COMPUTATIONAL STATISTICS: TIME SERIES AND DATA MINING

(Spine title: Plib)

(Thesis format: Monograph)

by

Tom Smith

Graduate Program in Statistics and Actuarial 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

THE UNIVERSITY OF WESTERN ONTARIO

School of Graduate and Postdoctoral Studies

CERTIFICATE OF EXAMINATION

Supervisor:

.....

Dr. A. I. McLeod

Joint Supervisor:

.....

Dr. A. Manning

Supervisory Committee:

.....

Dr. W. J. Braun

.....

Dr. A. Bing

Examiners:

.....

Dr. Q. Ring

.....

Dr. W. Fing

.....

Dr. G. Hing

The thesis by

Tom Smith

entitled:

Computational Statistics: Time Series and Data Mining

is accepted in partial fulfillment of the

requirements for the degree of

Masters of Science

. . . . .	. . . . .
Date	Chair of the Thesis Examination Board

Acknowledgement

I am deeply grateful toward to my supervisor, Dr. Olga Veksler. Her constant and gentle guidance has allowed me to enjoy the research I have done. She is always there whenever I needed help, and with her support, I overcame the difficulties in this work. I am thankful for all her insightful comments and for the very thorough revision she has made of this thesis. I consider myself very fortunate to have the chance to learn from her.

I would like to thank Dr. Yuri Boykov for his valuable and helpful suggestions to my research.

I have greatly enjoyed working with my colleagues: Andrew Delong, Chen Zhao, Yong Zhang, Robert Luong and Yu Liu. I am thankful to their help and support over the period of my M.Sc. program.

There are many others I've met along the way with whom I've shared part of this road. To all of you I'm thankful for sharing the thoughts and perspectives. Even though we have been apart for a while, may your trip be a joyful one until we meet again.

At last, I would express my deep gratitude to my parents, for their endless love and care through out my life. No matter how far away, they are always my biggest and best supporters.

Abstract

This is a really silly abstract.

Keywords: Time series analysis, data mining

Contents

List of Figures

List of Tables

List of Appendices

Chapter 1

Introduction

1.1 What Is Image Segmentation?

Image segmentation is a classical and fundamental problem in computer vision. As an integral step of many large vision problems, the quality of segmentation output largely influences the performance of the whole vision system. The success of some segmentation techniques makes them popular in a wide range of applications, such as medical image processing [**?** **?** **?**], object tracking [**?** **?**], recognition [**?** **?**], image reconstruction [**?** **?**] and so on.

Generally, image segmentation refers to partition an image into several disjoint subsets such that each subset corresponds to a meaningful part of the image. Pixels in a subset should possess a similar set of attributes such as intensity, color, or texture properties. Another formulation of image segmentation is called labeling, where each pixel in the image is assigned a unique label which indicates the region it belongs to. There may be a number of possible partitions, and selection of an

appropriate set of regions depends on the attributes associated with the regions.

Since the human visual system performs segmentation so well and is superior to even the most advanced computer vision system, image segmentation has been closely related with perceptual grouping or data clustering, which is the study of how human visual system performs segmentation. Gestalt theory [**?**] suggests how people perceive objects and how they discriminate between the objects and the background, so it can provide various principles for perceptual grouping. In this theory, a set of grouping laws such as similarity, proximity and good continuation are identified to explain the particular way by which the human perceptual system groups tokens together. The gestalt theory has inspired many approaches to segmentation, and it is hoped that a good segmentation can capture perceptually important clusters which reflect local and/or global properties of the image.

Despite many years of research, general purpose image segmentation is still a challenging task, because segmentation is inherently ambiguous. In real images, regions that a human perceives often contain different levels of detail. For different human observers, the level of detail at which the segmentation is produced might be different. In other words, there is no unique solution to the segmentation of a given image.

1.2 Graph based Image Segmentation Techniques

Among the existing image segmentation techniques, many successful ones are benefit from mapping the image elements (e.g. pixels, regions) onto a graph. The segmentation problem is then solved in a spatially discrete space by the efficient al-

gorithm from graph theory. One of the advantages of formulating the segmentation on a graph is that it might require no discretization step and thus incur no related errors. In this paper, we will investigate some graph based techniques for image segmentation, where the problem is generally modeled in term of partitioning a graph into several sub-graphs.

With a history dating back to 1960s, the earliest graph based methods stress the importance of the gestalt principles of similarity or proximity in capturing perceptual clusters. The graph is then partitioned according to these criteria such that each partition is considered as an object segment in the image. In these methods, fixed thresholds and local measures are usually used for computing the segmentation results, while global properties of segmentation are hard to guarantee. The introduction of graph as a general approach to segmentation with a global cost function was brought by Wu et al. [**?**] in 1990s. From then on, many of the research attentions moved on to the study of optimization techniques on the graph. It is known that one of the difficulties in image segmentation is its ill-posed nature. Since there are multiple possible interpretations of the image content, it might be difficult to find a single correct answer for segmenting a given image. This suggests that image segmentation should incorporate the mid- and high-level knowledge in order to accurately extract objects of interest. In the late 1990s, a prominent graph technique emerged in the use of a combination of model-specific cues and contextual information. An influential representation is the s/t graph cut algorithm [**?**]. Its technical framework is closely related to some variational methods [**? ?**] in terms of a discrete manner. Up to now, s/t graph cut and its variants have been extended to the majority of computer vision problems, and eventually acting as an optimization

tool in these areas.

1.2.1 Graph Based Automatic Image Segmentation Techniques

The automatic image segmentation techniques are very appealing when a large number of images are needed to be handled. These techniques can usually be classified into the following three categories:

Minimal spanning tree (MST) based methods

The clustering or grouping of image pixels are performed on the minimal spanning tree. The connection of graph vertices satisfies the minimal sum on the defined edge weights, and the partition of a graph is achieved by removing edges to form different sub-graphs. MST based segmentation methods are essentially related to the graph based clustering. The general study of graph clustering can be dated back to 1970s or earlier. In graph based clustering, the data to be clustered are represented by an undirected adjacency graph. To represent the affinity, edges with certain weights are defined between two vertices if they are neighbors according to a given neighborhood system. Clustering is then achieved by removing edges of the graph to form mutually exclusive subgraphs. The clustering process usually emphasizes the importance of the gestalt principles of similarity or proximity in the graph vertices. However, methods in this category can not deal with situations when there is a large variation inside a cluster. The gestalt principles play an important role in guiding the segmentation, while there lacks a precise measurement on the definition in the quantitative results.

Graph cut with cost functions

Similarly to the MST, graph cut is also a notion that explicitly defined on edge-

weighted graph. Graph cuts based methods possess a distinctive property against other methods in that a general framework of optimally partitioning the graph globally is presented. This brings the advantages that for different applications, different cost functions can be designed with a clear definition of segmented objects. Graph cut techniques provide us an opportunity for a clear and meaningful definition of graph partitioning: minimizing this cut makes vertices in different sets dissimilar. However, for a practical graph partition problem, it also requires vertices in the same set be similar.

Region merging based methods

Image regions are represented by the graph nodes. Starting from a set of primitive regions, the segmentation is conducted by progressively merging the similar neighboring regions according to a certain predicate, such that a certain homogeneity criterion is satisfied. Most region merging algorithms do not have some desirable global properties, even though some recent works in region merging address the optimization of some global energy terms, such as the number of labels [**?**] and the area of regions [**?**]. However, many of them are efficient for computation.

1.2.2 Graph Based Interactive Image Segmentation Techniques

Automatic image segmentation faces a big challenge due to the complexity of image content. A lot of work shows that the user guidance can help to define the desired content to be extracted and thus reduce the ambiguities produced by the automatic techniques. There are two classes of well-known graph based methods in this category.

Graph cut on Markov random field models

The goal is to combine the high level interactive information with the regularization of the smoothness characteristics in the graph cut function. The Markov random field (MRF) theory provides a useful and consistent way of modeling contextual information such as image pixels and features. With the combination of Bayes maximum a posterior (MAP), the MAP-MRF framework formulates the labeling problem into minimizing an energy function: $f^* = \arg\min_f E(f|d)$, where d is the observation of image elements, f is the unknown labeling, and $E(f|d)$ is thus the posterior energy function. Compared with the graph cuts methods introduced in Section **??**, the MAP-MRF framework can explicitly incorporate any desirable high-level contextual information in the energy function. Under the MAP-MRF framework[**? ? ?**], the optimization of a certain group of functionals can be obtained by the classical min-cut/max-flow algorithms or its nearly optimal variants. While for most of other functionals with various smoothness constraints, the minimization is NP-hard. Finding the optimal solutions to them is not a trivial work.

The shortest path based methods

In a weighted graph, the shortest path will connect the two vertices with the minimized sum of edge weights. The object boundary is then defined on a set of shortest path between pairs of graph vertices. These methods require user interactions to guide the segmentation. In practical applications, the modeling of this problem suggests an interactive guidance from the users such that the segmentation process becomes more effective. A potential problem is that one might search over the whole graph for the shortest paths, therefore a large amount of computational resource is needed when segments a high resolution image. Some known properties of graphs can be utilized to avoid the unnecessary shortest path computation, for

example, Falcão et al. [**?**] proposed an acceleration of graph searching algorithm. It is based on the fact that the results of computation from the selected point can make use of the previous position of the selected point. Their algorithm has the advantages that there is no restriction on the shape or size of the boundary and the boundary is oriented so that it has well defined inner and outer parts of the boundary.

1.3 Segmentation Evaluation

Evaluating the quality of image segmentations is an indispensable step for choosing an appropriate output of the image segmentation algorithms. Over the past decades, a large number of segmentation algorithms have been proposed with the hope that a reasonable segmentation should approach the human-level interpretation of the image. With the emergence and development of various segmentation algorithms, the evaluation of perceptual correctness on the output of these algorithms becomes a demanding task.

Visual evaluation of segmentation results can be rather subjective and time consuming, which often leads to inconsistent results among different observers. As an alternative, several quantitative evaluation methods have been proposed to design the objective segmentation measures. The existing objective evaluation methods proposed in the literature can be divided into two categories. In the first category, the empirical goodness measures are proposed to capture the heuristic criteria in the desirable segmentations. The second category of methods is based on measuring the difference between the machine segmentation result and the human-labeled ground truths. This comparison is more intuitive than the empirical based measures,

since the ground truth well represents the human-level interpretation of the image. So far, we do not yet have any standard procedure for the segmentation evaluation due to the ill-defined nature of image segmentation, i.e. there might be multiple acceptable outputs of a segmentation algorithm which are consistent to the human interpretation of the image. Furthermore, there exists a large diversity in the perceptually meaningful segmentations for different images. The above factors make the evaluation of image segmentation quite complex.

1.4 Thesis Contributions and Outline

The main contribution of this thesis lies in providing efficient graph based methods for both of the automatic and the interactive segmentation tasks.

For automatic image segmentation, we develop an algorithm in the region merging style, i.e. the image segmentation is performed by iteratively merging the regions according to a statistical test. The merging order follows the principle of dynamic programming. We can prove that the produced segmentation satisfies certain global properties. If we take the image segmentation as a clustering problem, an image can be partitioned into K sets of clusters that have coherent color and texture features. To this purpose, we propose a K-Sparse Clustering algorithm by which the K clustering centers are learned under the Spare Representation (SR) framework using l_1-norm minimization.

For interactive image segmentation, we address the problem under the graph cut framework. Rather than segmenting the entire image all at once, the segmentation is performed incrementally by extending the graph cut algorithm to a region

merging scheme. An iterated conditional mode (ICM) is studied and the maximum a posterior (MAP) estimation is obtained by virtual of graph cuts on each growing sub-graph. The segmentation process is stopped when all the regions are labeled.

Another contribution of this thesis is in the image segmentation evaluation. So far, there has been a limited number of studies in this field and only a small fraction of them is related to the multiple ground truths based measures. Our work is highly inspired by the human visual system (HVS) in the sense that human observes are highly adapted to extract structural information from natural scenes. Hence, instead of globally matching the segmentations, we propose a probabilistic measure which locally matches the segmentation as much as possible. This evaluation strategy generalizes the configurations of the segmentation and preserves the structural consistency in the ground truths, as a consequence, it provides an intuitive comparison as the HVS does.

In Chapter **??**, we present two automatic graph based methods, which use the region merging principle and the spare clustering technique respectively. In Chapter **??**, the interactive graph based method is proposed as an iterated version of the standard graph cut technique. In Chapter **??**, we present a new segmentation evaluation method. To the best of our knowledge, this is the first work that provides a framework to adaptively combine multiple ground truths in segmentation evaluation. Finally, the Chapter 6 concludes the thesis with some discussion on the future work.

Chapter 2

Automatic Image Segmentation by Dynamic Region Merging

2.1 Introduction

Dating back over decades, there is a large amount of literature on automatic image segmentation. For example, the edge detection algorithms [**? ? ? ?**] are based on the abrupt changes in image intensity or color, thus salient edges can be detected. However, due to the resulting edges are often discontinuous or over-detected, they can only provide candidates for the object boundaries. Another classical category of segmentation algorithms is based on the similarities among the pixels within a region, namely region-based algorithms. In order to cluster the collection of pixels of an image into meaningful groups of regions or objects, the region homogeneity is used as an important segmentation criterion. Many cut criteria in graph theory have been studied for this purpose. The most widely used cut criteria include normalized

cut [**?**], ratio cut [**?**], minimum cut [**?**] and so on. The aim of these algorithms is to produce a desirable segmentation by achieving global optimization of some cost functions. However, these cost functions only provide a characterization of each cut rather than the whole regions. Another problem is that the optimization processes are often computationally inefficient for many practical applications. In recent years, the success of combinatorial graph cut methods [**?** **?**] has been attracting significant research attention. These methods utilize the user input information along with the cut criteria in optimization and nearly global optima can be achieved in linear computational time. As a matter of fact, for most cut-based energy functionals a single optimal partition of an image is not easy to obtain. It makes a possibility of finding different level-based explanations of an image. From this aspect, there are methods [**?** **?**] tackling the image segmentation as a hierarchical bottom-up problem.

