (5.7b) at step t of the subgradient method. Among labellings $\boldsymbol{x}^t$, we choose the solution with the minimum cost $E(\boldsymbol{x}^t)$.

5.3.3 Experimental comparison of the optimisation methods

The goal of this section is to compare optimisation algorithms for energy based methods. We start by giving details about the experimental setup.

We use a simplified dataset of 20 pairs of images. These images are composites of 20

Figure 5.2: Dataset for cosegmentation. Examples of images used for comparison of the optimisation methods. These images are composites using the same foreground.

foreground objects from the database in [89] onto 40 different backgrounds. The dataset also contains ground truth. Representative images out of these 20 pairs are shown in Fig. 5.2.

We resized the images so that their maximum side is 150 pixels. Some of the models and optimisation methods discussed are limited to small images, in particular, QPBO method requires the construction of a graph that grows quadratically with the size of the image.

We use histograms over RGB colours, using 16 bins for each colour channel. Note that, in previous papers where some of the models were introduced, other appearance features were used [93, 45]. Since our dataset is constructed such that the foreground histograms over colour are very similar, extending the features used should not improve the results.

In this comparison of optimisation methods, we fix the weights for the different parts of the model in an ad-hoc way. For Model A, we choose $\lambda = 5$ and $\mu = -2$, and for Model B, $\lambda = 2$ and $\mu = -10$. The error rate obtained for these weights shows that they are a sensible choice.

We start by comparing Dual Decomposition with TRGC for Models A and B. Since TRGC is an iterative method that requires as input an initial segmentation, we test this method with three different starting points. First, we use the solution of Dual Decomposition as a starting point. The second starting point is a random segmentation whose foreground histogram is constructed by having each bin taking the minimum value over the corresponding bins in the full histogram of both images, i.e., $h_b = \min(|\mathcal{V}_{1b}|, |\mathcal{V}_{2b}|)$. Third, we initialise TRGC with the ground truth (GT). GT is not available at test time, and we report results only for comparison.

The results for Model A are shown in the first part of Table 5.1. Note that in [93], where TRGC was proposed, the Dual Decomposition solution was not used as a starting point. For this model, the difference between TRGC-DD and Dual Decomposition is very small, since TRGC starting with the Dual Decomposition solution only improves the energy for two images.

The results for Model B are shown in the second part of Table 5.1. Although QPBO also provides a lower bound, we use the lower bound obtained by Dual Decomposition since in our

		TRGC			DD	QPBO
		From DD	From hist	From GT		
Model A	Best energy: # cases	20	0	0	18	-
	Distance from LB	100.24	106.5	101.15	100.24	-
	Error rate	3.7%	8.1%	3.2%	3.7%	-
Model B	Best energy: # cases	13	0	7	3	0
	Distance from LB	101.59	107.56	101.77	104.20	197.29
	Error rate	3.93%	5.96%	2.85%	3.92%	51.77%

Table 5.1: Comparison of optimisation methods for Models A and B. We compare TRGC (using 3 different initial solutions), Dual Decomposition, and QPBO (only for Model B). For each model, the first row shows for how many pairs of images each method gives the best energy (out of 20 pairs). The second row is the gap between the energy and the lower bound (LB) obtained by Dual Decomposition. The values are normalised: first we add a constant to each term of the energy so that the minimum of each term becomes 0, and then we scale the energy so that the lower bound corresponds to 100. The last row is the error rate: percentage of misclassified pixels over the total number of pixels.

experiments, it was always better than the one provided by QPBO.

We conclude that a combination of Dual Decomposition and TRGC, using the Dual Decomposition solution as a starting point for TRGC, is the best performing method for both Models A and B.

Surprisingly, the performance obtained for the QPBO method contrasts with the one reported in [78], since for this experiment the number of pixels left unlabelled by this method was 90%. Note that in [78], the authors used a different spatially varying unary term which may induce differences. They also report that the performance of the method deteriorates when weight λ is increased. For the scenario considered, where w_p is constant, small values of λ lead to trivial solutions.

In order to better understand why QPBO fails, we ran the method with a fixed ballooning force, $\mu = -10$, and different values of λ. In Table 5.2, we show the percentage of pixels that were labelled one, zero, or left unlabelled. For intermediate values of λ, the number of unlabelled pixels is more than 90%. For such values, QPBO is not reliable as an optimisation method. On the other hand, for extreme values of λ, QPBO labels more pixels, but the resulting model is not meaningful. For example, for the case $\lambda = 10^{-3}$, all pixels for all images considered were labelled 1.

λ	10^{-3}	10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}	10^{3}
Labelled 1	100	64.49	9.52	0.18	0.03	0.03	0.03
Labelled 0	0	0	0	0	22.68	25.66	24.22
Unlabelled	0	35.51	90.48	99.82	77.30	74.31	75.75

Table 5.2: Optimising Model B using QPBO. Percentage of pixels labelled 1, 0 or left unlabelled by the QPBO method for different values of weight λ.

5.3.4 Experimental comparison of the models

As previously stated, the main focus of this section was to show that the Dual Decomposition method outperforms existing optimisation methods for previously used energy functions. For the experimental comparison of optimisation methods we use the same dataset of simplistic images. Although we are aware of the limitations of this dataset, the intuition for using it is that if the models fail in this scenario, they will also fail in a realistic scenario where the foreground histograms may differ.

In order to simulate more realistic scenarios, we present results not only for the original dataset, but also for modified versions of that dataset. We explore three different cases: the original dataset; altering the dataset by reducing one of the images in each pair to 90% and 80% of the original size; and altering the dataset by adding a constant (3 and 6 in the experiments) to all RGB values (ranging from 0 to 255) to one of the images in each pair, simulating differences in illumination.

Table 5.3 shows the results for this experiment. We report error rates for the different scenarios and the different models. Furthermore, in the last column we also report the average foreground histogram similarity for the different cases. This similarity is given by: $100 - 100 \times \frac{\sum_b |h_{1b}^{GT} - h_{2b}^{GT}|}{\sum_b h_{1b}^{GT} + h_{2b}^{GT}}$ where h_k^{GT} is the histogram of image k computed over foreground ground truth pixels. This similarity can be seen as a rough measure of the difficulty of the problem, and the higher it is, the simpler the problem. These results were obtained using leave-one-out cross validation of the free parameters λ and μ.

From the results presented in Table 5.3 we take the following statistically significant observations:

- Models A, B, and D perform similarly for the simplest case.
- Model C is the worst performing model since it produces in every case considerably higher error rates [2].

[2] A closer inspection of the properties of Model C reveals that this poor performance should not be surprising. Assume for simplicity that there are no pairwise terms. The energy for Model C can then be written as $E(\boldsymbol{x}) =$

	Model A	Model B	Model C	Model D	Histogram similarity
Original images	4.6% ±0.8	**3.9%** ±0.7	22.0% ±3.9	4.3% ±0.3	93.4
Resized to 90%	**4.7%** ±0.4	5.7% ±0.8	16.3% ±2.4	4.9% ±0.5	84.6
Resized to 80%	7.8% ±1.3	9.7% ±1.4	17.4% ±3.0	**5.1%** ±1.0	74.2
RGB +3	4.4% ±0.4	7.1% ±1.1	21.4% ±4.3	**3.7%** ±0.3	84.6
RGB +6	5.5% ±0.5	12.3% ±1.7	20.3% ±2.5	**4.0%** ±0.4	76.3

Table 5.3: Comparison of models based on histogram similarity. We report the error rate and the standard error of estimating the mean of the error rate for the different models and scenarios described in the text.

- Model D is the most robust to changes in size and illumination.
- Comparing both models based on histogram distances, the L1-norm (Model A) is more robust than the L2-norm (Model B), for the cases where there are small variations of foreground.

Since Model D performs better than the competitors for this dataset and it has the extra advantage of allowing the use of an effective and fast EM-style optimisation, we will use it as representative of histogram based methods in subsequent comparisons of models for cosegmentation.

5.3.5 Limitations of energy based approaches

In this section we discuss the limitations of previously proposed energy based models for more realistic images. In fact, despite the considerable attention these models have had recently [93, 45, 78], they suffer from many drawbacks.

The main assumption of these methods is that the foreground (colour) histograms of both images match. Furthermore, Models A and B have a ballooning force which prefers that as many pixels as possible are assigned to foreground. This leads to the less obvious assumption that the background histograms have to be distinct. Otherwise, if there are parts of the background with the same colour, those parts will be assigned to foreground.

Unfortunately these assumptions do not hold for many realistic scenarios. Take the examples in Fig. 5.3 where we show two pairs of images depicting the same object. These images are

$\sum_b E_b(h_{1b}, h_{2b})$ where

$$E_b(h_{1b}, h_{2b}) = \mu(h_{1b} + h_{2b}) - \lambda h_{1b} \cdot h_{2b}$$

We must have $\mu > 0$, otherwise all pixels would be assigned to the foreground. Minimising E_b over $[0, n_{1b}] \times [0, n_{2b}]$ where $n_{1b} = |\mathcal{V}_{1b}|$, $n_{2b} = |\mathcal{V}_{2b}|$ gives the following rule: if $n_{1b} \cdot n_{2b}/(n_{1b} + n_{2b}) \leq \mu/\lambda$ then assign pixels in $\mathcal{V}_{1b} \cup \mathcal{V}_{2b}$ to the background, otherwise assign these pixels to the foreground. This reliance on the harmonic mean of n_{1b} and n_{2b} can lead to unexpected results.

Figure 5.3: Cosegmentation results for energy based methods. Although both images contain the same object, the results obtained with energy based methods are far from satisfactory.

considerably more challenging than the dataset used so far. The results for both Model A and Model B are quite poor[3]. For the first pair of images, not only do the foreground histograms differ, due to changes in illumination, but also the background histograms overlap considerably, which leads to many background pixels being incorrectly labelled as foreground.

In Fig. 5.4 we show the foreground histograms for these images. These are obtained by fitting a GMM with 10 components to both images[4] and using ground truth segmentations to obtain a colour histogram corresponding to the foreground pixels. In both cases, but particularly for the first pair of images, it is clear that the foreground histograms are quite distinct and that a method that enforces similarity between these histograms would produce erroneous results.

Both pairs of images shown in Fig. 5.3 have been previously used as test cases for histogram based models [78, 45] and the results we obtain are visually worse than the original results included in [78, 45]. This difference is justified by the inclusion of user provided seeds in both previous works. Although the problem of interactive cosegmentation has interesting applications, discussed in section 5.2.2, it is not the focus of this chapter. Furthermore, methods designed specifically for the interactive scenario, e.g. [30, 5], are more adequate and produce better results than energy minimisation methods that were first introduced for automatic cosegmentation.

The results in Fig. 5.3 further suggest that priors stronger than the pairwise smoothness are needed for the individual images, since these segmentations are considerably fragmented. Note that, increasing the weight for the pairwise term would partially overcome this fragmentation, however, the weights for the different parts of the model were chosen by cross validation. Furthermore, we observed that increasing the weight for the pairwise term leads in many cases to

[3]We use histograms over colour with 16^3 bins, similar to the ones used in the experiments section

[4]Note that the results in Fig. 5.3 were not obtained using these histograms, however, these histograms are more appropriate for visualisation since they have a smaller number of bins.

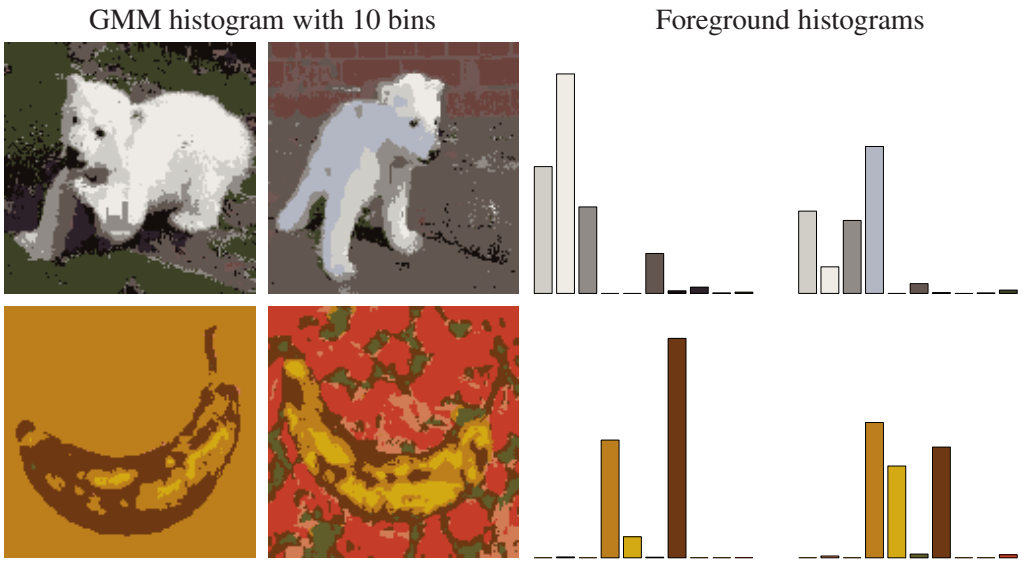


Figure 5.4: Comparison of foreground histograms. We show histogram assignment and the *foreground* histogram for the images in Fig. 5.3. The difference between both foreground histograms is considerable, especially for the first image.

trivial segmentations, where all pixels in an image are assigned the same label.

To summarise this review of energy minimisation methods, we have shown that Dual Decomposition outperforms previously used methods for optimisation of energy minimisation formulations of the cosegmentation task. However, these models cannot cope with the variation observed in real images. This observation motivates our new method described in the following section.

