12-16-2013 12:00 AM

Color Separation for Image Segmentation

Meng Tang
The University of Western Ontario

Supervisor
Yuri Boykov
The University of Western Ontario

Graduate Program in Computer Science
A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

COLOR SEPARATION FOR IMAGE SEGMENTATION

(Thesis format: Monograph)

by

Meng Tang

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Masters of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

Image segmentation is a fundamental problem in computer vision that has drawn intensive research attention during the past few decades, resulting in a variety of segmentation algorithms. Segmentation is often formulated as a Markov random field (MRF) and the solution corresponding to the maximum a posteriori probability (MAP) is found using energy minimization framework. Many standard segmentation techniques rely on foreground and background appearance models given a priori. In this case the corresponding energy can be efficiently optimized globally. If the appearance models are not known, the energy becomes NP-hard, and many methods resort to iterative schemes that jointly optimize appearance and segmentation. Such algorithms can only guarantee local minimum.

Here we propose a new energy term explicitly measuring the L_1 distance between object and background appearance models that can be globally maximized in one graph cut. Our method directly tries to minimize the appearance overlap between the segments. We show that in many applications including interactive segmentation, shape matching, segmentation from stereo pairs and saliency segmentation our simple term makes NP-hard segmentation functionals unnecessary and renders good segmentation performance both qualitatively and quantitatively.

Keywords: Image Segmentation, Appearance Model, Markov Random Fields, Color Separation, Submodular Function Minimization, Pesudo-boolean Function

Co-Authorship Statement

Section 1.3 in **Chapter 1** is primarliy written by my advisor Dr. Yuri Boykov. The motivation of this work is brought forward and contribution is summarized. Other parts in **Chapter 1, 2** and **3** are based on my own summary of related work.

Chapter 4 is collaboration work with Dr. Lena Gorelick. I programmed the code for all applications and wrote **Chapter 5** by my self. Dr. Olga Veksler gave me the valuable suggestion of comparing different color separation terms. The idea of combining color separation term with template shape prior is from Dr. Yuri Boykov, which leads to template shape matching application in **Section 5.2**.

Acknowlegements

I would like to thank my advisor Dr. Yuri Boykov and informal co-advisor Dr. Olga Veksler. When I need help in my research, they are always there, listening to my ideas, discussing with me and sparkling new ideas. They really care about my research and help me grow. I still remember the first meeting with Dr. Yuri Boykov and Dr. Olga Veksler when I had a small idea about dynamic graph cut and amazingly they quickly set up a meeting with me. They were willing to listen to and develop my idea even though I was just a first year graduate student at that time. Working with them is a lot of fun, highly efficient and I learnt how to do real research from scratch. Especially, I gained a sense of what is important research from them. I am fortunate to have them as advisors.

Thank you Dr. Lena Gorelick for helping me with my research project. I bothered you too many times and each time you patiently answered my questions and even helped me troubleshoot technical problem of my project. Thank you Dr. Andrew Delong, a former member of the computer vision group, for proving data used in your label cost paper when I requested.

I appreciate the hard-working of the thesis committee members, including Dr. Roberto Solis-Oba, Dr. Stephen Watt and Dr. Hristo Sendov. Their advices help me a lot in presenting this work.

I would also like to thank other members of the computer vision group at Western University including Liqun Liu, Xuefeng Chang, Igor Milevskiy and Yuchen Zhong. Even though they have different research projects from me, I also benefit from their outstanding research progress and they provided great pieces of advice on my research. Life without these lovely friends would be boring.

Last but not least, I would like to thank my parents and sister who support my study in Canada and give me this wonderful life I've ever experienced.

Contents

Abstract ii

Co-Authorship Statement iii

Acknowlegements iv

List of Figures vii

List of Tables x

1 Introduction 1

1.1 Markov Random Fields: Modeling and Inference 2

1.2 Markov Random Fields for Image Segmentation 6

1.3 Motivation of Color Separation Term for Image Segmentation 7

1.4 Contribution of the Thesis 13

1.5 Outline of the Thesis 14

2 Overview of Appearance Models 15

2.1 Histogram 15

2.2 Non-parametric Density Estimation 16

2.3 Gaussian Mixture Model 17

3 Related Work 20

3.1 GrabCut 20

3.2 Branch-and-Mincut 22

3.3 Dual Decomposition 23

3.4 Active Contour 23

4 Minimizing Appearance Overlap in One-cut 25

4.1 L_1 Color Separation Term 25

4.2 Minimizing Higher-order Pseudo-boolean Function 27

4.3 Relationship with P^n Potts Model 28
4.4 General Color Separation Term 29

5 Applications **31**
5.1 Interactive segmentation 31
5.1.1 Binary segmentation with bounding box 31
5.1.2 Comparison of Appearance Overlap Terms 38
5.1.3 Interactive Segmentation with Seeds 40
5.2 Template Shape Matching 41
5.3 Salient object segmentation 43
5.4 Foreground Segmentation from Stereo 45

6 Future Work and Conclusion **48**
6.1 Color Separation Term for GMM Appearance Model 48
6.2 Color Separation Term with Supermodular Term 48
6.3 Feature Separation Term for Multi-label Inference 50
6.4 Conclusion 51

Bibliography **52**

A Proofs of Theorems **58**

Curriculum Vitae **59**

List of Figures

1.1 Markov Property: The state of the black node is conditionally independent of all the white nodes, given the states of the gray nodes. This is called the Markov property. 2

1.2 Hidden Markov Model on a 4-connected graph. The upper layer represents observed value of nodes and the lower layer represents hidden state variables of nodes. 3

1.3 Take LP relaxation of the original energy with discrete variables and maximize the lower bound. 5

1.4 Graph Cut for Interactive Segmentation: (top-left) User specified seeds denoted by blue and red pixels, (top-right) Graph with pixels as nodes and S, T terminals, (bottom-left) Graph with t-links and n-links, (bottom-right) s/t mincut. [Image credit: Yuri Boykov] 8

1.5 Interactive image segmentation: user provided strokes (Left) and result (right). . 9

1.6 If we only optimize the first two terms in (1.15) and restrict the foreground to be a rectangle, we can get a rough bounding box of foreground and background in sliding-window fashion. The foreground is indicated by red box. 10

1.7 Color separation gives segments with low entropy. 10

1.8 Energy (1.15): volume balancing (a) and Jensen-Shannon color separation terms (b). Our L_1 color separation term (c). 11

1.9 ROC curves of thresholding log likelihood ratios with different number of color bins used in color histogram 13

2.1 Binning in RGB color space [1] 16

2.2 Window size is fixed for Parzen window (left) and number of neighboring points is fixed for k-NN(right). Window size or number of neighboring points should be chosen properly to get a good density estimation. © Olga Veksler. . . 17

2.3 Gaussian Mixture Model in RGB color space. In this example we have three Gaussian mixture components highlighted by three ellipses. 19

3.1 Scatter plots of energies versus error rates for different number of bins per channel. 21
3.2 Branch-and-Mincut [40] can find smaller global energy while GrabCut gets stuck in local minima with larger energy. 22
3.3 (top row) Intensity-based segmentation of Zebra at each iteration. (bottom row) Corresponding intensity distributions of Zebra and its background. *Pin* and *Pout* are intensity distributions of regions inside and outside the contour. [45] . 24

4.1 Graph construction for E_{L_1} in one color bin: nodes $v_1, v_2, \ldots, v_{n_k}$ corresponding to the pixels in bin k are connected to the auxiliary node A_k using undirected links. The capacity of these links is the weight of appearance overlap term $\beta > 0$. 26
4.2 Overall graph construction for energy with L_1 color separation term. 26
4.3 Our graph for minimizing pseudo-boolean function (4.3) 27
4.4 Graph for minimizing pseudo-boolean function (4.3) by Kohli et al. [30, 31] . . 29
4.5 Appearance overlap terms based on different metrics: L_1 norm, χ^2 distance, Bhattacharyya coefficient and Jensen-Shannon divergence 30
4.6 The original concave function (red) is approximated as a piece-wise linear function (blue, left) using three truncated components (blue,middle). Approximation with ten components (blue, right) is already very accurate. 30

5.1 Error-rates for different bin resolutions, as in Table 5.1. 34
5.2 Example of segmentation results. From left to right: (a) user input, (b) GrabCut [9, 51], (c) Dual Decomposition (DD) [59], (d) our One-Cut. For these examples we used 16^3 bins. 35
5.3 Example of segmentation results obtained with our One-Cut. For these examples we used 128^3 bins. 36
5.4 Example of segmentation results obtained with our One-Cut. For these examples we used 16^3 bins. 37
5.5 Error rates and running time comparison of different appearance overlap terms. 38
5.6 Left: Truncated appearance overlap term $D_{L_1^T}$ for a bin k. Right: Segmentation error rate as a function of parameter t in $D^T_{L_1}$. Best results are achieved for $t = 1$ (no truncation). 39
5.7 Interactive segmentation with seeds 40
5.8 Template shape matching examples, from left to right: Original images, contrast sensitive edge weights, shape matching results without and with the appearance overlap penalty. Input shape templates are shown as contours around the resulting segmentation. 41

5.9 Template shape matching examples: shape (top left) and pairs of original images + segmentations with $E_2(S)$. 42
5.10 Different saliency maps 43
5.11 Saliency segmentation results reported for dataset [1]: Precison-Recall and F-measure bars for $E_3(S)$, $E_4(S)$ are compared to FT[1], CA[24], LC[64], HC[19] and RC[19]. 44
5.12 Saliency segmentation examples: (a) Original image, (b) Saliency map from [49] with bright intensity denoting high saliency, (c-d) Graph cut segmentation without and with appearance overlap penalty term, (e) Ground truth. 46
5.13 Foreground segmentation from stereo pair (a) One of input stereo images, (b) Segmentation result when optimizing energy $E_5(S)$ (5.12) (c) Refine (b) with EM (d) Segmentation with color separation augmented energy $E_6(S)$ (5.14). . . 47

6.1 Color separation term for Gaussian Mixture Model. 49
6.2 Combine color separation term with FTR for volume balancing term. 50

List of Tables

4.1 Cut costs corresponding to four possible label assignments to the binary auxiliary nodes A_k^1 and A_k^0. The optimal cut must choose the minimum of the above costs, thus minimizing (4.3). 28

5.1 Error rates and mean runtime for GrabCut [9, 51], Dual Decomposition (DD) [59], and our method, denoted by *One-Cut*. 33

5.2 Template shape matching results with or without color separation term: TP, FP, misclassified pixels, and mean running time. 42

Chapter 1

Introduction

Image segmentation is the problem of partitioning the image into several segments. Pixels in the same segment should have similar characteristics, such as intensity, color, texture, etc. On the other hand, pixels in different segments should have distinct characteristic of the same measure. Sometimes image segmentation is a goal in itself. A user might want to segment an object from an image to paste in another image. Most often, segmentation is needed for other vision or medical imaging applications. For example, for medical diagnosis and prognosis of cancer patients, one often needs to accurately measure the tumor volume, which first requires segmenting a tumor from a medical MRF volume. Another application where image segmentation is useful is object detection. Segmentation is useful for feature extraction [53] and to limit the number of image patches to examine for a possible presence of an object [39].

Simply stated, image segmentation can be viewed as a labeling problem where each pixel is assigned a label. There are two labels for segmentation into two regions and multiple labels for segmentation into multiple regions.

Image segmentation can be performed in a supervised or unsupervised fashion. In unsupervised segmentation, no user assistance is available and typically, no additional knowledge about the scene contents is assumed. In supervised segmentation, user specifies either a bounding box containing the object of interest, or so called object and background ”seeds”, indicating some pixels that belong to the object and background, respectively. We may have prior knowledge of segments such as volume ratio of the segments, target distribution of segments learned from a training dataset or shape of the segments. The prior knowledge or user interaction is often incorporated into image segmentation algorithms.

Over the past few decades, numerous algorithms have been developed for image segmentation. Commonly used methods include live-wire [46], deformable models [28], normalized-cut [52], level sets [20, 43], graph cut [12, 14], etc. The focus of this thesis is energy minimization methods for image segmentation. We use the popular and well-known s-t mincut for optimization.

1.1 Markov Random Fields: Modeling and Inference

Markov Random Field (MRF) [10, 33] is a graphical model of joint probabilistic distribution on a set of random variables which are inter-dependent, and their dependences can be modeled with a graph. MRFs have been applied to a wide range of problems in computer vision such as image segmentation, image restoration, 3D reconstruction and image & video Synthesis [21, 38]. We often express a Markov Random Field as a graph $\mathcal{G} = \{V, \mathcal{E}\}$ where V is the set of vertexes and $\mathcal{E}$ is the set of edges. MRF satifisfies the Markov property that the state of one node is independent of all other nodes given the states of its neighboring nodes. For example, in Fig. 1.1, the state of the black node, when conditioned on the four gray neighboring nodes, is independent of all the other graph nodes.

The simplest Markov model- Markov Chain- is defined on a sequence of random variables $X = \{x_1, x_2, ...\}$ over time where the conditional probability of variable x_i only depends on x_{i-1}:

$$P(x_i|x_{i-1}, x_{i-2}, ..., x_1) = P(x_i|x_{i-1}). \tag{1.1}$$

A MRF can be seen as extension of a Markov Chain with higher-order clique (minimum set of connected nodes) and larger connectivity dimension. We often treat each pixel as one node and use 4 or 8 neighboring system as graph for Markov model in computer vision.

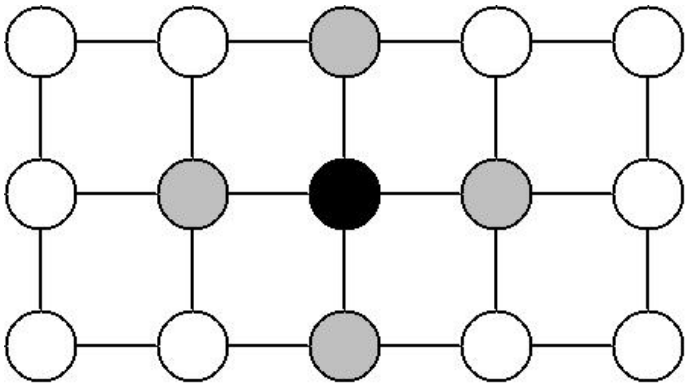


Figure 1.1: Markov Property: The state of the black node is conditionally independent of all the white nodes, given the states of the gray nodes. This is called the Markov property.

A Markov Random Field encodes the long-range correlation between states of variables by simply connecting nodes to a few neighboring nodes. By doing this we avoid densely connected graph which needs computationally expensive inference algorithms and yet capture the essential dependences between the pixels.

A typical 4-connected Hidden MRF Model (HMM) is shown in Fig. 1.2. Let us denote random

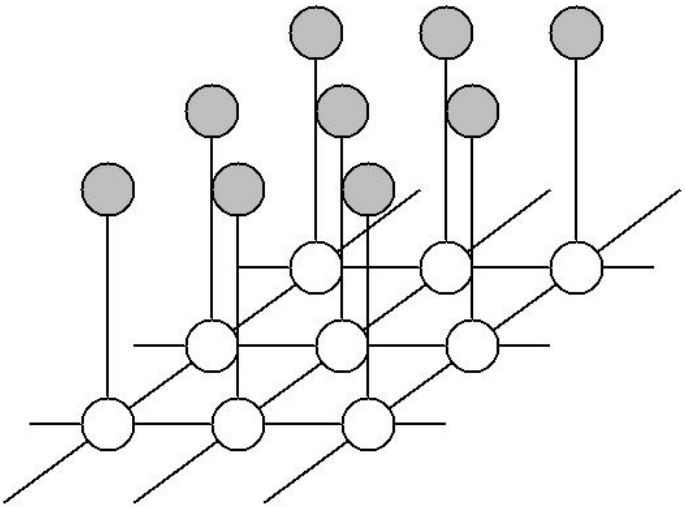


Figure 1.2: Hidden Markov Model on a 4-connected graph. The upper layer represents observed value of nodes and the lower layer represents hidden state variables of nodes.

variables by X and observations of nodes by Z for HMM. Applying Bayes' theorem, we have

$$P\,(X \mid Z) \propto P\,(Z \mid X)\; P\,(X). \tag{1.2}$$

For the likelihood $P(Z \mid X)$, each observed data z_i only depends on the state of hidden variable x_i. The distribution $P(X)$ is assumed to follow the Markov property.
The above posterior MRF is often expressed as energy by taking the log:

$$P\,(X \mid Z) = \frac{1}{\mathcal{Z}} \exp\left(-E(X \mid Z)\right), \tag{1.3}$$

$$E(X \mid Z) = \sum_{c \in C} \theta_c(X) + \sum_i \phi(x_i, z_i), \tag{1.4}$$

where $\mathcal{Z}$ is the normalization factor. Here C is the set of cliques.
Each $\theta_c(X)$ is called a clique potential. A clique contains fully connected subsets of nodes in the graph. The degree of a clique is the number of nodes in the clique. If the degree of cliques is more than three, we treat the MRF as high-order MRF and the corresponding optimization problem is often more challenging. A typical high-order clique is the term defined over number of pixels in segments, for example, term penalizing deviation of segment size from target size. The simplest cliques include one-degree clique and two-degree clique which get involved only one node or two nodes respectively in the clique.
A Conditional Random Field (CRF) directly models the conditional distribution $P(X|Z)$ as obeying the Markov property. In the context of CRF, a latent variable x_i only depends on

its neighboring nodes in the graph given the observed data Z, so the clique potential is of the form $\theta_c(X, Z)$ rather than $\theta_c(X)$. A CRF can be seen as a MRF globally conditioned on the observation. One application of CRF is binary image segmentation with edge-contrast sensitive smoothness term which will be explained later.
Inferrence of posterior MRF is a Maximum A Posteriori (MAP) problem :

$$X = arg\max P(X \mid Z) \tag{1.5}$$

or equivalently we can minimize the energy $E(X \mid Z)$.
The most commonly used MRF energy consists of an unary term and a pairwise term:

$$E(X \mid Z) = \sum_{i \in V} \phi_i(x_i, z_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i, x_j), \tag{1.6}$$

where variable x_i takes value from label set L and z_i is the observation of node i.
Generally, the MAP-MRF energy (1.6) optimization is NP-hard, even for binary case where the variable x_i can only take label 0 or 1. In the binary case, if the pairwise potential satisfies:

$$\theta_{ij}(0,0) + \theta_{ij}(1,1) \leq \theta_{ij}(0,1) + \theta_{ij}(1,0), \forall (i,j), \tag{1.7}$$

then the energy (1.6) is submodular and can be optimized with a graph cut [36]. Intuitively speaking, a submodular energy encourages nearby pixels to have same labels. Boykov and Kolmogorov [14] have developed a min-cut/max-flow algorithm that is particularly efficient in practice for graphs of small connectivity, that naturally arise in image segmentation problems. MRF optimization methods have two important groups: those in discrete and those in continuous domains. Graph cut is a popular discrete domain optimization method that can optimize submodular energy functions. For binary MRF optimization of submodular energy, graph cut gives global optimal solutions in polynomial time. For multi-label MRF when the pairwise term $\theta_{ij}(x_i, x_j)$ is metric or semi-metric, graph cut can find approximate solution by move-making algorithms such as $\alpha-\beta$ swap and α expansion. In each iteration of $\alpha-\beta$ swap and α expansion, the original multi-labeling problem reduces to a binary problem. For $\alpha - \beta$ swap, only nodes with current labels α and β are allowed to change their labels. In particular, they are allowed to change their labels to either α or β. This means that an $\alpha - \beta$ swap move finds an optimal reassignment of labels α and β in the current solution. For α expansion, nodes are only allowed to switch to label α or keep their current labels at each iteration. At each iteration of $\alpha - \beta$ swap (or α expansion), an optimal move finding the maximum energy decrease is found. The

algorithm converges when there is no further $\alpha-\beta$ swap (or α-expansion) move that decreases the energy.