In region-based methods, a lot of literature has investigated the use of primitive regions as a preprocessing step for image segmentation [**?** **?** **?**]. The advantages are twofold. First, regions carry on more information in describing the nature of objects. Second, the number of primitive regions is much fewer than that of pixels in an image and thus largely speeds up the region merging process. Starting from a set of primitive regions, the segmentation is conducted by progressively merging the similar neighboring regions according to a certain predicate, such that a certain homogeneity criterion is satisfied. In previous works, there are region merging algorithms based on statistical properties [**?** **?** **?** **?** **?**], graph properties [**?** **?** **?** **?**] and spatio-temporal similarity [**?**]. Although the segmentation is obtained by making local decisions, some techniques [**?** **?** **?** **?** **?**] have shown satisfying

results with efficient implementation. Most region merging algorithms do not have some desirable global properties, even though some recent works in region merging address the optimization of some global energy terms, such as the number of labels [**?**] and the area of regions [**?**]. As a good representation of morphological segmentation, watershed transform [**?**] can also be classified as region-based segmentation methods. The intuitive idea comes from geography, where watersheds are the dividing lines of different catchment basins. The major drawback of watershed transform is the over-segmentation of the image. To overcome this problem, one solution is "flooding" from the selected markers [**?** **?** **?**] such that only the most important regional minima are saved for the segmentation. The other [**?**] is based on a hierarchical process, where the catchment basins of the watershed image are merged until they belong to almost homogeneous regions.

In this chapter, we implement the segmentation algorithm in a region merging style for its merit of efficiency, where similar neighboring regions are iteratively merged according to a novel merging predicate. As stated above, homogeneity criteria (cues) are essential to the region merging process. Although a good enough cue is needed in order to obtain a valuable segmentation, our work does not focus on finding a more complex region model. Instead, we model the cues by a function of random variables. In this way, the properties of cues are not mainly concerned, but the reliability of the cues. In many traditional segmentation algorithms (e.g. [**?** **?** **?**]), the reliability is predetermined and thus researchers often try to use more reliable cues for implementation. Some existing statistical segmentation methods [**?** **?**] use parametric probability models to calculate the reliability of the cues. These methods cannot be used in a general scenario. In this work, image segmentation is

formulated as an inference problem, and there is no limitation on the distribution of random variables. We show that the performance of the segmentation algorithm can be enhanced by a statistical test on the reliability.

In the proposed algorithm, the evaluation of the homogeneity of regions follows the principle of Gestalt theory [**?** **?**], which suggests a preference to having consistent elements in the same data set. The proposed predicate can be therefore interpreted as a combination of the consistency measure and the similarity measure. More specifically, the extent of consistency tells whether or not the tested data belong to the same group. It is measured by two hypotheses according to the Sequential Probability Ratio Test (SPRT) [**?**]: a null hypothesis H_0 "the tested data are consistent" and an alternative hypothesis H_1 "the tested data are inconsistent". The similarity measure describes the affinity between two neighboring regions. In each iteration, a region finds its closest neighbor at the lowest value according to some cost function. With the consistency measure and the similarity measure, the problem of searching in a spatial-time space becomes finding an optimal path that mostly satisfies the data consistency. If taking the segmentation as a labeling problem [**?**], the label of a region will transit to the one of its closest neighbor as iteration goes on. As a result, a merging will happen when two neighboring regions exchange their labels. We can prove that with this merging scheme, a primitive region will be merged into its most similar group at the lowest cost in the final segmentation.

A good segmentation algorithm should preserve certain global properties according to the perceptual cues. This leads to another essential problem in a region merging algorithm: the order that is followed to perform the region merging. Since

the merging process is inherently local, most existing algorithms have difficulties to possess some global optimality. However, we demonstrate that the proposed algorithm holds certain global properties, i.e., being neither over-merged nor under-merged, using the defined merging predicate. In addition, to speed up the region merging process, we introduce the structure of nearest neighbor graph (NNG) to accelerate the proposed algorithm in searching the merging candidates. The experimental results indicate the efficiency of the acceleration algorithm.

2.2 The Region Merging Predicate

Automatic image segmentation can be phrased as an inference problem [**?**]. For example, we might observe the colors in an image, which are caused by some unknown principles. In the context of image segmentation, the observation of an image is given but the partition is unknown. In this respect, it is possible to formulate the inference problem as finding some representation of the pixels of an image, such as the label that each pixel is assigned. With these labels, an image is partitioned into a meaningful collection of regions and objects. The Gestalt laws in psychology [**? ?**] have established some fundamental principles for this inference problem. For example, they imply some well-defined perceptual formulations for image segmentation, such as homogeneous, continuity and similarity. In the family of region merging techniques, some methods have used statistical similarity tests [**? ?**] to decide the merging of regions, where a predicate is defined for making local decisions. These are good examples of considering the homogeneity characteristics within a region, from which we can see that an essential attribute for region

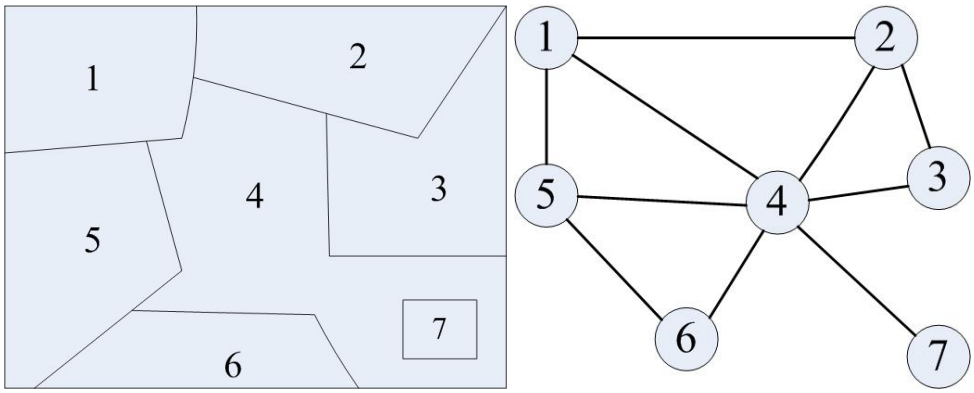
merging is the consistency of data elements in the same region. In other words, if neighboring regions share a common consistency property, they should belong to the same group. However, most of the existing region merging algorithms cannot guarantee a globally optimal solution of the merging result. As a consequence, the region merging output is over-merged, under-merged or a hybrid case. In this section, we propose a novel predicate which leads to certain global properties for the segmentation result.

The proposed predicate is based on measuring the dissimilarity between pixels along the boundary of two regions. For the convenience of expression, we use the definition of region adjacency graph (RAG) [**?**] to represent an image. Let $G = (V, E)$ be an undirected graph, where $v_i \in V$ is a set of nodes corresponding to image elements (e.g. super-pixels or regions). E is a set of edges connecting the pairs of neighboring nodes. Each edge $(v_i, v_j) \in E$ has a corresponding weight $w((v_i, v_j))$ to measure the dissimilarity of the two nodes connected by that edge. In the context of region merging, a region is represented by a component $R \subseteq V$. We obtain the dissimilarity between two neighboring regions $R_1, R_2 \subseteq V$ as the minimum weight edge connecting them. That is,

$$S(R_1, R_2) = \min_{v_i \in R_1, v_j \in R_2, (v_i, v_j) \in E} w((v_i, v_j)) \tag{2.1}$$

The graph structure of an example partition is shown in Fig.**??**, where the image has 7 partitioned regions and its RAG is accordingly shown on the right. The advantage of RAG is that it can provide a "spatial view" of the image.

Since the merging predicate will decide whether there is an evidence of merging between the pair of regions, it involves two aspects: a dissimilarity measure which



(a) (b)

Figure 2.1 An example of region partition and the corresponding region adjacency graph (RAG)

is used to determine the candidate region for merging, and the consistency property which checks if the regions are homogenous. We define the following region merging predicate P:

$$P(R_1, R_2) = \begin{cases} true & \text{if (a) } S(R_1, R_2) = \min\limits_{R_i \in \Omega_1} S(R_1, R_i) \\ & = \min\limits_{R_j \in \Omega_2} S(R_2, R_j); \text{ and} \\ & \text{(b) } R_1, R_2 \text{ are consistent} \\ false & \text{otherwise} \end{cases} \tag{2.2}$$

where Ω_1 and Ω_2 are the neighborhood sets of R_1 and R_2, respectively. The merging predicate on regions R_1 and R_2 could thus be "merge R_1 and R_2 if and only if they are the most similar neighbors in each other's neighborhood and follow the principle of consistency." The condition (a) is stronger than that of only requiring the connecting edge between R_1 and R_2 to be the minimum one in either of the neighborhood. This leads to an interesting property of the proposed region merging algorithm, i.e., the

candidates of the pairs of regions for merging are uniquely decided by the given graph. We shall see hereafter that such a condition uniquely decides the pairs of regions to be merged at a given merging level. Moreover, in Section **??** we will prove that there is always at least one pair of regions which satisfies condition (a). Clearly, without condition (b), all the regions will merge into one big region at the end of region merging process. Therefore, condition (b) acts as a stopping criterion. Fig. **??** illustrates an example when the predicate P between regions R_1 and R_2 is true.

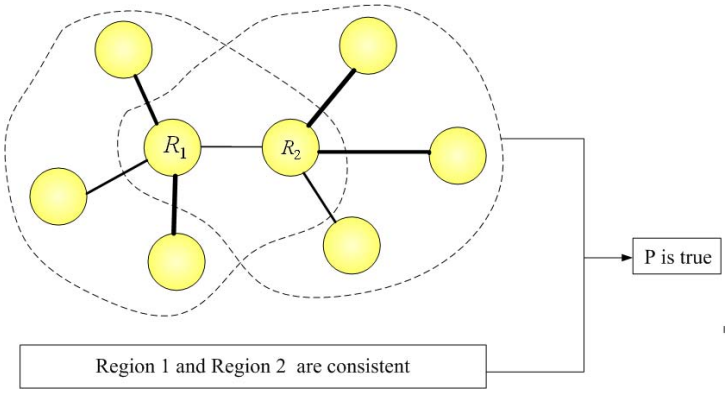


Figure 2.2 An example that the predicate P between R_1 and R_2 is true. The thickness of lines indicates the weights of the edges. The most similar pair of regions is connected by an edge with the minimum weight.

According to the definition of P, the test of consistency is based on the visual cues extracted from the image data. In the next section, the SPRT method is introduced for a reliable decision on the consistency of regions.

2.3 Consistency Test of Cues

In order to obtain the homogenous regions in region merging, the proposed predicate P in Eq. (**??**) checks the consistency of regions. Region information is usually presented by the cues extracted from the observed data. The choice of cues can be intensity, color, texture and so on. If we view the cue as a random variable, the distribution of the cue depends on the consistency of pairs of regions. In this paper, we formulate the evaluation of the region consistency as a sequential test process. Suppose parameter θ is related to the distribution of random cues x. More specifically, we gather information of parameter θ by observing random variables in successive steps. Since every sample of the cues carries statistical information on parameter θ , we may collect the information at the end of observation. This is one of the interesting problems studied in sequential analysis, where θ is called a hypothesis. In the context of region merging, two hypotheses are involved in the evaluation task: a pair of regions is "consistent", and is "inconsistent", which are denoted by a null hypothesis $H_0 : \theta = \theta_0$ against an alternative hypothesis $H_1 : \theta = \theta_1$, respectively. The property of the hypotheses is a hidden state that is not directly observable, but is statistically linked to the observable cues. To decide whether or not a pair of regions belongs to the same group, we look for the solution of its hypothesis test.

An efficient and popular procedure for integrating the statistical evidence is the sequential probability ratio test (SPRT) which was proposed by Wald [**?**]. SPRT shows that the solution to the hypothesis can be found by making the smallest number of observations and satisfying the bounds on the predefined probabilities of two errors. SPRT is purely sequential in nature, i.e. continuing sampling on the instances of a random variable will eventually lead to a reliable inference about

parameter θ.

The application of SPRT to the consistency test of cues is described as follows. We observe the distribution of random cues x in a sequence until a likelihood ratio δ goes out of the interval (B, A) for the first time by a random walk, where the real numbers A and B satisfy $B < 0 < A$. The sequence of successive likelihood ratio δ_i is:

$$\delta_i = log\frac{P_0(x_i|\theta_0)}{P_1(x_i|\theta_1)}, i = 1, 2, \ldots, N \tag{2.3}$$

where $P_0(x|\theta_0)$ and $P_1(x|\theta_1)$ are the distributions of visual cues. $P_0(x|\theta_0)$ and $P_1(x|\theta_1)$ should be different so as to make a convincing decision. We use the Gaussian distribution model to approximate the cue distributions. The two conditional probabilities are given as follows:

$$\begin{aligned} P_0(x|\theta_0) &= \lambda_1 \cdot exp^{(-(I_b-I_{a+b})^T S_I^{-1}(I_b-I_{a+b}))}, \\ P_1(x|\theta_1) &= 1 - \lambda_2 \cdot exp^{(-(I_b-I_a)^T S_I^{-1}(I_b-I_a))} \end{aligned} \tag{2.4}$$

where I_a, I_b are the average color of sampled data in region a, b respectively, and I_{a+b} is the average value of samples' union. S_I is the covariance matrix of the regions, and λ_1, λ_2 are scalar parameters. If each test is independent, the composition of the likelihood ratios is the sum of the individual δ_i:

$$\delta = \sum_{i=1}^{N} \delta_i \tag{2.5}$$

where N is the first integer for which $\delta \geq A$ or $\delta \leq B$. We can see that the solution to the hypothesis is decided by the relationship between δ and an upper and lower limits, denoted by A and B, respectively. If δ goes out of one of these limits, the hypothesis is made and thus the test stops. Otherwise, the test is carried on with a new random sampling.

Intuitively, the likelihood ratio δ is positive and high when the terminal decision is in favor of H_0, while it is negative and low when the situation is reversed. In the SPRT theory [**?**], Wald recommended implementing the test with a practical approximation: $A = log\frac{1-\beta}{\alpha}$ and $B = log\beta(1-\alpha)$, where α and β are probabilities of the decision error given by:

$$\alpha = Pr\{\text{Reject } H_1 \text{ when } H_1 \text{ is true}\},$$
$$\beta = Pr\{\text{Accept } H_1 \text{ when } H_0 \text{ is true}\}$$

The selection of α and β affects the region merging quality. Intuitively, as error rates decrease, the region merging quality grows. However, at the same time the computational effort will increase accordingly. In our implementation, both α and β are set as a fixed value 0.05.