5.4 Object Cosegmentation

In this section we develop a new cosegmentation approach, motivated by the following observation: in most applications of cosegmentation the regions of interest are **objects**, i.e. "things" (such as a bird or a car) as opposed to "stuff" (such as grass or sky). Although this assumption was implicit in most of the work reviewed in the section 5.2 it was not directly incorporated in any of the models.

This observation is quite relevant, since a segment that corresponds to an object has very different properties from an indiscriminate segment, particularly in terms of shape and extent. Some of these properties have been used before for segmentation [87, 66, 107], saliency detection [73] and object detection [3] and are summarised in Table 5.4.

Ideally, we would encode the preference for object-like segmentations in an energy minimisation framework by defining an energy function that favours segmentations that follow this criteria. Properties like contour alignment with image edges are local properties of the segmen-

Object properties	Description
Contour aligned with images edges	The boundary of the segmentation should align with strong image edges.
Shape	The foreground region should have a limited spatial extent, be connected and close to convex.
Intra region similarity	The appearance of the object should be close to homogenous, presenting only few variations in colour and texture.
Inter region similarity	The colour and texture of the object should be distinct from the appearance of the surroundings.

Table 5.4: Summary of object properties. A segment that fully encloses a single object is expected to satisfy certain properties that make it distinct from an arbitrary segment.

tation that can be encoded using pairwise potentials, as it is done in graph cut based methods [14]. The remaining properties are global and depend on the label assigned to all pixels in the image. Some of these properties relate to the models discussed in the previous chapters. Connectivity was discussed in chapter 3 and the intra and inter region similarities are part of the joint model discussed in chapter 4.

An energy formulation for cosegmentation of two images would have the form:

$$E^{MRF}(\boldsymbol{x}_1) + E^{MRF}(\boldsymbol{x}_2) + distance(\boldsymbol{x}_1, \boldsymbol{x}_2) \tag{5.8}$$

where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are the segmentation for the first and second image respectively and $distance(.)$ is a function comparing the foreground of both segmentations. In the previous section, $E^{MRF}(.)$ was a pairwise energy function encoding a contrast sensitive smoothness term and $distance(.)$ a function measuring the difference between foreground histograms. As previously discussed, this formulation revealed to be inadequate for real images.

To include object properties in this formulation, we would extend the definition of $E^{MRF}(\boldsymbol{x})$, for example to a function of the form:

$$E^{MRF}(\boldsymbol{x}) = P(\boldsymbol{x}) + C(\boldsymbol{x}) + Conv(\boldsymbol{x}) + Extent(\boldsymbol{x}) + Joint(\boldsymbol{x}) + ... \tag{5.9}$$

where $P(\boldsymbol{x})$ is the contrastive sensitive smoothness term, $C(\boldsymbol{x})$ corresponds to the connectivity constraint **C0** and takes values in $\{0, \infty\}$, $Conv(\boldsymbol{x})$ penalises non-convex segmentations, $Extent(\boldsymbol{x})$ penalises foreground regions that occupy a large part of the image and the last term ($Joint(\boldsymbol{x})$) prefers that the appearance of the foreground and the background are distinct and

could be similarly defined as the joint model in the previous chapter[5].

This energy function encloses the higher-order models that we have previously considered. However, the two terms $C(\boldsymbol{x})$ and $Joint(\boldsymbol{x})$ were considered individually in the previous chapters and we showed that they both lead to NP-hard optimisation problems. We can then envision the difficulty in fully formulating and optimising an energy function of the form of equation (5.9) that encodes *all* the properties in table 5.4.

Furthermore, we have seen in the previous section the limitations of defining the term $distance(\boldsymbol{x}_1, \boldsymbol{x}_2)$ in equation (5.8) as a direct distance between the unnormalised foreground histograms. A distance of this form is not robust to variations between the input images, for example in terms of illumination, object pose or object size. We would like to define this term in a more robust way, accounting for these variations. We would also like to include several appearance properties (such as colour and texture) and to include a measure of similarity in terms of shape.

This motivates the use of an alternative approach that takes into account all these requirements without formulating the problem as an energy minimisation. We build on the work in [23], which we now review.

Object proposal methods

Given a single image, the goal of *object proposal* methods is to generate a set of binary segmentations that are plausible segmentations of objects in the image. This is very different from the more common task of *multiple region segmentation* where the goal is to divide the image into coherent regions.

An example of an object proposal approach is [23]. The method starts by extracting multiple segmentations using parametric maxflow. Parametric maxflow efficiently solves a sequence of pairwise energy minimisation problems, where the unary terms of the energy depend linearly on a parameter λ. Variations in the parameter λ influence the area of the segmentation obtained. The parametric maxflow procedure is run multiple times with different seed nodes to cover the full extent of the image. After the extraction of proposals, the method reduces the number of proposals by discarding the ones that differ by only a very small number of pixels.

The method proceeds by extracting features for the proposals retained. Examples of the features used are: area, perimeter, inter-region colour similarity, and convexity (area of foreground region over the area of its convex hull). This feature vector is then used to score the quality of the proposal. In a training stage, the method learns the scoring function based on

[5]The energy function in (5.9) is only a schematic representation of a possible energy function that encodes object properties. There are different ways of formulating each of the terms and extra terms could be added. A meaningful definition would also require a careful choice of the weights for the different parts of the model

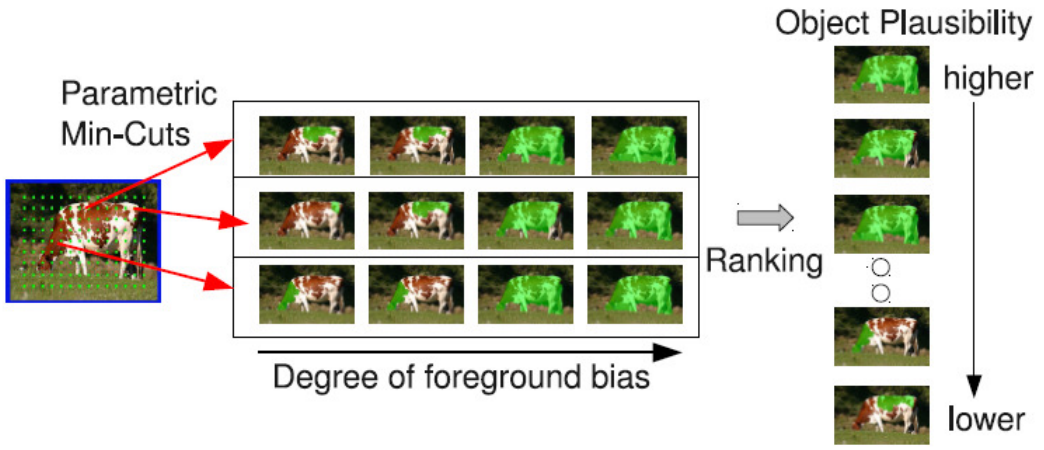


Figure 5.5: Illustration of the object proposal method of [23]. Multiple segmentations of a single image are obtained by using parametric maxflow with different foreground seeds and different foreground biases. Each row in the middle figure corresponds to segmentations obtained with the same foreground seed and with increasing foreground bias. The resulting binary segmentations are then filtered and ranked according to how object-like they are. The score is based on several features: graph partition (e.g. value of the cut), region (e.g. area and perimeter) and gestalt (e.g. convexity). Figure reproduced from [23].

ground truth segmentations.

This method can be seen as a two-step heuristic approximation to the problem of minimising the energy in equation (5.9). Using the parametric maxflow procedure in the first step ensures that the proposals are smooth and the contour aligns with image edges. The other properties are enforced in the second step, by selecting the best scoring proposals. The main drawback of a heuristic procedure of this form is that there is no guarantee that the energy minimised in the first step captures the properties desired in the second step. In the second step, the method cannot recover if the proposals from the first step were poor. Similar procedures have been proposed for the task of interactive segmentation [29], where the main goal was to improve computational efficiency for large images.

Although object proposal methods have been successfully used as a building block for object segmentation and recognition systems (in particular [23] was part of the system that won the segmentation competition of the VOC Pascal Challenge 2009), it is not clear how to use them as a standalone method for image segmentation. Indeed, their accuracy relies on the use of multiple binary segmentations for each image, which leaves the question of choosing the optimal one.

Learning the similarity between proposals

In general, the preference for object-like segmentations alone is not sufficient to segment a single image if, for example, an image contains multiple objects. Cosegmenting multiple images can help disambiguating what is the object of interest.

Cosegmentation can be loosely defined as the task of segmenting "similar looking objects" and a crucial part of any cosegmentation system is the definition of this *similarity* (or distance) measure. We have seen that previously used similarity measures are not robust to variations which occur in natural images. To obtain a more robust similarity measure, we rely on machine learning techniques to learn the similarity measure between segmentation proposals.

Another important observation is that "similar looking objects" has been used to refer to very distinct scenarios, with different degrees of variability of object appearances. Our goal is to define an approach that can adapt to these different scenarios. For each scenario, the system is trained on an adequate dataset, with the appropriate variability of appearance. We will consider two scenarios, differing in what "similar looking objects" refers to: objects of the same class or the same physical object. The "same physical object" scenario can still be very challenging if, for example, the images capture different physical parts of the object (viewpoint or zoom change) or the object is deformable. Examples of such variations are the Stonehenge, Statue and Alaskan bear classes in Fig. 5.8.

In practice, our approach creates a system for each of the cosegmentation scenarios. Assume we are given a group of input images containing objects of the same class, for example horses. We do not have information that the images depict horses, only that they have objects of the same class. For this scenario, we would use a system that has been previously trained on pairs of images depicting objects of the same class.

To summarise, our main contribution is to add two new aspects to the task of cosegmentation: the region of interest has to be an object, and "similar looking objects" are defined by learning a similarity measure.

5.4.1 Problem formulation

We now give further details of our formulation of the cosegmentation task. Assume that we are given K images containing the same object[6] and the goal is to segment the common object. For each image $I_k, k = 1, ..., K$, we retrieve 200 proposal segmentations using the implementation of [23] and retain the 50 highest scoring ones. We denote by $\mathcal{S}_k = \{S_k^1, ..., S_k^{50}\}$ the set of proposal segmentations for image I_k. A proposal segmentation S_k^i is a binary labelling of the

[6]We use "same object" to refer both to objects of the same class or the same physical object under different viewing or lighting conditions.

Figure 5.6: Top scoring proposals obtained with [23]. Each proposal corresponds to a binary segmentation of the image. The result of our method for this image is shown in Fig. 5.11.

image I_k, which assigns to each pixel one of two possible labels: 0 for background, and label 1 for foreground. For all the objects in the image, we expect that one of these proposals contains only the full object. Fig. 5.6 shows examples of proposals obtained with [23].

We formulate the task of cosegmentation as a labelling problem in a complete graph. Each image corresponds to a node in the graph and each proposal segmentation to a label. The goal is to find a labelling $\boldsymbol{s} = (s_k | k = 1, ..., K; s_k \in \{1, ..., 50\})$ that maximises the scoring function:

$$Score(\boldsymbol{s}) = \sum_{(k,k'),k \neq k'} P(s_k, s_{k'}). \tag{5.10}$$

Assigning label $s_k = i$ corresponds to selecting the proposal S_k^i as the segmentation for image I_k. The function in equation (5.10) is a pairwise function similar to the ones previously considered. However, in this formulation a label is associated to each *image* instead of each *pixel*. To emphasise this difference, we use the term "scoring function" to refer to the function in equation (5.10) instead of "energy function".

The pairwise term $P(s_k, s_{k'})$ is learned using a Random Forest regressor [19]. This term encodes both how similar and how close to the ground truth the two proposals are, and is described in detail in section 5.4.2. Since the problem is defined in a complete graph, we compute the pairwise term for all pairs of proposals for all pairs of images.

Optimisation

To find the labelling $\boldsymbol{s}$ that maximises the scoring function (5.10) we use an exact A^*-search algorithm introduced in [4, 7] for labelling in complete graphs. The use of an exact inference algorithm limits the number of images that can be jointly segmented. Alternatively we could use an approximate inference method, such as loopy belief propagation. For the simplest case, when $K = 2$, the inference reduces to choosing the pair of proposals with the highest score.

5.4.2 Learning the pairwise term between proposals

At training time, our method requires ground truth segmentations of pairs of images depicting similar objects. The test images belong to different classes than the ones used to train the system.

For each training image, we consider 50 proposals obtained using [23]. For each pair of proposals, we extract features depending both on the proposals and on the corresponding images. We extract a total of 33 features. There are two different types of features. The first type takes into account the two proposals and images simultaneously. The second type only considers one of the images. The features are inspired by previously used features for similar tasks [23, 87].

We train a Random Forest regressor based on these features. For the two proposals being considered, we compute the overlap of each proposal with ground truth and regress on the sum of the overlaps, where the overlap is given by: $Overlap(S_k^i, GT_k) = (S_k^i \cap GT_k)/(S_k^i \cup GT_k)$. At test time, the score of the Random Forest regressor is used as a pairwise term between proposals.

Features including both images

We start by describing the features that are based on both images. Most of those features are based on histogram similarity.

Given two normalised histograms h_1 and h_2 with b bins, we use as histogram similarity the χ^2-distance measure: $\chi^2(h_1, h_2) = \sum_b (h_1^b - h_2^b)^2/(h_1^b + h_2^b)$.

We consider a total of seven features in this category. The first three features depend on the foreground segment of both images and they measure how similar the foreground of both proposals is with respect to colour, patches and SIFT features. The last features only depend on the proposal segmentations and they measure how similar the two proposal segmentations are in terms of shape.

- **Similarity between the foreground colour histograms of both proposals:** The colour histograms are computed by fitting a Gaussian Mixture Model (GMM) to the RGB colour of both images simultaneously, where each Gaussian in the mixture model corresponds to a bin.