Belief propagation [22, 48, 54, 63] is another important early discrete optimization method for energy (1.6). BP can be seen as re-parameterization of the original energy and gives exact solution on trees, but it gives local minima if there are loops in the graph and may not even converge. BP usually returns an energy which is worse than that of a graph cut.

In the continues optimization domain [2, 18, 29, 37], the MRF optimization problem is written as an integer program (IP). Denote the set of possible labels as L, by relaxing the integration constraints of the integer program, the IP can be further written as the following Linear Program (LP) and solved by LP solver such as interior point methods. However, the solution of LP needs to be rounded and can be far from the optimal solution of IP and LP solver is relatively slow in practice.

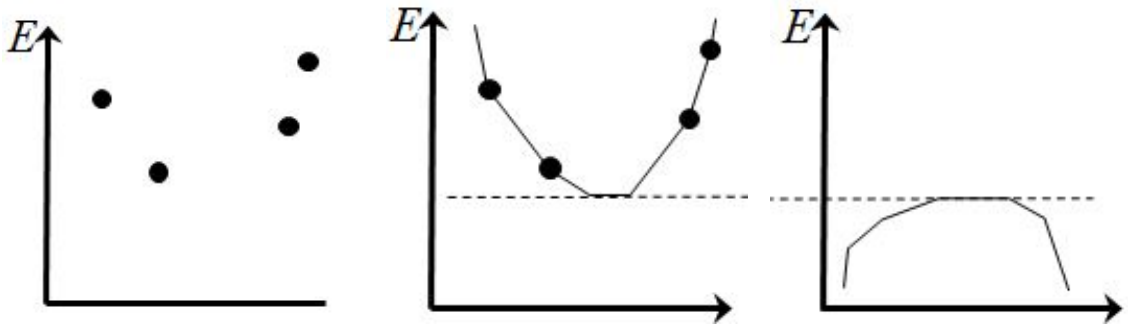
$$\min \sum_{i\in V, x_i\in L} u_i(x_i)\phi_i(x_i) + \sum_{(i,j)\in E, x_i, x_j\in L} u_{ij}(x_i,x_j)\phi_{ij}(x_i,x_j) \tag{1.8}$$

subject to:

$$\sum_{x_i,x_j\in L} u_{ij}(x_i,x_j) = 1, \tag{1.9}$$

$$\sum_{x_i\in L} u_{ij}(x_i,x_j) = u_j(x_j), \tag{1.10}$$

$$u_{ij}(x_i,x_j), u_i(x_i) \in [0,1]. \tag{1.11}$$



(a) Energy (b) LP relaxation (c) Lower bound

Figure 1.3: Take LP relaxation of the original energy with discrete variables and maximize the lower bound.

Tree-reweighted message passing (TRW) [34, 62] maximizes the upper bound of the LP relax-

ation (Fig. 1.3). TRW represents the graph as convex combination of trees. The summation of minimum of trees gives lower bound of the original energy. The two major steps of TRW include performing BP on trees and averaging nodes selected by particular scheme. TRW iterates between the two steps until convergence, which gives local maximum of lower bound. A nice property of TRW-S [34] is that the lower bound never decreases.

1.2 Markov Random Fields for Image Segmentation

S-t maxflow/mincut is first used by Greig et al. [26] as optimization algorithm for computer vision and image processing. There it is used for the task of binary image reconstruction from noisy images. Then Boykov and Jolly first employed graph cut for image segmentation [11, 12, 13, 14]. Bellow we show an example of graph cut for binary interactive image segmentation.

We denote $s_p \in \{0, 1\}$ as binary indicator variables for pixel p, 1 for foreground and 0 for background. The most commonly used single-variable potential is log-likelihood term $\ln \Pr(I_p|\theta^{s_p})$ for each pixel p, where θ^1 and θ^0 are fixed foreground and background appearance models, usually based on color distributions. I_p is the color of pixel p. In the case of interactive segmentation, we can estimate the initial appearance model through user input strokes. Sometimes the appearance model of foreground and background are known a priori, for example from a training set. Commonly used pairwise potential is edge-contrast sensitive smoothness penalty. The higher the intensity contrast between two adjacent pixels, the smaller the smoothness penalty.

The basic object segmentation energy [12, 57] combines boundary length regularization ∂S with log-likelihood term

$$E(S|\theta^1, \theta^0) \;=\; -\sum_{p\in\Omega} \ln \Pr(I_p|\theta^{s_p}) \;+\; |\partial S| \tag{1.12}$$

where Ω is the set of all image pixels and S is the set of foreground pixels labelled as $s_p = 1$. The most commonly used boundary length regularization term is $|\partial S| = \sum_{\{p,q\}\in N} \omega_{pq}|s_p - s_q|$ and N is the set of all pairs of neighboring pixels. If ω_{pq} is constant, then the smoothness term is data independent and the model is MRF. If ω_{pq} is edge-contrast sensitive, then the smoothness term depends on the observed data and the model becomes a CRF. The log-likelihood term, or data term is a unary term, and the smoothness term is a pairwise term. A real example of interactive image segmentation is shown in Fig. 1.5

Example: Interactive image segmentation with graph cut (Fig. 1.4)

1. The user specifies hard-constrained pixels that have to be segmented as foreground or background. For example, the user can put strokes on object and background. We can estimate foreground and background appearance model (color histogram) accrording to the strokes.

2. A four or eight connected graph is constructed with each node representing each pixel and there are two additional terminal nodes in the graph: source node S and sink node T.

3. Then we connect nodes in the graph through links. The links between pixels and terminals are denoted by t-links and links between pixels themselves are denoted by n-links. The hard-constrained pixels are linked to terminal nodes S or T with infinity edge weights. Other pixels are linked to terminal nodes through soft-constrain t-link, the weight of which will be explained later. One way of setting soft-constrain t-link is through appearance model of foreground and background.

4. Adjacent pixels in neighboring system are connected through n-links. The weight of the smoothness term can be edge-contrast sensitive.

5. After the graph is constructed, we can use any available maxflow/mincut optimization algorithm and get the cut of minimum weight. The min cut specifies whether the pixels belong to foreground and background.

1.3 Motivation of Color Separation Term for Image Segmentation

Appearance models are critical for many image segmentation algorithms. One important practical advantage of this basic energy is that there are efficient methods for their global minimization using graph cuts [14] or continuous relaxations [16, 50].
In many applications the appearance models may not be known *a priori*. Some well-known approaches to segmentation [17, 51, 66] consider model parameters as extra optimization variables in their segmentation energies. E.g.,

$$E(S, \theta^1, \theta^0) \;=\; -\sum_{p\in\Omega} \ln \Pr(I_p|\theta^{s_p}) \;+\; |\partial S|, \tag{1.13}$$

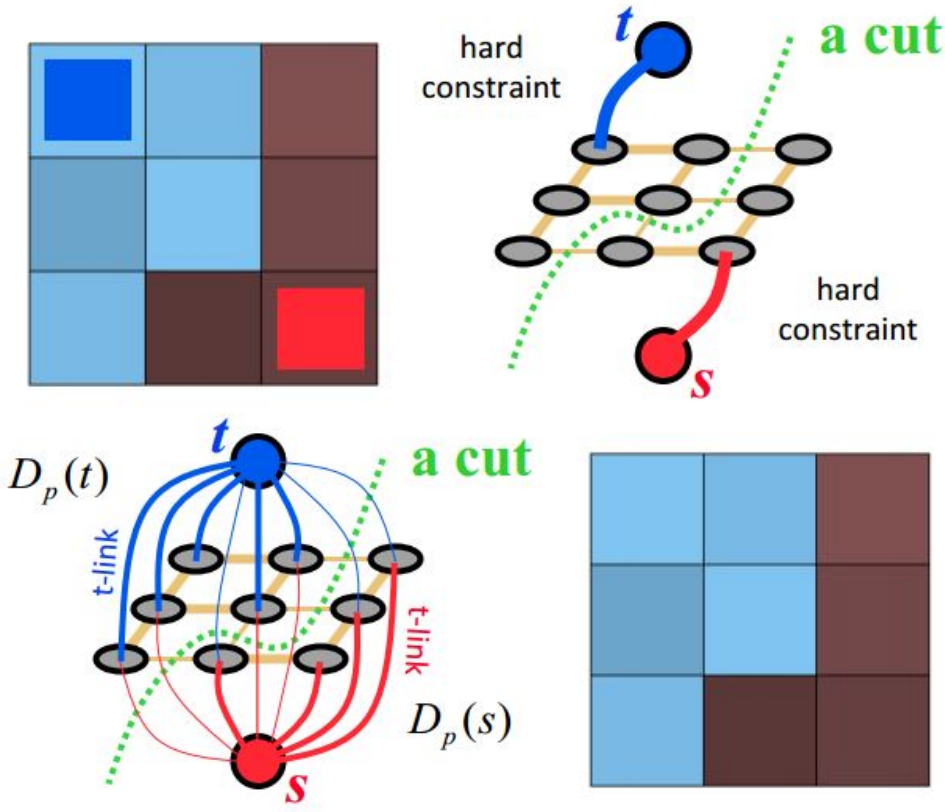


Figure 1.4: Graph Cut for Interactive Segmentation: (top-left) User specified seeds denoted by blue and red pixels, (top-right) Graph with pixels as nodes and S, T terminals, (bottom-left) Graph with t-links and n-links, (bottom-right) s/t mincut. [Image credit: Yuri Boykov]

which is known to be NP-hard for optimization [59], is used for interactive segmentation in GrabCut [9, 51] where initial appearance models θ^1, θ^0 are computed from a given bounding box. The most common approximation technique for minimizing (1.13) is a block-coordinate descent [51] alternating the following two steps. First, they fix model parameters θ^1, θ^0 and optimize over S, e.g. using a graph cut algorithm for energy (1.12) as in [12]. Second, they fix segmentation S and then optimize over model parameters θ^1 and θ^0. Two well-known alternatives, *dual decomposition* [59] and *branch-and-mincut* [40], sometimes find a global minimum of energy (1.13), but these methods are too slow in practice. Please refer to **Chapter 3** for detailed description on GrabCut, *dual decomposition* and *branch-and-mincut*.

We observe that when appearance models θ^1, θ^0 are represented by (non-parametric) color

Figure 1.5: Interactive image segmentation: user provided strokes (Left) and result (right).

histograms, we can rewrite (1.13) as :

$$\begin{aligned} E(S,\theta^1,\theta^0) &= -\sum_{s_p=1} \ln \Pr(I_p|\theta^1) \; - \sum_{s_p=0} \ln \Pr(I_p|\theta^0) \; + \; |\partial S| \\ &= -\sum_k n_k^S \ln \theta_k^1 \; - \sum_k n_k^{\bar{S}} \ln \theta_k^0 \; + \; |\partial S| \\ &= -|S| \sum_k \theta_k^S \ln \theta_k^1 \; - |\bar{S}| \sum_k \theta_k^{\bar{S}} \ln \theta_k^0 \; + \; |\partial S| \\ &= |S| \cdot H(\theta^S|\theta^1) + |\bar{S}| \cdot H(\theta^{\bar{S}}|\theta^0) + |\partial S|, \end{aligned} \tag{1.14}$$

where n_k^S is the number of pixels in k^{th} color bin in foreground and $n_k^{\bar{S}}$ in background. Here θ^S and $\theta^{\bar{S}}$ are histograms inside object S and background $\bar{S} = \Omega \setminus S$. $H(\theta^S|\theta^1)$ and $H(\theta^{\bar{S}}|\theta^0)$ are cross entropies of probability distributions. According to well-known cross entropy inequality $H(\theta^S|\theta^1) \geq H(\theta^S)$ where $H(\cdot)$ is the *entropy* functional for probability distributions, minimization of (1.13) is equivalent to minimization of energy

$$E(S) \; = \; |S| \cdot H(\theta^S) + |\bar{S}| \cdot H(\theta^{\bar{S}}) + |\partial S| \tag{1.15}$$

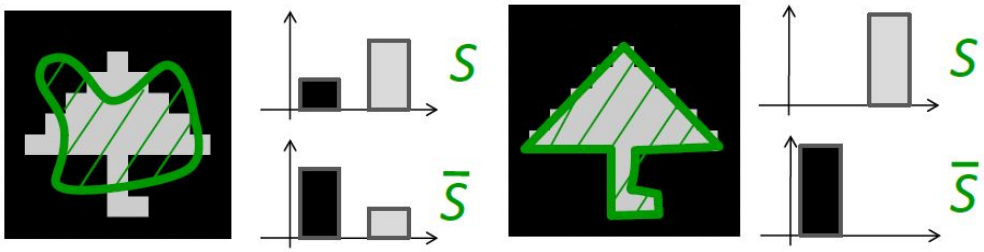
that depends on S only. Interestingly, the global minimum of segmentation energy (1.15) does not depend on the initial color models provided by the user. Thus, the interactivity of GrabCut algorithm is primarily due to the fact that its solution is a local minimum of (1.15) sensitive to the initial bounding box.

If we ignore the smoothness term and use a sliding window to find the bounding box that minimizes energy (1.15), we would get a box that splits the object of distinct appearance from the background in the image. See Fig. 1.6 for example. We used integral image [60] which is originally used for face detection to help accelerate the optimization.

Formulation (1.15) is useful for analyzing the properties of energy (1.13). The entropy

Figure 1.6: If we only optimize the first two terms in (1.15) and restrict the foreground to be a rectangle, we can get a rough bounding box of foreground and background in sliding-window fashion. The foreground is indicated by red box.

terms of this energy prefer segments with more peaked color distributions that give lower entropy. Intuitively, this should also imply that the optimal foreground and background distributions have a small overlap. For example, consider a simple case of black-&-white image when color histograms θ^1 and θ^0 have only two bins (Fig. 1.7). Clearly, the lowest value (zero) for the entropy terms in (1.15) is achieved when black and white pixels are completely separated between the segments, e.g. all white pixels are inside the object and all black pixels are inside the background.



(a) High entropy example (b) Low entropy example

Figure 1.7: Color separation gives segments with low entropy.

The intuitive observation that separating pixels of the same color into different segments renders segments with low entropy can also be derived by analytically rewriting the energy [59].

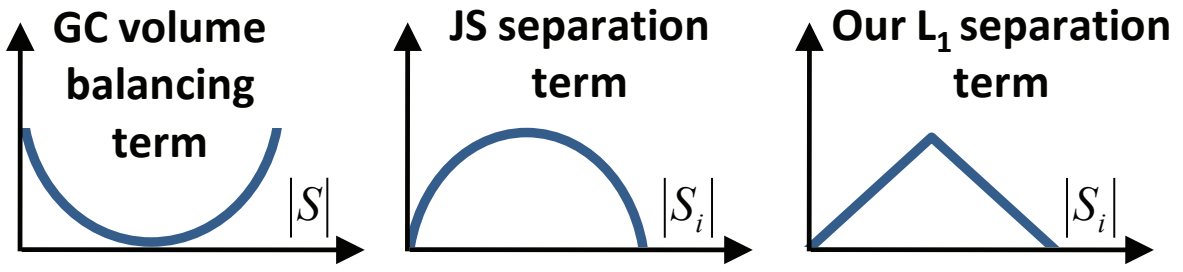
We can further rewrite energy (1.15) as:

$$\begin{aligned} E(S) &= -|S| \sum_k \theta_k^S \ln \theta_k^S \; - |\bar{S}| \sum_k \theta_k^{\bar{S}} \ln \theta_k^{\bar{S}} \; + \; |\partial S| \\ &= -\sum_k n_k^S \ln \theta_k^S \; - \sum_k n_k^{\bar{S}} \ln \theta_k^{\bar{S}} \; + \; |\partial S| \\ &= -\sum_k n_k^S \ln \frac{n_k^S}{|S|} \; - \sum_k n_k^{\bar{S}} \ln \frac{n_k^{\bar{S}}}{|\bar{S}|} \; + \; |\partial S| \\ &= |S| \ln S \; + |\bar{S}| \ln |\bar{S}| - \sum_k (n_k^S \ln n_k^S + n_k^{\bar{S}} \ln n_k^{\bar{S}}) \; + \; |\partial S| \end{aligned} \tag{1.16}$$

So the color separation bias in energy (1.15) is shown by equivalently rewriting its two entropy terms as

$$h_\Omega(S) - \sum_i h_{\Omega_i}(S_i) \tag{1.17}$$

where $h_A(B) = |B| \cdot \ln |B| + |A \setminus B| \cdot \ln |A \setminus B|$ is standard Jensen-Shannon (JS) divergence functional for subset $B \subset A$. We also use Ω_i to denote the set of all pixels in color bin i (note $\Omega = \cup_i \Omega_i$) and $S_i = S \cap \Omega_i$ is a subset of pixels of color i inside object segment (note $S = \cup_i S_i$). The plots for functions $h_\Omega(S)$ and $-h_{\Omega_i}(S_i)$ are illustrated in Fig.1.8.