For the uncertainty of the (worst-case) number of tests in SPRT, a truncated SPRT [**?**] is used here by presetting an upper bound N_0 on the number of tests. In Wald's theory, the expected number of tests is given by:

$$\begin{aligned} E\{n \mid \theta_0\} &= [A(1-\beta) + B\beta]/\eta_0 \\ E\{n \mid \theta_1\} &= [A\alpha + B(1-\alpha)]/\eta_1 \end{aligned} \tag{2.6}$$

where η_0, η_1 are the conditional expected number of trails from a single test: $\eta_0 = E\{\delta \mid \theta_0\}$ and $\eta_1 = E\{\delta \mid \theta_1\}$. We set N_0 to be a constant which is greater than $max\{E\{\delta \mid \theta_0\}, E\{\delta \mid \theta_1\}\}$. The proposed SPRT based consistency test of cues is summarized in Table **??**.

Table 2.1 Algorithm 1: Consistency test of cues

Preset λ_1;

Let $\lambda_2 = 1, \alpha = 0.05, \beta = 0.05$;

Compute parameters:

N_0: be a constant greater than $max\{E\{\delta \mid \theta_0\}, E\{\delta \mid \theta_1\}\}$;

$A = log\frac{1-\beta}{\alpha}$,$B = log\beta(1-\alpha)$;

$P_0(x \mid \theta_0)$,$P_1(x \mid \theta_1)$ are computed using Eq. (**??**);

Input: a pair of neighboring regions.

Output: the decision D that the two regions are "consistent" ($D = 1$) or "inconsistent" ($D = 0$).

---

1. Set evidence accumulator δ and trials counter n to be 0.

2. Randomly choose m pixels in each of the pair of regions, where m equals the half size of the region.

3. Calculate the distributions of visual cues x using Eq. (**??**) based on these pixels.

4. Update the evidence accumulator $\delta = \delta + log\frac{P_0(x|\theta_0)}{P_1(x|\theta_1)}$.

5. If $n \leq N_0$,

   If $\delta \geq A$, return $D = 1$ (consistent),

   If $\delta \leq B$, return $D = 0$ (inconsistent)

   If $n > N_0$,

   If $\delta \geq 0$, return $D = 1$ (consistent)

   If $\delta < 0$, return $D = 0$ (inconsistent)

6. Go back to step 2.

2.4 Dynamic Region Merging (DRM)

2.4.1 The dynamic region merging algorithm

In this section, we explain the proposed region merging algorithm as a dynamic region merging (DRM) process, which is proposed to minimize an objective function with the merging predicate P defined in Eq. (**??**). As mentioned in Section **??**, the proposed DRM algorithm is started from a set of over-segmented regions. This is because a small region can provide more stable statistical information than a single pixel, and using regions for merging can improve a lot the computational efficiency. For simplicity and in order to validate the effectiveness of the proposed DRM algorithm, we use the watershed algorithm [**?**] (with some modification) to obtain the initially over-segmented regions (please refer to Section **??** for more information), yet using a more sophisticated initial segmentation algorithm (e.g. mean-shift [**?**]) may lead better final segmentation results.

Given an over-segmented image, there are many regions to be merged for a meaningful segmentation. By taking the region merging as a labeling problem, the goal is to assign each region a label such that regions belong to the same object will have the same label. There are two critical labels for a region R_i : the initial label l_i^0, which is decided by the initial segmentation, and the final label l_i^n, which is assigned to the region when the merging process stops. In our problem, the final label l_i^n for a given region is not unique, which means that the same initialization l_i^0 could lead to different solutions. This uncertainty mainly comes from the process of SPRT with a given decision error. The test of consistency/inconsistency depends on the error probabilities of the cue decisions α and β. In general, these decisions are precise for

homogenous regions. If a region contains a small part of non-homogenous data, the SPRT might add a few more times of tests to verify its decision. With reasonably small error probabilities, the segmentation results will be more reliable. According to our observation, in most cases, the segmentation result is stable for a given image and it can be guaranteed that all the results satisfy the merging predicate P defined in Eq. (**??**). In the process of region merging, the label of each region is sequentially transited from the initial one to the final one, which is denoted as a sequence $(l_i^0, l_i^1, \ldots, l_i^n)$.

To find an optimal sequence of merges which produce a union of optimal labeling for all regions, the minimization of a certain objective function F is required. According to predicate P, the transition of a region label to another label corresponds to a minimum edge weight connects the two regions. In this case, a sequence of transitions will be defined on a set of local minimum weights, i.e., in each transition the edge weight between the pair of merged regions should be the minimum one in the neighborhood. As a result, the objective function F used in this work is defined as the measure of transition costs in the space of partitions. In other words, as the whole image is a union of all regions, F is the sum of transition costs over all regions. That is:

$$F = \sum_{R_i} F_i \tag{2.7}$$

where F_i is the transition costs of one region R_i in the initial segmentation. Minimizing F in Eq. (**??**) is a combinatorial optimization problem and finding its global solutions is in general a hard task. Since the exhaustive search in the solution space is impossible, an efficient approximation method is desired. The solution adopted here is based on the stepwise minimization of F, where the original problem is

broken down into several sub-problems by using the dynamic programming (DP) technique [**?**].

The DP is widely used to find the (near) optimal solution of many computer vision problems. The principle of DP is to solve a problem by studying a collection of sub-problems. Indeed, there have been some works in image segmentation that benefit from this efficient optimization technique, such as DP snake [**? ?**]. In the proposed DRM algorithm, we apply DP on discrete regions instead of line segments [**? ?**]. The minimization problem for region starting at labeling l_i^0 is defined as:

$$\begin{aligned} & minF_i(l_i^0, \cdots, l_i^n) \\ & = minF_i(l_i^0, l_i^{n-1}) + d_{n-1,n} \\ & = minF_i(l_i^0, l_i^{n-2}) + d_{n-2,n-1} + d_{n-1,n} \\ & = \cdots \\ & = \sum_{k=0}^{n-1} d_{k,k+1} \end{aligned} \tag{2.8}$$

where $F_i(l_i^0, \cdots, l_i^n)$ is the transition cost from l_i^0 to l_i^n, $d_{k,k+1}$ is the minimum edge weight between the regions with labeling l_i^k and l_i^{k+1}, respectively. In conjunction with Eq. (**??**), we have

$$d_{k,k+1} = \min_{R_{k+1} \in \Omega_k} S(R_k, R_{k+1}) \tag{2.9}$$

The overall path length from l_i^0 to l_i^n is the sum of minimum edges $d_{k,k+1}$ for each node in that path. This problem reduces to a search for a shortest path problem, whose solution can be found at a finite set of steps by the Dijkstra's algorithm in polynomial time. At this point, a minimization process of object function F is

exactly described by the predicate P defined in Eq. (**??**), where P is true if the nodes are connected by the edge with the minimum weight in their neighborhood. It means that the closest neighbors will be assigned to the same label, which is the cause of the merging.

In Fig. **??**, an example process of region merging is shown by embedding it into a 3D graph. Between two adjacent layers there is a transition which indicates the costs of a path. Clearly, this is also a process of label transitions. The neighborhood of the highlighted region (in red) is denoted as the black nodes in the graph and the closest neighbor is denoted as the red nodes. The directed connections with the lowest cost between adjacent layers are made (shown as blue arrows). Note that the connectivity between regions in the same layer is represented by the RAG, which is not explicitly shown in Fig. **??**.

The proposed DRM algorithm is summarized in Table **??**. Minimization of the objective function follows the principle of DP, which is exactly expressed in terms of predicate P. In the k-th decision step, a merging occurs when two neighboring regions are connected by the minimum weight edge in each other's neighborhood. With the observation of this characteristic, we will propose an accelerated region merging algorithm in Section **??**.

2.4.2 Properties of the proposed DRM algorithm

Although the proposed DRM scheme is conducted in a greedy style, some global properties of the segmentation can be obtained. More specifically, it can be proved that the proposed DRM algorithm produces a segmentation S which is neither over-merged nor under-merged according to the proposed predicate P. Similar to the

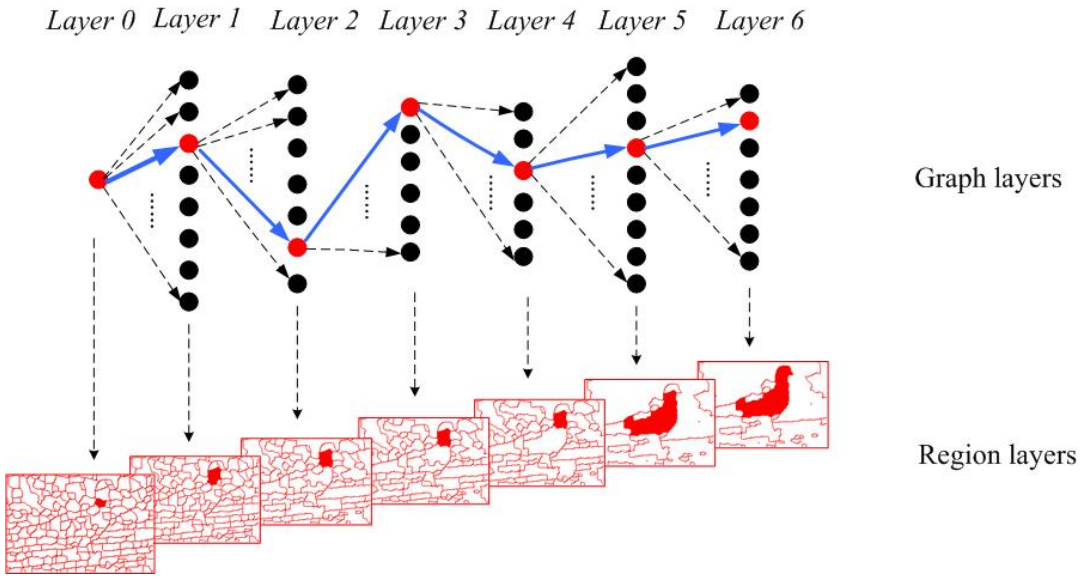


Figure 2.3 The dynamic region merging process as a shortest path in a layered graph. The upper row shows the label transitions of a graph node. The lower row shows the corresponding image regions of each label layer. Starting from layer 0, the highlighted region (in red) obtains a new label from its closest neighbor (in red). If the region is merged with its neighbor, they will be assigned to the same label. The shortest path is shown as the group of the directed edges (in blue).

Table 2.2 Algorithm 2: Segmentation by Dynamic Region Merging

Input: the initially over segmented image S_0 .
Output: region merging result.
1. Set $i = 0$.
2. For each region in segmentation S_i, use **Algorithm 1** to check the value of predicate P with respect to its neighboring regions.
3. Merge the pairs of neighboring regions whose predicate P is true, such that segmentation S_{i+1} is constructed.
4. Go back to step 2 until $S_{i+1} = S_i$.
5. Return S_i.

definitions of over-segmentation and under-segmentation in [**?**], we define the concepts of over-merged segmentation and under-merged segmentation as below.

Definition 1. Under-merged segmentation. A segmentation S is under-merged if it contains some pair of regions for each the predicate P is true.

Definition 2. Over-merged segmentation. A segmentation S is over-merged, if there is another segmentation S_r which is not under-merged and each region of S_r is contained in some component of S. Or saying, S_r can be obtained by splitting one or more regions of S. We call that S_r is a refinement of S.

Definition 3. The evidence of boundary. There is an evidence of boundary between a pair of regions R_1 and R_2 if the predicate P is false with condition (a) being satisfied but not condition (b)in Eq. (**??**).

Lemma 1. If two adjacent regions are not merged for an evidence of a boundary between them in the k-th iteration, they will be in the different regions in the final segmentation. Denote by R_i^k and R_j^k two neighboring regions in the k-th iteration. Then $R_i = R_i^k$ and $R_j = R_j^k$, where R_i is the region whose label is L_i and R_j is the region whose label is L_j in the final segmentation S.

Proof. If there is an evidence of a boundary between R_i^k and R_j^k , according to **Definition 3**, we have

$$S(R_i, R_j) = \underset{a \in \Omega_i}{minS}(R_i, R_a) = \underset{b \in \Omega_j}{minS}(R_j, R_b)$$

where Ω_i and Ω_j are the neighborhood of R_i^k and R_j^k. Since R_i^k and R_j^k are not merged, they will not be merged with any other regions in the remaining steps before they are merged with each other. So we have $R_i = R_i^k$ and $R_j = R_j^k$. ■

Lemma 1 holds because our merging predicate relies on comparing the minimum weight edge between regions. We can prove that with this measure, some global properties of the segmentation can be easily obtained.

Theorem 1. Given the predefined error bounds, the segmentation S by **Algorithm 2** is not under-merged according to **Definition 1**.

Proof. If S is under-merged, there must be some pair of regions R_i^k and R_j^k that did not cause a merge in the merging process. Therefore, the evidence of a boundary does not hold for R_i^k and R_j^k. According to **Lemma 1**, if R_i^k and R_j^k are not merged, $R_i = R_i^k$ and $R_j = R_j^k$. This implies that **Algorithm 2** does not merge R_i^k and R_j^k. The evidence of a boundary holds for them, which is a contradiction. ■

Theorem 2. Given the predefined error bounds, the segmentation S by **Algorithm 2** is not over-merged according to **Definition 2**.

Proof. If S is over-merged, there must be a proper refinement T that is not under-merged. Let a region $C \in S$, and there are two adjacent regions $A \subset C$, $B \subset C$, such that A and B satisfy the refinement T. According to **Algorithm 2**, A and B will not merge with any other regions in C before they merge with each other. Then C does not contain A and B, which is a contradiction. ■

In **Theorem 1** and **Theorem 2**, we claim that the proposed DRM method brings global properties to the segmentation result. The under-merging and over-merging issues are essential problems to deal with in many region merging algorithms. Although the predicate P is locally defined in Eq. (**??**), it can lead to some desirable properties which are held in the mediate segmentation after several iterations, and especially in the final segmentation. The under-merging error in **Definition 1** is easier to overcome, because once two regions are merged, there is no chance to judge them again. While for the over-merging error in **Definition 2**, the proposed DRM method forces the predicate P to make the same decision on the evidence of a boundary, and therefore it can bring a global property for the final segmentation result. This is not a trivial problem in region merging techniques. Many existing algorithms have difficulties to preserve the global property. For example, the method in [**?**] cannot overcome the over-merging error, which means that the merging of a pair of regions may contradict to some previous judgments on them, and in [**? ?**], the under-merging and over-merging problems are not solved with theoretical guarantees.

2.5 Algorithm Acceleration by Nearest Neighbor Graph (NNG)

The DRM process presented in **Algorithm 2** depends on adjacency relationships between regions. At each merging step, the edge with the minimum weight in a certain neighborhood is required. This requires a scan of the whole graph by which the relations between neighboring regions are identified. The edge weights and nodes of RAG are calculated and stored for each graph layer. Since the positions of the edges are unknown, the linear search for these nodes and edges requires $O(||E||)$ time. After each merging, at least one of the edges must be removed from RAG, the positions and edge weights are updated. Then a new linear search is performed for constructing the next graph layer. If the number of regions to be merged is very large, the total computational cost in the proposed DRM algorithm will be very high.