- **Similarity between the foreground histograms of patches:** We use the implementation of [32] to compute a patch codebook with 100 clusters for each pair of images. The foreground histogram of each proposal is obtained from this codebook.

- **Similarity between the foreground histograms of SIFT descriptors:** For each pair of images, we compute SIFT descriptors [74] over a regular grid and cluster them in 100 clusters. We use the code from [65].

- **Similarity between the curvature histograms of the segmentation (2 features):** To compute a histogram over curvature, we use an integral representation of the curvature

[20]. For each point in the boundary we compute the number of foreground pixels inside a circle centred at that point. We use two circles of different radius, obtaining two different histograms.

- **Similarity between the histograms of the boundary orientation:** For each point in the boundary we compute the orientation at that point and cluster them into eight bins.

- **Segmentation overlap:** Overlap of both segmentations when constrained to the tightest possible bounding box and reshaped to have 64×64 pixels.

Features for a single image

Following [23], we also consider features that are computed individually for each image and the corresponding proposal. These features measure how well separated the foreground and background histograms are and provide geometric information of the proposal, such as location and size.

- **Foreground and background similarity (3 features):** Distance between the foreground and background histograms of colour, patches and SIFT. We use the histograms described in the previous section.

- **Alignment with image edges:** Average edge strength on the segmentation boundary.

- **Centroid (2 features):** Coordinates of the centre of mass of the foreground region, normalised by each dimension.

- **Major and minor axis length (2 features):** Lengths of the major and the minor axes of the ellipse that has the same normalised second central moments as the segmentation.

- **Convexity and area (2 features):** Ratio of the number of foreground pixels over the area of the convex hull and over the total area of the image.

- **Bounding box (2 features):** Size of the bounding box (2 dimensions), normalised by the size of the image.

- **Boundary pixels:** Percentage of boundary pixels that belong to the segmentation.

Note that, an important object property, connectivity, is imposed by construction of the proposals.

5.4.3 Learning a single image scoring function

For completeness, in the experimental section (section 5.5), we report the results of training a regression Random Forest for single images, using the features that depend only on a single image.

This method is similar to [23], differing slightly in some of the features and in the dataset used for training. The results of a single image classifier, trained with the same features used for the pairwise classifier, provide information of how much gain in performance, comparing with existing cosegmentation methods, comes from single image measures alone.

5.5 Experimental results

In this section we report results for our system. We start by describing in section 5.5.1 the three different datasets we use. We show both quantitative and qualitative results for three experiments that differ in the datasets used for training and for testing. The measure reported for the quantitative results is the accuracy, i.e. the percentage of pixels in the image (both foreground and background) correctly classified. Since the performance of our method varies substantially for different object classes, we separate the results per class. Note, our algorithm does not use any information about the class of the object.

5.5.1 Datasets

We now provide detailed information about the datasets used. They differ in difficulty and in the amount of intra-class variation. The first two datasets are examples of the "same physical object" scenario, while the last dataset is an example of the "same class" scenario.

Cosegmentation dataset

The *cosegmentation dataset* contains 20 image pairs with the exact same object in similar poses and typically in very different backgrounds. i.e. the ideal setting for cosegmentation. Most of these images have been used before to test other cosegmentation methods [93, 45, 50].

iCoseg dataset

The *iCoseg dataset* was introduced in [5] in the context of interactive cosegmentation and to the best of our knowledge, it was never used in a fully automatic setting.

The dataset is organised in 38 groups in a total of 643 images. Each group contains images of the same object instance or very similar objects from the same class[7]. The iCoseg dataset is a challenging dataset because the objects are deformable, change considerably in terms of viewpoint and illumination, and in some cases, only a part of the object is visible. This contrasts

[7]Since this dataset was not collected by us, we cannot certify that the images correspond to the exact some physical object. However, a visual inspection suggests that this is the case for most of the classes.

Our method	93.3	94.7	92.6	94.5
Upper bound	96.2	95.5	98.1	95.3
Competitors	98 [45]	99 [50]	96 [45]	97 [78]

Figure 5.7: Results of Experiment 1. For each pair of images we report the accuracy for our method, the upper bound corresponding to the accuracy of the best proposal and the accuracy of the best competitor.

significantly with the images typically used to test cosegmentation systems, like the ones in Fig. 5.7. The diversity of the dataset can be seen in the Fig. 5.8, 5.9 and 5.10.

We use a subset of the full dataset. We selected 16 groups of images and, for each of the selected groups, we consider only a subset of the images in order to make it feasible to use A^*-search for maximising function (5.10). In total, we use 122 images from this dataset. We also resized the images to half the size.

MSRC dataset

The *MSRC dataset* was first introduced in the context of supervised class segmentation [102]. It contains objects of 23 different classes in a total of 591 images.

We use a subsect of the images, selecting 7 classes (or groups) and 10 images per class, such that there is a single object in each image.

5.5.2 Experiment 1: the cosegmentation dataset

Training set: cosegmentation dataset (leave one out cross validation)

Test set: cosegmentation dataset

We start by reporting results for the cosegmentation dataset. Since some of these images have been used before for evaluating cosegmentation algorithms, we can compare the results obtained by those methods with our method. The results are obtained by using leave one out cross validation, i.e. we train the system with 19 pairs and use it to evaluate the remaining pair. The average accuracy for this experiment was 91.9% for our single image implementation and 91.8% for our joint method applied to pairs of images.

	Our method		Competitors			Baselines	
	1 image	All images	[23]	Model D	[50]	Upper bound	Uniform
Alaskan bear (9)	79.0	**90.0**	60.4	58.2	74.8	96.4	79.0
Balloon (8)	79.5	90.1	**97.5**	89.3	85.2	99.3	86.8
Baseball (8)	84.5	**90.9**	74.6	69.9	73.0	96.5	88.8
Bear (5)	78.2	**95.3**	83.5	87.3	74.0	97.5	68.4
Elephant (7)	**75.4**	43.1	74.3	62.3	70.1	96.5	82.9
Ferrari (11)	84.8	**89.9**	71.8	77.7	85.0	97.1	73.9
Gymnastics (6)	82.1	**91.7**	72.2	83.4	90.9	96.4	83.4
Kite (8)	89.3	**90.3**	81.5	87.0	87.0	96.7	83.5
Kite panda (7)	80.2	**90.2**	87.7	70.7	73.2	97.8	68.7
Liverpool (9)	87.4	**87.5**	83.2	70.6	76.4	92.7	76.0
Panda (8)	87.8	**92.7**	79.5	80.0	84.0	96.3	62.0
Skating (7)	78.4	77.5	73.4	69.9	**82.1**	85.8	62.7
Statue (10)	92.9	**93.8**	91.5	89.3	90.6	97.8	73.7
Stonehenge (5)	**84.2**	63.3	83.3	61.1	56.6	96.1	78.2
Stonehenge 2 (9)	**88.9**	88.8	79.7	66.9	86.0	93.8	64.4
Taj Mahal (5)	80.7	**91.1**	82.2	79.6	73.7	96.5	82.2

Table 5.5: Segmentation accuracy for experiment 2. We compare our results for the iCoseg dataset with existing methods. Our method outperforms competitors for 11 out of 16 classes. The values in brackets correspond to the number of images used for that class.

The results shown in Fig. 5.7 are visually comparable to the ones presented in previous work on cosegmentation [45, 50, 78, 93]. The accuracy for our method is slightly lower compared with the best accuracy previously reported for each of the images. However, the performance of our method is upper bounded by the accuracy of the best segmentation in the pool of proposals (also reported in Fig. 5.7). A post-processing step, using e.g. [91], could further improve our results, by recomputing a pixel-accurate segmentation. A similar post-processing step was used in [50]. Note that, in [45, 78] the authors used information about the object's colour, based on user seeds and incorporated it in the model as unary terms.

5.5.3 Experiment 2: images with the same object

Training set: cosegmentation dataset

Test set: iCoseg dataset

For this experiment, we use the iCoseg dataset as test set and the cosegmentation dataset as training set. The goal is to show that training our model on a small and distinct dataset still gives good performance.

Table 5.5 shows the segmentation accuracy of different methods for the iCoseg dataset. We compare our results with three previously proposed methods and also report two different baselines, that we now describe in detail.

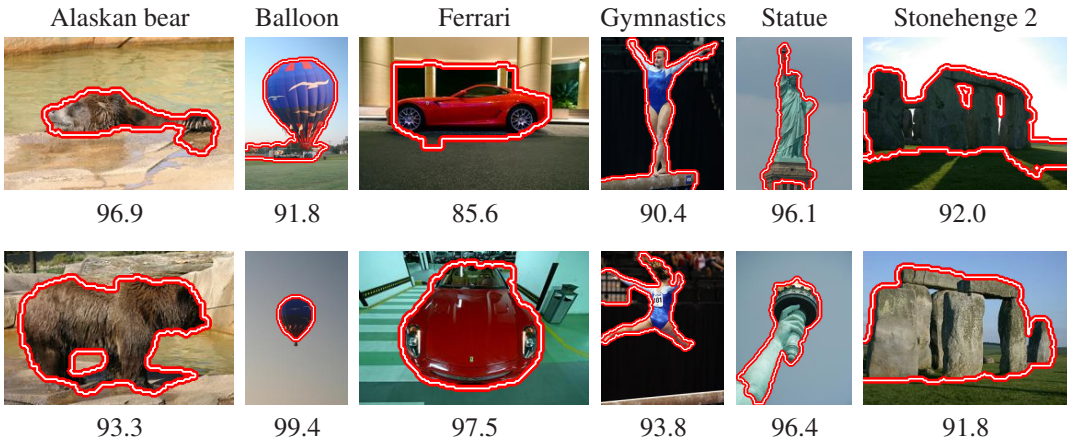


Figure 5.8: Qualitative results for experiment 2. Our method is robust to changes in object size (Balloon), viewpoint (Ferrari and Gymnastics) and partial occlusions of the object (Alaskan bear and Statue). Below each image we report the accuracy of the segmentation.

Competitors

The method of [23] was designed for single image segmentation and we select the highest scoring segmentation as the result. This method is comparable with our method for a single image, differing only slightly in the features used and in the training set.

The second method corresponds to the *Model D* described in section 5.3, i.e. an histogram based approach. We apply the method to all possible pairs of images in each class and reported the average accuracy for all pairs. We use two different initialisations: (1) the histogram intersection previously described and (2) the best scoring segmentations from [23]. From the two results provided by the two initialisations, we select the one with lower energy.

We also compare with [50]. We use the reference implementation of the method and set the only free parameter to 0.001. Since the superpixel code used in [50] is not freely available, we use mean shift to compute the superpixels. For each class, we tested this method using SIFT and colour features, with and without graph cut post-processing and report results for the best of the 4 settings.

Baselines

The last two columns show two different baselines. First, we report the accuracy upper bound for our method. This is given by choosing the best segmentation according to ground truth from the 50 used proposals. For most of the classes, there is a gap between the accuracy of the upper bound and of our method, suggesting that the use of proposals is not considerably limiting the performance of our method.

Finally, we report results considering a uniform segmentation, i.e. for each image, we take the full and the empty segmentations and choose the one with the highest accuracy. For

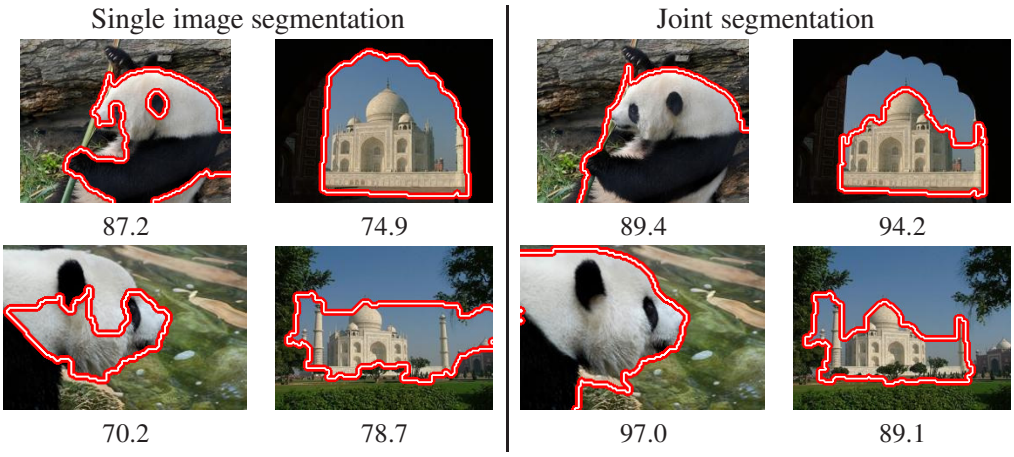


Figure 5.9: Comparison of our single image segmentation and joint segmentation. Single image segmentation fails to correctly segment the object due to strong internal edges (Panda) and strong edges in the background (Taj Mahal).

some classes, the accuracy of a segmentation where all the pixels are labelled background is surprisingly high (e.g. 89% for the Baseball class) even if such a segmentation is meaningless.

From the results reported in 5.5 we can conclude that our method using all images is the best for 11 out of 16 classes. For four classes (Balloon, Elephant, Skating and Stonehenge) other methods are clearly better, which we discuss below, while for the remaining class (Stonehenge 2) the difference in performance is not significant.

Fig. 5.8 shows qualitative results of jointly segmenting all the images in a class. The dataset contains considerable variation within each class and our method is robust to that variation. For example, the Alaskan bear and Statue classes have images with significant object occlusion while the Ferrari images have great variations in terms of viewpoint.