(a) $h_\Omega(S)$ (b) $-h_{\Omega_i}(S_i)$ (c) $-|S_i - \bar{S}_i|$

Figure 1.8: Energy (1.15): volume balancing (a) and Jensen-Shannon color separation terms (b). Our L_1 color separation term (c).

The first term in (1.17) shows that energies (1.13) or (1.15) implicitly bias image segmentation to two segments of equal size, see Fig.1.8(a). The remaining terms in (1.17) show bias to color separation between the segments, see Fig.1.8(b). Note that a similar analysis in [59] is used to motivate their convex-concave approximation algorithm for energy (1.13).

Relation with Normalized Cuts: The combination of color separation term and volume balancing term is analogous to Normalized Cuts [52]. In Normalized Cuts, the graph partition criteria is given by:

$$Ncut\ (A,\ B) \;=\; \frac{cut(A,\ B)}{Vol(A)} + \frac{cut(A,\ B)}{Vol(B)} \tag{1.18}$$

where $cut(A,\ B)$ is the sum of weights of connections between groups A and B and $Vol(A)$ is the total weight of the edges originating from group A and $Vol(B)$ is similarly defined. The term $cut(A,\ B)$ plays a similar role as smoothness term $|\partial S|$ that minimizes the boundary length. If there's only the term $cut(A,\ B)$ in the energy, then trivial solutions of $A = \emptyset$ or $B = \emptyset$ would be global optimal solutions. The volume terms $Vol(A)$ and $Vol(B)$ have the same effects as volume balancing term here $h_\Omega(S)$ that prefers balanced foreground and background. Note that normalized cut does not have a color separation term. The lack of color separation can lead to significant artifacts in segmentation, where volume balancing plays too much of a role. This often results in segments that are almost equal in volume, but perceptually not distinct.

Volume balancing $h_\Omega(S)$ is the only term in (1.17) and (1.13) that is not submodular and makes optimization difficult. Our observation is that in many applications this volume balancing term is simply unnecessary [55], see Sections 5.1.3, 5.2-5.3. In other applications we propose to replace it by other easier to optimize terms.

Moreover, it is known that JS color separation term $-h_{\Omega_i}(S_i)$ is submodular (any concave function of cardinality (number of pixels in segment) is submodular [42]. This applies to JS, χ^2, Bhattacharyya, and our L_1 color separation terms in Figs.1.8, 5.5.), so it can be optimized by graph cuts [30, 31, 59]. We propose to replace it with a simpler L_1 separation term [55] in Fig.1.8(c). We show that it corresponds to a simpler construction with fewer auxiliary nodes leading to higher efficiency while capturing the essence of a more general color separation term. Interestingly, it also gives better color separation effect in practice for some applications, see Section 5.1.2. A Bhattacharyya gradient flow driven active contour can also maximize the discrepency between distribution of regions inside and outside the active contour [45], but optimization of the level set energy is very slow.

We also observe one practical limitation of block-coordinate approach to (1.13), as in GrabCut [9, 51], could be due to increased sensitivity to local minima when the number of color bins for models θ^S and $\theta^{\bar{S}}$ is increased, see **Section 3.1**, Table 5.1 and Fig.5.1. The reason is that with more color bins, the dimensionality of the histogram gets larger, and there are more local minimums of the energy function. This is because there are more histogram-based color models for foreground and background that result in a good color separation. In practice, however, finer bins better capture the information contained in the full dynamic range of color images (8-bit per channel or more). Our ROC curves show that even a difficult camouflage image in Figure 1.9 has a good separation of intensities between the object and background if larger number of bins is used. With 16^3 bins, however, the overlap between the "fish" and the background is too strong making it hard to segment. Since GrabCut algorithm is more likely to get stuck at weak local minima for larger number of bins, it may not benefit from higher color resolution.

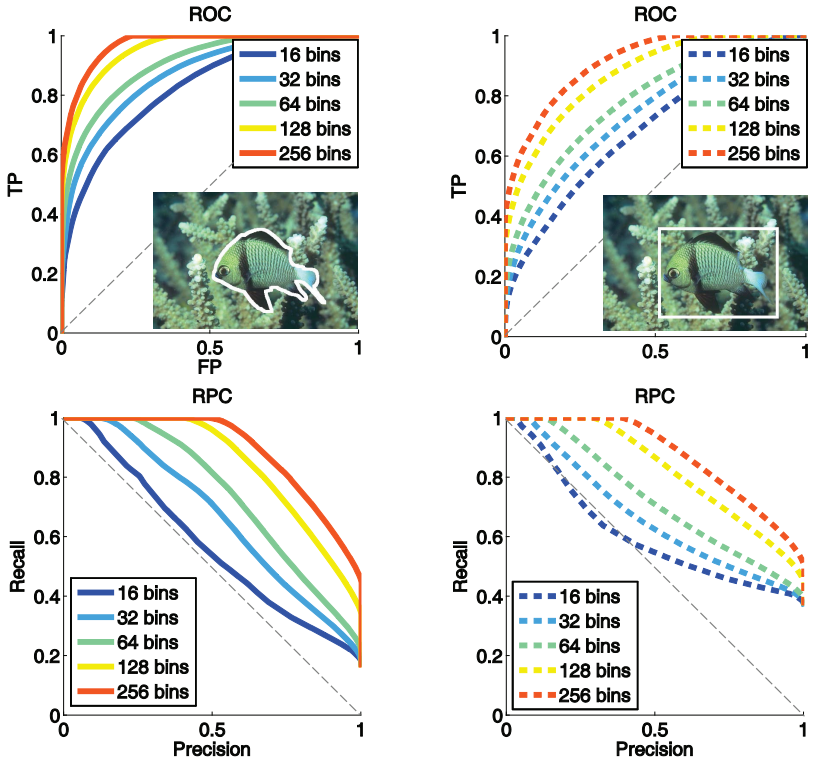


Figure 1.9: Given appearance models θ^o, θ^b extracted from the ground truth object/background segments (white contour, top-left), we can threshold log-likelihood ratios $\ln \frac{\theta^o(I_p)}{\theta^b(I_p)}$ at each pixel p and compare the result with the same ground truth segmentation for different thresholds. The corresponding ROC (top-left) curves and RPC curves (bottom-left) show that the color separation between the object and background increases for finer bins. The same procedure is repeated for an arbitrary chosen rectangle within the same image (top-right, bottom-right) with far less pronounced improvement. It is clearly seen that using higher number of bins to represent appearance can help separate objects from the background even in the case of camouflage images.

1.4 Contribution of the Thesis

The contribution of the thesis is summarized as follows:

- We propose a simple energy term penalizing L_1 measure of appearance overlap between segments. While it can be seen as a special case of a high-order *label consistency* term introduced by Kohli et al. [30, 31] we propose a simpler construction for our specific constraint. Unlike NP-hard multi-label problems discussed in [30, 31], we focus on binary segmentation where such high-order constraints can be globally minimized. Moreover, we show that our L_1 term works better for separating colors than other concave separators (including JS, Bhattacharyya, and χ^2).

- We are first to demonstrate fast globally optimal binary segmentation technique explicitly minimizing overlap between un-normalized object/background color histograms. In one graph cut we get similar or better results at faster running times w.r.t. earlier methods, e.g. [19, 40, 51, 59].
- We show general usefulness of the proposed appearance overlap penalty by showing different practical applications: binary segmentation, shape matching, etc.

1.5 Outline of the Thesis

The thesis is organized as follows: **Chapter 2** is an overview of appearance models for segments, including non-parametric density estimation such as Parzen window and k-NN density estimation and parametric density estimation Gaussian mixture model. In **Chapter 3** realated work of *GrabCut*, *branch-and-mincut*, *dual decompostion* and *active contour* is analysed and limitations of these approaches are shown. In **Chapter 4** our proposed L_1 color separation term is introduced, we also explain the relationship bettween L_1 color separation term and general color separation term. We show the graph construction for minimizing these color separation terms. Furthermore we explain the difference of our L_1 color separation term from P^n Potts model. **Chapter 5** presents several applications of our color separation term. We apply the color separation term to segmentation with bounding box or seeds, shape matching with a simple template and salient object segmentation. Our algorithm based on color separation term outperforms the state of the art. **Chapter 6** concludes the thesis by pointing out several promising directions of future work.

Chapter 2

Overview of Appearance Models

An appearance model is a model of distribution of intensity, color, texture, shape, etc. inside a segment. In this thesis, we model appearance based on the color feature. In particular, we use the RGB color space representation. The separation term for the color model can be easily generalized to other appearance models. One simply has to quantize the features that are being used into an appropriate number of bins. In this chapter we start with the simplest color model, namely a color histogram and further introduce non-parametric techniques including Parzen window and k-NN. Finally, we discuss the Gaussian Mixture Model (GMM).

2.1 Histogram

One way to view a histogram is as a graphical presentation of data distribution. First, the range of all possible feature values is divided into ”bins”, usually at uniform intervals. A histogram then simply counts how many pixels are in each bin. The simplest and most commonly used color model is a histogram over all unique colors. That is, each color gets its own bin. Thus in this case, a histogram simply counts how many pixels of each unique color are there in a segment. In this thesis, we used the RGB color space, which is an additive color space based on RGB color model. We can also have histograms for other color spaces such as LAB color space, where L stands for intensities and A, B stand for color opponent dimensions. In the RGB color space, each color is represented by a 3-dimensional the RGB feature vector. The number of colors in RGB color space is commonly 256^3.

Histograms of colors simply count the number of points in each color bin and normalize the number by number of sample points (Fig. 2.1). In this thesis we experiment with dividing each color channel (R, G or B) into 1, 2, 4, 8, 16, and 32 equal intervals. This gives us, respectively, 1, 2^3, ..., 32^3 distinct bins in the histogram. The problem with color binning is

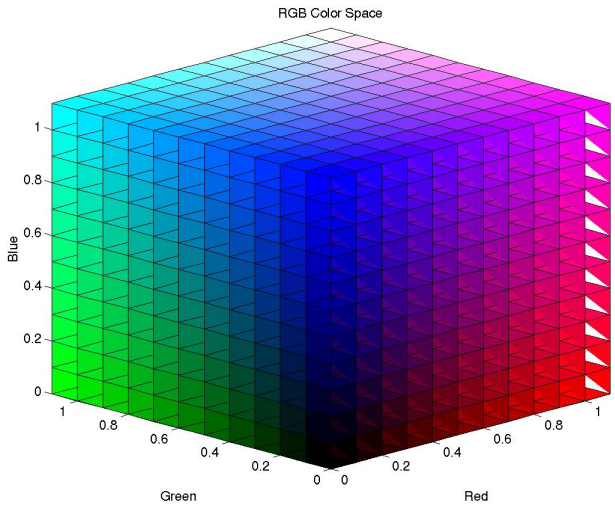


Figure 2.1: Binning in RGB color space [1]

that similar colors may fall into different color bins thus the affinity between similar colors is not completely preserved. If we take a smaller bin size, the histogram may not have enough samples per bin, resulting in an unreliable appearance model. To fix the problems of quantized histograms, non-parametric density estimation and Guassian mixture models are often used. These give a smoother distribution with no artifacts due to hard decisions made when deciding how to bin a histogram.

2.2 Non-parametric Density Estimation

The goal of non-parametric density estimation is to estimate the probability distribution that generated given training samples, given only a limited number of training samples n. Non-parametric techniques can be used for estimation of samples coming from any distribution. The probability density at sample point x is estimated as:

$$P(x) \approx \frac{k/n}{V}$$

where n is the number of training samples, V is the volume of region R around point x and k is the number of points inside region R. There are two commonly used non-parametric techniques, Parzen window and k-Nearest Neighbor (k-NN).
For Parzen window, the region size V is fixed, so the number of points k differs for different x.

[1] https://www.clear.rice.edu/elec301/Projects02/artSpy/color.html

Instead of counting the number of points inside region R, we can also apply a kernel function, most often a Gaussian kernel, to weigh each sample in proportion to its distance from x. The k-NN method takes an opposite approach, namely the number of neighboring points k is fixed and what's changing is region size V. Fig. (2.2) shows the windows for Parzen and k-NN.

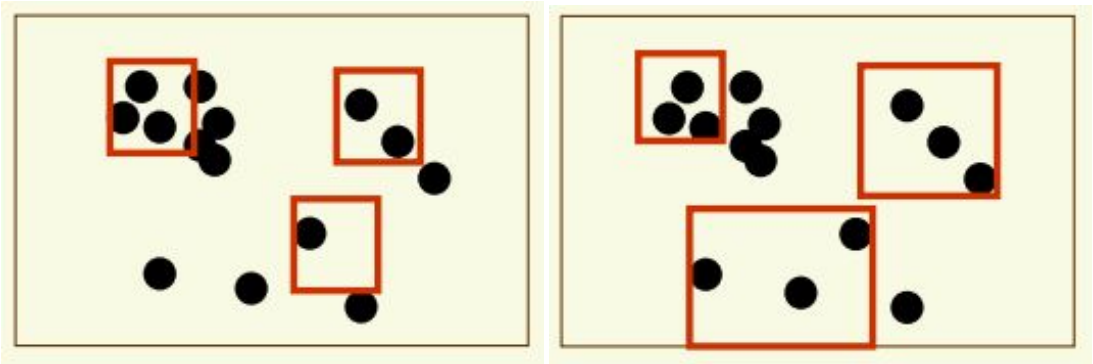


Figure 2.2: Window size is fixed for Parzen window (left) and number of neighboring points is fixed for k-NN(right). Window size or number of neighboring points should be chosen properly to get a good density estimation. © Olga Veksler.

For Parzen window, if the window size is too small, the resulting density estimation is very noisy, giving us similar undesirable results as histograms with small bin size. If the window size is too large, each sample would affect many other samples' density estimation and the distribution is over-smoothed. It's not easy to select a proper window size for the Parzen window technique.
Analogously, for k-NN density estimation, we have to choose the appropriate number of neighboring pixels. What's more, finding the k nearest neighbors such as via Voronoi diagram [3] increases the computational cost. In general, if we have enough training samples and choose a proper window size or number of neighboring pixels, non-parametric density estimation is better than histograms because it can be shown to approach the true distribution of the samples.

2.3 Gaussian Mixture Model

The Gaussian Mixture Model [8, 9, 44] is a parametric probability density function based on weighted sum of Gaussian components. It maximizes the likelihood of the training samples given the model. Suppose we have K Gaussian components indexed by $1, 2, ..., K$, the Gaussian mixture model for 3-dimensional RGB color feature $\overrightarrow{c} = (r, g, b)$ can be represented as:

$$Pr(\overrightarrow{C} = \overrightarrow{c}) \propto \sum_{k=1}^{K} w_k \cdot N(\overrightarrow{c} \mid \overrightarrow{\mu}_k, \overrightarrow{\sigma}_k), \tag{2.2}$$

where w_k is the weight of the k^{th} component, means are

$$\overrightarrow{\mu}_k = (\mu_k^r, \mu_k^g, \mu_k^b) \tag{2.3}$$

standard deviations are

$$\overrightarrow{\sigma}_k = (\sigma_k^r, \sigma_k^g, \sigma_k^b) \tag{2.4}$$

and

$$\begin{aligned} N(\overrightarrow{c} \mid \overrightarrow{\mu}_k, \overrightarrow{\sigma}_k) &= N(r \mid \mu_k^r, \sigma_k^r) \cdot N(g \mid \mu_k^g, \sigma_k^g) \cdot N(b \mid \mu_k^b, \sigma_k^b) \\ N(i \mid \mu_k, \sigma_k) &= \frac{1}{\sqrt{2\pi}\,\sigma_k} e^{-\frac{(i-\mu_k)^2}{2\sigma_k^2}} \end{aligned} \tag{2.5}$$

Note that we assume a diagonal covariance matrix in this section for simplicity, but the case of the full covariance matrix is very similar. Suppose we have N pixels in the image domain Ω. Given training samples, for example all the pixels in one segment, we wish to find the parameters for GMM that maximize the following likelihood:

$$P(I \mid G) = \prod_{p=1}^{N} Pr(\overrightarrow{c_p} \mid G) \tag{2.6}$$

where G is the set of parameters for Gaussian mixture modes including weights w_k, means $\overrightarrow{\mu}_k$ and standard deviations $\overrightarrow{\sigma}_k$ for $k = 1, 2, ..., K$. $\overrightarrow{c}_p = (r_p, g_p, b_p)$ is the color of pixel p. The parameters can be estimated through Expectation-Maximization (EM) algorithm. We can get the initial Gaussian mixture model through k-means algorithm [27]. Then we iterate between E-step and M-step 10-20 times or until convergence.

Procedure: Iterate between E-step and M-step until convergence.