Based on the observation that only a small portion of RAG edges counts for the merging process, we can find an algorithm for accelerating the region merging process. The implement of the algorithm relies on the structure of nearest neighbor graph (NNG), which is defined as follows. For a given RAG, where $G = (V, E)$, the NNG is a directed graph $G_m = (V_m, E_m)$, where $V_m = V$. If we define a symmetric dissimilarity function S to measure the edge weights, the directed edge is defined as:

$$E_m = \{(v_i, v_j) \mid w((v_i, v_j)) = minS(v_i, v_k), (v_i, v_k) \in E\}$$

Fig. **??** shows an example of NNG with respect to the RAG in Fig. **??**. From the definition we can see that the out-degree of each node equals to one and the number

of edges in an NNG is $\|V\|$. The edge starting at a node is pointing toward its most similar neighbor. For a sequence of graph nodes, if the starting and ending nodes coincide, we call it a cycle (see Fig. **??**).

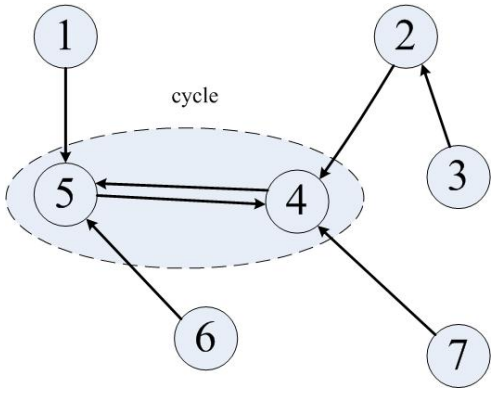


Figure 2.4 A possible NNG of the RAG in Fig. **??** and a cycle in the NNG.

It is easy to verify that the NNG has the following properties [**?**]:

1. Along any directed edge in NNG, the weights are non-increasing.

2. The maximum length of a cycle is two.

3. NNG contains at least one cycle.

4. The maximum number of cycles is $\lfloor \|V\|/2 \rfloor$.

From the definition of merging predicate in Eq.(**??**), we see that when the predicate value between two regions is true, there is exactly a cycle between them. This demonstrates the stop criterion for the proposed DRM algorithm, i.e., if there is no cycle in the NNG, the region merging will stop. In other words, before the process stops, we can always have at least one pair of regions to merge according to the

Table 2.3 Algorithm 3: Accelerating the dynamic region merging process.

Input: the initial RAG and NNG of the image.
Output: The region merging result (in the form of RAG).
1. Set $i = 0$.
2. For the NNG in the i-th graph layer, find the minimum weight edge of the RAG using the cycles.
3. Use **Algorithm 1** to check the value of the predicate P. If P is between the cycle is true, merge the corresponding pair of regions.
4. Update the RAG, NNG and the cycles.
5. Set $i = i + 1$.
6. Go back to step 2 until no cycle can be found.
7. Return RAG.

above property (3). This suggests that we can keep the NNG cycles during the region merging process instead of searching over the whole RAG. The original RAG is constructed from the over-segmented image, and the NNG is formed by searching for the most similar neighbors of each graph node. The NNG cycles are identified by a scan of the NNG. The accelerated region merging process is described in Table **??**.

Now we have the following theorem to demonstrate that the regions merging process will go on until the predicate is false for all pairs of regions in the image.

Theorem 3. In the DRM algorithm, there is at least one pair of regions to be

merged in each iteration before the stopping criterion is satisfied.

Proof. According to the property (3) of the NNG, there is always at least one cycle in the graph (the number of graph nodes should be greater than 1). Therefore, the region merging process will continue until the condition (b) in Eq.(**??**) is not satisfied. ■

If only use RAG in the merging process, update of the RAG requires $O(\|V\|)$ searching time. The computation is usually expensive due to a large number of nodes in RAG. In Algorithm 3, after the nodes of a cycle are merged, the weights of the neighboring RAG and the structure of NNG are modified. There is an observation that the new cycles can only form in the second order neighborhood of the merged nodes, which is illustrated in Fig. **??**. After merging nodes 1 and 2, the closest neighbor of node 3 becomes node 4, and that of node 4 becomes node 5. A new cycle is formed between nodes 4 and 5, and by no means to be between nodes 5 and 6. In such a way, the causes of new NNG cycles can only be detected in the second order neighborhood of merged nodes. Hence, the computational effort for updating the NNG at each merge of region pair depends on the distribution of the second order neighborhood size in the RAG. The computation time for a merge of NNG cycle is $O(\gamma^{(2)} + 1)$, where $\gamma^{(2)}$ is the size of the second order neighborhood of the new node. In most cases, $\gamma^{(2)}$ is far less than $\|V\|$, which indicates the reduction of computation time by the accelerating algorithm.

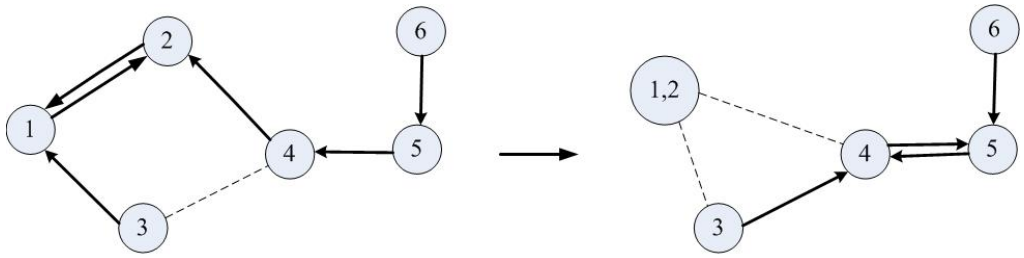


Figure 2.5 An example of NNG modification. Dotted lines represent the RAG edges, while directed lines represent NNG edges.

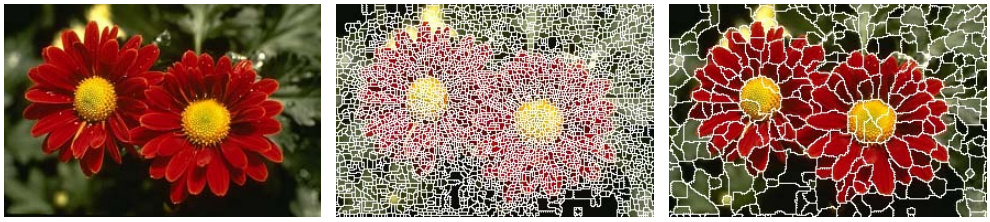
2.6 Experimental Results and Discussions

In this section, we evaluate the proposed DRM algorithm on the Berkeley Segmentation Dataset (BSDS) [1] , which contains 100 test images with 5-10 human segmentations on each one of them as the ground-truth data. In Section **??**, we test the DRM algorithm with several representative examples. In Section **??**, we compare the well known mean-shift algorithm [**?**] and the graph-based region merging method [**?**] with the DRM algorithm. In Section **??**, the performance of the accelerated DRM algorithm is evaluated. There are several free parameters in the proposed algorithm, and we test the effects of them in Section **??**. Some discussions on the DRM method and its potential extensions will be referred in Section **??**.

[1] http://www.cs.berkeley.edu/projects/vision/grouping/segbench/

2.6.1 Analysis on DRM by representative examples

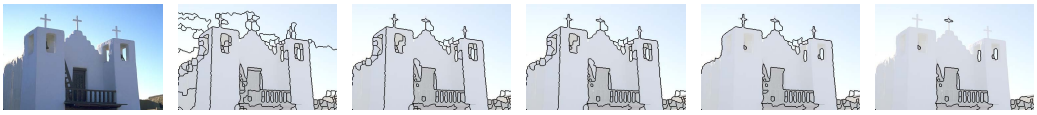
The proposed DRM algorithm starts from an initially over-segmented image. For simplicity, we use the watershed algorithm to obtain the initial segmentation. Certainly, a more sophisticated initial segmentation method, such as the mean-shift algorithm [**?**], may lead to a better final segmentation result. Since the standard watershed algorithm is very sensitive to noise and hence leads to severe over-segmentation (see Fig **??** for an example), to reduce noise and trivial structures we apply median filtering on the gradient image before using the watershed algorithm. Fig. **??** shows the result of the modified watershed segmentation, where the over-segmentation is reduced a lot while preserving the desired the object boundaries.



(a) (b) (c)

Figure 2.6 (a) Original image; (b) initial segmentation by standard watershed algorithm; and (c) initial segmentation by modified watershed algorithm with median filtering on the gradient image.

As stated in Section **??**, the values of λ_1, λ_2 in the cue distribution functions (refer to Eq. (**??**)) control the degree of consistency between two regions, and hence decide when to terminate the region merging. In practice, we set λ_2 to be a constant value 1, and hence there is only one parameter λ_1 to be set. By experimental experience, we set λ_1 between 0.1 and 5. Fig. **??** shows the segmentation results



(a) (b) $\lambda_1 = 0.5$ (c) $\lambda_1 = 1$ (d) $\lambda_1 = 1.2$ (e) $\lambda_1 = 1.4$ (f) $\lambda_1 = 1.8$

Figure 2.7 (a) Original image (b-f) Segmentation results with λ_1=0.5,1,1.2,1.4,1.8, respectively.

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

Figure 2.8 Region merging process. (a) the original image; (b) the result of initial watershed segmentation; (c)-(f) the merging results in different stages.

of an example image with different values of λ_1. When λ_1 is large, the algorithm is more likely to take two neighboring regions as consistency. Therefore it may produce over-merged regions (see the result where $\lambda_1 = 1.8$). When λ_1 is small, the case is inverse and as a result produces under-merged regions (see the results where $\lambda_1 = 0.5, 1$). To clearly show the region merging process, we give an example in Fig. **??**, in which primitive regions (in Fig. **??**) are merged iteratively until the stop criterion is satisfied (in Fig. **??**).

We illustrate the proposed DRM algorithm using more example images in Fig. **??**. It is clear that neighboring regions with coherent colors are merged into one, whereas the boundaries are well located on the reasonable places. Some of the large regions have substantial variations inside (Figs. **??**, **??**,**??**), however, with relatively slow changes of colors along the boundaries. This indicates that the DRM algorithm can tolerate some variations for grouping meaningful regions in an image.

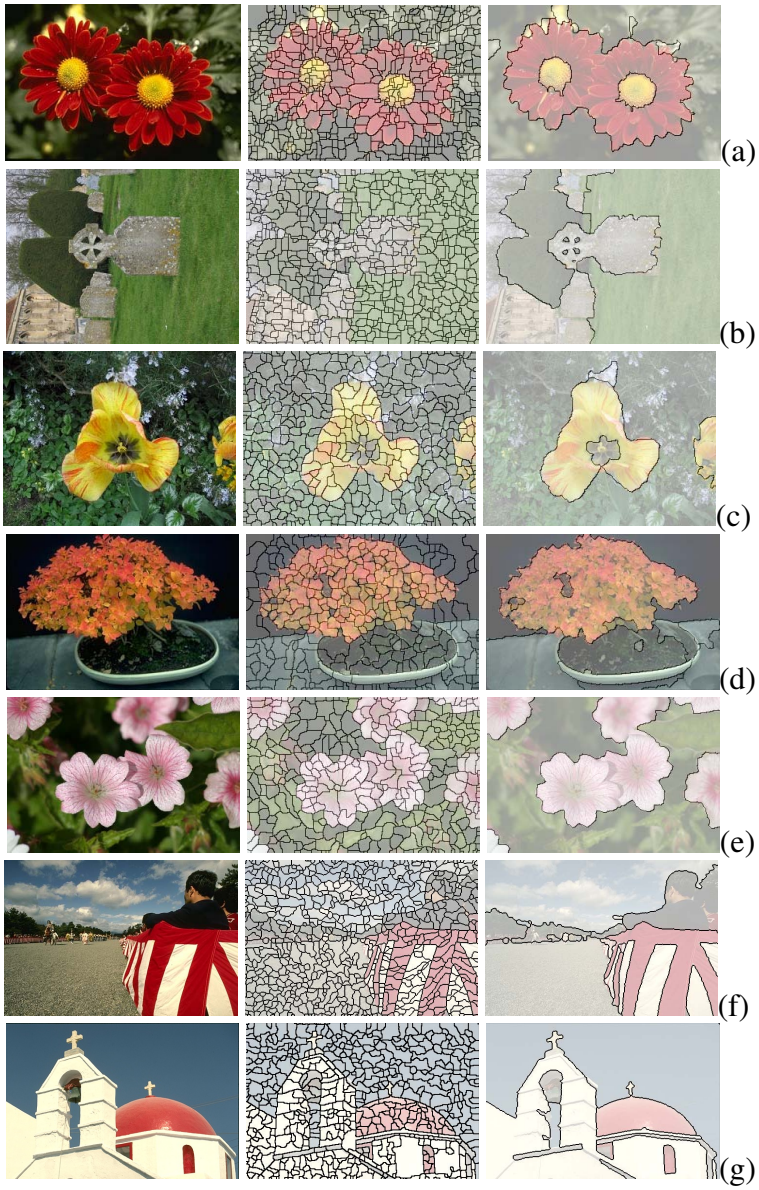


Figure 2.9 Segmentation results by the proposed algorithm. From left to right, the first column shows the original images. The second column shows the over-segmentation produced by watershed algorithm. The third column shows our segmentation results.