Comparison with single image segmentation

Fig. 5.9 compares the result of segmenting the images individually and jointly. For both classes, there is an increase in accuracy if the images are segmented jointly. For example, for the Panda images, the single image method aligns with the strong boundaries inside the object, while the joint segmentation, correctly retrieves the full panda.

Failure cases

Table 5.5 shows that for some classes the joint method is outperformed by our implementation of single image segmentation. This is particulary noticeable for the Elephant, Skating and Stonehenge classes. Fig. 5.10 shows segmentations for some images in those classes. As it can be seen in Fig. 5.10, the object is very complex in the Skating class, since all the skaters are considered foreground. The proposal segmentations we use are connected and therefore not suitable for segmenting this type of complex foreground. In this particular example there is a

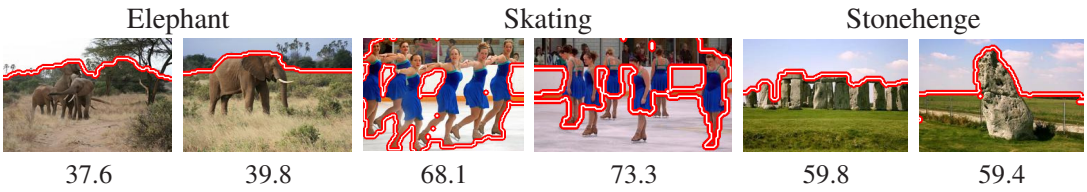


Figure 5.10: Failure cases for experiment 2. Joint segmentation fails for these classes due to the high similarity of the background in all the images (Elephant and Stonehenge) and the complexity of the object (Skating).

considerable amount of background incorrectly labelled foreground.

For the Elephant and the Stonehenge classes, the failure of our algorithm is explained by the similarity of the backgrounds. Recall that most of the pairs used for training (in the cosegmentation dataset) have very distinct backgrounds.

Note that, for the other group with Stonehenge images (Stonehenge 2), some of the images have very different lighting conditions which helps to disambiguate the object. This can be seen in the last column of Fig. 5.8.

Overcoming limitations by "duplicating" the training set

To improve on classes where the background is very similar, we could extend the training set in order to account for those cases. Recall that the goal of *Experiment 2* is to show that the model can be trained on a limited and distinct dataset. In order to keep with this goal, we extend the training set by using extra pairs with the same image, where the images belong to the *cosegmentation dataset*. This procedure does not need extra Ground Truth data and, intuitively, mimics the scenario where the images have the same background.

The accuracy for this experiment is 83.7% for the Elephant class (previous accuracy 43.1%) and 93.4% for the Stonehenge class (previous accuracy 63.3%), while achieving comparable results for all the other classes.

5.5.4 Experiment 3: unsupervised object class segmentation

Training set: MSRC dataset (leave one out cross validation)

Test set: MSRC dataset

For the last experiment we consider the task of unsupervised object class segmentation. We use the MSRC dataset and a leave one out cross validation procedure for training and testing, i.e. we train in 6 classes and test on the remaining one, repeating this procedure for all the classes. The results of this experiment are shown in table 5.6.

For the MSRC dataset, jointly segmenting the 10 images from a class gives comparable accuracy to segmenting independently each image using our single image classifier. We believe

	Our method		Competitors			Baselines	
	1 image	All images	[23]	Model D	[50]	Upper bound	Uniform
Bird	90.8	**95.3**	90.7	88.0	62.2	97.4	84.0
Car	**80.2**	79.6	72.3	64.9	78.6	89.4	62.3
Cat	91.9	**92.3**	87.8	77.5	80.8	96.2	73.0
Cow	93.9	**94.2**	92.9	91.9	80.8	95.3	73.5
Dog	92.9	**93.0**	88.7	86.7	75.6	96.7	78.5
Plane	82.7	**83.0**	78.2	65.7	80.3	90.7	78.9
Sheep	**94.6**	94.0	94.3	89.8	92.5	96.5	74.1

Table 5.6: Segmentation accuracy for the MSRC dataset. For this dataset, the results of our joint method are comparable with single image segmentation.

that this is due to the characteristics of the dataset, where objects tend to be centred in the image, have a good contrast with background, and are homogeneous in terms of colour. In this scenario, the usefulness of the extra information provided by using a set of images with objects of the same class is less obvious, since there is large intra-class variability in terms of appearance. However, note that the improvement over other cosegmentation methods is considerable.

In Fig. 5.11 we show qualitative results for the MSRC dataset. For each class, we show the best, the worst and an average result in terms of accuracy. The figure shows the considerable intra-class variability in this dataset and that our method performs well for the task of unsupervised object class segmentation. The exceptions are the car and plane classes. Besides intra-class variability, these classes have additional characteristics that make them more challenging: the objects are not easily distinguishable from the background (planes) or have very heterogenous appearance and strong internal edges (cars).

5.6 Discussion and limitations

We have discussed in the previous sections how methods for the cosegmentation task can benefit from explicitly imposing the constraint that the region of interest is an *object*.

From the results reported for the different datasets we conclude that our method outperforms existing methods for cosegmentation.

For unsupervised object class segmentation (section 5.5.4) using single image methods (e.g. [23]) already outperforms existing methods for cosegmentation. Intuitively, using multiple images should provide more information and make the problem easier to solve; however, that is not the case for the *Experiment 3* due to the properties of the MSRC dataset: large intra-class variation and objects very distinct from the rest of the image. Although our method uses multiple images, it is capable of adapting to such situation by weighting the importance of single image features accordingly.

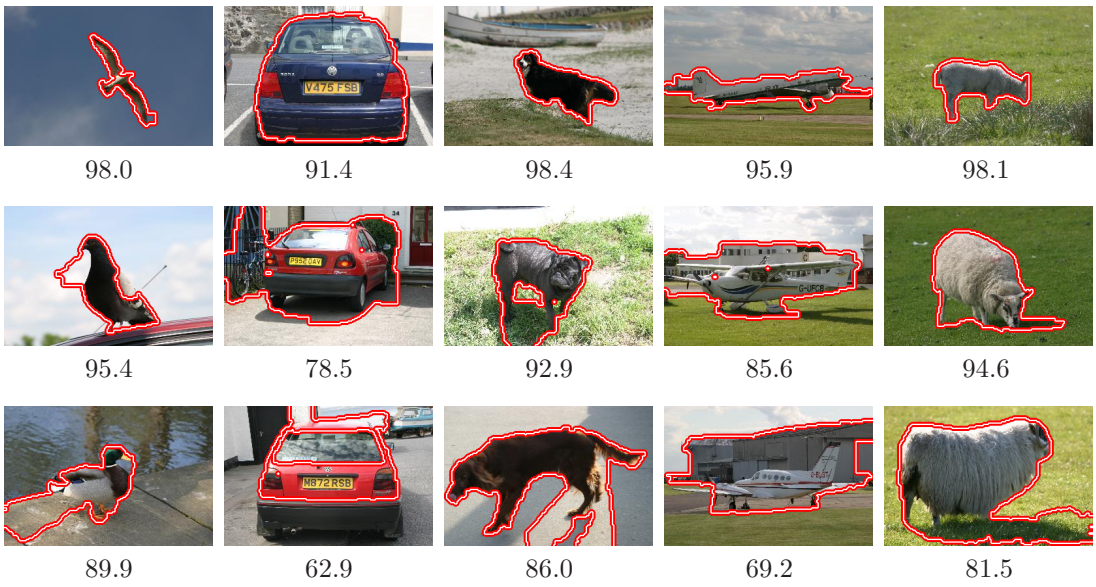


Figure 5.11: Qualitative results for the MSRC dataset. Each row corresponds respectively to the best, average and worse result obtained for the class.

We also showed that for the MSRC dataset our single image version outperforms [23]. This is probably due to the fact that they use the Pascal VOC dataset for training, which contains high variability in terms of object properties, while we perform leave-one-out cross validation in the MSRC dataset.

In the *Experiment 2*, we showed that our method considerably outperforms both state-of-the-art methods for cosegmentation and single image approaches.

In summary, our method presents several advantages compared to existing methods for the same or similar tasks: (1) it can be applicable to sets with a small number of images (in contrast to generative methods for unsupervised object class segmentation); (2) it does not require images of the same class for training (as opposed to supervised object segmentation methods) and (3) it can be adapted to different cosegmentation scenarios, by using a different training set (as opposed to cosegmentation methods that have a "fixed" concept of distance between foreground segments).

Despite these advantages, the method has some limitations:

- We assume that the set of input images contains the same object. This is a limitation compared with other methods that "decide" if the images have the same object [81, 33].

- The fact that the similarity measure is learned can be seen as both an advantage and a disadvantage. On one hand, our method requires training data, contrasting to previous cosegmentation methods. On the other hand, it can adapt to more realistic scenarios while histogram based approaches struggle to succeed for real images, since their assumption

of histogram similarity is very strict and unrealistic.

- The random forest classifier provides a measure of importance for each of the 33 features used and we observed that features relating to size, like the dimensions of the bounding box, have a significant importance. This may be a limitation of our method when extending it to other datasets with more variation on object size.

- As previously discussed, the optimisation technique used is limited to a small number of images. In the case of an enlarged dataset, we would need to resort to an approximate inference technique.

Possible extensions

The accuracy of the method is upper bounded by the quality of the proposals. Therefore, it would be desirable to improve the quality of these proposals by considering other forms of generating them. Alternatively, the method could be combined with a post-processing refining step that would take into account the appearance of the individual image. Such post-processing has been used before for methods that work at the super-pixel level, e.g. [50].

Comparing the results of our method with the upper bound baseline in tables 5.5 and 5.6, we observe that there is still a gap between the two. Ideally we would like to reduce this gap, possibly by including additional features or by using an alternative learning method, for example a structured learning approach that takes into account the full graph construction, as opposed to learning the pairwise potentials individually for pairs of images.

Our method can be easily extended by incorporating extra terms in the scoring function (5.10). For example, the score of the single image classifier can be directly included in the scoring function as a unary term. However, this requires an extra parameter that weights the unary and the pairwise parts of the model.

Another possible extension is to address the case when the variability between pairs of objects in the set is high. Consider the following example scenario, where the set has three images A, B and C. The objects in images A and B are very similar, but the object in C is quite different from both A and B. We expect that $\mathcal{P}(A, B)$ is large, and both $\mathcal{P}(A, C)$ and $\mathcal{P}(B, C)$ are small, where $\mathcal{P}(A, B) = P(s^*_A, s^*_B)$, i.e. the similarity between the two selected proposals in each image. Currently, the segmentation of the object in image B is influenced *equally* by the pairwise term $\mathcal{P}(A, B)$ and $\mathcal{P}(B, C)$. The idea is to down-weight the importance of the term $\mathcal{P}(B, C)$. To achieve this, the sum over P in (5.10) could be replaced by a sum over $f(P)$, where f is some learned robust function, e.g. truncated linear.

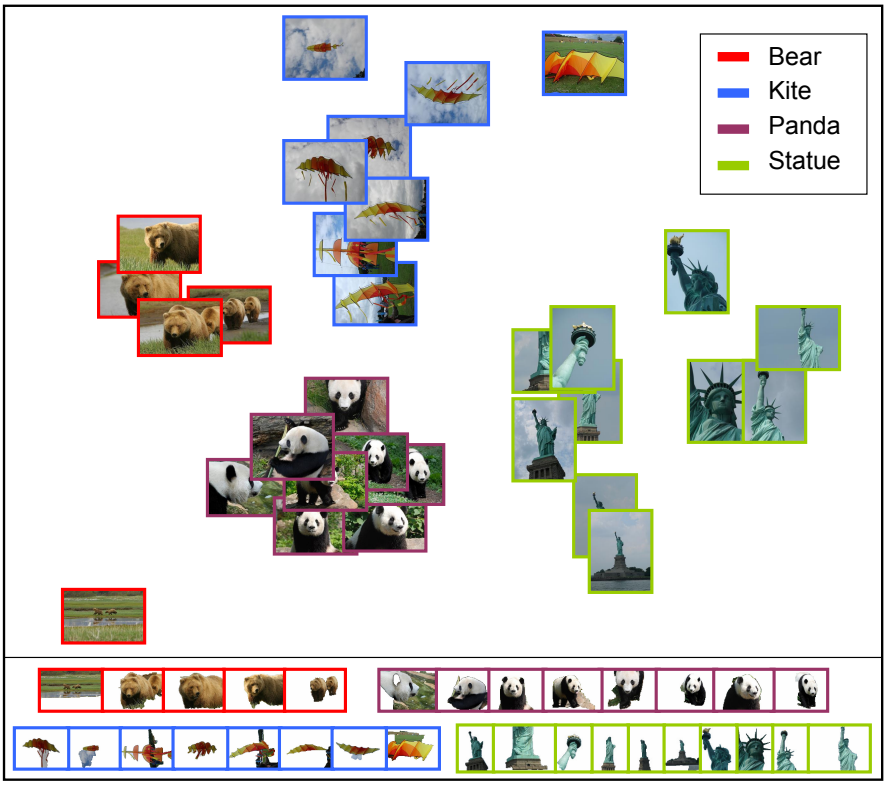


Figure 5.12: Illustrating the idea of an object-sensitive clustering system. We selected 31 images from 4 classes of the iCoseg dataset. All possible pairwise distances between the images were computed with our method, and used to map the images onto a 2D map (using multi-dimensional scaling) - the corresponding segmentations are in the bottom part. We see that the 4 classes are nicely separated and images with very similar foreground objects are close (e.g. top 2 bears). A retrieval system could visualise only the cluster means to illustrate the variability in the dataset. For each image we choose the closest image using our pairwise similarity measure and also show the segmentation corresponding to segmenting that pair.