E-step: For each pixel p, compute the probability that its color $\overrightarrow{c}_p$ belongs to the k^{th} component.

$$\phi_p^k = \frac{w_k \cdot N(\overrightarrow{c_p} \mid \overrightarrow{\mu}_k, \overrightarrow{\sigma}_k)}{\sum_{j=1}^{K} w_j \cdot N(\overrightarrow{c_p} \mid \overrightarrow{\mu}_j, \overrightarrow{\sigma}_j)} \tag{2.7}$$

M-step: Simutaneourly update parameters w_k, $\overrightarrow{\mu}_k = \mu_k^{r,g,b}$ and $\overrightarrow{\sigma}_k = \sigma_k^{r,g,b}$ for all Gaussian

components $k = 1, 2, ..., K$.

$$\begin{aligned} w_k &= \frac{1}{N}\sum_{p=1}^{N}\phi_p^k, \\ \overrightarrow{\mu}_k &= \frac{\sum_{p=1}^{N}\overrightarrow{c_p}\cdot\phi_p^k}{\sum_{p=1}^{N}\phi_p^k}, \\ (\sigma_k^{r,g,b})^2 &= \frac{\sum_{p=1}^{N}(c_p^{r,g,b}-\mu_k^{r,g,b})^2\cdot\phi_p^k}{\sum_{p=1}^{N}\phi_p^k}. \end{aligned} \tag{2.8}$$

Here we assume the covariance matrix of the Guassian components to be diagonal, which implies there is no correlation between the R,G and B channels of RGB color space. This assumption is acceptable for the description of image appearance model. The major configuration of Gaussian mixture model is the number of components. Here we usually set 10-30 components for images. Fig. 2.3 gives an example of estimating a Gaussian mixture model with 3 components in the RGB color space. The three components are denoted by ellipses centered at the mean vectors. The scale of the ellipses represents the scale of covariance of Gaussian components.

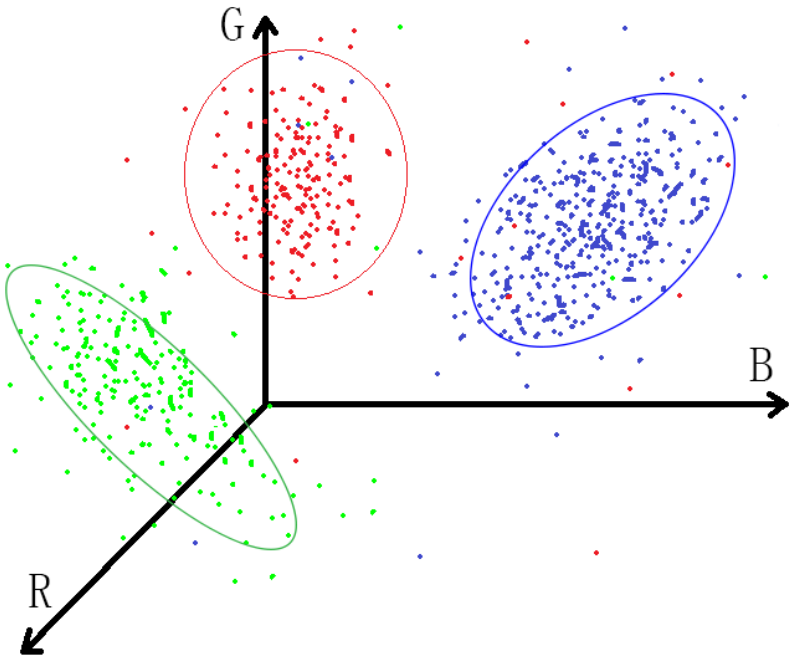


Figure 2.3: Gaussian Mixture Model in RGB color space. In this example we have three Gaussian mixture components highlighted by three ellipses.

Chapter 3

Related Work

In this chapter we review segmentation methods that address the problem of joint optimization $E(S, \theta^1, \theta^0)$ in (1.13) over appearance models and segmentation. The following methods are discussed in detail: GrabCut [9, 51], Branch-and-Mincut [40], Dual Decomposition [59] and active contour [45]. We talk about the limitations of these works that motivate our approach.

3.1 GrabCut

GrabCut [9, 51] is a commonly used method for interactive foreground segmentation. An extension of GrabCut has been shipped into Microsoft Office 2010. A traditional way of user interaction is through bounding box provided by the user. The method is iterative where at each iteration there are two steps: (1) Segmentation via maxflow algorithm given fixed appearance models θ^1 and θ^0; (2) Re-estimation the appearance models based on current segmentation. Appearance can be modeled with either histograms or Gaussian mixture models (GMM). GrabCut is an upper bound optimizer of the energy (1.13) because the following inequality holds:

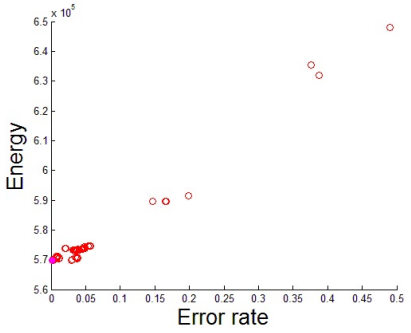
$$E(S \mid \theta^{S_0}, \theta^{\bar{S_0}}) \geq E(S \mid \theta^{S}, \theta^{\bar{S}}), \ \forall S_0 \tag{3.1}$$

At iteration with current solution S_0, GrabCut takes the upper bound of the original energy. The energy is guaranteed to decrease at each iteration.

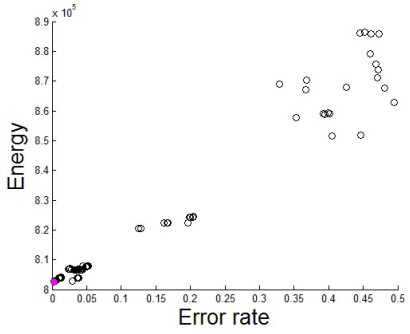
The GrabCut method can be seen as a block-coordinate descend and as such is prone to local minimum. This problem is especially prominent when the number of parameters used to model appearance is high, which is confirmed empirically in our experiments (see Fig. 3.1). We randomly select box as initial solution. Region inside the box is taken as foreground and outside as background. We run block-coordinate descent until convergence and do this experiment for

500 randomly generated boxes. For the 500 solutions we get, we compute their error rates and energy. As we can see from the scatter plots, when the number of color bins increases, there are more distinct solutions. This implies there are more local minima with more color bins and GrabCut is more prone to getting stuck in local minima.

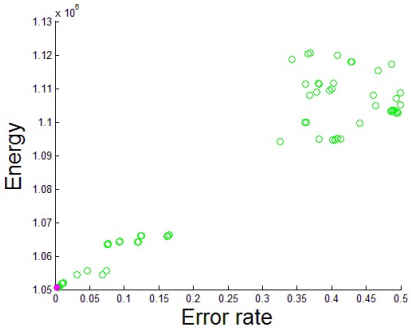
(a) Original Image



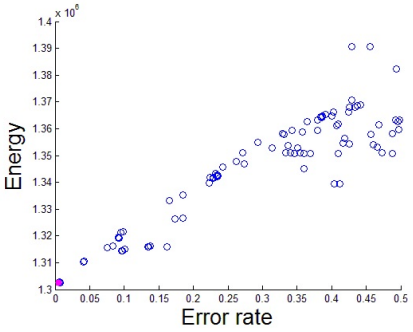
(b) 16 bins per channel



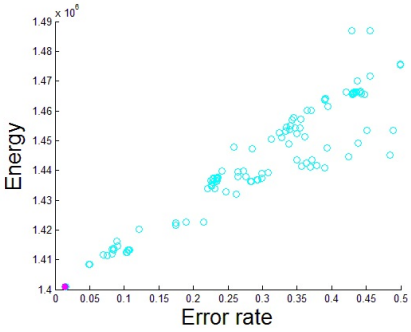
(c) 32 bins per channel



(d) 64 bins per channel



(e) 128 bins per channel



(f) 256 bins per channel

Figure 3.1: Scatter plots of energies versus error rates for different number of bins per channel. We randomly select box as initial solution and run block-coordinate descent until convergence. We perform the experiment for 500 times and plot the energies versus error rates. The pink dot shows the error rate and energy of ground-truth solution.

3.2 Branch-and-Mincut

Branch-and-Mincut [40] combines two powerful techniques: Graph cut and Branch-and-Bound. It can find global optimal of the energy (1.13), rather than local minima when using EM-style GrabCut (Fig. 3.2). Branch-and-Bound is a popular optimization technique for combinatorial optimization and discrete optimization. The basic idea of Branch-and-Bound is to divide the space of all solutions into subsets and obtain a lower bound for each subset. Whenever the lower bound of some subset is greater than the function value of current best solution, we can prune those subsets and the search space is reduced. The search space is split, bounded and pruned until only one solution is left, guaranteed to be the global optimum.

Figure 3.2: Branch-and-Mincut [40] can find smaller global energy while GrabCut gets stuck in local minima with larger energy.

For image segmentation with Graph cut, the search space is divided into subsets based on the parameters of the appearance models. The lower bound of the appearance model subset can be computed using a single run of maxflow algorithm.

If we use K bin color histogram as appearance model, the size of the search space for foreground and background appearance models would be 2^{2K}. The time complexity of Branch-and-Mincut is exponential with respect to the number of color bins K. While finer color histograms can better describe appearance models, Branch-and-MinCut method cannot be used due to exponential complexity.

Note that Branch-and-Mincut is not limited to optimization acceleration for choosing better appearance model. It can also be applied to a wider range of graph cut problems as long as the graphs are parameterized and have similar structure. For example, the shape matching problem with a simple binary shape template in Sec. 5.2 can also be accelerated by Branch-and-Mincut.

3.3 Dual Decomposition

Vincente et al. [59] proposed dual decomposition method to optimize the energy (1.13). First, the energy is rewritten as:

$$E(S) = -\sum_i h_{\Omega_i}(S_i) + |\partial S| - < \mathbf{y}, S > + h_\Omega(S) + < \mathbf{y}, S > \tag{3.2}$$

where $\mathbf{y} \in R^N$ is a vector, $N = |\Omega|$ and $< \cdot, \cdot >$ is the dot product between two vectors. The following $\Phi(\mathbf{y})$ gives a lower bound of $E(S)$:

$$\Phi(\mathbf{y}) = \min_S[-\sum_i h_{\Omega_i}(S_i) + |\partial S| - < \mathbf{y}, S >] + \min_S[h_\Omega(S) + < \mathbf{y}, S >]. \tag{3.3}$$

It suffices to consider $\mathbf{y} = \lambda \cdot \mathbf{1}$ where λ is a scalar and $\mathbf{1}$ is a unit vector. So we can also rewrite the original energy as

$$E(S) = -\sum_i h_{\Omega_i}(S_i) + |\partial S| - \lambda < 1, S > + h_\Omega(S) + \lambda < 1, S > . \tag{3.4}$$

We denote $-\sum_i h_{\Omega_i}(S_i) + |\partial S| - \lambda < 1, S >$ as $E^1(S, \lambda)$ and $h_\Omega(S) + \lambda < 1, S >$ as $E^2(S, \lambda)$, then we have

$$\phi(\lambda) = arg\min_S E^1(S, \lambda) + arg\min_S E^2(S, \lambda) \leq E(S). \tag{3.5}$$

$\phi(\lambda)$ renders a lower bound of $E(S)$ and is called the dual function of E(S). We can explore all values of λ to get the tightest lower bound. In order to optimize over λ, Vicente et al. [59] proposed using parametric maxflow, which is very slow in practice. If there is no discrepancy between the lower bound at the optimal λ and the original energy for the corresponding S, we obtain a global optimal solution. The final labeling is chosen among all solutions according to the original energy. Dual decomposition is very slow in practice because to explore all breakpoints via parametric-maxflow is slow.

3.4 Active Contour

Michailovich et al. in [45] proposed an active contour method that maximizes the Bhattacharyya distance between foreground and background color distributions. The energy functional is based on Bhattacharyya distance. Gradient flow of the energy is used to drive the

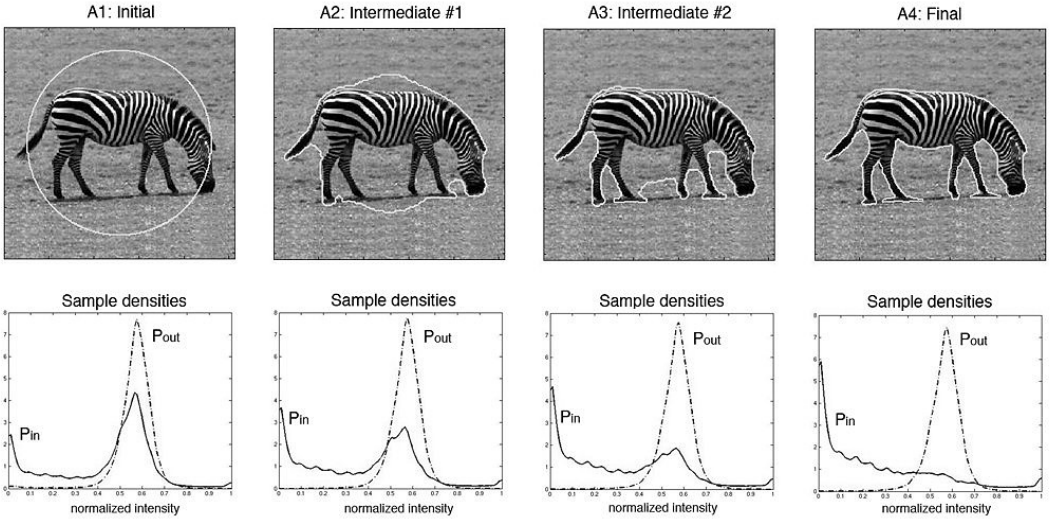


Figure 3.3: (top row) Intensity-based segmentation of Zebra at each iteration. (bottom row) Corresponding intensity distributions of Zebra and its background. *Pin* and *Pout* are intensity distributions of regions inside and outside the contour. [45]

evolution of the active contours at each iteration. Fig. 3.3 illustrates intermediate active contours at different iterations and the corresponding foreground and background distributions. As we can see, the final segmentation yields large discrepancy between foreground and background appearance distributions. The level set based method does not guarantee global optimal solution and is very slow in practice.

Chapter 4

Minimizing Appearance Overlap in One-cut

In this chaper we introduce the L_1 color separation term and show how it can be optimized in one graph cut. We also talk about general color separation term that is not based on L_1 metric. Particularly, we address the difference between our color separation term and P^n potts model which was originally proposed for enforcing labeling consistency within superpixels. Note that the color separation terms here are all used for color histogram appearance model. In this chapter, color separation terms are formulated over color histograms. **Chapter 6** shows possible extensions to GMM appearance models.

4.1 L_1 Color Separation Term

Let $S \subset \Omega$ be a segment of the image plane Ω and denote by θ^S and $\theta^{\bar{S}}$ the unnormalized color histograms for the foreground and background appearance respectively. Let n_k be the number of pixels in the image that belong to bin k and let n_k^S and $n_k^{\bar{S}}$ be the according number of the foreground and background pixels in bin k. Our appearance overlap term penalizes the intersection between the foreground and background bin counts by incorporating the simple yet effective high-order L_1 term into the energy function:

$$E_{L_1}(\theta^S, \theta^{\bar{S}}) = -\|\theta^S - \theta^{\bar{S}}\|_{L_1}, \tag{4.1}$$

Theorem 4.1.1. *The L_1 color separation term we proposed here is submodular.*

Below we explain how to incorporate and optimize the term $E_{L_1}(\theta^S, \theta^{\bar{S}})$ using one graph

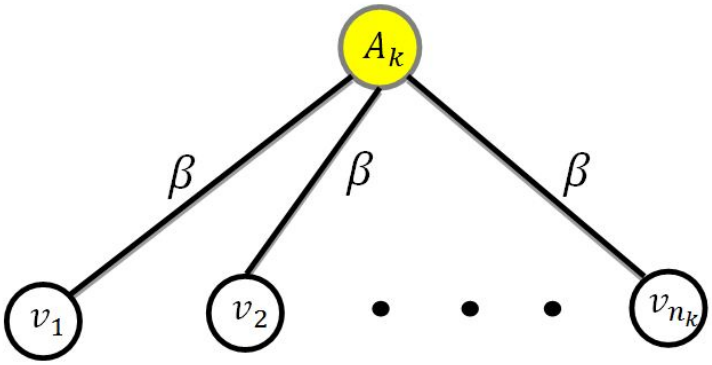


Figure 4.1: Graph construction for E_{L_1} in one color bin: nodes $v_1, v_2, \ldots, v_{n_k}$ corresponding to the pixels in bin k are connected to the auxiliary node A_k using undirected links. The capacity of these links is the weight of appearance overlap term $\beta > 0$.

cut. For clarity of explanation we rewrite the term as

$$E_{L_1}(\theta^S, \theta^{\bar{S}}) = \sum_{k=1}^{K} \min(n_k^S, n_k^{\bar{S}}) - \frac{|\Omega|}{2}. \tag{4.2}$$

It is easy to show that the two sides of (4.2) are equivalent. It's obvious that the L_1 color separation term encourages labeling inconsistency among pixels in the same color bin. The details of the graph construction for the above term over one color bin are shown in Fig. 4.1. In the graph we ignore links for other terms such as smoothness term.

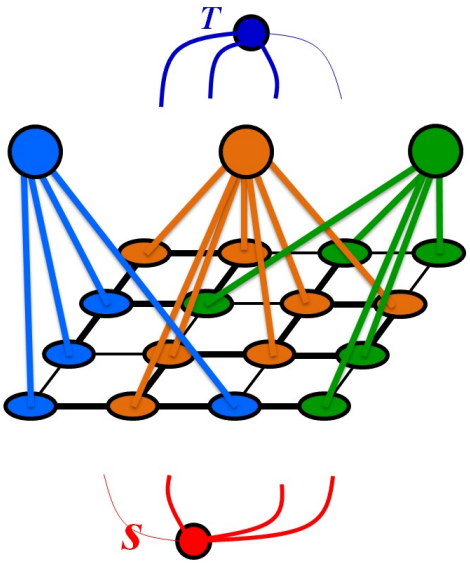


Figure 4.2: Overall graph construction for energy with L_1 color separation term. We have three color bins blue, green and orange in this example. Add one auxiliary node for each color bin and connect the auxiliary node to all pixels of certain color.

Fig. 4.2 gives the overall graph construction of energy with L_1 color separation term. We add K auxiliary nodes $A_1, A_2, ..., A_K$ into the graph and connect k^{th} auxiliary node to all the pixels that belong to the k^{th} bin. In this way each pixel is connected to its corresponding auxiliary node. The capacity of these links is set to $\beta = 1$. Assume that bin k is split into foreground and background, resulting in n_k^S and $n_k^{\bar{S}}$ pixels accordingly. Then any cut separating the foreground and background pixels must either cut n_k^S or $n_k^{\bar{S}}$ links that connect the pixels in bin k to the auxiliary node A_k. The optimal cut must choose $\min(n_k^S, n_k^{\bar{S}})$ in (4.2).