2.6.2 Comparison with some well-known methods

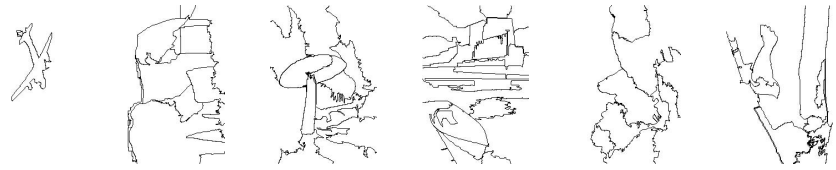
In this section we compare the segmentation results of DRM with the mean-shift algorithm [**?**] and the graph-based algorithm [**?**], which are among the well-known works in automatic image segmentation. Mean-shift performs the segmentation by clustering the data into several disjoint groups. The cluster centers are computed by first defining a spherical window, then shifting the window to the mean of the data points with a radius r. This shifting process repeats until convergence. The segmented image is constructed using the cluster labels. The mean-shift algorithm takes the feature (range) bandwidth, the spatial bandwidth, and the minimum region area (in pixels) as input. We need to adjust these three parameters for a good segmentation. In the graph-based algorithm [**?**], the predicate is based on both of the external and the internal intensity differences of the regions. More specifically, the external difference is calculated as the minimum weight edge between two neighboring regions and the internal difference of a region is the largest weight in the minimum spanning tree of that region. The predicate is then defined by comparing these two values to make an adaptive decision with respect to the local characteristics of an image. The graph-based algorithm [**?**] also contains three parameters, which are used to smooth the image, to control the minimum region size and to set the expected number of regions. All of these parameters are in different ranges, making the tuning of parameters tricky. As stated in Section **??**, the proposed DRM algorithm contains five free parameters (m,λ_1,λ_2,α,β) in the consistency test. In implementation, we fix λ_2 to be 1, both α and β to be 0.05 and m to be half size of the region. Hence only the value of λ_1 is needed to tune. In this sense, DRM is much more convenient to control than the mean-shift algorithm [**?**] and the graph-based

algorithm [**?**].

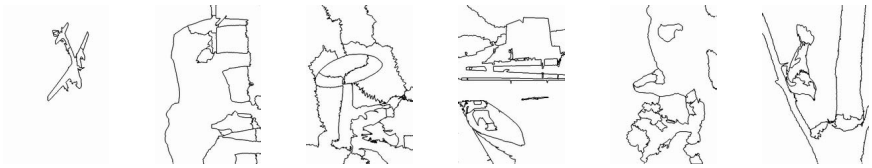
Fig. **??**-**??** show the segmentation results by the graph-based method [**?**], the mean-shift method [**?**] and the proposed DRM method. Segmentations are obtained by choosing the results with roughly the same number of regions from the three methods. By this setting, we can compare the performance of them in the same segmentation granularity. In Figs. **??**-**??**, some major boundaries in the ground truth are missed, and some boundaries are over-detected. In Fig. **??**, we can see that the proposed method can retain good localization of boundaries. In Fig. **??**, the human labeled segmentation is used as the reference (ground truth) for visual evaluation of the segmentation quality. For each image, 5-10 segmentations produced by different human observers are chosen, and the consistency among different human segmentations is indicted by the probability-of-boundary (Pb) images (Fig. **??**), where the darker pixels means more people marked them as a boundary. For more information about the generation of human labeled ground truth segmentation, please refer to [**?**].

Let's then discuss about the quantitative measure of segmentation quality. This task might be as hard as segmentation itself. In the past decades, many researchers have been studying the supervised evaluation methods, where the segmented images are compared with a ground truth image. However, in unsupervised image segmentation, often there is no ground truth available. In order to build a common platform for researchers to evaluate various image segmentation methods, a group of human segmented images for each test sample are provided in the Berkeley Segmentation Dataset (BSDS), and a boundary-based evaluation method was proposed by Martin et al. [**?**] on this dataset. In [**?**], the precision-recall frame-

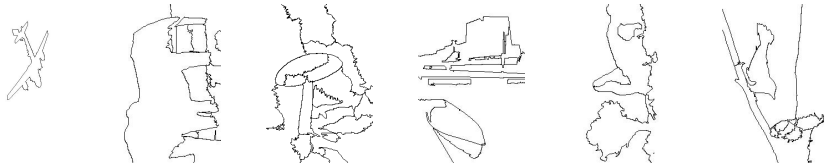
(a) Original images



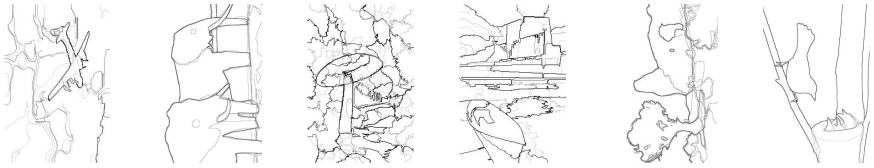
(b) Segmentation results by the graph-based region merging method in [**?**].



(c) Segmentation results of mean-shift method in [**?**].



(d) Segmentation results of the proposed method with watershed initializations.



(e) Ground truth.

Figure 2.10 Segmentation results by different methods. The first row shows the original images. The second row shows the results by the graph-based image segmentation method [**?**]. The third row shows the results by the mean-shift method [**?**]. The forth row shows the proposed method. The fifth row shows probability-of-boundary images of the human segmentation, where the darker boundaries indicate more subjects marked a boundary [**?**].

work is applied in conjunction with the human-marked boundaries. Precision-recall is a well-accepted measure technique in image segmentation, which considers two aspects of boundary qualities: *precision* is the fraction of detections that are true positives rather than false positives, while *recall* is the fraction of true positives that are detected rather than missed. A combination of these two quantities can be summarized as the F-measure in [**?**]:

$$F = PR/(\tau R + (1 - \tau)P)$$

where τ is a relative cost between P and R. An F-measure curve can be obtained by changing the algorithm parameters. Since the F-measure curve is usually unimodal, the maximal one may be taken as a summary of the algorithm's performance in the sense that large value of F-measure indicates high quality of image segmentation. In our experiment, the optimal parameter is chosen for each image and the corresponding F-measure is recorded for the overall evaluation.

There are two issues needed to address when use BSDS for evaluating the segmentation results. First, the ground truth is defined by a collection of 5 to 10 human segmentations for each image. Simply uniting the humans' boundary maps is not a reasonable choice because it ignores the high overlapping boundaries among different humans. Second, when matching boundary pixels, one should avoid over-penalizing the slightly mislocalized boundaries. We use the method proposed by Martin et al. to compute the F-measure for all algorithms. In this method, the segmentation results are compared with each human segmentation separately. False positives are counted as the boundary pixels that do not match any human boundary. The recall depends on all the human segmentations, i.e. averaged over different human data. A merit of Martin et al.'s method is that it can tolerate some local-

Table 2.4 The best F-measures by the competing methods in the BSDS dataset. The average F-measure for the human subjects is 0.79 [**?**], which represents the human performance for the segmentation task. The DRM algorithm is tested when initialized by watershed and mean-shift respectively.

Method	Human	[**?**]	[**?**]	[**?**]	DRM (watershed)	DRM (mean-shift)
F-measure	0.79	0.62	0.62	0.65	0.65	0.66

ization errors in the matching process, which is a reasonable consideration in the correspondence of ground-truth and segmentation results. In Table **??**, we compare the F-measures of the Canny edge detector [**?**], the graph-based method [**?**], the mean-shift method [**?**] and the DRM algorithm. The best parameter is chosen for each of these methods, and we have the F-measures as 0.62 for Canny edge detector, 0.62 for the graph-based method [**?**], 0.65 for mean-shift method [**?**] and 0.65 (with watershed as initial segmentation) and 0.66 (with mean-shift as initial segmentation) for the proposed DRM method.

2.6.3 The performance of accelerated DRM algorithm

In Fig. **??**, we show the number of RAG regions and the number of NNG cycles for images Fig. **??**-**??** at different region merging stages. Apparently, the number of cycles (in red) is much smaller than that of the RAG edges (in blue). By using the accelerated region merging algorithm, the computational effort is largely saved. The update of NNG cycles depends on the size of RAG in the second order neighborhood. In Fig. **??**, we show the histogram of RAG node degrees for the over-segmented images in Fig. **??**-**??** (in the second column). We can see that

most of the nodes have the degree less than 15, therefore the computational cost for updating the NNG cycles is not expensive. This acceleration algorithm would be very helpful in the high-resolution environment where a massive graph containing billions of vertices will exist. In our test of 100 Berkeley images, the running time of the proposed algorithm ranges from 5 to 15 seconds on a PC with Intel Core 2 Duo 2.66 GHz CPU, 2GB memory. It is largely depended on the number of regions in the initial segmentation and that in the final segmentation.

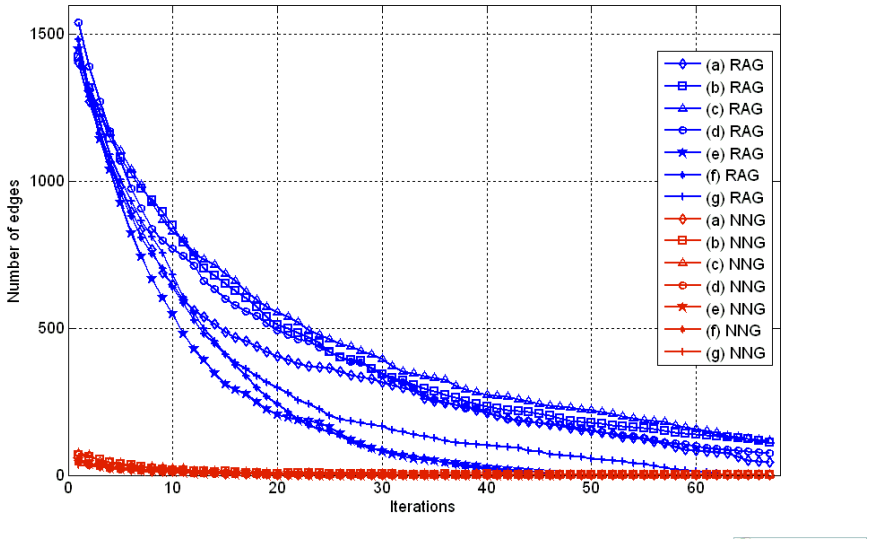


Figure 2.11 The number of RAG edges (in blue) vs. the number of NNG cycles (in red) for images in Fig. **??**-Fig. **??** at different region merging stages.

2.6.4 The choices for parameters

In the proposed DRM method, there are five free parameters $\{m, \lambda_1, \lambda_2, \alpha, \beta\}$ which control the consistency hypothesis evaluation. In implementation, we fixed four of

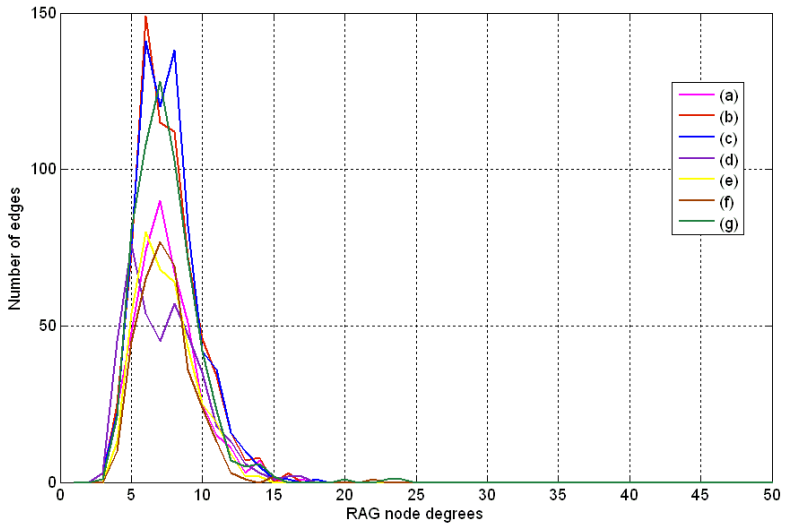


Figure 2.12 Histograms of the RAG node degrees for images in Fig. **??**-Fig. **??** (shown in different colors).

them (i.e., m, λ_2, α, β) to be constant. However, it is still worth investigating their impact on the stability and robustness of the whole segmentation process.

In SPRT, m is used to decide the amount of data selected for the random test. We empirically choose it to be the half size of the region, as in the experiments we found that the segmentation results are not sensitive to the moderate changes of m.

Then we test the choices of parameters α and β, which represent the probability of accepting an "inconsistent" model as "consistent" and rejecting a "consistent" model as "inconsistent", respectively. In Fig. **??**, we show 120 segmentation results of Fig. 9(a) in the form of F-measures, which are produced by different pre-specified (α, β) values. In the experiment, α ranges from 0.01 to 0.26 and β ranges from 0.01 to 0.4. In general, we can see that the lower the test error is, the higher the obtained segmentation quality is. In our implementation, both α and β are set to

a fixed value 0.05. Theoretically, the values of α and β will also affect the number of tests needed for the SPRT, as shown in Eq. (**??**) proposed by Wald. The upper bound of the number of tests N_0 is set based on the Turncated SPRT [**?**]. This has been explained in Section **??**.

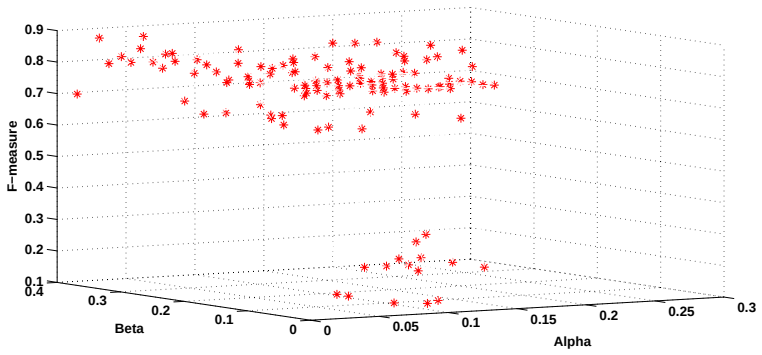


Figure 2.13 The F-measures of Fig. **??** with 120 pre-specified (α, β) values. In the experiment, α ranges from 0.01 to 0.26 and β ranges from 0.01 to 0.4.

The last experiment is performed to test the choices of parameter λ_1, which is the only user input parameter in the implementation. As defined in Eq. (**??**) and Eq. (**??**), if we set $\lambda_2 = 1$, λ_1 can be used to balance the relative weight of P_0 and P_1 so that only λ_1 needs to be tuned. For a small value of λ_1, it requires stronger evidence for a boundary. In other words, the over-merging is not likely to happen with small values of λ_1. The control of this parameter leads to a hierarchy of segmentations at different scales. Table **??** shows the number of segmented regions under different values of λ_1 (the rest parameters are fixed in this experiment). As λ_1 increases, the number of regions tends to decrease. In our experiments, the value of λ_1 is chosen from 0.1 to 5. Notice that it is hard to find a generally appropriate λ_1 for all the images, since the variety and multiscale interpretation of the image content.