5.6.1 Applications of cosegmentation

In this section we discuss potential application areas of our system. Although we do not address these specific applications in this thesis, they are worthy of mention.

One interesting scenario is the use of our system to re-rank images in an image retrieval system. For example, if the input to the retrieval system is one image, our system may provide a ranking which focuses on the similarity of the common object as opposed to the similarity of the full image. If the input is a text query, e.g. “animals”, our system can be used to provide object sensitive image pair-distances within the retrieved set. Clustering the images based on these distances can help visualise the variety of images within the results retrieved (see Fig. 5.12).

Note that, in our current method the pairwise distances are trained using only pairs of images that have the same object. Extending the training set to include pairs of images that do not match would probably improve the performance for this particular application.

Another possible application is motivated by interactive cosegmentation [5], where a user provides the system with a set of photographs from a photo collection which contain the same object, for example the iCoseg dataset in Fig. 5.8. Our system automatically provides a solution before any user interaction is done.

5.7 Conclusion

In this chapter we have focused on the task of cosegmentation. We started by reviewing the different tasks that have been previously referred to as cosegmentation and corresponding techniques.

We discussed energy minimisation approaches in more detail in section 5.3 where we showed that Dual Decomposition can be used to improve the optimisation of these models. Although, energy based approaches provide elegant models for cosegmentation, these models are too restrictive for practical scenarios. The assumption that the foreground histograms are similar is not always valid and the method produces segmentations with unrestricted shape.

Many application scenarios of cosegmentation assume that the region of interest is an object. However, this assumption has not been previously included in the models. Unfortunately, the properties expected in an object-like segmentation cannot be easily included in an energy minimisation framework. For this reason, we resorted to a proposal generation approach that has been successfully used as a building block of an object recognition system. This approach also allows to learn a similarity measure for the proposals, which is more robust than the histogram distance used in energy minimisation approaches.

We show state-of-the-art results in a recently introduced challenging dataset. In this dataset, the objects present large variations of viewpoint, scale and illumination.

Chapter 6

Conclusion

In this thesis we have investigated three higher-order models for object segmentation: a connectivity constraint, a joint model for segmentation and appearance, and a model for cosegmentation. We have shown that these models are useful to encode assumptions and impose constraints that go beyond the commonly used pairwise model. We have discussed several energy formulations and introduced corresponding global optimisation methods. In this chapter, we summarise the conclusions of the thesis, discuss the limitations of the methods introduced and point to some future research directions.

6.1 Summary of findings

In chapter 3 we discussed connectivity constraints for image segmentation. We proposed two different optimisation algorithms for minimising an energy function under those constraints: DijkstraGC, a heuristic algorithm inspired by the Dijkstra algorithm for finding shortest paths, and a Dual Decomposition approach. We demonstrated that these connectivity constraints are useful in an interactive scenario for segmentation, in particular in the extraction of long elongated structures that are typically cut off by existing methods.

In chapter 4 we provided insights and a better optimisation algorithm for a commonly used model that jointly optimises segmentation and appearance. We rewrite this model as a function of the segmentation only, by observing that the extra variables that encode the appearance can be written as a function of the segmentation. This transformation was previously unknown and reveals some properties of the model, such as the preference towards balanced segmentations. More interesting, this new formulation allows for global optimisation methods that contrast with the previously used coordinate descent algorithms. The new optimisation procedure based on Dual Decomposition not only outperforms existing methods for more than half of the images in our experiments, but also provides a lower bound to assess optimality.

In chapter 5 we addressed the task of cosegmentation from two different perspectives.

We reviewed existing energy based formulations and showed how Dual Decomposition can be used to improve existing optimisation procedures for those energy formulations. We also discussed that these methods are not appropriate for many realistic images and introduced a proposal generation approach for cosegmentation. This proposal generation approach contrasts with the energy minimisation framework used in the rest of the thesis. This approach allowed us to incorporate several higher-level properties in an easy way. It is unclear how to incorporate these properties in an energy minimisation framework. We showed that this method provides state of the art results for cosegmentation in a challenging dataset containing images of the same object with significant deformations, occlusions and changes in viewpoint.

Dual Decomposition proved to be an effective optimisation approach for a diverse set of problems. This framework is generic enough to be adapted to the different energy functions by careful selection of the subproblems, and it has the extra benefit of providing a lower bound to evaluate the optimality of the solution on a per instance basis. Its success is highly dependent on the existence of efficient algorithms to solve each of the subproblems independently. In particular, we made extensive use of graph cut methods to solve the subproblems arising from Dual Decomposition.

A common topic throughout all the chapters was the use of better and global optimisation methods for energy minimisation. We showed that good optimisation methods are a crucial part of the energy minimisation framework. The use of less powerful methods leads, in many cases, to erroneous conclusions about the properties and applicability of the models. For example, in chapter 3 we showed that a previously proposed bounding box tightness constraint can take advantage of our method (DijkstraGC) for optimisation. DijkstraGC outperforms the previously proposed optimisation method (pinpointing) and reveals the inadequacy of the model in many scenarios.

We also showed that, although energy based formulations provide an elegant and probabilistically sound framework for many interesting vision problems, some constraints are difficult to include in this framework and they lead to hard optimisation problems. We have seen such a scenario in chapter 5 for the cosegmentation task, where we not only wanted to impose object properties, but we also wanted to measure the similarity of the foreground region in multiple images.

6.2 Limitations and future work

In terms of optimisation, although Dual Decomposition provided a good optimisation framework for the models described, this method is unsuitable for real-time or interactive applica-

tions. The alternative methods discussed (DijkstraGC for the connectivity constraint and the EM-style approach for the joint model) have better performance in terms of running time but have several drawbacks: they do not provide any guarantee in terms of optimality, are application specific and are not easily generalisable to other energy functions. An interesting goal would be to devise generic and time efficient algorithms for energy functions with global terms. Recent work in this direction, e.g. [106], has looked at higher-order cliques of a special form that can be efficiently incorporated in traditional optimisation methods, like belief propagation.

An obvious extension of our work is to combine the connectivity constraint described in chapter 3 with the joint model of chapter 4. A unified energy function that encloses both energy formulations should outperform the stand alone versions and could potentially overcome the limitations of the individual models: the "1-pixel width bias" of the connectivity model and the weakness of the joint model in the unconstrained scenario.

Furthermore, this energy function could be complemented with other properties, such as convexity and compactness. We showed in chapter 5 that these properties are relevant to identify object-like segmentations. An interesting question left open is whether it is possible to properly define and globally optimise an energy function that includes all these properties, and how this could be done.

For the cosegmentation task, the next open question would then be how to define and incorporate in the same energy minimisation framework a robust similarity measure between the images.

The models we considered in this thesis do not require information about the object class and are applicable to a wide range of objects. At the other end of the spectrum are models that incorporate information about the object class, for example by imposing shape constraints. Exploring models that fall in between these two cases is a possible direction for future research. For example, models that are suitable for segmenting objects with complicated boundaries, where length regularisation is not applicable (such as plants or fences), or models that are suitable for more specific but still broad object classes like man-made objects or four legged animals.

In this thesis, we showed how some higher level constraints for object segmentation can be encoded in the energy minimisation framework and be often globally optimised. We also showed that some important constraints are not easy to include in an energy minimisation framework, given current state-of-the-art methods. We hope this thesis motivates future research in expanding the boundaries of what is possible to formulate in the energy minimisation framework.

Appendix A

Proofs

The following proofs refer to Theorems within the thesis. They were first introduced in [108, 109] and they were contributed by Vladimir Kolmogorov.

A.1 Theorem 1

Connectivity constraints:

C0 *The set* $[\boldsymbol{x}]$ *corresponding to segmentation* $\boldsymbol{x}$ *must form a single connected component in the graph* $(\mathcal{V}, \mathcal{F})$.

C1 Nodes s, t must be connected in the segmentation set $[\boldsymbol{x}]$, i.e. there must exist a path in the graph $(\mathcal{V}, \mathcal{F})$ from s to t such that all nodes p in the path belong to the segmentation, i.e. $x_p = 1$.

C2 *There must exist a path in the graph* $(\mathcal{V}, \mathcal{F})$ *from* s *to* t *such that for all nodes* p *in the path the subset* $\mathcal{Q}_p$ *belongs to* $[\boldsymbol{x}]$*, i.e.* $x_q = 1$ *for* $q \in \mathcal{Q}_p$.

P0, **P1**, **P2** denote the problems of minimising function (3.2) under constraints **C0**, **C1**, **C2**, respectively.

Theorem. *Problems* **P0**, **P1**, **P2** *are NP-hard.* **P0** *and* **P2** *remain NP-hard even if the set* $\mathcal{N}$ *is empty, i.e. function* (3.2) *does not have pairwise terms.*

Proof. **NP-hardness of P0**

Let us show that the minimum Steiner tree problem (**ST**), which is known to be NP-hard, can be reduced to **P0** with a function $E(\boldsymbol{x})$ containing only unary terms. An instance of **ST** is given by an undirected weighted graph $(\mathcal{V}^\circ, \mathcal{N}^\circ, c)$ with non-negative weights $c : \mathcal{N}^\circ \to \mathbb{N}$ and a subset of nodes $\mathcal{S}^\circ \subseteq \mathcal{V}^\circ$. The goal is to find a subset of edges $\mathcal{X} \subseteq \mathcal{N}^\circ$ of minimum cost such that the set $\mathcal{S}^\circ$ is connected in $(\mathcal{V}^\circ, \mathcal{X})$. (Clearly, there exists a minimum subset which is a tree.)

We construct an instance of **P0** as follows. We start with the graph $(\mathcal{V},\mathcal{F}) = (\mathcal{V}^\circ,\varnothing)$ and the function $E(\boldsymbol{x}) = \sum_{p\in\mathcal{S}^\circ} C(1-x_p)$ where C is a sufficiently large constant, e.g. $C > \sum_{e\in\mathcal{N}^\circ} c_e$. Then for every edge $(p,q)\in\mathcal{N}^\circ$ we add a new node $e=(p,q)$ to $\mathcal{V}$ and two edges (p,e), (e,q) to $\mathcal{F}$. We also add a unary term $c_{pq}x_e$ for the new node.

Let us call a labelling $\boldsymbol{x}\in\{0,1\}^{\mathcal{V}}$ "feasible" if (i) it satisfies **C0**, (ii) $E(\boldsymbol{x}) < C$, i.e. $x_p = 1$ for all nodes $p\in\mathcal{S}^\circ$, and (iii) $e\in[\boldsymbol{x}]$ implies $p,q\in[\boldsymbol{x}]$ for nodes $e=(p,q)\in\mathcal{N}^\circ$. We can make any labelling $\boldsymbol{x}$ satisfying (i) and (ii) feasible by removing nodes $e=(p,q)$ from $[\boldsymbol{x}]$ for which $x_p = 0$ or $x_q = 0$. This operation preserves the connectivity of $[\boldsymbol{x}]$ and does not increase the cost $E(\boldsymbol{x})$. Thus, **P0** has an optimal feasible solution.

There is a one-to-one mapping between feasible solutions and subsets $\mathcal{X}\subseteq\mathcal{N}^\circ$ which form a single connected components and cover all nodes in $\mathcal{S}^\circ$. Furthermore, $E(\boldsymbol{x})$ equals the cost of $\mathcal{X}$ for such solutions. Thus, solving problem **P1** will also solve **VC**.

NP-hardness of P1

Let us show that the minimum vertex cover problem (**VC**), which is known to be NP-hard, can be reduced to **P1**. An instance of **VC** is specified by an undirected graph $(\mathcal{V}^\circ,\mathcal{N}^\circ)$. (We assume that $\mathcal{V}^\circ = \{1,2,\dots,n\}$ where $n = |\mathcal{V}^\circ|$.) The goal is to find a subset $\mathcal{X}\subseteq\mathcal{V}$ of minimum cardinality such that for each edge $(i,j)\in\mathcal{N}^\circ$ at least one of the nodes i,j is in $\mathcal{X}$.

We construct an instance of **P1** as follows. For each node $i\in\mathcal{V}^\circ$ we add two nodes $i,\bar{i}$ to $\mathcal{V}$. We say that solution $\boldsymbol{x}$ specifies subset $\mathcal{X}\subseteq\mathcal{V}$ as follows: $i\in\mathcal{X}$ iff $x_i = 1$. We also add the terminal nodes s and t to $\mathcal{V}$. Thus, $|\mathcal{V}| = 2n+2$. For each pair of consecutive nodes i,$j = i+1$, $1\le i\le n-1$ we add four edges (i,j), $(i,\bar{j})$, $(\bar{i},j)$, $(\bar{i},\bar{j})$ to the connectivity graph $(\mathcal{V},\mathcal{F})$. We also add edges $(s,1)$, $(s,\bar{1})$, (n,t), $(\bar{n},t)$ to $(\mathcal{V},\mathcal{F})$. Thus, $|\mathcal{F}| = 4n$. The connectivity constraint **C1** for the terminal nodes $\{s,t\}$ is equivalent to the following: for each node $i\in\mathcal{V}^\circ$ at least one of the nodes $i,\bar{i}\in\mathcal{V}$ must have label 1. The function $E(\boldsymbol{x})$ is constructed as follows:

- Add unary terms Cx_p for all nodes $p\in\mathcal{V}-\{s,t\}$ where C is a sufficiently large constant, e.g. $C > n$. (These terms will ensure that in the optimal solution exactly one of the nodes $i,\bar{i}$ has label 1.)

- Add pairwise terms $C(1-x_i)x_{\bar{j}}$ for all edges $(i,j)\in\mathcal{N}^\circ$. (These terms will ensure that subset $\mathcal{X}$ corresponding to $\boldsymbol{x}$ satisfies the constraint of the **VC** problem.)