4.2 Minimizing Higher-order Pseudo-boolean Function

The L_1 color separation term can be seen as a special case of the following higher-order psudo-boolean function:

$$f(X_c) = \min\{\theta_0 + \sum_{i \in c} w_i^0(1 - x_i), \theta_1 + \sum_{i \in c} w_i^1 x_i, \theta_{max}\} \tag{4.3}$$

where $x_i \in \{0, 1\}$ are binary variables in clique c, $w_i^0 \geqslant 0$, $w_i^1 \geqslant 0$, and θ_0, θ_1 and θ_{max} are parameters satisfying the constraints $\theta_{max} \geqslant \theta_0$ and $\theta_{max} \geqslant \theta_1$. Consider each color bin as a clique and set parameters $w_i^0 = 1$, $w_i^1 = 1$, $\theta_0 = 0$, $\theta_1 = 0$ and $\theta_{max} = n_k/2$, where n_k is the number of pixels in bin k. Then $f(X_c)$ reduces to $E_{L_1}(\theta^S, \theta^{\bar{S}}) + |\Omega|/2$.

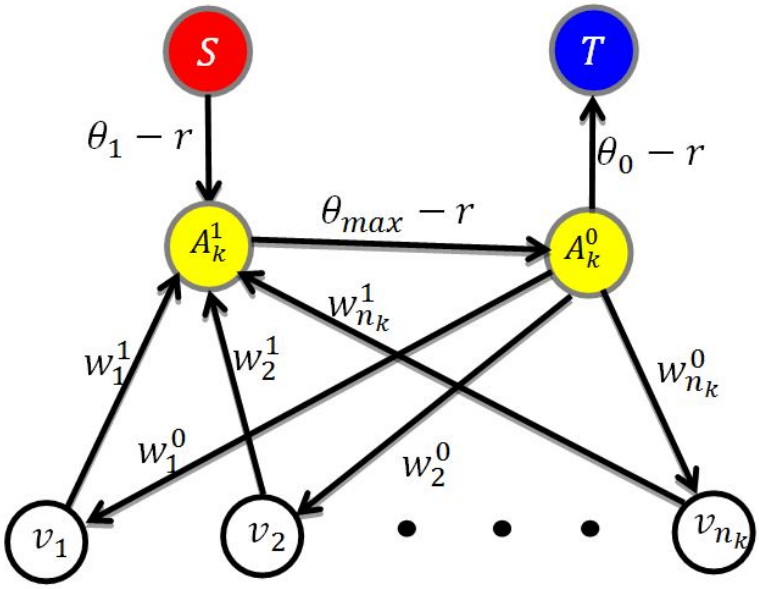


Figure 4.3: Our graph for minimizing pseudo-boolean function (4.3), $r = \min\{\theta_0, \theta_1\}$. Note that nodes and links in the graph for other energy terms are not shown in this figure.

We also propose graph construction (Fig. 4.3) for minimizing pseudo-boolean functions in (4.3) using directed links. Note how the graph in Fig. 4.3 degenerates to the graph in Fig. 4.1: If $\theta_{max} = +\infty$ and $\theta_0 = \theta_1 = 0$, then the two nodes A_k^1 and A_k^0 emerge to one node, directed link becomes undirected and the auxiliary nodes are disconnected from source or sink nodes because $\theta_1 - r = 0$ and $\theta_0 - r = 0$.

To see how the graph in Fig. 4.3 works we consider four possible label assignments for the auxiliary nodes A_k^1 and A_k^0. Table 4.1 shows the cost of corresponding cuts. The minimum cut renders optimization of the function (4.3).

(A_k^1, A_k^0)	the cost of cut
(0,0)	$\theta_1 + \sum_{i\|x_i=1} w_i^1 - r$
(0,1)	$\theta_0 + \sum_{i\|x_i=0} w_i^0 + \theta_1 + \sum_{i\|x_i=1} w_i^1 - 2r$
(1,0)	$\theta_{max} - r$
(1,1)	$\theta_0 + \sum_{i\|x_i=0} w_i^0 - r$

Table 4.1: Cut costs corresponding to four possible label assignments to the binary auxiliary nodes A_k^1 and A_k^0. The optimal cut must choose the minimum of the above costs, thus minimizing (4.3).

4.3 Relationship with P^n Potts Model

A similar graph construction with auxiliary nodes (Fig. 4.4) is proposed in [30, 31] to minimize higher order pseudo-boolean functions (4.3).

Unlike our construction, the method in [30, 31] requires that the parameter θ_{max} in $f(X_c)$ should satisfy the following constraint:

$$\theta_{max} \leqslant \max(\theta_0 + \sum_{i\in c} w_i^0(1 - x_i), \theta_1 + \sum_{i\in c} w_i^1 x_i). \tag{4.4}$$

In contrast, we can optimize high-order functions in (4.3) with arbitrary θ_{max} provided that $\theta_{max} \geqslant \theta_0$ and $\theta_{max} \geqslant \theta_1$. Even though the constraint is not problematic for color separation term as long as we set θ_{max} to $n_k/2$, but we believe eliminating the constrain is important for some other applications when the constraint cannot be easily satisfied.

The graph construction in Fig. 4.4 can be used to minimize E_{L_1}. However, the advantage of our graph construction in Fig. 4.1 is that only one auxiliary node is needed for each color bin, thus our graph construction renders faster inference.

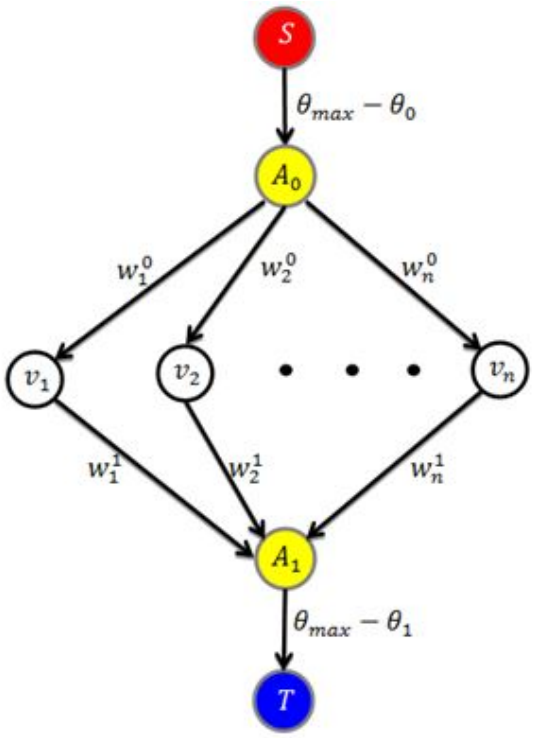


Figure 4.4: Graph for minimizing pseudo-boolean function (4.3) by Kohli et al. [30, 31]

4.4 General Color Separation Term

In general, color separation term does not have to be based on L_1 distance. We can define color separation term using other distance metric such as Jensen-Shannon distance, χ^2 distance or *Bhattacharyya* distance. For example, below we define four appearance overlap terms based on the L_1 norm, χ^2 distance, Bhattacharyya coefficient and Jensen-Shannon divergence between histograms.

$$D_{L_1}(\theta^S, \theta^{\bar{S}}) = \sum_{k=1}^{K} \min(n_k^S, n_k^{\bar{S}}) \tag{4.5}$$

$$D_{\chi^2}(\theta^S, \theta^{\bar{S}}) = \sum_{k=1}^{K} (n_k/2 - (n_k^S - n_k^{\bar{S}})^2/(2n_k)) \tag{4.6}$$

$$D_{Bha}(\theta^S, \theta^{\bar{S}}) = \sum_{k=1}^{K} \sqrt{n_k^S n_k^{\bar{S}}} \tag{4.7}$$

$$D_{JS}(\theta^S, \theta^{\bar{S}}) = \sum_{k=1}^{K} \frac{n_k \log n_k - n_k^S \log n_k^S - n_k^{\bar{S}} \log n_k^{\bar{S}}}{2} \tag{4.8}$$

where θ^S and $\theta^{\bar{S}}$ are unnormalized histograms of the foreground and background respectively.

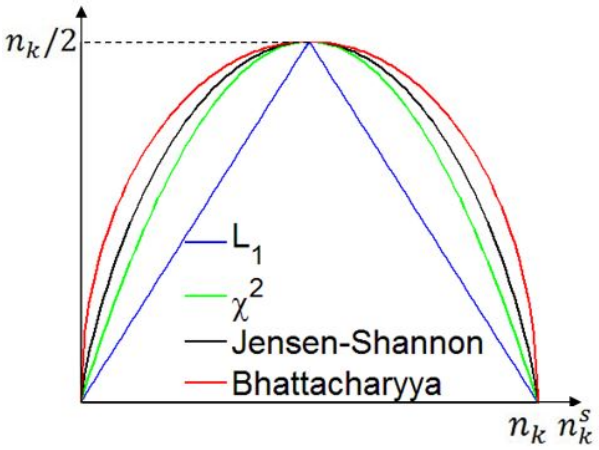


Figure 4.5: Appearance overlap terms based on different metrics: L_1 norm, χ^2 distance, Bhattacharyya coefficient and Jensen-Shannon divergence

All four terms above are concave functions of n_k^s attaining maximum at $n_k/2$. See Fig. 4.5 for the visualization of the terms and comparison with D_{L_1}.
Similarly to [30] we observe that any concave function can be approximated as a piece-wise linear function by using a summation of specific (pyramid-like) truncated functions, each having a general form as in (4.3). For example, Fig. 4.6 illustrates one possible approximation using three components. These truncated components take the form of high-order psudo-boolean function in (4.3) and thus can be incorporated into our graph using the construction shown in Fig. 4.3. Note, D_{L_1} is equivalent to D_{χ^2}, D_{Bha} or D_{JS} when approximated using one truncated component.

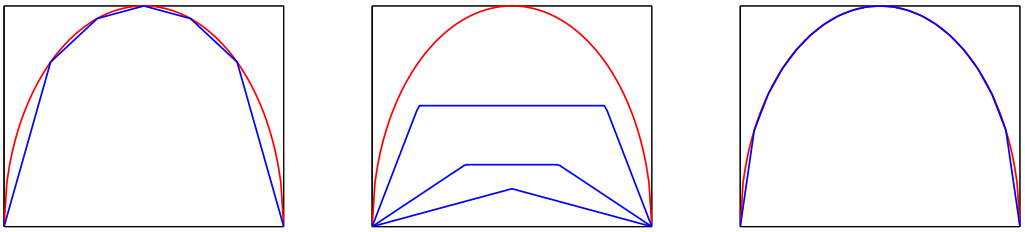


Figure 4.6: The original concave function (red) is approximated as a piece-wise linear function (blue, left) using three truncated components (blue,middle). Approximation with ten components (blue, right) is already very accurate.

Chapter 5

Applications

In this section we apply our appearance overlap penalty term in a number of different practical applications including interactive binary segmentation with bounding boxes or strokes in Sec.5.1, shape matching with simple shape templates in Sec.5.2, saliency segmentation with saliency maps in Sec.5.3 and segmentation from stereo pairs in Sec. 5.4. We show general usefulness of our proposed color separation term.

5.1 Interactive segmentation

First, we discuss interactive segmentation with several standard user interfaces: bounding box [51] in Section 5.1.1 and seeds [12] in Section 5.1.3. We compare different color separation terms including L_1, Jenson-Shannon, Bhattacharyya, χ^2 and truncated L_1 color separation terms in Section 5.1.2.

5.1.1 Binary segmentation with bounding box

First, we use appearance overlap penalty in a binary segmentation experiment with the same setting as GrabCut [9, 51]. For GrabCut GMM is used as appearance model but here color histogram is the appearance model. A user provides a bounding box around an object of interest and the goal is to perform binary image segmentation within the box. The pixels outside the bounding box are assigned to the background using hard constraints. The hard constrain is achieved by having infinity edge weights for links connecting these pixels and sink node. Let $R \subseteq \Omega$ denote the binary mask corresponding to the bounding box, $S_{GT} \subseteq \Omega$ be the ground truth foreground and $S \subseteq \Omega$ be a segment. Denote by $1_S = \{s_p | p \in \Omega\}$ the characteristic

function of S. The segmentation energy function $E(S)$ used here is given by

$$E(S) = |\bar{S} \cap R| - \beta \|\theta^S - \theta^{\bar{S}}\|_{L_1} + \lambda |\partial S|, \tag{5.1}$$

where the first term is a standard ballooning term inside the bounding box R that favors larger foreground, the second term is the appearance overlap penalty as in (4.1), and the last term is a contrast-sensitive smoothness term. We use $|\partial S| = \sum \omega_{pq} |s_p - s_q|$ with $\omega_{pq} = \frac{1}{\|p-q\|} \cdot e^{\frac{-\Delta I^2}{2\sigma^2}}$ and σ^2 set as average ΔI^2 over the image. If we only have color separation term and smoothness term, the trivial solution would be global minima. So we need balooning term to avoid trivial solution. This energy can be optimized with one graph cut.

The energy parameters here are weight of ballooning term β and smoothness term λ. We choose β according to heuristic trick. The input bounding box contains useful information about the object to be segmented other than that it bounds the object. We use the measure of appearance overlap between the box R and its background $\bar{R}$ to automatically find image specific relative weight β of the appearance overlap term w.r.t. the first (ballooning) term in (5.1). In our experiments, we adaptively set an image specific parameter β_{Img} based on the information within the provided bounding box:

$$\beta_{Img} = \frac{|R|}{-\|\theta^R - \theta^{\bar{R}}\|_{L_1} + |\Omega|/2} \cdot \beta^{'}. \tag{5.2}$$

Here $\beta^{'}$ is a global parameter tuned for each application and dataset. It is common to tune the relative weight of each energy term for a given dataset [59]. We found it to be more robust compared to tuning β.

Consider the following two extreme cases. In the case of a trivial solution in which $S = R$, the energy of the solution becomes

$$|\bar{S} \cap R| + E_{L_1}(\theta^S, \theta^{\bar{S}}) = \beta D_{L_1}(\theta^R, \theta^{\bar{R}}). \tag{5.3}$$

In the case of an ideal solution in which $S = S_{GT}$, assuming the object is distinct from its background, we have the energy

$$|\bar{S} \cap R| + E_{L_1}(\theta^S, \theta^{\bar{S}}) \approx |R| - |S_{GT}| \tag{5.4}$$

assuming that the foreground appearance is separated well from the background appearance. Therefore, to avoid the trivial solution $S = R$, we should enforce $\beta D_{L_1}(\theta^R, \theta^{\bar{R}}) > |R| - |S_{GT}|$. Note that if β is too large, the appearance overlap term will dominant the energy and yield

another trivial solution $S = \emptyset$. Therefore, in our experiments, we adaptively set an image specific parameter β_{Img} based on the information within the provided bounding box:

$$\beta_{Img} = \frac{|R|}{D_{L_1}(\theta^R, \theta^{\bar{R}})} \cdot \beta^{'}. \tag{5.5}$$

Here $\beta^{'}$ is a global robust parameter, we set $\beta^{'} = 0.9$ empirically in our experiments.
We experiment on the well known Grab-cut database [51]. There are 50 images in the dataset, but we exclude the cross image for the sake of comparison with [59]. The error rate is defined as the number of misclassified pixels within the bounding box R divided by the size of the box $|R|$. We average error rates for all the images.
We test with different number of color bins and provide quantitative comparison with the grab-cut method [51] (our implementation, modified to work with histograms as in [59]) and the dual decomposition method [59] (results reported by the authors). The Table 5.1 and the plots in Fig. 5.1 report the respective error rates and the average running times. The error rate for our implementation of the GrabCut method is slightly different from 8.1% reported in [59], since we use a different smoothness term and do not downscale images. We tune λ separately for each method and number of bins by minimizing error rate.

	Error rate	Mean runtime
GrabCut (8^3 bins)	8.54%	2.48 s
GrabCut (16^3 bins)	7.1%[1]	1.78 s
GrabCut (32^3 bins)	8.78%	1.63s
GrabCut (64^3 bins)	9.31%	1.64s
GrabCut (128^3 bins)	11.34%	1.45s
GrabCut (256^3 bins)	13.59%	1.46s
DD (16^3 bins)	10.5%	576 s
One-Cut (8^3 bins)	9.98%	18 s
One-Cut (16^3 bins)	8.1%	5.8 s
One-Cut (32^3 bins)	6.99%	2.4 s
One-Cut (64^3 bins)	6.67%	1.3 s
One-Cut (128^3 bins)	6.71%	0.8 s
One-Cut (256^3 bins)	7.14%	0.8 s

Table 5.1: Error rates and mean runtime for GrabCut [9, 51], Dual Decomposition (DD) [59], and our method, denoted by *One-Cut*.

With 16^3 bins, the GrabCut method is the most accurate and fast. However, it is important to see the effect of working with larger number of bins, as some objects might only be distinguishable from the background when using a higher dynamic rage. There is a dip of the error rates for

GrabCut at 16^3 bins. The reason is that finer binning provides better appearance model but GrabCut is more likely to get stuck in local minima. With more color bins, there are more variables for appearance model and GrabCut is more prone to local minima.

As we increase the number of color bins from 16^3, the error rate for the GrabCut method increases, while the error rate of One-Cut decreases. When using 128^3 bins One-Cut runs twice as fast, while obtaining much lower error rate. This is because with more bins, more auxiliary nodes are used, but each auxiliary node is connected to less pixels. The connectivity density decreases and the mincut/maxflow algorithm runs faster.

DD is hundreds of times slower, while its error rate is quite high. Note that in [59], images are down-scaled to maximum side-length of 250 pixels while the method here deals with the original image.