Table 2.5 The number of regions w.r.t. different values of λ_1. Indices of images are corresponding to the original images in Fig. **??** (from the left to the right)

	λ_1	0.3	0.4	0.8	1.0	1.01	1.02
Fig.10-1	No. of regions	28	27	24	15	7	2
	λ_1	0.3	0.8	0.92	0.96	0.98	1.0
Fig.10-2	No. of regions	167	150	134	117	82	67
	λ_1	0.9	0.98	1.2	1.4	2.0	2.3
Fig.10-3	No. of regions	375	285	130	126	93	72
	λ_1	1.0	1.2	1.4	2.1	3.5	4.0
Fig.10-4	No. of regions	235	124	95	91	80	17
	λ_1	1.0	1.1	1.2	1.5	1.9	2.0
Fig.10-5	No. of regions	445	225	178	148	92	21
	λ_1	0.8	0.9	0.94	1.0	1.2	1.3
Fig.10-6	No. of regions	192	181	174	121	102	27

2.6.5 Discussions

This paper presents a framework for region merging. The simple cue, i.e., region color, is used in the implementation. Naturally, different cues (e.g., color, texture, shape, etc.) or a combination of them can be explored by changing the definition of conditional probabilities in SPRT (Eq. (**??**)). The region merging predicate uses the minimal weight edge between two regions to measure the difference between them. It is guaranteed that some global properties can be obtained under this predicate. However, the predicate might lead to a limitation for capturing the perceptual differences between two neighboring local regions. Some failure examples are given

in Fig. **??**. We can see that the proposed DRM algorithm may miss some long but weak boundaries (1^{st} example), and it may also merge the regions with short but high contrast boundary (2^{nd} example).

There are several potential extensions to DRM to solve the above problems. For example, we can add a global refinement step to correct the miss-classified regions due to local decisions. Many merging errors in DRM are due to the insufficient local (perceptual) information used to making the merging decision. However, if we view the outputted labels of DRM as initial markers of the image content, then these initial markers can be used to calculate some global statistics of the image so that a global refinement can be defined to correct the DRM errors according to some criterion. Such a DRM with refinement scheme can exploit both local and global image features, and hence better results can be expected.

Another potential extension is the introduction of user interaction. With some user guidance, the initial label of part of the image regions can be assigned beforehand, which will provide useful information for the region merging process. An interactive DRM algorithm can then be developed to accomplish the image segmentation.

2.7 Conclusions

In this chapter, we proposed a novel method for segmenting an image into distinct components. The proposed algorithm is implemented in a region merging style. We defined a merging predicate P for the evidence of a merging between two neighboring regions. This predicate was defined by the sequential probability

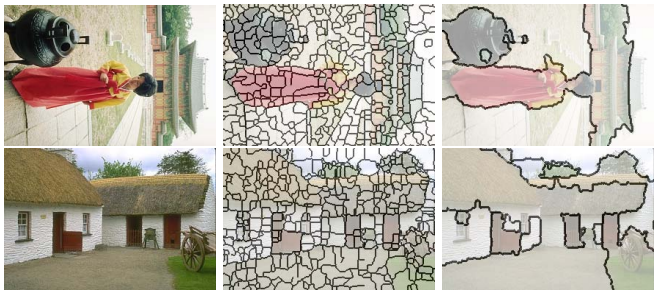


Figure 2.14 Failure examples of the proposed DRM method. The first column shows the original images, the second column shows the initial segmentation by watershed algorithm, and the third column shows the region merging results by the proposed method. We see that DRM methods may miss some weak boundaries (in the 1^{st} example) and merge the regions with short but high contrast boundary (in the 2^{nd} example).

ratio test (SPRT) and the maximum likelihood criterion. A dynamic region merging (DRM) was then presented to automatically group the initially over-segmented many small regions. Although the merged regions are chosen locally in each merge stage, some global properties are kept in the final segmentations. For the computational efficiency, we introduce an accelerated algorithm by using the data structure of region adjacency graph (RAG) and nearest neighbor graph (NNG). Experiments on natural images show the efficiency of the proposed algorithm. There are several potential extensions to this work, such as the introduction of global refinement and user interaction, etc. Those will be further investigated in our future work.

Chapter 3

Image Segmentation by Iterated Region Merging with Localized Graph Cuts

3.1 Introduction

While it has been widely studied for many decades, automatic image segmentation is still a big challenge due to the complexity of image content. A lot of work shows that the user guidance can help to define the desired content to be extracted and thus reduce the ambiguities produced by the automatic methods. In this chapter we consider the most common type of interactive segmentation: segmenting the object of interest from its background.

In the past few years, various approaches to interactive segmentation have been proposed. For example, livewire [**?**] allows the user to interactively select certain

pixels where the segmentation boundary should pass. However, high complexities of object shapes (e.g. intricate shapes with lots of protrusions and indentations) might lead to many interactions for an acceptable segmentation. And images with the large size will require more computational time. To obtain real time response to the user's actions, independent of the image size, Falcão [**?**] proposed a modified livewire method, which exploits three properties of Dijkstra's algorithm to compute minimum-cost paths in sub-linear time. Active contour, or snake [**?**], is defined as an energy-minimizing spline. After initializing the contour close to the original object boundary, the contour will fit the actual object boundary iteratively. Level sets based segmentation method [**?**] uses implicit active contour models, in which the numerical computation involving curves and surface is performed without having to parameterize the objects.

Another preferable interactive segmentation method based on combinatorial optimization is graph cuts [**? ?**]. It addresses segmentation in a global optimization framework and guarantees a globally optimal solution to a wide class of energy functions. In addition, the user interface of graph cuts is convenient - seeds can be loosely positioned inside the object and background regions, which is easier compared to placing seeds exactly on the boundary, like in livewire [**?**]. Because graph cuts can involve a wide range of visual cues, a number of recent literature further extended the original work of Boykov and Jolly [**?**] and developed the use of regional cues [**? ?**], geometric cuesv[**? ?**], shape cues [**? ?**], stereo cues [**?**], or even topology priors [**?**] as global constraints in the graph cuts framework. When foreground and background color distributions are not well separated, the traditional graph cuts [**?**] can not achieve satisfying segmentation. Some advanced versions

of graph cuts are developed [**?** **?** **?** **?**], which are more robust and substantially simplify the user interaction. In [**?**], the user interaction can be applied on both coarse and fine scales, which inherit the advantages in region and boundary based methods for image segmentation. The work proposed in [**?**] makes a progressive local selection on the object of interest. Instant visual feedback is provided to the user for a quick and effective image editing.

In the classical graph-based framework, most of segmentation methods consider pixels or groups of pixels as the nodes in a graph. The edge weight estimation usually takes into account image attributes, for example color, gradient and texture. An efficient edge weight assignment method was proposed by Miranda et al. [**?**], where the object information obtained from user interaction as well as the image attributes are both used for estimating edge weights. Separating from the image segmentation process, it can act as a basic step for high accuracy image segmentation. Some other works studied graph structures for designing image processing operators. Image foresting transform (IFT) [**?** **?**], for example, defines a minimum-cost path forest in a graph, and provides a mathematically sound framework for many image processing operations. Based on similar graphs, a theoretical analysis between optimum-path forests and minimum cut was given in [**?**]. Under some conditions, the two algorithms were proven to produce the same result.

In our preliminary work [**?**], we explore the graph cuts algorithm by extending it to a region merging scheme. Starting from seed regions given by the user, graph cuts is conducted on a propagated sub-graph where the regions are regarded as the nodes of the graph. An iterated conditional mode (ICM) is studied and the maximum a posterior (MAP) estimation is obtained by virtual of graph cuts on

each growing sub-graph. The segmentation process is stopped when all the regions are labeled. In [**?**], the initial segmentation is obtained by Mean-shift algorithm, which is a sophisticated segmentation technique. While in this paper, the initial segmentation is obtained by the simple watershed algorithm [**?**]. In each iteration, a semi-supervised algorithm is applied to learn a classifier. Consequently, the most confident labels will contribute for new seed regions in the next iteration.

The proposed method is a novel extension of the standard graph cuts algorithm. Rather than segmenting the entire image all at once, the segmentation is performed incrementally. It has many advantages to do this. First of all, using sub-graph significantly reduce the complexity of background content in the image. The many unlabeled background regions in the image may have unpredictable negative effect on graph cuts optimization. This is why the global optimum obtained by graph cuts often does not lead to the most desirable result. However, by using a sub-graph and blocking those unknown regions far from the labeled regions, the background interference can be much reduced, and hence better results can be obtained under the same amount of user interaction. Second, the algorithm is run on the sub-graph that comprises object/background regions and the surrounding un-segmented regions, thus the computational cost is significantly less than running graph cuts on the whole graph which is based on image pixels. Third, as a graph cuts based region merging algorithm, our method obtains the optimal segmentation on each sub-graph. In interactive image segmentation, user input information helps to enhance the discontinuities between object and background by constructing color data models [**?**], which represent object and background respectively. Some simple methods such as color histograms can be used to calculate these models. In this work, the

construction of the object and the background color models are obtained from the most confident labels by a learned classifier. This scheme automatically collects more reliable information for the next round of segmentation.

Although the user input is helpful in steering the segmentation process to reduce the ambiguities, too much interaction will lead to a tedious and time-consuming work. If the object is in a complex environment from which the background can not be trivially subtracted, a significant amount of interaction may be required. Moreover, the complex content of an image also makes it hard to provide user guidance for accurate segmentation while keeping the interaction as less as possible. Therefore, some algorithms allow further user edit based on the previous segmentation results [**? ? ? ?**] until the desired result is achieved. In comparison to the traditional graph cuts algorithm, the proposed method is able to reduce the amount of user interaction needed for a desirable segmentation result, or that given a fixed amount of user interaction it increases the quality of the final segmentation result. Experiments show that with poor initialization (i.e. user inputs), the segmentation results of standard graph cuts algorithm might be far from what we expect, while the proposed method can still offer good results. In addition, much better segmentation results can be achieved by the proposed method for images with complex background.

3.2 Image Segmentation by Graph Cuts

Image segmentation can be naturally taken as a labeling problem. Given a set of labels L and a set of sites S (e.g, image pixels or regions),our goal is to assign

each of the sites $p \in S$ a label $f_p \in L$. The graph cuts framework proposed by Boykov and Jolly [**?**] addresses the segmentation on binary images, which solves a labeling problem with two labels. The label set is $L = \{0, 1\}$, where 0 corresponds to the background and 1 corresponds to the object. Therefore, labeling is a mapping from S to L and is denoted by $f = \{f_p | f_p \in L\}$, i.e. label assignments to all pixels. An energy function in a "Gibbs" form is formulated as:

$$E(f) = E_{data}(f) + \lambda E_{smooth}(f) \tag{3.1}$$

The data term E_{data} consists of constraints from the observed data and measures how sites like the labels that f assigns to them. It is usually defined to be:

$$E_{data}(f) = \sum_{p \in S} D_p(f_p) \tag{3.2}$$

where D_p measures how well label f_p fits site p. For example, we can use intensities of marked sites (seeds) to learn the histograms for the object and the background intensity distributions $Pr(I|"obj")$ and $Pr(I|"bkg")$. Then D_p can be expressed as follows:

$$D_p("obj") = -lnPr(I_p|"obj") \tag{3.3}$$

and

$$D_p("bkg") = -lnPr(I_p|"bkg") \tag{3.4}$$

D_p is the penalty of assigning the label f_p to site $p \in S$. The negative log-likelihoods should be small if p likes f_p and vice versa. E_{smooth} is called the smoothness term and measures the extent to which f is not piecewise smooth. The typical form of E_{smooth} is:

$$E_{smooth} = \sum_{\{p,q\} \in \mathcal{N}} V_{pq}(f_p, f_q) \tag{3.5}$$

where $\mathcal{N}$ is a neighborhood system, such as a 4-connected neighborhood system or an 8-connected neighborhood system. The smoothness term typically used for image segmentation is the Potts Model [**?**], which is

$$V_{pq}(f_p, f_q) = \omega_{pq} \times T(f_p \neq f_q) \tag{3.6}$$

where:

$$T(f_p \neq f_q) = \begin{cases} 1 & \text{if } f_p \neq f_q \\ 0 & \text{otherwise} \end{cases}$$

The model (**??**) is a piecewise constant model because it encourages labelings consisting of several regions where sites in the same region have the same labels. In image segmentation, we want the boundary to lie on the intensity edges in the image. A typical choice for $\omega_{p,q}$ is as follows:

$$\omega_{pq} = e^{-\frac{|I_p - I_q|^2}{2\delta^2}} \cdot \frac{1}{dist(p,q)} \tag{3.7}$$

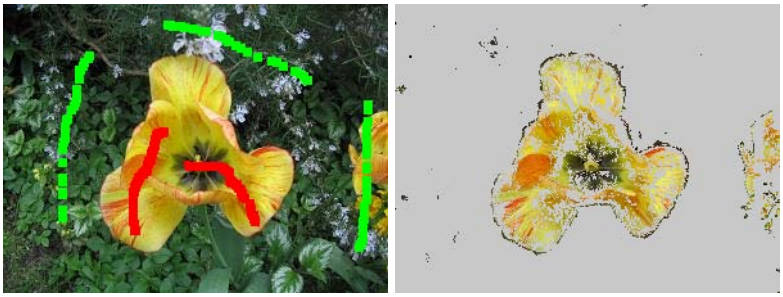
For gray images, I_p and I_q are the intensities of site p and q. For color images, they are taken placed by the notations of $\vec{I_p}$ and $\vec{I_q}$, which can be the LAB color vectors of sites p and q. $dist(p,q)$ is the distance between sites p and q. Parameter δ is related to the level of variation between neighboring sites within the same object. The parameter λ is used to control the relative importance of the data term versus the smoothness term. If λ is very small, only the data term matters. In this case, the label of each site is independent from the other sites. If λ is very large, all the sites will have the same label. Minimization of the energy function can be done using the min-cut/max-flow algorithm as described in [**?**].

Now we need to construct a graph corresponding to the energy function (**??**). There are two additional nodes: the source terminal s and the sink terminal t, representing the object and the background respectively. Each node in the graph is

connected to s and t by two t-links. And each pair of neighboring nodes is connected by an n-link. The weights of t-links for seed pixels can be seen as hard constraint imposed on the segmentation. In initialization, the user will mark some pixels as the object or the background so that these pixels will keep their initial labels in the final result. If pixel p is marked as an object label, the edge between p and s should be set to infinity and the edge between p and t should be set to zero. N-links correspond to the penalty for discontinuity between the two neighboring pixels. They are derived from the smoothness term E_{smooth} in energy function (**??**). And the weight of a t-link corresponds to a penalty for assigning the label to the pixel. It will be derived from the data term E_{data} in the energy function (**??**).