- Add unary terms $1\cdot x_i$ for all nodes $i\in\mathcal{V}^\circ$. (These terms will "count" the cardinality of $\mathcal{X}$.)

Let us call solution $\boldsymbol{x}$ "feasible" if it satisfies the connectivity constraint **C1** and $E(\boldsymbol{x}) < nC + C$. It is easy to see that there is a one-to-one mapping between feasible solutions $\boldsymbol{x}$ and subsets $\mathcal{X} \subseteq \mathcal{V}$ satisfying the constraint of the **VC** problem, and $E(\boldsymbol{x}) = nC + |\mathcal{X}|$ for such solutions. Thus, solving problem **P0** will also solve **VC**.

NP-hardness of P2 without pairwise terms

We will use a reduction from the minimum vertex cover problem similar to the one described above. Given an instance $(\mathcal{V}^\circ, \mathcal{N}^\circ)$ of **VC** we start constructing the graph $(\mathcal{V}, \mathcal{F})$ and the function $E(\boldsymbol{x})$ as before, except that instead of adding a pairwise term $C(1 - x_p)x_q$ where $p = i$, $q = \bar{j}$ we do the following. First, we add a new node r to the graph. Second, we add this node to the sets $\mathcal{Q}_p$ and $\mathcal{Q}_q$. (We assume that in the beginning $\mathcal{Q}_p = \{p\}$ for all nodes p.) Finally, we add unary terms $C(x_r - x_p)$ to the function.

We claim that these operations "simulate" the pairwise term $C(1 - x_p)x_q$. Indeed, if $x_p = x_q = 0$ then the connectivity constraint **C2** does not affect the node r, therefore the contribution of the new term will be $\min_{x_r \in \{0,1\}} C(x_r - 0) = 0$. If $x_p = 1$ or $x_q = 1$ then the connectivity constraint **C2** will imply $x_r = 1$, so the contribution of the new term will be $C(1 - x_p)$ which equals $C(1 - x_p)x_q$ if x_p, x_q are binary and $(x_p, x_q) \neq (0, 0)$. □

A.2 Theorem 2

Theorem. *Suppose that $\boldsymbol{x}$ is a global minimum of function* (3.2) *without any constraints.*

(a) There exists an optimal solution $\boldsymbol{x}^*$ of **P2** *which includes $\boldsymbol{x}$, i.e. $[\boldsymbol{x}] \subseteq [\boldsymbol{x}^*]$. The same holds for the problem* **P1** *since the latter is a special case.*

(b) Suppose that $\mathcal{N} \subseteq \mathcal{F}$. Let $\mathcal{C}_1, \ldots, \mathcal{C}_k \subseteq V$ be the connected components of the set $[\boldsymbol{x}]$ in the graph $(\mathcal{V}, \mathcal{F})$. Then there exists an optimal solution $\boldsymbol{x}^*$ of **P0** *such that each component $\mathcal{C}_i$ is either entirely included in $[\boldsymbol{x}^*]$ or entirely excluded. In other words, if $\mathcal{C}_i$ and $[\boldsymbol{x}^*]$ intersect then $\mathcal{C}_i \subseteq [\boldsymbol{x}^*]$.*

Proof. **Part (a)**

Let $\boldsymbol{y}$ be a global minimum of problem **P2**. Consider solution $\boldsymbol{x}^* = \boldsymbol{y} \vee \boldsymbol{x}$, with $[\boldsymbol{x}^*] = [\boldsymbol{y}] \cup [\boldsymbol{x}]$. It is a global minimum of **P2** since it satisfies the connectivity constraint **C2** and

$$E(\boldsymbol{x}^*) \leq E(\boldsymbol{y}) + [E(\boldsymbol{x}) - E(\boldsymbol{y} \wedge \boldsymbol{x})] \leq E(\boldsymbol{y}).$$

(The first inequality follows from submodularity of function E, and the second inequality holds since $\boldsymbol{x}$ is a global minimum of E.) It remains to notice that $[\boldsymbol{x}] \subseteq [\boldsymbol{x}^*]$.

Part (b)

Let $\boldsymbol{y}$ be a global minimum of **P0**, $\mathcal{C}_1, \ldots, \mathcal{C}_j$ be all connected components of the set $\mathcal{C} = [\boldsymbol{x}]$ that intersect $[\boldsymbol{y}]$, and $\mathcal{C}_{j+1}, \ldots, \mathcal{C}_k$ be the connected components of $\mathcal{C}$ that do not intersect $[\boldsymbol{y}]$. We denote $\boldsymbol{x}'$ and $\boldsymbol{x}''$ to be respectively the labellings corresponding to the sets $\mathcal{C}' = \mathcal{C}_1 \cup \ldots \mathcal{C}_j$ and $\mathcal{C}'' = C - \mathcal{C}'$, i.e. $[\boldsymbol{x}'] = \mathcal{C}'$ and $[\boldsymbol{x}''] = \mathcal{C}''$.

Consider solution $\boldsymbol{x}^* = \boldsymbol{y} \vee \boldsymbol{x}'$, and denote $\boldsymbol{z}' = \boldsymbol{y} \wedge \boldsymbol{x}'$, $\boldsymbol{z} = \boldsymbol{z}' \vee \boldsymbol{x}''$. We claim that $\boldsymbol{x}^*$ is a global minimum of **P0**. Indeed, the set $[\boldsymbol{x}^*] = [\boldsymbol{y}] \cup \mathcal{C}_1 \cup \ldots \cup \mathcal{C}_j$ is connected and

$$\begin{aligned} E(\boldsymbol{x}^*) &\leq E(\boldsymbol{y}) + [E(\boldsymbol{x}') - E(\boldsymbol{z}')] \\ &= E(\boldsymbol{y}) + [E(\boldsymbol{x}) - E(\boldsymbol{z})] \leq E(\boldsymbol{y}). \end{aligned}$$

(The first inequality follows from submodularity of function E, and the last inequality holds since $\boldsymbol{x}$ is a global minimum of E. Let us show the equality in the middle. We can assume without loss of generality that unary and pairwise terms of function E satisfy $D_p(0) = 0$, $V_{pq}(0,0) = 0$. Then $E(\boldsymbol{x}) = E(\boldsymbol{x}') + E(\boldsymbol{x}'')$ since the sets $[\boldsymbol{x}]$ and $[\boldsymbol{x}']$ are disconnected in the graph $(\mathcal{V}, \mathcal{F})$ and $\mathcal{N} \subseteq \mathcal{F}$. Similarly, $E(\boldsymbol{z}) = E(\boldsymbol{z}') + E(\boldsymbol{x}'')$. This implies the desired result.)

It remains to notice that $\mathcal{C}_1, \ldots, \mathcal{C}_j \subseteq [\boldsymbol{x}^*]$ and $\mathcal{C}_{j+1}, \ldots, \mathcal{C}_k$ do not intersect $[\boldsymbol{x}^*]$. □

A.3 Theorem 3

Theorem. *If function $E(\boldsymbol{x})$ does not have pairwise terms and $\mathcal{Q}_p = \{p\}$ for all nodes p (i.e it is an instance of* **P1***) then the algorithm in Fig. 3.5 produces an optimal solution.*

Proof. Suppose that function $E(\boldsymbol{x})$ has only unary terms, i.e. the set $\mathcal{N}$ is empty. We can write it as

$$E(\boldsymbol{x}) = const + \sum_{p \in \mathcal{V}} c_p x_p$$

For the purpose of the proof the constant can be chosen arbitrarily. Let us set it as follows:

$$E(\boldsymbol{x}) = -c^- + \sum_{p \in \mathcal{V}} c_p x_p$$

where $c^- = \sum_{p \in \mathcal{V}} \min\{c_p, 0\}$. Clearly, for any subset $\mathcal{P} \subseteq \mathcal{V}$ we have

$$\min\{E(\boldsymbol{x}) \mid \mathcal{P} \subseteq [\boldsymbol{x}]\} = -c^- + \sum_{p \in \mathcal{P} \vee c_p < 0} c_p = \sum_{p \in \mathcal{P}} c_p^+$$

where we denoted $c_p^+ = \max\{c_p, 0\}$.

Let us prove by induction on the number of steps that $d(p) = d^*(p)$ for all nodes $p \in \mathcal{S}$

where $d^*(p)$ is the optimal solution of problem **P1** for nodes $\{s, p\}$. It is clear that this property holds after initialisation. Consider the step that adds a new node p° to $\mathcal{S}$. Let $\mathcal{P}^*$ be an optimal path from s to p°, then $d^*(p^\circ) = \sum_{r\in\mathcal{P}^*} c_r^+$. Let (p, q) be an edge in this path such that $p \in \mathcal{S}$ and $q \notin \mathcal{S}$. Let $\mathcal{P}_p^*$ be the subset of the path $\mathcal{P}^*$ which goes from s to p. We can write

$$\begin{aligned} d(p^\circ) &\overset{(1)}{\leq} d(q) \overset{(2)}{\leq} d(p) + c_q^+ \overset{(3)}{=} d^*(p) + c_q^+ \\ &\overset{(4)}{\leq} \sum_{r\in\mathcal{P}_p^*} c_r^+ + c_q^+ \overset{(5)}{\leq} \sum_{r\in\mathcal{P}^*} c_r^+ = d^*(p^\circ) \end{aligned}$$

(1) holds since node p° was added to $\mathcal{S}$ rather than q. (2) holds since the edge (p, q) was explored when node p was added to $\mathcal{S}$, and the cost of the proposed solution for node q was $d(p) + c_q^+$. (3) holds by the induction hypothesis. (4) holds since $d^*(p)$ is the optimal distance for node p. (5) holds since path $\mathcal{P}^*$ contains $\mathcal{P}_p^* \cup \{q\}$. Therefore, $d(p^\circ) = d^*(p^\circ)$, as claimed.

Note that if $c_p \geq 0$ for all nodes p then DijkstraGC is equivalent to the standard Dijkstra algorithm which looks for minimum paths from s to all other nodes, if we define the length of edge $(p \to q)$ to be c_p. □

A.4 NP-hardness of the joint model

As discussed in section 4.3, the problem of minimising energy (4.2) with histograms as colour models is equivalent to that of minimising energy (4.6). We will consider a restricted version of the problem in which all pixels are assigned to unique bins. Thus, $B = n$ and $n_b = 1$ for all bins b. Since $n_b^1 \in \{0, 1\}$ and $g_b(0) = g_b(1) = 0$, the energy reduced to

$$E(\boldsymbol{x}) = g(n^1) + \sum_{(p,q)\in\mathcal{N}} w_{pq}|x_p - x_q| \tag{A.1}$$

Suppose that n is even and all weights w_{pq} equal to a sufficiently small constant w so that $g(k) > g(n/2) + w|\mathcal{N}|$ for integers $k \neq n/2$, $k \in [0, n]$. (Such w exists since function $g(\cdot)$ is strictly concave and attains the minimum at $n/2$. Since $g''(z) = 1/z + 1/(n - z) \geq 4/n$ for $z \in [0, n]$ we conclude that $g(z) - g(n/2) \geq 2(z - n/2)^2/n$, so it suffices to take $w < 2/(n|\mathcal{N}|)$.) Then any minimum $\boldsymbol{x}$ is a bisection, i.e. $n^1 = \sum_{p\in\mathcal{V}} x_p = n/2$. The problem of minimising (A.1) is thus equivalent to finding a bisection in an undirected unweighted graph that cuts the smallest number of edges. This *minimum graph bisection* problem is known to be NP-hard.

A.5 Lemma 4

Lemma. *Let $\mathcal{V}_b$ be the set of pixels that fall in bin b. Suppose that pixels in $\mathcal{V}_b$ are not involved in pairwise terms of the energy, i.e. for any $(p,q) \in N$ we have $p,q \notin \mathcal{V}_b$. Also suppose that energy* (4.6) *is minimised under user-provided hard constraints that force a certain subset of pixels to the background and another subset to the foreground. Then there exists a global minimiser $\boldsymbol{x}$ in which all unconstrained pixels in $\mathcal{V}_b$ are assigned either completely to the background or completely to the foreground.*

Proof. Let $\boldsymbol{x}$ be a global minimum of (4.6). Let us fix the labelling of all pixels in $\mathcal{V} - \mathcal{V}_b$, and let us allow the labelling of pixels in $\mathcal{V}_b$ to vary. Let n_b^l be the number of pixels in $\mathcal{V}_b$ with label l, and a_b^l be the number of pixels in $\mathcal{V} - \mathcal{V}_b$ with label l. The energy can then be written as a constant plus

$$f(n_b^1) = g_b(n_b^1) + g(a_b^1 + n_b^1)$$

It is easy to see that function $f(\cdot)$ is concave in $[0, n_b]$. Indeed,

$$\begin{aligned} f''(n_b^1) &= -\left[\frac{1}{n_b^0} + \frac{1}{n_b^1}\right] + \left[\frac{1}{a_b^0 + n_b^0} + \frac{1}{a_b^1 + n_b^1}\right] \\ &= -\frac{a_b^0}{n_b^0(a_b^0 + n_b^0)} - \frac{a_b^1}{n_b^1(a_b^1 + n_b^1)} \;\leq\; 0 \end{aligned}$$

Function (4.6) is minimised under constraints $n_b^1 \in [c_b^1, n_b - c_b^0]$ where c_b^l is the number of pixels in $\mathcal{V}_b$ constrained to have label l. The concavity of $f(\cdot)$ implies that it attains a minimum at one of the ends of the interval $[c_b^1, n_b - c_b^0]$, therefore setting all unconstrained pixels in $\mathcal{V}_b$ either to 0 or to 1 will not increase the energy. □

A.6 Theorem 5

Theorem. *Suppose that continuous functions $\Phi^1, \Phi^2 : \mathbb{R}^{|\mathcal{V}|} \to \mathbb{R}$ have the following properties:*

(a)

$$\Phi^1(\boldsymbol{\lambda} + \delta \cdot \chi_p) \geq \Phi^1(\boldsymbol{\lambda}) + \min_{x \in \{0,1\}} \{-x\delta\} \tag{A.2}$$

for all vectors $\boldsymbol{\lambda}$ and nodes $p \in \mathcal{V}$, where χ_p is the vector of size $|\mathcal{V}|$ with $(\chi_p)_p = 1$ and all other components equal to zero;

(b)

$$\Phi^2(\boldsymbol{\lambda}) = \min_{\boldsymbol{x} \in \{0,1\}^{|\mathcal{V}|}} E^2(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{x} \rangle \tag{A.3}$$

where $E^2(\boldsymbol{x}) = g(\sum_{p \in \mathcal{V}} x_p)$ and function $g(\cdot)$ is convex on $[0, n]$ where $n = |\mathcal{V}|$, i.e.