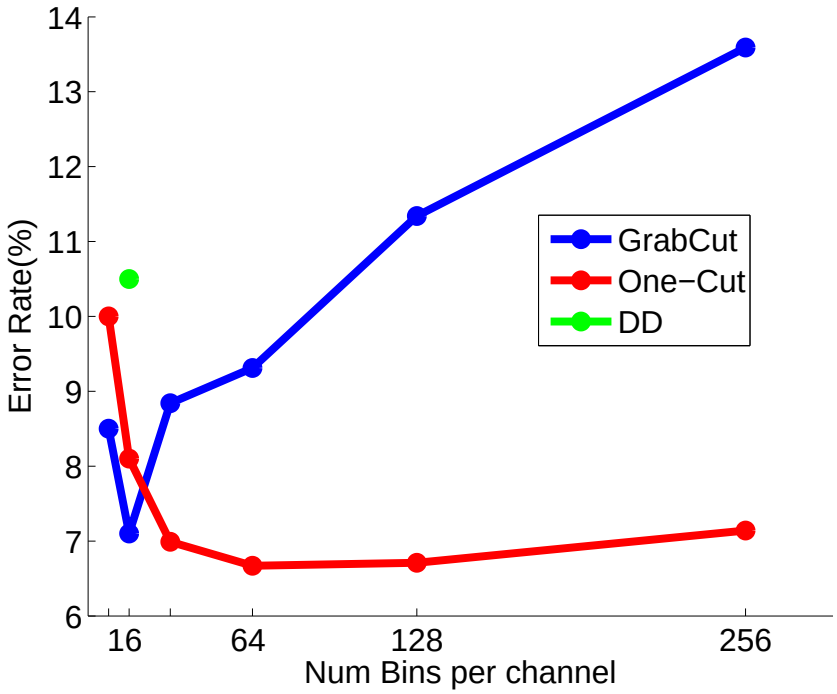
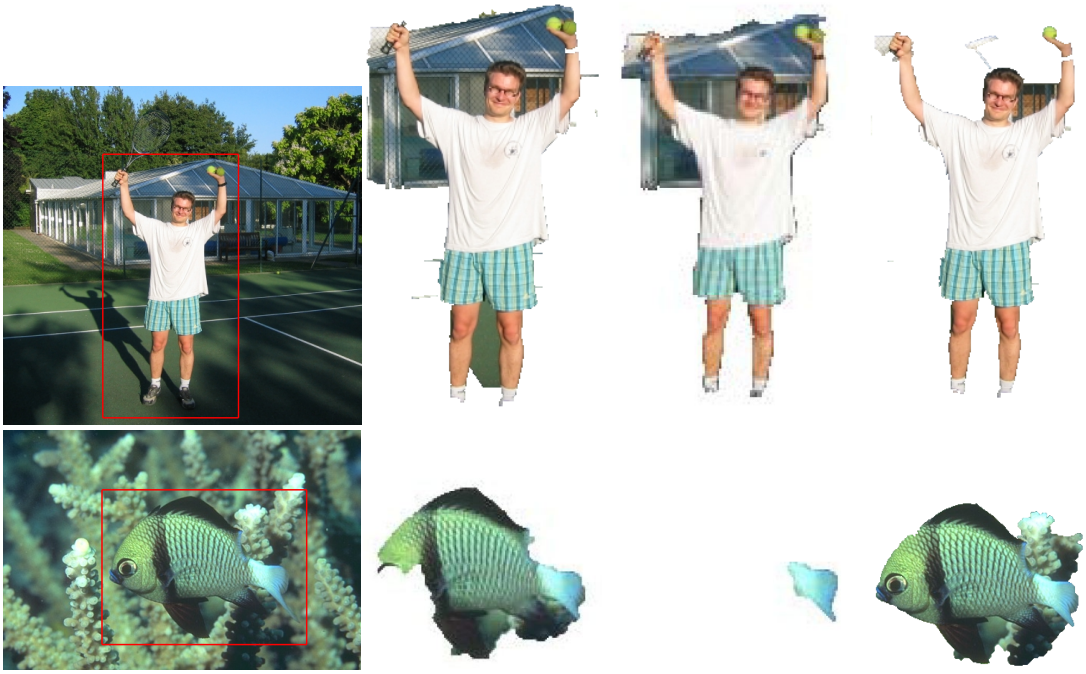


Figure 5.1: Error-rates for different bin resolutions, as in Table 5.1.

Finally, Figures 5.2 - 5.4 show examples of input bounding boxes and segmentation results with the GrabCut [9, 51], Dual Decomposition [59] and our One-Cut method.



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

Figure 5.2: Example of segmentation results. From left to right: (a) user input, (b) GrabCut [9, 51], (c) Dual Decomposition (DD) [59], (d) our One-Cut. For these examples we used 16^3 bins.

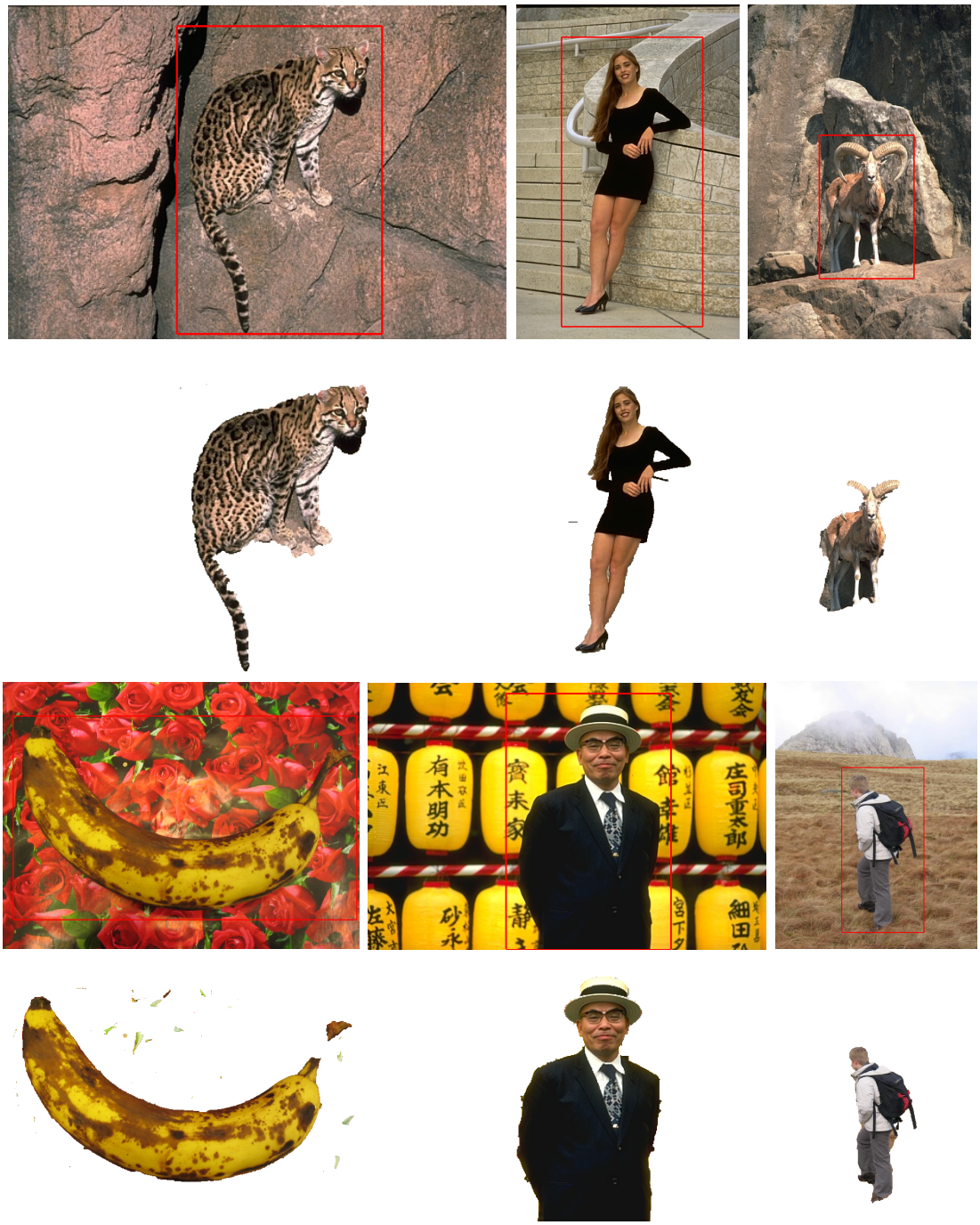


Figure 5.3: Example of segmentation results obtained with our One-Cut. For these examples we used 128^3 bins.

Figure 5.4: Example of segmentation results obtained with our One-Cut. For these examples we used 16^3 bins.

5.1.2 Comparison of Appearance Overlap Terms

In this part we are trying to answer the question of whether our proposed L_1 color separation term is better than the original Jenson-Shannon color separation term. Below we discus additional variants for appearance overlap penalty term. In Sec. 4.4 we've explained how they all can be implemented with the construction proposed in Fig. 4.3. We compare their performance in the task of binary segmentation applied to the GrabCut dataset [9, 51] with the same bounding boxes. We consider four appearance overlap terms based on the L_1 norm, χ^2 distance, Bhattacharyya coefficient and Jensen-Shannon divergence between histograms. The D_{L_1} term above is equivalent to $-\|\theta^S - \theta^{\bar{S}}\|_{L_1}$, but we use this notation for easiness of comparison with other overlap terms.

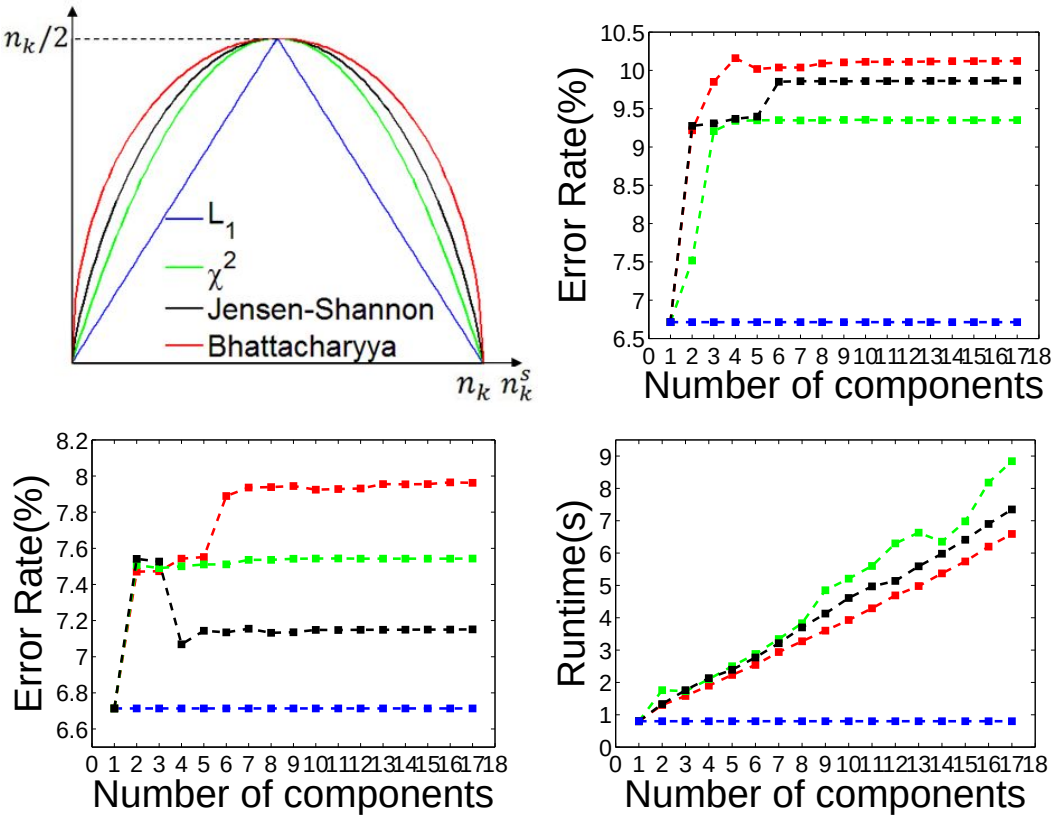


Figure 5.5: Comparison of appearance overlap terms: (top-left) shows the functions plotted for one bin k, (top-right) shows segmentation error rates using the same β_{Img} as in (5.5) for all overlap terms and (bottom-left) shows segmentation results when using a term-specific β_{Img}. The running time is shown on (bottom-right).

All four appearance overlap terms above can be optimized with one graph cut. We can approximate these terms by their piece-wise linear approximation. **Sec. 4.4** has shown how to optimize these terms. We wish to find out which color separation term and what level of approximation

accuracy are optimal for the task of binary segmentation. Therefore, for each color separation term we vary the approximation accuracy with different number of breakpoints (the number of truncated components used) and compare the performance of D_{χ^2}, D_{Bha}, D_{JS} with that of D_{L_1} color separation terms.

In the first experiment setting, we use an adaptive image specific weight β_{Img} for the appearance overlap term as in (5.5) and set $\beta' = 0.9$ which was found optimal for D_{L_1} overlap term. As we can see from Fig. 5.5 (top-right), as the approximation accuracy (the number of components used) increases, the error rate goes up. Note that approximation of D_{χ^2}, D_{Bha}, D_{JS} with one component is the same as D_{L_1}.

In the second experiment, we choose the optimal β_{Img} separately for each appearance overlap term by replacing the denominator of (5.5) with either $D_{\chi^2}(\theta^R, \theta^{\bar{R}})$, $D_{Bha}(\theta^R, \theta^{\bar{R}})$ or $D_{JS}(\theta^R, \theta^{\bar{R}})$ according to the appearance overlap term used. We also tune parameter β' separately for each appearance overlap term. As shown in Fig. 5.5, D_{L_1} achieves the lowest error rate and has the shortest running time than any other overlap term with any level of approximation accuracy. The running time is almost proportional to the number of breakpoints used to approximate the color separation terms.

In the third experiment we replace D_{L_1} with the truncated version $D_{L_1^T} = \sum_{k=1}^{K} \min(n_k^S, n_k^{\bar{S}}, t \cdot n_k/2)$ where $t \in [0, 1]$ is the truncation parameter. Our D_{L_1} term can be seen as a special case of the truncated $D_{L_1^T}$ where $t = 1$. Again, for each value of t we replace the denominator in (5.5) by $D_{L_1^T}(\theta^R, \theta^{\bar{R}})$ and tune β'. Fig. 5.6 reports the error rates of the segmentation as a function of the parameter t. It can be seen that the non-truncated version ($t = 1$) yields the best performance. This further proves the benefit of our proposed D_{L_1} appearance overlap term.

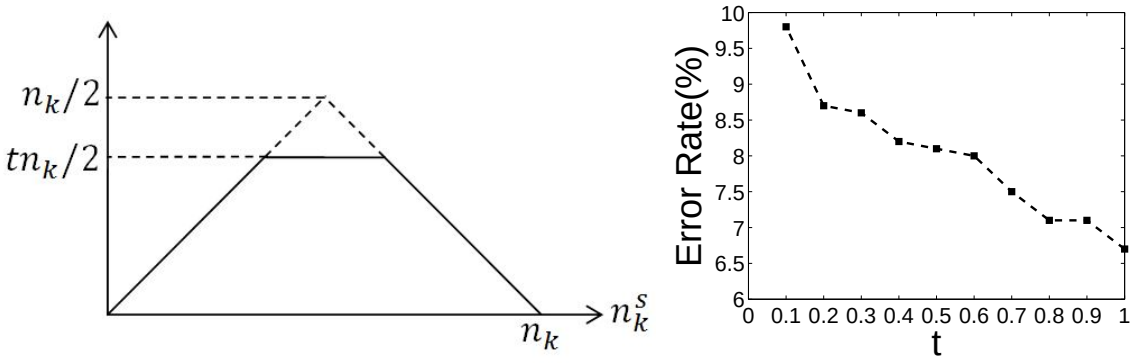


Figure 5.6: Left: Truncated appearance overlap term $D_{L_1^T}$ for a bin k. Right: Segmentation error rate as a function of parameter t in $D_{L_1}^T$. Best results are achieved for $t = 1$ (no truncation).

Figure 5.7: Interactive segmentation with seeds

5.1.3 Interactive Segmentation with Seeds

Unlike interactive segmentation with a bounding box, using seeds *a la* Boykov-Jolly [12] makes volumetric balancing unnecessary due to hard constraints enforced by the user. Therefore, the segmentation energy is quite simple:

$$E_{seeds}(S) = -\beta||\theta^S - \theta^{\bar{S}}||_{L_1} + \lambda|\partial S|$$

subject to the hard constraints given by the seeds. Figure 5.7 shows several qualitative segmentation results with user provided strokes. It is possible to generalize this approach to multilabel interactive segmentation, where color separation terms are minimized between each pair of labels in an $\alpha\beta$-swap move [15].

5.2 Template Shape Matching

Below we discuss how appearance overlap penalty term can be used for template shape matching [5, 6, 7, 56]. Several prior methods rely on graph-cut based segmentation with shape prior [23, 40, 58, 61]. Most commonly, these methods use a binary template mask M and combine the shape matching cue with contrast sensitive smoothness term via energy

$$E_1(S) = \min_{\rho \in \Phi} E_{Shape}(S, M^\rho) + \lambda |\partial S|. \tag{5.6}$$

where ρ denotes a transformation in parameter space Φ, translation for example, and M^ρ is a transformed binary mask. Possible transformation includes but is not limited to translation, rotation and scaling. The term $E_{Shape}(S, M^\rho)$ measures the similarity between segment S and the transformed binary mask M^ρ. Possible metric include Hamming distance or L_2 distance. We further incorporate the appearance overlap into the energy:

$$E_2(S) = E_1(S) - \beta \|\theta^S - \theta^{\bar{S}}\|_{L_1} \tag{5.7}$$

and compare the performance of $E_1(S)$ and $E_2(S)$ in the task of template shape matching.

Figure 5.8: Template shape matching examples, from left to right: Original images, contrast sensitive edge weights, shape matching results without and with the appearance overlap penalty. Input shape templates are shown as contours around the resulting segmentation.