3.3 Iterated Region Merging with Localized Graph Cuts

3.3.1 Initial Segmentation by Modified Watershed Algorithm



(a) (b)

Figure 3.1 (a) Original image with user input seeds. The background seeds are in green, and object seeds are in red. (b) The segmentation results by standard graph cuts.

In the original graph cuts algorithm [**?**], the segmentation is directly performed on the image pixels. There are two problems for such a processing. First, each pixel will be a node in the graph so that the computational cost will be high; second, the segmentation result may not be smooth, especially along the edges. Fig.**??** shows an example of the graph cuts segmentation result. It can be seen that although there should be clear boundary between the object and background, the graph cuts fails to give a smooth segmentation map by labeling some object pixels as background, or vice versa. Actually, in the early work of Wu and Leahy [**?**], it was noticed that the minimum cut criteria favored cutting small sets of isolated nodes in the graph.

To alleviate this problem, Veksler [**?**] included a shape constraint in the graph cuts energy function, which encourages a long object boundary. Some other segmentation criterions were also proposed to solve this problem, such as normalized cuts [**?**] and ratio cut [**?**]. In this paper, we adopted a relatively simple but effective strategy to solve this problem by introducing some low level image processing techniques to graph cuts. In [**?**], Li et al. used watershed [**?**] for initial segmentation to speed up the graph cuts optimization process in video segmentation. With such initialization, the image can be partitioned into many small homogenous regions, and then each region, instead of each pixel, is taken as a node in the graph. In this way the computational cost can be reduced significantly, while the object boundary can be better preserved. The watershed technique is also used in this paper with some modification. Watershed algorithm produces coherent over-segmented regions which preserve most structures of the interest object. However, the standard watershed algorithm is very sensitive to noise and thus leads to severe over-segmentation (see Fig. **??**). There are some edge-preserving smoothing

techniques, such as median filtering, can help to reduce noise and trivial structures. Therefore, to reduce over-segmentation, we apply median filtering on the gradient image before conducting the watershed algorithm. Fig. **??** shows an example. Fig. **??** is the gradient image of the original image in Fig. **??** and Fig. **??** is the watershed segmentation of it. Clearly, there is a severe over-segmentation in Fig. **??**. Such small regions are not reliable for calculating the region statistics and they will also increase the computational cost in our region merging algorithm Fig. **??** is the median filtering output of the gradient image in Fig. **??**, and Fig. **??** is the watershed segmentation result on it. We see that the over-segmentation is significantly reduced, while the contour of the object is well preserved. Note that we can use more sophisticated initial segmentation techniques in the proposed method. To weaken the importance of initial segmentation, the watershed algorithm is adopted for its simplicity.

3.3.2 Iterated Conditional Mode

Although graph cuts technique provides an optimal solution to the energy function (**??**) for image segmentation, the complex content of an image makes it hard to precisely segment the whole image all at once. In the proposed region merging based segmentation algorithm, the one-shot minimum cut estimation algorithm is replaced by a novel iterative procedure, in which the object/background distributions are updated according to the previous segmentation results and new nodes are added until the whole image is segmented. This problem is studied in a way like the iterated conditional mode (ICM) proposed by Besag [**?**], where the local conditional probabilities is maximized sequentially.

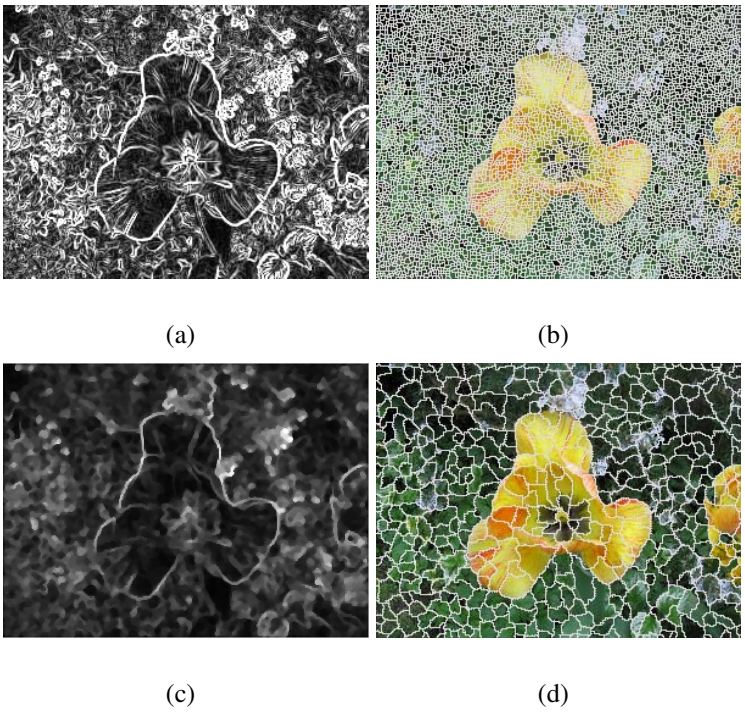


Figure 3.2 Initial segmentation using modified watershed algorithm. (a) is the gradient image of Fig. **??**; (c) is the median filtering result of (a); (b) and (d) are the watershed segmentation results of (a) and (c) respectively. We see that the over-segmentation is significantly reduced in (d).

In computer vision, an image can be represented by a graph $G =< V, E >$, where V is a set of nodes corresponding to image elements (e.g. pixels, regions), and E is a set of edges connecting the pairs of nodes. We say two nodes are incident with an edge and that these nodes are adjacent or neighbors of each other. Edge weights of the graph are computed as the dissimilarity between the connected nodes (e.g. the distance of region histograms). A sub-graph $G' =< V', E' >$ can be defined such that $V' \subseteq V$ and $E' \subseteq E$. In this paper, we consider image regions as the graph nodes, and the neighborhood of a node in V' corresponds to its adjacent regions in the image. Inspired by ICM, we consider the graph-cuts algorithm in a "divide and conquer" style: finding the minima on the sub-graph and extending the sub-graph successively until reach the whole graph. The proposed method works iteratively, in place of the previous one-shot graph cuts algorithm [**?**].

Given the observed data d_p of site p, the label f_p of site p and the set of labels $f_{S-\{p\}}$ which is at the site in S-$\{p\}$, where $f_p \in L$ and S-$\{p\}$ is the set difference. We sequentially assign each f_i by maximizing conditional probability $P(f_p|d_p, f_{S-\{p\}})$ under the MAP-MRF framework. There are two assumptions in calculating $P(f_p|d_p, f_{S-\{p\}})$. First, the observed data $d_1, \ldots, d_m$ are conditionally independent given f and that each d_p depends only on f_p. Second, f depends on labels in the local neighborhood, which is Markovianity, i.e. $P(f_p|d_p, f_{S-\{p\}}) = P(f_p|f_{N_p})$, where N_p is a neighborhood system of site p. Markovianity depicts the local characteristics of labeling. With the two assumptions we have:

$$P(f_p|d_p, f_{S-\{p\}}) = \frac{P(d_p|f_p) \cdot P(f_p|f_{N_p})}{P(d)} \tag{3.8}$$

where $P(d)$ is a normalizing constant when d is given. There is:

$$P(f_p|d_p, f_{S-\{p\}}) \propto P(d_p|f_p) \cdot P(f_p|f_{N_p}) \tag{3.9}$$

where $\propto$ denotes the relation of direct proportion. The posterior probability satisfies:

$$P(f_p|d_p, f_{S-\{p\}}) \propto e^{-U(f_p|d_p, f_{N_p})} \tag{3.10}$$

where $U(f_p|d_p, f_{N_p})$ is the posterior energy and satisfies:

$$\begin{aligned} U(f_p|d_p, f_{N_p}) &= U(d_p|f_p) + U(f_p|f_{N_p}) \\ &= U(d_p|f_p) + \sum_{p' \in N_p} U(f_p|f_{p'}) \end{aligned} \tag{3.11}$$

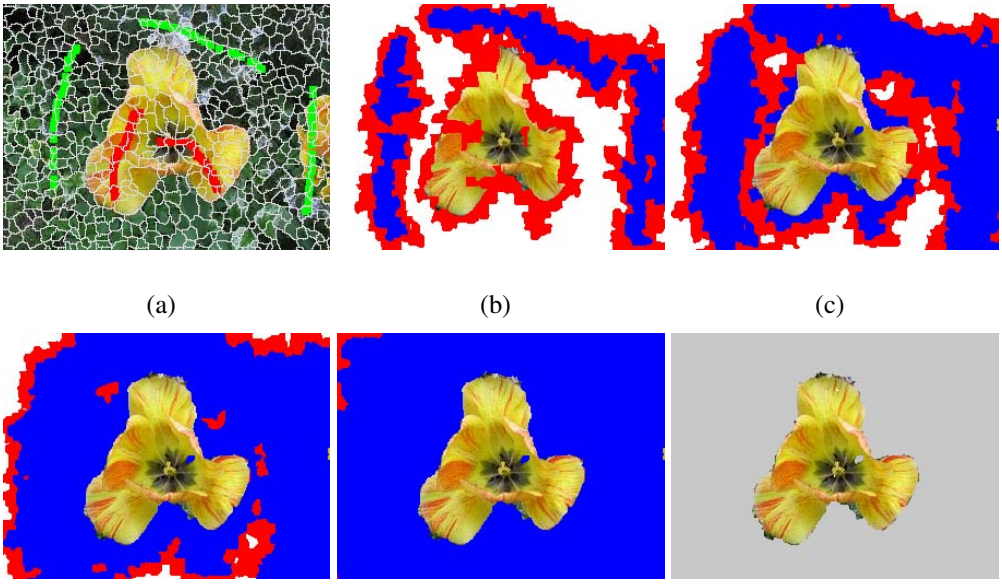
$U(d_p|f_p)$ is the data term corresponding to function (**??**), and $\sum_{p' \in N_p} U(f_p|f_{p'})$ is the smoothness term which relates to the number of neighboring sites whose labels $f_{p'}$ differ from f_p. The MAP estimate is equivalently found by minimizing the posterior energy:

$$f^{k+1} = \arg\min_{f} U(f|d, f_N^k) \tag{3.12}$$

where f_N^k is the optimal labeling of graph nodes obtained in previous k iterations. The labeling result in each iteration is reserved for later segmentation. This process is done until the whole image is labeled.

3.3.3 Iterated Region Merging

The proposed iterated region merging method starts from the initially segmented image by the modified watershed algorithm in Section **??**. Fig. **??** illustrates the iterative segmentation process by using an example. In each iteration, new regions



(a) (b) (c)

(d) (e) (f)

Figure 3.3 The iterative segmentation process. (a) Initial segmentation. (b)-(e) show the intermediate segmentation results in the 1st, 2nd, 3rd and 4th iterations. The newly added regions in the sub-graphs are shown in red color and the background regions are in blue color. We can see the target object is well segmented from the background in (f).

which are in the neighborhood of newly labeled object and background regions are added into the sub-graph, while the other regions keep their labels unchanged.

The proposed algorithm is summarized in Table **??**. The inputs consist of the initial segmentation from watershed segmentation and user marked seeds. The object and background data models are updated based on the labeled regions from the previous iteration. In Section **??**, the algorithm to construct data models will be discussed in detail.

3.3.4 Update Object/Background Models

Incorporating user input information in segmentation is one of the most interesting features of graph cuts method [**?**]. There is a lot of flexibility in how the information can be used to adjust the algorithm for a desired segmentation, for example, initializing the algorithm or editing the results. With the given information, the object and background models can be learned for formulating the data term in function (**??**), which describes how well label f_p fits site p. In step 2 of the proposed Algorithm 1 (Table **??**), the models are updated based on the previously labeled regions. However, if all the labeled regions are used to update the models, the misclassified regions will probably reinforce themselves in the next round of iteration. Therefore, we propose a semi-supervised approach in which the labeled regions in the $(i-1)^{th}$ iteration are partially selected to be the seeds for the i^{th} iteration. This model updating process is independent of the graph cuts optimization algorithm, aiming is to increase the confidence levels of the color models. The main idea of our object/background models updating process can be summarized as follows: in each iteration, a set of confident labels is chosen by a semi-supervised approach,

Table 3.1 Iterated region merging with localized graph cuts

Algorithm1 : RegionMergingGraphCuts()

Input:

– Initial segmentation of the given image.

– User labeled object regions R_o and background regions R_b.

Output: Segmentation result.

1. Build object and background data models based on labeled regions R_o and R_b.

2. Build subgraph $G' =< V', E' >$, where V' consist of R_o, R_b and their adjacent regions.

3. Update object and background data models using the SelectLabels() algorithm (refer to section **??**).

4. Use graph cuts algorithm to solve the min-cut optimization on G', i.e.

$$\arg\min_{f} U(f|d, f_N^k).$$

5. Update object regions R_o and background regions R_b according to the labeling results from step 4.

6. Go back to step 2, until no adjacent regions of R_o and R_b can be found.

7. Return the segmentation results.

such that the corresponding regions are taken as confident regions. Based on these confident regions, new object/background models are constructed for the graph cuts segmentation, as an integral step of the proposed Algorithm 1.

There are a number of semi-supervised algorithms which use both labeled and unlabeled data to build classifiers. With the merits of less human effort and higher accuracy, they are of great interest in practice. The Yarowsky algorithm [**?**] is a well-known semi-supervised algorithm, which is widely used in computational linguistics. Some variants of the original Yarowsky algorithm [**? ?**] were also developed to optimize specific objective functions. In this section, we adopt it to build better object/background models for the proposed iterated segmentation algorithm.

Suppose $\phi_x(j)$ is the probability that instance x belongs to the j^{th} class, and $\pi_x(j)$ is the score of the model in predicting label j for the region x. An object function based on cross-entropy is defined as [**?**]:

$$l(\phi, \pi) = \sum_{x \in X} H(\phi_x || \pi_x) = \sum_{x \in X} \sum_{j} \phi_x(j) log \frac{1}{\pi_x(j)} \tag{3.13}$$

The minimization of function (**??**) encourages the unlabeled data becomes labeled, and its assigned label agrees with the model prediction. Since the goal is to build color models based on previously labeled regions, we would like to choose the regions whose predictions are most confident according to the Yarowsky algorithm. With the fact that seeds regions in the $(i-1)^{th}$ iteration are confident for the graph cut segmentation, we only have to decide which are the confident regions resulting from the graph cut in the $(i-1)^{th}$ iteration. The Algorithm 2 in Table **??** describes the process of how to choose the labeled regions in the $(i-1)^{th}$ iteration for constructing color models of the i^{th} iteration, which is corresponding to step 3

Table 3.2 Algorithm of label selection for constructing color model in the i^{th} iteration

Algorithm2: SelectLabels()

Input:

– Seeds regions $Y^0 = R_o \cup R_b$

– Labeled regions X after the 1^{st} iteration, which contain Y^0 and their adjacent regions $\perp$.