$2g(k) \leq g(k-1) + g(k+1)$ *for* $k = 1, \ldots, n-1$.

Under these conditions function $\Phi(\boldsymbol{\lambda}) = \Phi^1(\boldsymbol{\lambda}) + \Phi^2(\boldsymbol{\lambda})$ *has maximiser* $\boldsymbol{\lambda}$ *such that* $\lambda_p = \lambda_q$ *for any* $p, q \in \mathcal{V}$.

Proof. Let $\boldsymbol{\lambda}^\circ$ be a maximiser of $\Phi(\cdot)$, and let Ω be the set of vectors $\boldsymbol{\lambda}$ such that $\Phi(\boldsymbol{\lambda}) = \Phi(\boldsymbol{\lambda}^0)$ and $\min_{q\in\mathcal{V}} \lambda_q^\circ \leq \lambda_p \leq \max_{q\in\mathcal{V}} \lambda_q^\circ$ for $p \in \mathcal{V}$. Clearly, Ω is a non-empty compact set. Let $\boldsymbol{\lambda}$ be a vector in Ω with the minimum value of $\Delta(\boldsymbol{\lambda}) = \max_{p\in\mathcal{V}} \lambda_p - \min_{p\in\mathcal{V}} \lambda_p$. (The minimum is achieved in Ω due to compactness of Ω and continuity of function $\Delta(\cdot)$.) If there are multiple vectors $\boldsymbol{\lambda} \in \Omega$ that minimise $\Delta(\boldsymbol{\lambda})$, we will choose a vector such that the cardinality of the set $\{p \in \mathcal{V} \,:\, \min_{q\in\mathcal{V}} \lambda_q < \lambda_p < \max_{q\in\mathcal{V}} \lambda_q\}$ is maximised. We need to prove that $\Delta(\boldsymbol{\lambda}) = 0$. Suppose that $\Delta(\boldsymbol{\lambda}) > 0$. Let $p^- \in \mathcal{V}$ and $p^+ \in \mathcal{V}$ be nodes with the minimum and maximum values of λ_p, respectively, so that $\lambda_{p^+} - \lambda_{p^-} = \Delta(\boldsymbol{\lambda}) > 0$.

Denote $\bar{E}^2(\boldsymbol{x}) = E^2(\boldsymbol{x}) + \langle \boldsymbol{x}, \boldsymbol{\lambda}\rangle$, and let $\mathcal{X}$ be the set of minimisers of $\bar{E}^2(\cdot)$. We claim that there exist labellings $\boldsymbol{x}^-, \boldsymbol{x}^+ \in \mathcal{X}$ such that $x^-_{p^-} = 0$, $x^+_{p^+} = 1$. Indeed, suppose that all labellings $\boldsymbol{x} \in \mathcal{X}$ have $x_{p^-} = 1$, then there exists sufficiently small $\delta \in (0, \Delta(\boldsymbol{\lambda}))$ such that increasing λ_{p^-} by δ will not affect the optimality of labellings in $\boldsymbol{x} \in \mathcal{X}$. As a result of this update, $\Phi^2(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}}[E^2(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{x}\rangle]$ will increase by δ and $\Phi^1(\boldsymbol{\lambda})$ will decrease by no more than δ due to (A.2), therefore vector $\boldsymbol{\lambda}$ will remain a maximiser of $\Phi(\cdot)$. After this update either $\Delta(\boldsymbol{\lambda})$ will decrease or the cardinality of the set $\{p \in \mathcal{V} \,:\, \min_{q\in\mathcal{V}} \lambda_q < \lambda_p < \max_{q\in\mathcal{V}} \lambda_q\}$ will increase. This contradicts to the choice of $\boldsymbol{\lambda}$, which proves the existence of labelling $\boldsymbol{x}^- \in \mathcal{X}$ with $x^-_{p^-} = 0$. Similarly, suppose that all labellings $\boldsymbol{x} \in \mathcal{X}$ have $x_{p^+} = 0$, they will remain optimal if we decrease λ_{p^+} by a sufficiently small amount $\delta \in (0, \Delta(\boldsymbol{\lambda}))$. As a result of this update, $\Phi^2(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}}[E^2(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{x}\rangle]$ will not change and $\Phi^1(\boldsymbol{\lambda})$ will not decrease, therefore vector $\boldsymbol{\lambda}$ will remain a maximiser of $\Phi(\cdot)$. This contradicts to the choice of $\boldsymbol{\lambda}$, and proves the existence of labelling $\boldsymbol{x}^+ \in \mathcal{X}$ with $x^+_{p^+} = 1$.

Next, we will establish some useful properties about the structure of $\mathcal{X}$. Let us call labelling $\boldsymbol{x} \in \{0,1\}^n$ *monotonic* if it satisfies the following property: if $\lambda_p < \lambda_q$ for nodes $p, q \in \mathcal{V}$ then $x_p \geq x_q$. Clearly, any labelling $\boldsymbol{x} \in \mathcal{X}$ must be monotonic. Indeed, if $\lambda_p < \lambda_q$, $x_p = 0$ and $x_q = 1$ then swapping the labels of p and q would decrease $\bar{E}^2(\boldsymbol{x})$.

Let us introduce function

$$\bar{g}(k) = \min_{\boldsymbol{x}:|\boldsymbol{x}||=k} \bar{E}^2(\boldsymbol{x}) = g(k) + \min_{\boldsymbol{x}:||\boldsymbol{x}||=k} \langle \boldsymbol{x}, \boldsymbol{\lambda}\rangle$$

where we denoted $||\boldsymbol{x}|| = \sum_{p\in\mathcal{V}} x_p$. It is easy to see that $\boldsymbol{x} \in \mathcal{X}$ if and only if two conditions hold: (i) $\bar{g}(k)$ achieves the minimum at $k = ||\boldsymbol{x}||$; (ii) labelling $\boldsymbol{x}$ is monotonic. (Note, all

monotonic labellings $\boldsymbol{x}$ with the same count $||\boldsymbol{x}||$ have the same value of $\langle \boldsymbol{x}, \boldsymbol{\lambda} \rangle$.)

Let $(\lambda^1, \ldots, \lambda^n)$ be the sequence of values λ_p, $p \in \mathcal{V}$ sorted in the non-decreasing order. In other words, λ^k is the k-th smallest element among values λ_p, $p \in \mathcal{V}$. Clearly, we have

$$\bar{g}(k) = g(k) + \sum_{i=1}^{k} \lambda^i \,, \qquad k = 0, 1, \ldots, n$$

Functions $g(k)$ and $s(k) = \sum_{i=1}^{k} \lambda^i$ are convex, so $\bar{g}(k)$ is convex as well. Therefore, the set of values of k that minimise $\bar{g}(k)$ form an interval $[k^-, k^+]$ where $0 \le k^- \le k^+ \le n$. Furthermore, if $k^- < k^+$ then $\lambda^{k^-+1} = \lambda^{k^+}$. Indeed, we have $\bar{g}(k) = const$ for $k \in [k^-, k^+]$, i.e. function $\bar{g}(\cdot)$ is linear on $[k^-, k^+]$. It is a sum of two convex functions, so both functions must be linear on $[k^-, k^+]$. This implies that $s(k^- + 1) - s(k^-) = s(k^+) - s(k^+ - 1)$, i.e. $\lambda^{k^-+1} = \lambda^{k^+}$.

Let us show that $\lambda_{p^+} = \lambda^{k^+}$. Suppose not: $\lambda^{k^+} < \lambda_{p^+}$. Then there are at least k^+ nodes $p \in \mathcal{V}$ with $\lambda_p < \lambda_{p^+}$. They must satisfy $x_p^+ = 1$, since $x_{p^+}^+ = 1$ and $\boldsymbol{x}^+$ is monotonic. Thus, there are at least $k^+ + 1$ nodes $p \in \mathcal{V}$ with $x_p = 1$, so $||\boldsymbol{x}^+|| \ge k^+ + 1$ - a contradiction.

Similarly, we can show that $\lambda_{p^-} = \lambda^{k^-+1}$. (Note, we have $k^- \le ||\boldsymbol{x}^-|| \le n-1$.) Suppose not: $\lambda_{p^-} < \lambda^{k^-+1}$. Then there are at least $n - k^-$ nodes $p \in \mathcal{V}$ with $\lambda_p > \lambda_{p^-}$. They must satisfy $x_p^- = 0$, since $x_{p^-}^- = 0$ and $\boldsymbol{x}^-$ is monotonic. Thus, there are at least $n - k^- + 1$ nodes $p \in \mathcal{V}$ with $x_p = 0$, so $||\boldsymbol{x}^-|| \le k^- - 1$ - a contradiction.

The arguments above imply that if $k^- < k^+$ then $\lambda_{p^-} = \lambda^{k^-+1} = \lambda^{k^+} = \lambda_{p^+}$, and if $k^- = k^+$ then $\lambda_{p^-} = \lambda^{k^-+1} \ge \lambda^{k^+} = \lambda_{p^+}$. This contradicts to the assumption $\lambda_{p^-} < \lambda_{p^+}$ made earlier. □

Appendix B

Illustration of the DijkstraGC algorithm

We provide an illustration of the DijkstraGC algorithm applied to a specific example. Recall that the DijkstraGC algorithm aims to minimise a pairwise energy function under the constraint that the terminal nodes s and t are connected. We discussed two related connectivity constraints **C1** and **C2** and for simplicity of presentation we consider only **C1**.

The algorithm starts with an initialisation step, depicted in Fig. B.1, where all nodes are set free and with distance ∞. For node s (initial terminal node), the algorithm computes the minimum of the energy under the constraint that s belongs to the foreground, i.e. it solves $min_{\boldsymbol{x}} E(\boldsymbol{x}|s \in [\boldsymbol{x}])$. (Recall that $[\boldsymbol{x}]$ is the set of nodes with label 1. The distance of node s is initialised with the energy of this solution.

A later iteration is illustrated in Fig. B.2. It is composed of three steps. In the first step, the algorithm selects the free node p with smallest distance. In the second step it adds that node to the set of fixed nodes and it fixes its *PARENT* node. Finally, in the third step it updates the distance of the free neighbours. For each neighbour q, the energy is minimised under the constraint that all the nodes in the path $\mathcal{P}$ connecting s and q through p belong to foreground, i.e. it solves $min_{\boldsymbol{x}} E(\boldsymbol{x}|\mathcal{P} \in [\boldsymbol{x}])$. The path $\mathcal{P}$ is obtained from the *PARENT* pointers. The distance for node q is updated with the value of this energy, if it is smaller than the current distance. In Fig. B.2 the distance for the first neighbour visited is not updated, while the distance for the second neighbour visited and corresponding *PARENT* pointer is updated.

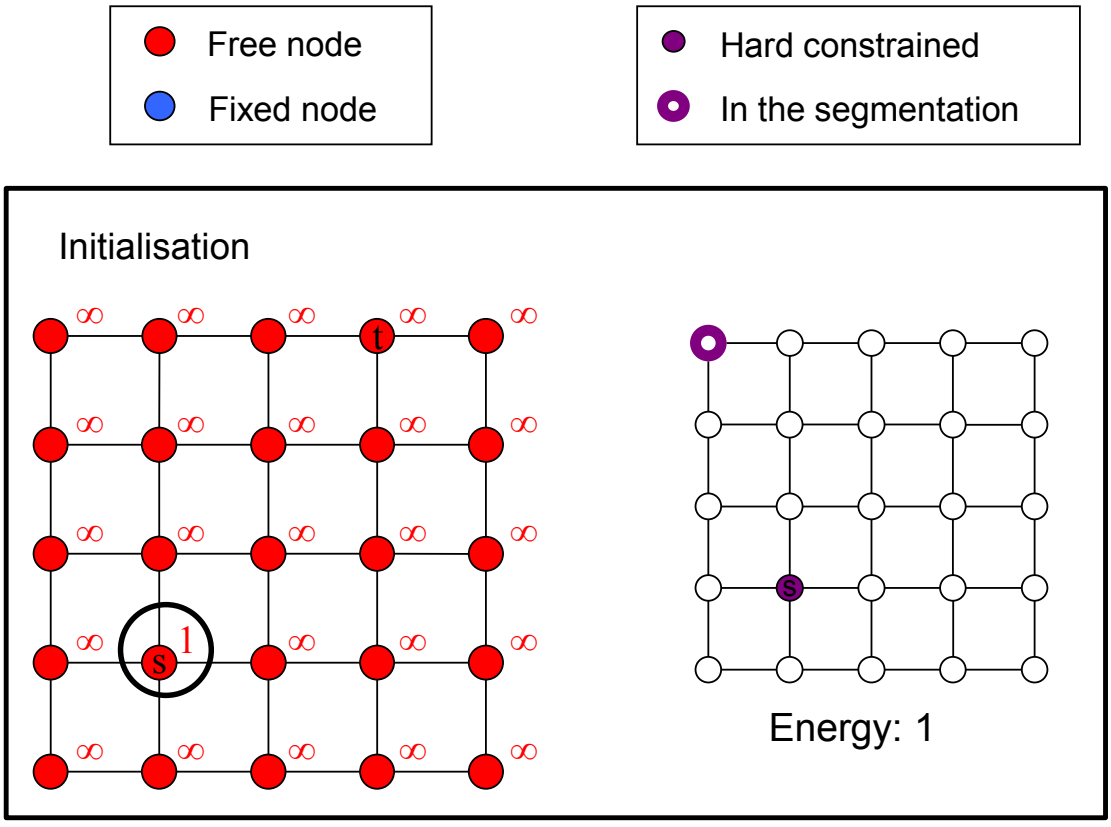


Figure B.1: Initialisation of the DijkstraGC Algorithm.