Fig. 5.8 shows few examples of input image, smoothness term, input template and matching results. The templates used are polygon and ellipses. Without the appearance overlap term shape matching yields erroneous segmentation due to the clutter edges in the background.
We experiment on Microsoft Research Cambridge Object Recognition Image Database[2]. There are 282 side view images of cars, roughly of the same scale. We down-scaled all images to 320×240 and used 128^3 color bins per image. We generate ground truth segmentation by ourselves. For this experiment, Φ is defined to be the set of all possible translations and horizontal flip. For the template matching, we scan the image in a sliding-window fashion and compute maxflow/mincut with dynamic graph cut [32]. Each time we shift the template by a little bit minority of the graph edges are changed so we can reuse the maxflow via dynamic graph cut [32]. Note that we did not skip any position when sliding the templates thus we are able to get global minima of the energy. We use Hamming distance for the energy (5.6). In principle, branch-and-mincut [40] can speed up optimization of both energies (5.6) and (5.7), but this is outside the scope of our paper.

Figure 5.9: Template shape matching examples: shape (top left) and pairs of original images + segmentations with $E_2(S)$.

Energy	TP	FP	Error pixels	Runtime
$E_1(S)$	76.97%	6.96%	10106	4.1 s
$E_2(S)$	81.88%	3.86%	7480	13.0 s

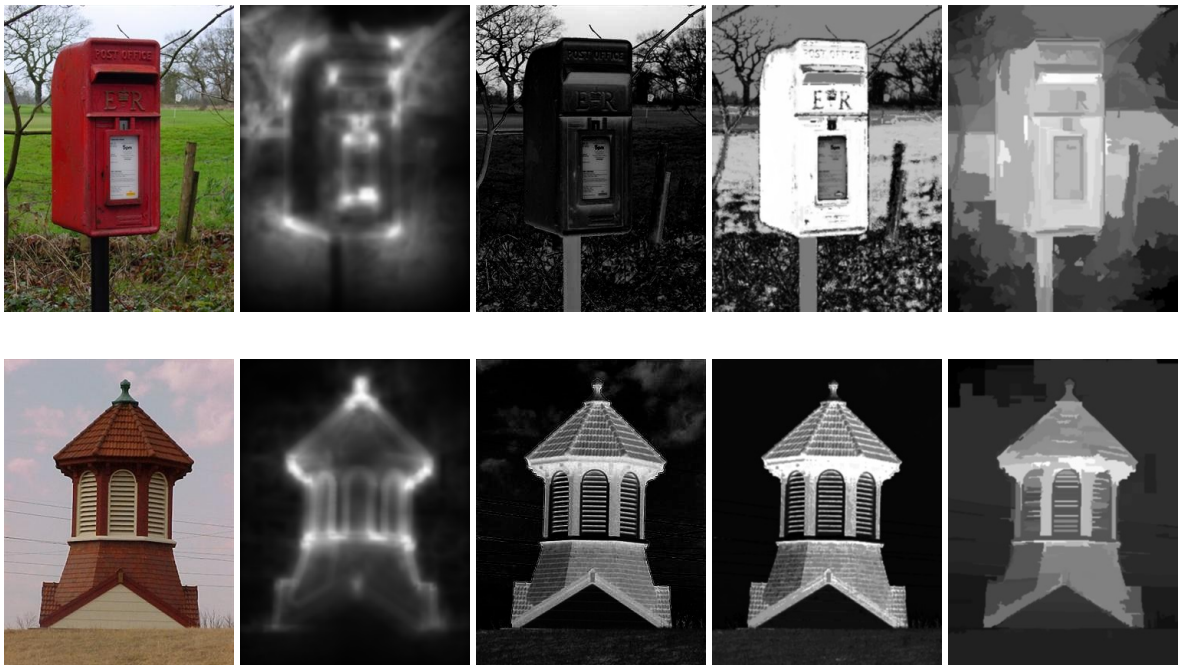
Table 5.2: Template shape matching: comparing $E_1(S)$ and $E_2(S)$ in terms of TP, FP, misclassified pixels, and mean running time. We used $\lambda = 5$ for $E_1(S)$ and $(\lambda = 5, \beta = 1.1)$ for $E_2(S)$.

Fig. 5.9 shows the coarse car template used for this experiment and some qualitative results.

[2]http://research.microsoft.com/en-us/projects/objectclassrecognition

Table 5.2 provides quantitative comparison of the results obtained with and without incorporating the appearance overlap term, namely using $E_2(S)$ and $E_1(S)$. The results are reported with respect to manually outlined ground truth segmentations and clearly point to the benefit of incorporating the overlap term E_{L_1} into the segmentation energy. When using $E_2(S)$ we achieve higher true positive (TP) rate of 81.88%, lower false positive (FP) rate of 3.86% and less misclassified pixels without compromising much the running time.

5.3 Salient object segmentation



(a) Input (b) CA (c) LC (d) HC (e) RC

Figure 5.10: Different saliency maps

Saliency is the measure of pixels by visual attention they attract. Salient region detection and segmentation is an important preprocessing step for object recognition and adaptive compression. Salient objects usually have an appearance that is distinct from the background [1, 19, 49, 65]. Fig. 5.10 shows saliency maps obtained by CA [24], LC [64], HC [19] and RC [19]. A saliency map is a pixel-wise map whose intensity corresponds to the degree of saliency. Below we show how our appearance overlap penalty term can be used for the task of salient object segmentation. We use the saliency map provided by [49] because it yields the best precision/recall curve when thresholded and compared to the ground truth. Let $A : \Omega \rightarrow [0, 1]$

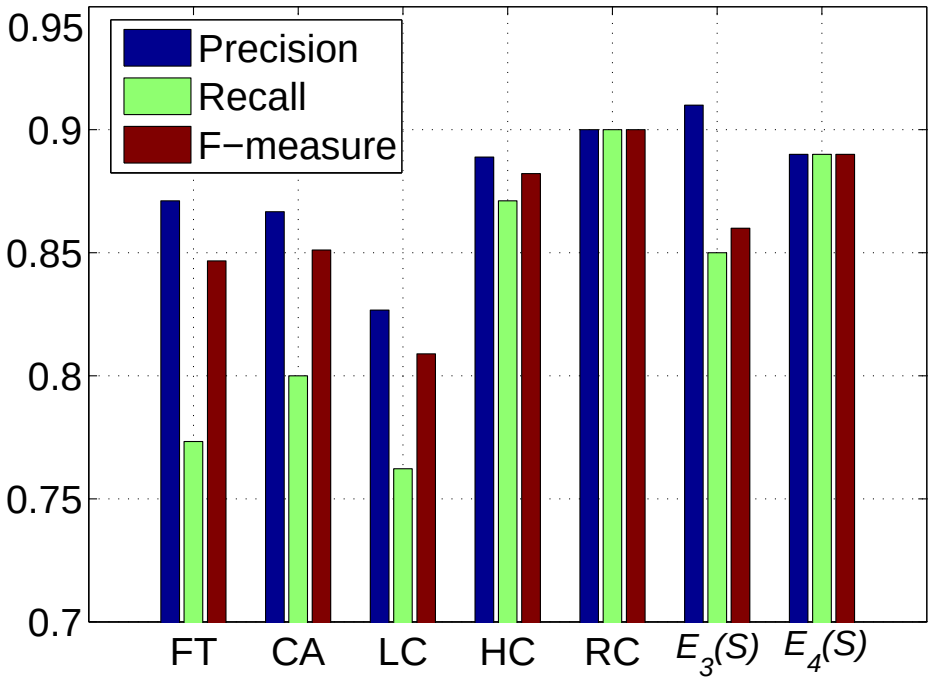


Figure 5.11: Saliency segmentation results reported for dataset [1]: Precison-Recall and F-measure bars for $E_3(S)$, $E_4(S)$ are compared to FT[1], CA[24], LC[64], HC[19] and RC[19].

denote the normalized saliency map and <A> be its mean value. Then let

$$E_{Salience}(S) = \sum_{p\in\Omega} <A> -(A(p)) \cdot s_p \tag{5.8}$$

denote an energy term measuring the saliency of a given segment. We now define two segmentation energies, with and without the appearance overlap term. Let $E_3(S)$ be the energy combining the saliency and smoothness terms

$$E_3(S) = E_{Salience}(S) + |\partial S|, \tag{5.9}$$

and $E_4(S)$ be the energy with the appearance overlap term

$$E_4(S) = E_3(S) - \beta \|\theta^S - \theta^{\bar{S}}\|_{L_1}. \tag{5.10}$$

$E_4(S)$ can be optimized in one graph cut using the construction shown in Fig. 4.1. We use 128^3 color bins, $\beta = 0.3$ and smoothness term $\omega_{pq} = 3(e^{\frac{-\partial I^2}{2\sigma^2}}/\|p-q\| + 0.1)$.
We experiment on publicly available dataset [1] which provides ground-truth segmentation of

1000 images from MSRA salient object database [41]. Fig. 5.11 compares the performance of $E_3(S)$ and $E_4(S)$ with that of FT [1], CA [24], LC [64], HC [19] and RC [19] in terms of *precision*, *recall* and *F-measure* defined as

$$F = \frac{1.3 \cdot Precision \cdot Recall}{0.3 \cdot Precision + Recall}. \tag{5.11}$$

Optimizing $E_3(S)$ results in *precision* = 91%, *recall* = 85% and *F-measure* = 86%, whereas incorporating the appearance term in $E_4(S)$ yields *precision* = 89%, *recall* = 89% and *F-measure* = 89%, which is comparable to the state-of-the-art results reported in literature [19](*precision* = 90%, *recall* = 90%, *F-measure* = 90%). Note that our optimization requires one graph-cut only, rather than the iterated EM-style grab-cut refinement in [19]. Assuming the saliency map is precomputed, the average running time for optimizing $E_4(S)$ is 0.43s and for optimizing $E_3(S)$ is 0.39s. Fig. 5.12 shows qualitative results for our saliency segmentation with and without the appearance overlap term.

5.4 Foreground Segmentation from Stereo

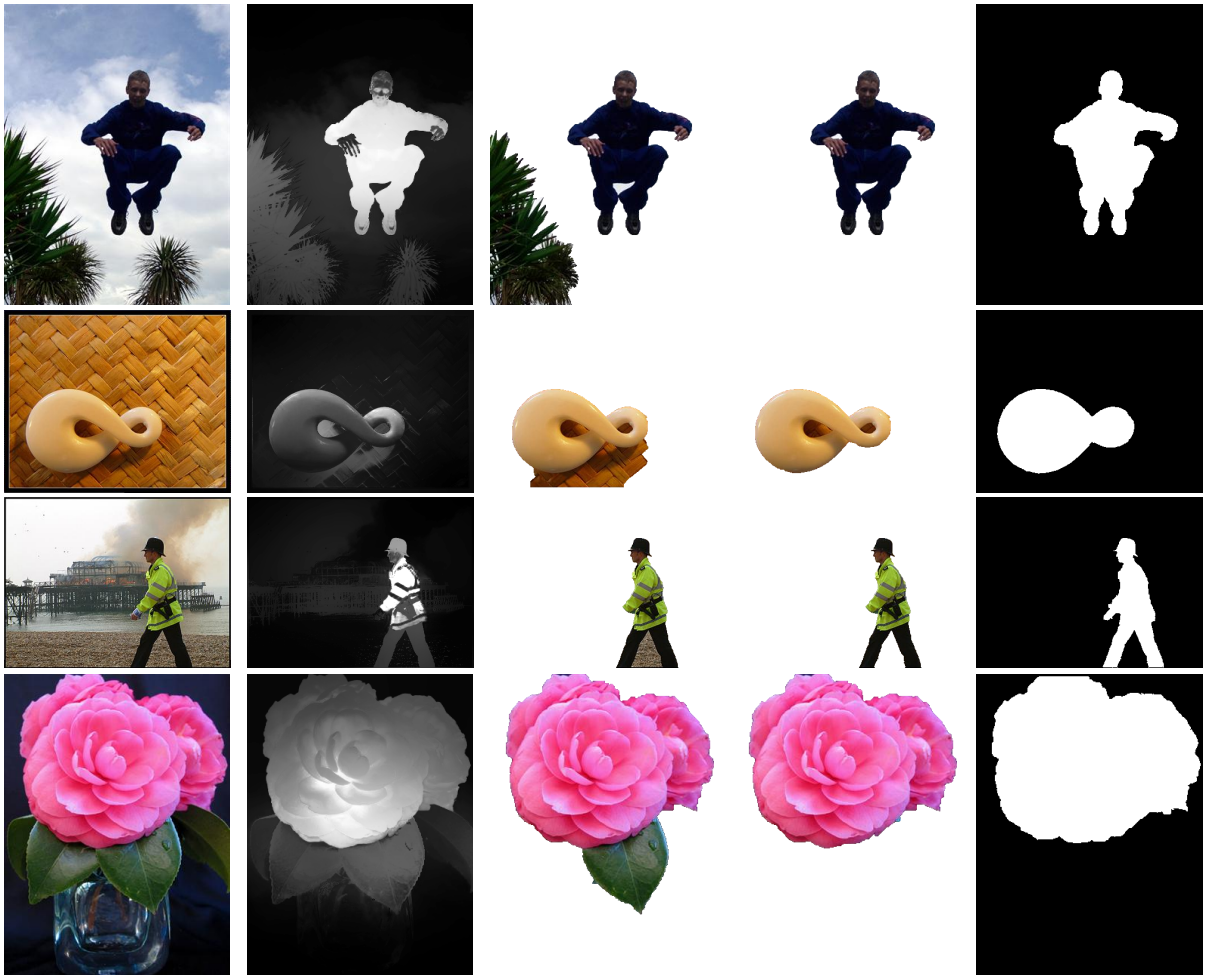
In the task of segmentation from stereo, we are given a stereo pair of left and right views of the same scene. The goal is to segment the foreground that is closer to the camera. Below we formulate the segmentation energy. Given two images denoted by I and I', the energy consists of photo-consistency term and spatial coherence term:

$$E_5(S) = \sum_{i \in \Omega} D_p(s_p) + |\partial S| \tag{5.12}$$

where the photo-consistency term $D_p(s_p)$ can be defined as

$$\begin{aligned} D_p(s_p = 1) &= \min_{0 \le d \le d_1} |I_p - I'_{p \bigoplus d}|, \\ D_p(s_p = 0) &= \min_{d_2 \le d \le d_3} |I_p - I'_{p \bigoplus d}|. \end{aligned} \tag{5.13}$$

Here d_1, d_2 and d_3 are predefined disparities of foreground and background. $p \bigoplus d$ means to shift pixel p to its left or right by d pixels. $D_p(s_p)$ tends to favor label 1 for foreground pixels and label 0 for background pixels. If we optimize energy (5.12) with graph cut, we would get result like (b) in Fig. 5.13 where the foreground and background is not well separated because the photo-consistency term is noisy. We can refine the result with EM to have better result like (c) in Fig. 5.13. At each iteration of EM we re-estimate foreground and background



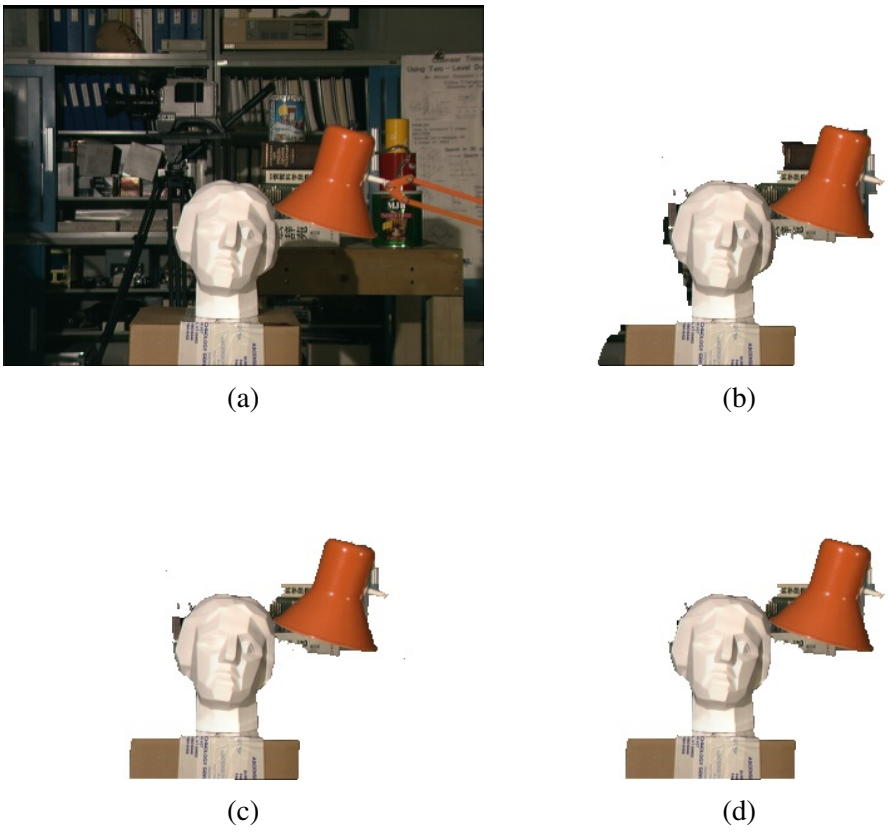
(a) (b) (c) (d) (e)

Figure 5.12: Saliency segmentation examples: (a) Original image, (b) Saliency map from [49] with bright intensity denoting high saliency, (c-d) Graph cut segmentation without and with appearance overlap penalty term, (e) Ground truth.

appearance model and re-segment using log-likelihood term. It takes some iterations for EM to converge. We propose to incorporate the L_1 color separation term into the energy:

$$E_6(S) = E_5(S) - \beta \|\theta^S - \theta^{\bar{S}}\|_{L_1} \tag{5.14}$$

The above color separation augmented energy can be optimized in one graph cut and we get comparable results ((d) in Fig. 5.13) to the result (c) in Fig. 5.13.



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

Figure 5.13: Foreground segmentation from stereo pair (a) One of input stereo images, (b) Segmentation result when optimizing energy $E_5(S)$ (5.12) (c) Refine (b) with EM (d) Segmentation with color separation augmented energy $E_6(S)$ (5.14).

Chapter 6

Future Work and Conclusion

In this chapter we introduce several possible extensions of color separation term and conclude our work.

6.1 Color Separation Term for GMM Appearance Model

One limitation of the L_1 color separation term is that the appearance model is based on color histograms. Using color histograms does not account for similarity between colors in different bins. While distant colors are more likely to belong to different segments, similar colors should be encouraged to group together, even when belonging to different bins. In this section we explain how GMM can be used for color separation term. This is one of our future work.
We can relax the rigid color binning of the histograms by using GMM. Each mixture component can be represented by an auxiliary node and each pixel is assigned to one of the mixture components. The weight of the link between a pixel and an auxiliary node can be set using, e.g., the likelihood that the pixel color is drawn from the corresponding component. Other variants are also possible. Fig. 6.1 shows an example of color separation term with GMM. The graph construction in Fig. 6.1 is very similar to that in 4.1 and have similar complexity. Similarly Kyoungup et. al [47] defined high-order consistency potentials on mean-shift clusters.

6.2 Color Separation Term with Supermodular Term

The segmentation energy (1.15) consists of color separation term and volume balancing term which is supermodular. In the interactive segmentation application (5.1), the supermodular term is simply replaced by hard constraints or heuristic ballooning term to avoid trivial so-

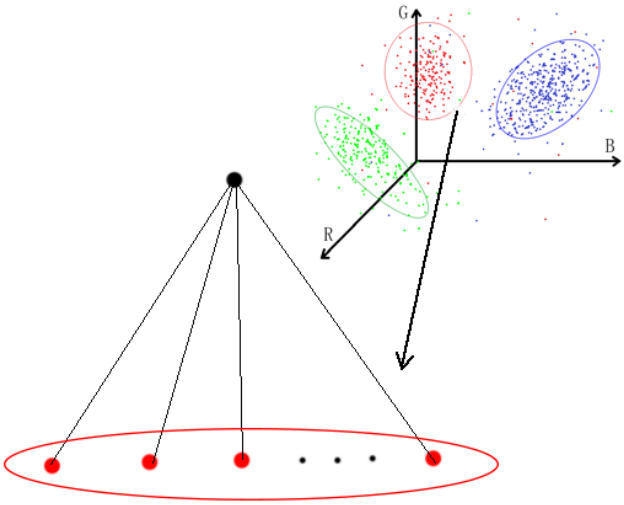


Figure 6.1: Color separation term for Gaussian Mixture Model. For each Gaussian Mixture component, we add an auxiliary node to the graph. Each pixel is connected to the maximum likelihood mixture component. The weight of the link can be the likelihood that the pixel was generated by that particular mixture component.

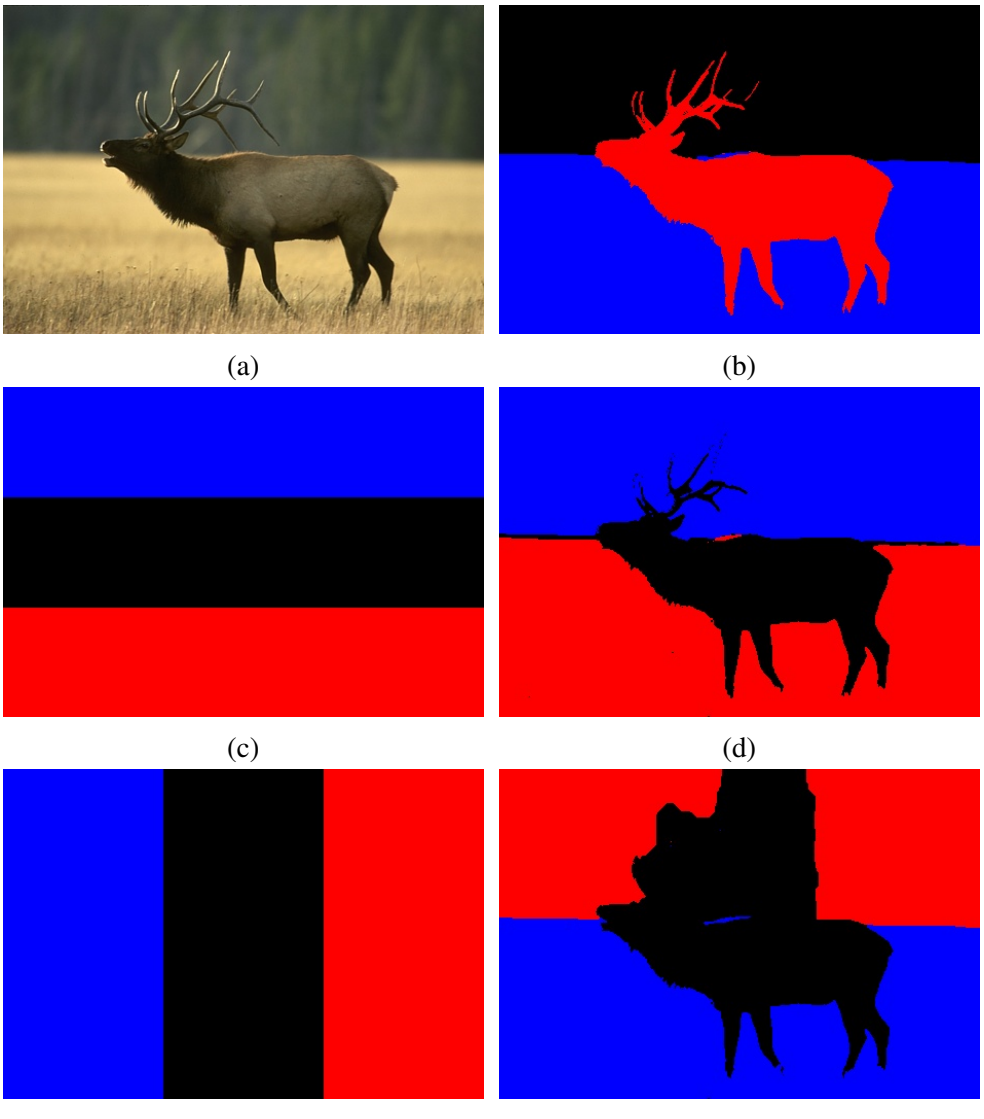
lution. Future work may include combining submodular L_1 color separation term with problematic non-submodular volume balancing terms like h_Ω in (1.17). The resulting energy is non-submodular and cannot be directly optimized using graph-cuts. However, recently published optimization methods such as *Fast Trust Region (FTR)* [25] and *Auxiliary Cuts* [4] have proven to work well for such energies.

Fast Trust Region is an iterative optimization algorithm. At each step the energy is approximated around the current solution. Since in general approximations are only accurate locally, the approximated model is globally optimized within some (trust) region. The current solution is updated and the trust region size is adjusted based on the quality of the approximation model. Auxiliary cut is essentially an upper bound optimizer that tries to approximate the energy by a submodular upper bound. The upper bound approximation and the actual energy agree on the current solution. We believe that color separation term can be combined with other supermodular terms and efficiently optimized using FTR or auxiliary cuts. This is one of the topics in our future work.

In our interactive segmentation with bounding box application, we replace the volume balancing term with linear foreground ballooning term. There we heuristically choose the weight of the ballooning term. We can also explore a range of linear ballooning terms parameterized by their weight. Parametric maxflow [35] will give us solutions of all breakpoints and we choose the solution according to the original energy with nonsubmodular volume prior term.

We propose to efficiently explore a range of simple models for optimization of nonsubmodular hard-to-inference models in the future.

6.3 Feature Separation Term for Multi-label Inference



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

Figure 6.2: Combine color separation term with FTR for volume balancing term: (a) Image, (b) Color separation term + volume balancing term (optimized with FTR, init from trivial solution) (c) Initial solution 1 (d) Result with EM for initial solution 1, expansion move is used (e) Initial solution 2 (f) Result with EM for initial solution 2, expansion move is used.

This thesis focused on binary image segmentation. In principle color separation term can also be applied to multi-label segmentation where appearance overlap should be minimized

between different segments. A straightforward way of incorporating color separation term for multi-label segmentation is through $\alpha - \beta$ swap. During each swap move, only segments of label α and β may change so the problem degenerates to binary segmentation problem and our L_1 color separation term can be incorporated trivially. We only need to replace color binning with other feature binning, then we can minimize overlap of other features for a wide range of applications.

Fig. 6.2 gives preliminary results of multilabel segmentation with color separation term. We used Fast Trust Region (FTR) [25] to optimize the volume balancing term and started from trivial solution where all pixels took the same label. Note that EM is sensitive to initialization. Only good initialization can give good results for EM.

Finally, color separation term can be generalized to incorporate histograms or distribution of other features. Consequently features overlap between different segments can be minimized using similar techniques.

6.4 Conclusion

We proposed an appearance overlap term for graph-cut based image segmentation. This term is based on L_1 distance between unnormalized histograms of foreground and background. The optimization of this term is easier to implement as is shown in Fig. 4.1. What's more, the proposed term is more effective at separating colors compared to other concave (submodular) separators. We show a simpler graph construction than P^n-Potts model that can be easily incorporated into any graph cut based segmentation method. In several applications including interactive image segmentation, shape matching and saliency region detection we achieve comparable or better results with respect to the state-of-the-art. We show that our term is a good fit for interactive segmentation (with bounding box or user seeds interfaces). In contrast to other appearance adaptive methods (e.g. GrabCut) our approach finds guaranteed global minimum in one cut. Our color separation term can be used for multi-label segmentation and easily generalized to other image features, e.g., texture. We hope this work would have an impact on more applications in the future.

Bibliography

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[2] Aaron Archer, Jittat Fakcharoenphol, Chris Harrelson, Robert Krauthgamer, Kunal Talwar, and Éva Tardos. Approximate classification via earthmover metrics. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '04, pages 1079–1087. Society for Industrial and Applied Mathematics, 2004.

[3] Franz Aurenhammer. Voronoi diagramsła survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[4] Ismail Ben Ayed, Lena Gorelick, and Yuri Boykov. Auxiliary cuts for gen. classes of higher order func. In *CVPR*, pages 1304–1311, 2013.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, April 2002.

[6] Serge Belongie, Jitendra Malik, and Jan Puzicha. Matching shapes. In *ICCV*, volume 1, pages 454–461, July 2001.

[7] Serge Belongie, Greg Mori, and Jitendra Malik. *Matching with Shape Contexts*. Birkhäuser, 2005.

[8] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[9] 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)*, Prague, Chech Republic, 2004.

[10] Andrew Blake, Pushmeet Kohli, and Carsten Rother. *Advances in Markov Random Fields for Vision and Image Processing*. MIT Press, 2011.

[11] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision (IJCV)*, 70(2):109–131, 2006.

[12] Yuri Boykov and Marie-Pierre Jolly. *Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images. In *ICCV*, volume I, pages 105–112, July 2001.

[13] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, number 2134 in LNCS, pages 359–374, Sophia Antipolis, France, September 2001. Springer-Verlag.

[14] Yuri Boykov and Vladimir 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):1124–1137, September 2004.

[15] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *International Conference on Computer Vision*, volume I, pages 377–384, 1999.

[16] T. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.

[17] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Trans. Image Processing*, 10(2):266–277, 2001.

[18] Chandra Chekuri, Sanjeev Khanna, Joseph (Seffi) Naor, and Leonid Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, SODA '01, pages 109–118. Society for Industrial and Applied Mathematics, 2001.

[19] Ming-Ming Cheng, Guo-Xin Zhang, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. Global contrast based salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[20] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72:215, 2007.

[21] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, August 2001.

[22] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. *Int. J. Comput. Vision*, 70(1):41–54, 2006.

[23] Daniel Freedman and Tao Zhang. Interactive graph cut based segmentation with shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[24] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[25] L. Gorelick, F. R. Schmidt, and Y. Boykov. Fast trust region for segmentation. In *CVPR*, pages 1714–1721, 2013.

[26] 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.

[27] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[28] M. Kass, A. Witkin, and D. Terzolpoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[29] Jon Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM*, 49(5):616–639, September 2002.

[30] Pushmeet Kohli, Lubor Ladicky, and Philip H. S. Torr. Robust higher order potentials for enforcing label consistency. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[31] Pushmeet Kohli, Lubor Ladicky, and Philip H. S. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. *International Journal of Computer Vision (IJCV)*, 82(3):302C324, 2009.

[32] Pushmeet Kohli and Philip H.S. Torr. Efficiently solving dynamic markov random fields using graph cuts. In *ICCV*, October 2005.

[33] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2011.

[34] Vladimir Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, October 2006.

[35] Vladimir Kolmogorov, Yuri Boykov, and Carsten Rother. Applications of parametric maxflow in computer vision. In *International Conference on Computer Vision (ICCV)*, Nov. 2007.

[36] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts. In *7th European Conference on Computer Vision*, volume III of *LNCS 2352*, pages 65–81, Copenhagen, Denmark, May 2002. Springer-Verlag.

[37] Nikos Komodakis and Georgios Tziritas. Approximate labeling via graph cuts based on linear programming. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(8):1436–1453, 2007.

[38] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, 22(3):277–286, July 2003.

[39] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, May 2008.

[40] Victor Lempitsky, Andrew Blake, and Carsten Rother. Image segmentation by branch-and-mincut. In *ECCV*, 2008.

[41] Tie Liu, Jian Sun, Nan-Ning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[42] László Lovász. Submodular functions and convexity. *Mathematical programming: the state of the art*, pages 235–257, 1983.

[43] R. Malladi, J. A. Sethian, and B. C. Vemuri. A topology independent shape modeling scheme. In *SPIE Conf. on Geometric Methods in Comp. Vision II*, volume 2031, pages 246–258, 1994.

[44] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[45] Oleg Michailovich, Yogesh Rathi, and Allen Tannenbaum. Image segmentation using active contours driven by the bhattacharyya gradient flow. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 16(11), 2007.

[46] E. N. Mortensen and W. A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60:349–384, 1998.

[47] Kyoungup Park and Stephen Gould. On learning higher-order consistency potentials for multi-class pixel labeling. In *European Conference on Computer Vision (ECCV)*, 2012.

[48] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[49] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient reg. detect. In *CVPR*, 2012.

[50] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. Global solutions of variational models with convex regularization. *SIAM Journal on Imaging Sciences*, 3:1122–1145, 2010.

[51] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut - interactive foreground extraction using iterated graph cuts. In *ACM transactions on Graphics (SIGGRAPH)*, August 2004.

[52] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–737, 1997.

[53] Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman, and William T. Freeman. Discovering objects and their location in images. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[54] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(7):787–800, July 2003.

[55] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[56] Alexander Toshev, Ben Taskar, and Kostas Daniilidis. Shape-based object detection via a boundary structure segmentation. *Int. J. Comput. Vision*, 99(2):123–146, September 2012.

[57] M. Unger, T. Pock, D. Cremers, and H. Bischof. Tvseg - interactive total variation based image segmentation. In *British Machine Vision Conference (BMVC)*, Leeds, UK, September 2008.

[58] Olga Veksler. Star shape prior for graph-cut image segmentation. In *European Conference on Computer Vision (ECCV)*, 2008.

[59] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Joint optimization of segmentation and appearance models. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[60] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[61] Nhat Vu and B.S. Manjunath. Shape prior segmentation of multiple objects with graph cuts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[62] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. Map estimation via agreement on (hyper) trees: Message-passing and linear-programming approaches. *IEEE Transactions on Information Theory*, 51(11):3697–3717, November 2005.

[63] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized belief propagation. In *IN NIPS 13*, pages 689–695. MIT Press, 2000.

[64] Yun Zhai and Mubarak Shah. Visual attention detection in video sequences using spatiotemporal cues. In *ACM international conference on Multimedia (ACM Multimedia)*, 2006.

[65] Jianming Zhang and Stan Sclaroff. Saliency detection: A boolean map approach. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[66] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE transactions on PAMI*, 18(9):884–900, September 1996.

Appendix A

Proofs of Theorems

Proof of Theorem 4.1.1. Suppose we have two labeling S_1 and S_2. To prove $E_{L_1}(S_1 \cap S_2) + E_{L_1}(S_1 \cup S_2) \leq E_{L_1}(S_1) + E_{L_1}(S_2)$, we only need to prove the color separation term is submodular for each color bin. We denote A the number of pixels in k^{th} color bin for set $S_1 \cap S_2$, B for set $S_1 \setminus S_1 \cap S_2$, C for set $S_2 \setminus S_1 \cap S_2$ and D for set $\Omega_k \setminus S_1 \cup S_2$. We let:

$$\begin{aligned} t_1 \times A + (1 - t_1) \times (A + B + C) &= A + B \\ t_2 \times A + (1 - t_2) \times (A + B + C) &= A + C \end{aligned} \tag{A.1}$$

then we can see $t_1 + t_2 = 1$. The function $f(x) = \min(x, |\Omega_k| - x)$ is concave, so we have two inequalities:

$$\begin{aligned} t_1 \times f(A) + (1 - t_1) \times f(A + B + C) &\leq f(A + B) \\ t_2 \times f(A) + (1 - t_2) \times f(A + B + C) &\leq f(A + C) \end{aligned} \tag{A.2}$$

the sum of which gives us:

$$\min(A, B + C + D) + \min(A + B + C, D) \leq \min(A + B, C + D) + \min(A + C, B + D). \tag{A.3}$$

Then $E^k_{L_1}(S_1 \cap S_2) + E^k_{L_1}(S_1 \cup S_2) \leq E^k_{L_1}(S_1) + E^k_{L_1}(S_2)$ is proved. □

Curriculum Vitae

Name:	Meng Tang
Post-Secondary Education and Degrees:	University of Western Ontario London, ON, CA 2012 - Present, M.Sc. in computer science Huazhong University of Science and Technology Wuhan, China 2008 - 2012 B.E. in Automation
Honours and Awards:	WGRS, Western University, 2012-2013 Outstanding Graduate, Huazhong Univ. of Sci. and Tech., 2012
Related Work Experience:	Teaching Assistant, Western University, 2012-2013 Research Assistant, Computer Vision Group, Western University, Sep. 2012 - Dec. 2013

Publications:

Meng Tang, Lena Gorelick, Olga Veksler, Yuri Boykov, ”Grabcut in One Cut”, In *IEEE International Conference on Computer Vision (ICCV)*, 2013.