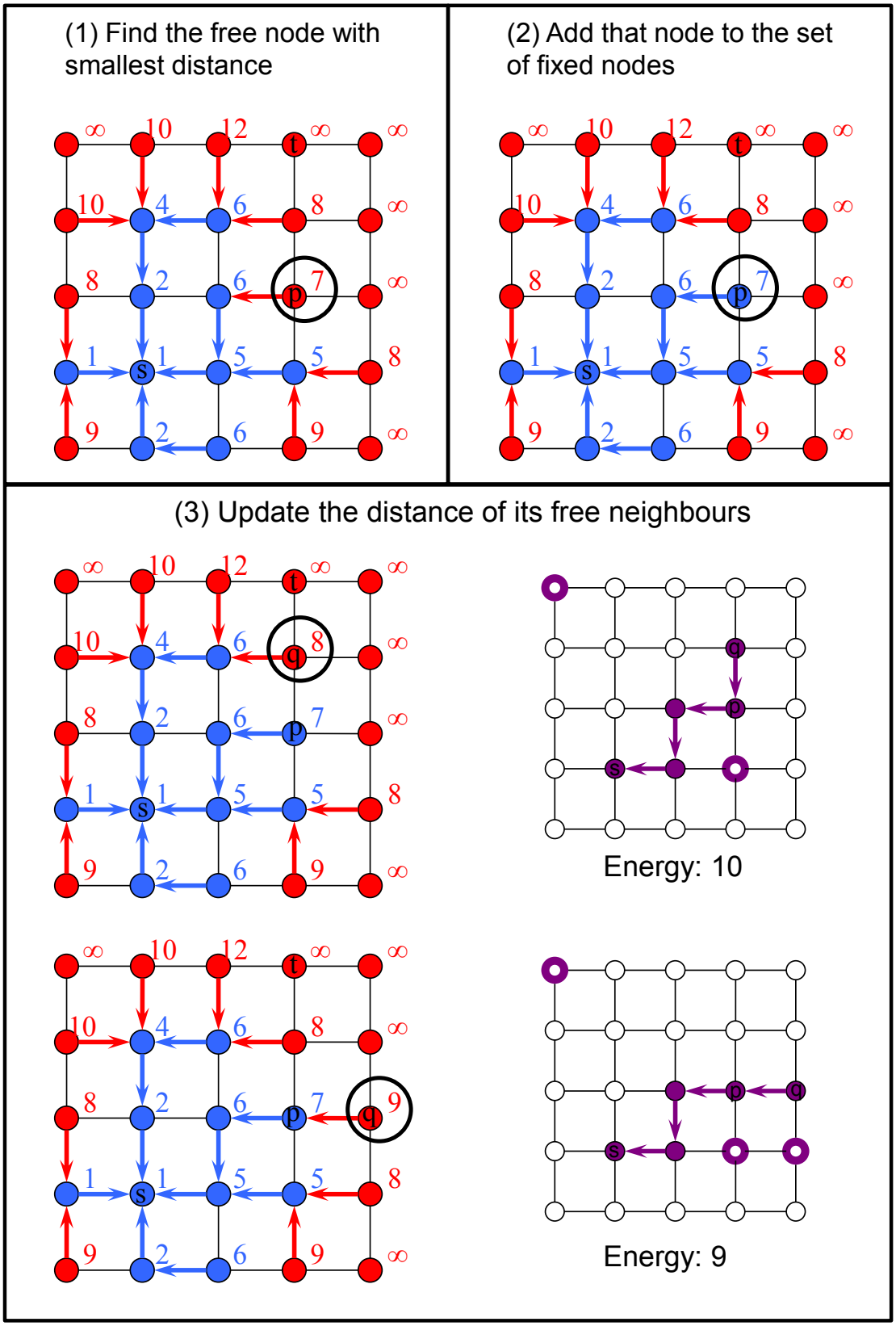


Figure B.2: Sample iteration of the DijkstraGC algorithm.

Bibliography

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Classcut for unsupervised class segmentation. In *European Conference on Computer Vision (ECCV)*, 2010.

[3] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[4] B. Andres, J. H. Kappes, U. Koethe, C. Schnörr, and F. A. Hamprecht. An empirical comparison of inference algorithms for graphical models with higher order factors using OpenGM. In *DAGM*, 2010.

[5] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive co-segmentation with intelligent scribble guidance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[6] D. Batra, D. Parikh, A. Kowdle, T. Chen, and J. Luo. Seed image selection in interactive cosegmentation. In *IEEE International Conference on Image Processing (ICIP)*, 2009.

[7] M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnörr. A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision*, 87(1-2):93–117, 2010.

[8] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[9] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–302, 1986.

[10] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2008.

[11] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *European Conference on Computer Vision (ECCV)*, 2004.

[12] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *European Conference on Computer Vision (ECCV)*, 2002.

[13] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155 – 225, 2002.

[14] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *IEEE International Conference on Computer Vision (ICCV)*, 2001.

[15] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *IEEE International Conference on Computer Vision (ICCV)*, October 2003.

[16] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9), September 2004.

[17] Y. Boykov and O. Veksler. Graph cuts in vision and graphics: Theories and applications. In *Handbook of Mathematical Models in Computer Vision*. Springer, 2005.

[18] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11), November 2001.

[19] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[20] J. W. Bullard, E. J. Garboczi, W. C. Carter, and E. R. Fuller Jr. Numerical methods for computing interfacial mean curvature. *Computational Materials Science*, 4:103–116, 1995.

[21] N. D. F. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Automatic 3D object segmentation in multiple views using volumetric graph-cuts. *Image and Vision Computing*, 28(1):14 – 25, 2010.

[22] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[23] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[24] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[25] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266 –277, February 2001.

[26] C. Chen, D. Freedman, and C.H. Lampert. Enforcing topological constraints in random field image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[27] D. Cremers, T. Pock, K. Kolev, and A. Chambolle. Convex relaxation techniques for segmentation, stereo and multiview reconstruction. In *Advances in Markov Random Fields for Vision and Image Processing*. MIT Press, 2011.

[28] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007.

[29] A. Criminisi, T. Sharp, and A. Blake. Geos: Geodesic image segmentation. In *European Conference on Computer Vision (ECCV)*, 2008.

[30] J. Cui, Q. Yang, F. Wen, Q. Wu, C. Zhang, L. Van Gool, and X. Tang. Transductive object cutout. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[31] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[32] T. Deselaers and V. Ferrari. Global and efficient self-similarity for object classification and detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[33] T. Deselaers and V. Ferrari. Visual and semantic similarity in imagenet. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[34] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[35] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004.

[36] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[37] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18:30–55, 1989.

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

[39] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, November 2006.

[40] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.

[41] M. Guignard and S. Kim. Lagrangean decomposition: a model yielding stronger Lagrangean bounds. *Mathematical programming*, 39(2):215–228, 1987.

[42] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[43] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0-1 optimization. *Mathematicl Programming*, 28:121–155, 1984.

[44] X. Han, C. Xu, and J. L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):755–768, 2003.

[45] D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[46] H. Ishikawa. Exact optimization for Markov Random Fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, October 2003.

[47] H. Ishikawa. Higher-order clique reduction in binary graph cut. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2993 –3000, 2009.

[48] S. Jegelka and J. Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[49] I. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10), October 2001.

[50] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[51] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[52] P. Kohli, M.P. Kumar, and P.H.S. Torr. P3 beyond: Solving energies with higher order cliques. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[53] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[54] P. Kohli, J. Rihan, M. Bray, and P.H.S. Torr. Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *International Journal of Computer Vision*, 79(3):285–298, 2008.

[55] P. Kohli and P. H. S. Torr. Efficiently solving dynamic Markov random fields using graph cuts. In *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[56] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[57] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1274–1279, 2007.

[58] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004.

[59] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[60] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[61] A. Kowdle, YJ Chen, D. Batra, and T. Chen. Scribble based interactive 3d reconstruction via scene co-segmentation. In *IEEE International Conference on Image Processing (ICIP)*, 2011.

[62] M. P. Kumar, P. H. S Torr, and A. Zisserman. Obj cut. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[63] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, 2001.

[64] X. Lan, S. Roth, D. Huttenlocher, and M. Black. Efficient belief propagation with learned higher-order markov random fields. In *European Conference on Computer Vision (ECCV)*, pages 269–282, 2006.

[65] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[66] I. Leichter and M. Lindenbaum. Boundary ownership by lifting to 2.1D. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[67] V. Lempitsky, A. Blake, and c. Rother. Image segmentation by branch-and-mincut. In *European Conference on Computer Vision (ECCV)*, 2008.

[68] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[69] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[70] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[71] Y. Li, J. Sun, C.K. Tang, and H.Y. Shum. Lazy snapping. *ACM SIGGRAPH*, August 2004.

[72] J. Liu, J. Sun, and H.Y. Shum. Paint selection. *ACM SIGGRAPH*, July 2009.

[73] T. Liu, J. Sun, N.N. Zheng, X. Tang, and H.Y. Shum. Learning to detect a salient object. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

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

[75] T. Malisiewicz and A. Efros. Improving spatial support for objects via multiple segmentations. In *British Machine Vision Conference (BMVC)*, 2007.

[76] 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 *IEEE International Conference on Computer Vision (ICCV)*, July 2001.

[77] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. *ACM SIGGRAPH*, 1995.

[78] L. Mukherjee, V. Singh, and C. R. Dyer. Half-integrality based algorithms for cosegmentation of images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[79] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[80] H. Nickisch, C. Rother, P. Kohli, and C. Rhemann. Learning an interactive segmentation system. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, 2010.

[81] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[82] S. Nowozin and C. H. Lampert. Global connectivity potentials for random field models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[83] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(3-4):185–365, 2010.

[84] C.H. Papdimitriou and K. Steiglitz. *Combinatorial optimization: Algorithms and complexity*. 1982.

[85] S. Prince. *Computer vision: models, learning and inference*. Cambridge University Press (to appear).

[86] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *IEEE Transactions on Image Processing*, 16(4):1046–1057, April 2007.

[87] X. Ren and J. Malik. Learning a classification model for segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[88] C. Rhemann, C. Rother, P. Kohli, and M. Gelautz. A spatially varying PSF-based prior for alpha matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[89] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[90] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009.

[91] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *ACM SIGGRAPH*, August 2004.

[92] C. Rother, V. Kolmogorov, Y. Boykov, and A. Blake. Interactive foreground extraction using graph cut. Technical Report MSR-TR-2011-46, Microsoft Research, 2011.

[93] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[94] B. C. Russell, W. T. Freeman, A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[95] M. I. Schlesinger. Syntactic analysis of two-dimensional visual signals in noisy conditions (in russian). *Kibernetika*, 4(113-130):1868, 1976.

[96] M. I. Schlesinger and V. V. Giginyak. Solution to structural recognition (MAX,+)-problems by their equivalent transformations. Part 1. *Control Systems and Computers*, (1):3–15, 2007.

[97] M. I. Schlesinger and V. V. Giginyak. Solution to structural recognition (MAX,+)-problems by their equivalent transformations. Part 2. *Control Systems and Computers*, (2):3–18, 2007.

[98] T. Schoenemann. Minimizing count-based high order terms in markov random fields. In *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2011.

[99] T. Schoenemann and D. Cremers. Introducing curvature into globally optimal image segmentation: Minimum ratio cycles on product graphs. In *IEEE International Conference on Computer Vision (ICCV)*, October 2007.

[100] T. Schoenemann, F. Kahl, and D. Cremers. Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In *IEEE International Conference on Computer Vision (ICCV)*, October 2009.

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

[102] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision (ECCV)*, 2006.

[103] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[104] P. Strandmark and F. Kahl. Curvature regularization for curves and surfaces in a global optimization framework. In *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2011.

[105] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov

random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1068–1080, 2008.

[106] D. Tarlow, I. Givoni, and R. Zemel. Hop-map: Efficient message passing with high order potentials. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[107] O. Veksler. Star shape prior for graph-cut image segmentation. In *European Conference on Computer Vision (ECCV)*, 2008.

[108] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008.

[109] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[110] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on trees: Message-passing and linear-programming approaches. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005.

[111] T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7), 2007.

[112] T. Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[113] J. Winn and N. Jojic. LOCUS: learning object classes with unsupervised segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[114] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[115] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Neural Information Processing Systems Conference (NIPS)*, pages 689–695, 2000.

[116] Y. Zeng, D. Samaras, W. Chen, and Q. Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images. *Computer Vision and Image Understanding*, 112(1):81–90, 2008.

[117] S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9), 1996.