Output: labeling Y^{i+1}.

1. For $i \in \{0, 1, \dots\}$ do.

2. $\wedge^i = \{x \in X | Y^i \neq \perp\}$.

3. Train classifier on $(\wedge^i, Y^i)$; resulting in π^i.

4. For each example $x \in X$

   4.1 set $\hat{y} = argmax_j \pi_x^{i+1}(j)$

   4.2 set

$$Y^{i+1} = \begin{cases} Y_x^0 & \text{if } x \in \wedge^0 \\ \hat{y} & \text{if } x \in \pi^i \vee \pi_x^{i+1}(\hat{y}) > 1/L \\ \perp & \text{otherwise} \end{cases}$$

5. If $Y^{i+1} = Y^i$, stop. Otherwise, go 1.

6. return Y^i.

in Algorithm 1.

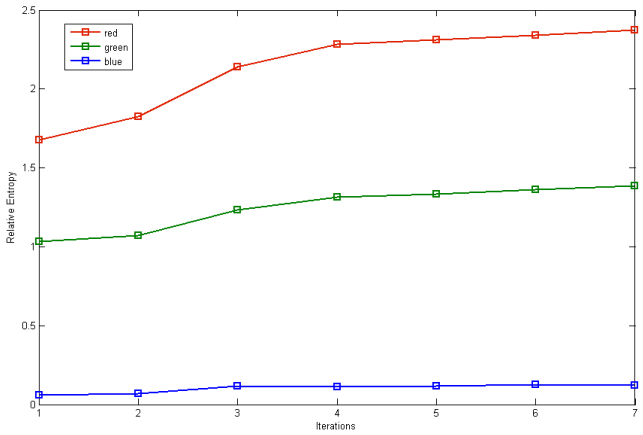


Figure 3.4 Relative entropy of the object and background distributions (Fig.3) in different iterations. The three plots represent the red, green and blue color channels respectively.

In Algorithm 2, the outer loop is given a seed set Y^0 to start with. In step 2, a labeled training set $\wedge^i$ is constructed from the most confident predictions Y^i. The score $\pi_x(j)$ is related to all the feature values in a sample x , and is given by:

$$\pi_x(j) = \frac{1}{|F_x|} \sum_{f \in F_i} \theta_{fj} \tag{3.14}$$

where $\theta_{fj} = \frac{|\wedge_{fj}|+1/L|V_f|}{|\wedge_f|+|V_f|}$, $|F_x|$ is the number of features of a region x, L is the number of labels, $|\wedge_{fj}|$ is the number of regions with label j and feature f; $|V_f|$ and $|\wedge_f|$ are respectively the numbers of unlabeled and labeled regions that have feature f. The feature used here is the average RGB color of a region. Abney [**?**] proved that the definition of score $\pi_x(j)$ can promise the object function (**??**) to decrease with the iteration number until it reaches a minimum. The predicted label for region is given in step 4.1 in Algorithm 2, where it is assumed that the classifier makes confidence-weighted predictions.

To check the relationship between the object and background distributions, we use the relative entropy to evaluate the distance between them. It is defined as the Kullback-Leibler distance from the distribution of foreground to that of the background, i.e. $D_{KL}(p||q) = \sum_{x \in X} p(x) log \frac{p(x)}{q(x)}$, where $p(x)$ and $q(x)$ are the probability density functions of the object and background respectively. Fig.**??** shows the value of relative entropy in all the 7 iterations for the image in Fig.**??**. As the value of relative entropy goes up from the first iteration, the data models of the object and background become more and more distinguishable. This leads to a higher probability of well separating the object from the background.

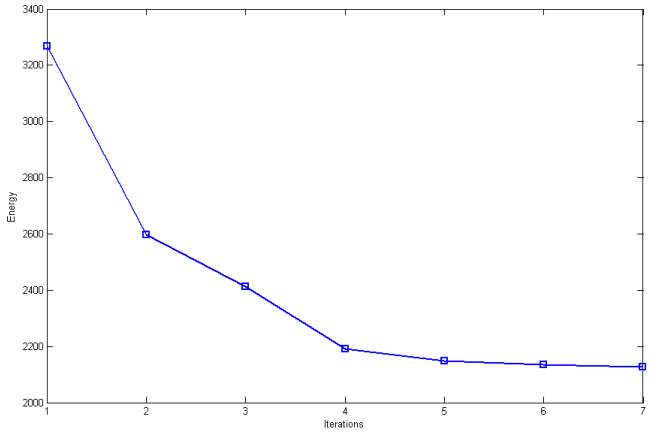


Figure 3.5 The energy evolution of the segmentation results in Fig. 3. Graph cuts energy decreases in the iterated segmentation process.

In the proposed algorithm, segmentation is obtained on different levels of sub-graphs. In light of graph cuts, the segmentation keeps the property of global optimality on each sub-graph. Adding new seeds according to the previous optimal labeling, it increases the amount of useful information that can be used for further

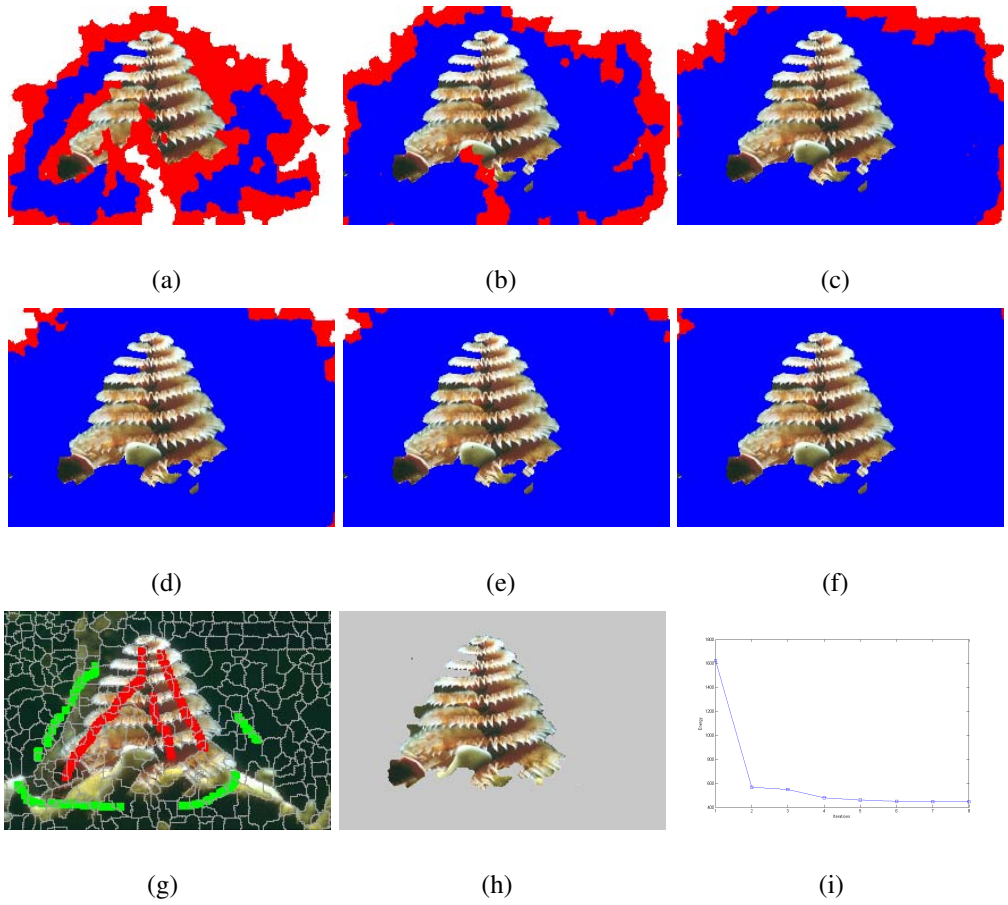


Figure 3.6 Another example of energy evolution. (a)-(f) show the object and background seeds in different iterations based on the user input seeds shown in (g). (h) shows the final segmentation result, and (i) shows the energy values, which are calculated on the whole graphs by using the seeds obtained in each iteration. We see that the energy decreases monotonically.

segmentation while avoiding introducing much interference information from unknown regions. Fig.**??** shows the energy evolution of the image segmentation process in Fig.**??**. Fig.**??** shows another example. With the user input seeds (Fig.**??**), the amount of object and background seeds increases automatically based on the segmentation result in each iteration. It is straightforward that our algorithm guarantees the monotonic decrease of energy because iterative minimization can be taken as a multi-step minimization of the total energy.

3.4 Experimental Results

We evaluate the segmentation performance of the proposed method in comparison with the graph cuts algorithm [**?**] and *GrabCut* [**?**]. Since we use watershed for initial segmentation, for a fair comparison, we also extend the standard graph cuts to a region based scheme, i.e. we use the regions segmented by watershed, instead of the pixels, as the nodes in the graph. *GrabCut* algorithm is also an interactive segmentation technique based on graph cuts and has the advantage of reducing user's interaction under complex background. It allows the user to drag a rectangle around the desired object. Then the color models of the object and background are constructed according to this rectangle. Hence in total we have four algorithms in the experiments: the pixel based graph cuts (denoted by GC_p), the region based graph cuts (GC_r), the *GrabCut* and the proposed iterated region merging method with localized graph cuts (denoted by *IRM-LGC*).

In Sections **??** and **??**, the four algorithms are evaluated qualitatively. In Section **??**, the segmentation results are evaluated quantitatively. Some discussions are

made in Section **??**. Our experiment database contains 50 benchmark test images selected from online resources [1] [2], where 10 of them contain objects with simple background and the others are images with relatively complex background. Every image in our database has a figure-ground assignment labeled by human subjects.

3.4.1 Comparison with Graph Cuts

In this subsection, the segmentation results are compared between the proposed algorithm and algorithms GC_p and GC_r. Note that GC_r algorithm is used as the first step in lazy snapping [**?**]. This experiment can thus partially compare the performance of lazy snapping and *IRM-LGC*. However, a direct comparison of the two methods is not a fair choice, since lazy snapping has another refinement step which adjusts the mis-located boundaries produced by the first step. Fig.**??** shows some images with simple background. In these examples, it is relatively easy to extract the objects from the background. Therefore some of the results by GC_p or GC_r are not too bad, while the proposed method works better.

Extracting objects of interest from complex background is a more challenging task. Fig.**??** shows some images with relatively complex background and their segmentation results. In these images, the objects contain weak boundaries due to poor contrast and noise, and the colors of some background regions are very close to those of the objects. Given the same amount of user input, the proposed *IRM-LGC* achieves much better segmentation results than the GC_p and GC_r algorithms.

[1] http://www.research.microsoft.com/vision/cambridge/segmentation/

[2] http://www.cs.berkeley.edu/projects/vision/grouping/segbench/

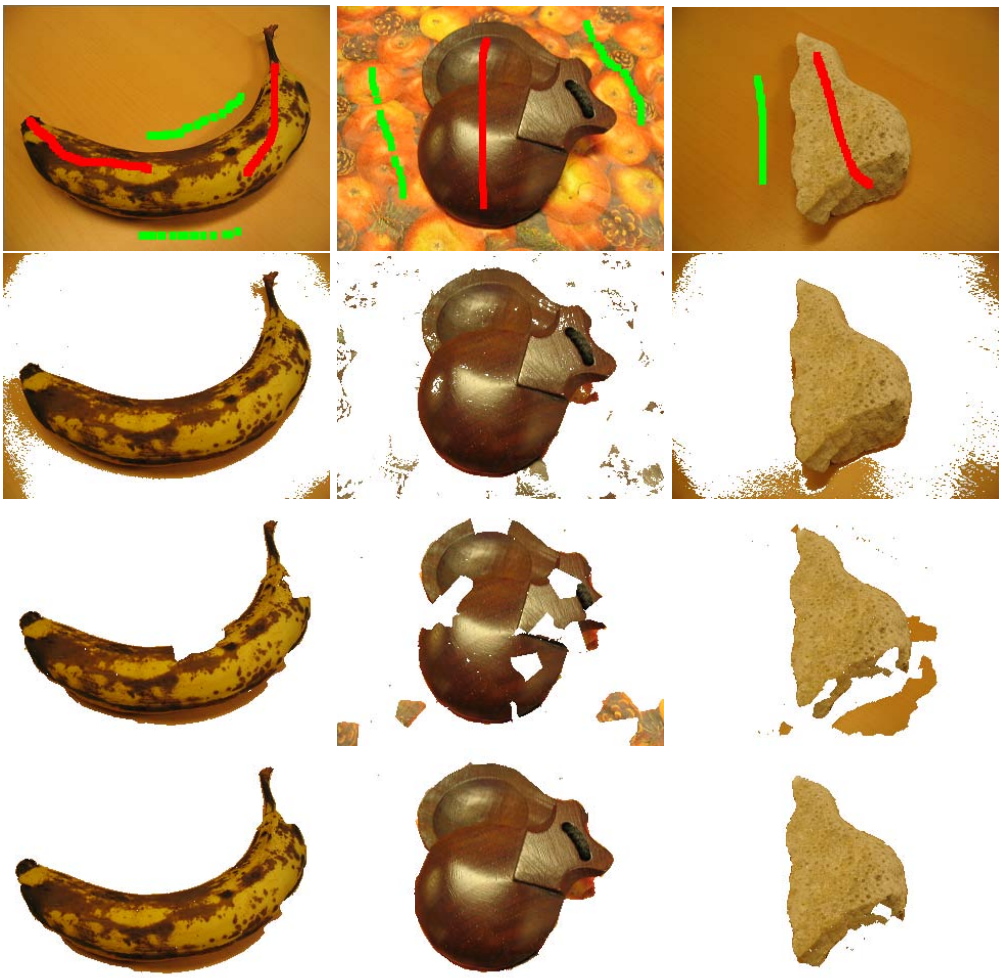


Figure 3.7 Segmentation results of images with simple background. The first row shows the original images with seeds. Red strokes are for the object and the green strokes are for the background. The second to the forth row show the segmentation results by GC_p, GC_r and *IRM-LGC* respectively.

Figure 3.8 Segmentation results of images with complex background. The first row shows the original images with seeds. From the second to the forth row, there are the segmentation results obtained by GC_p , GC_r and *IRM-LGC* respectively.

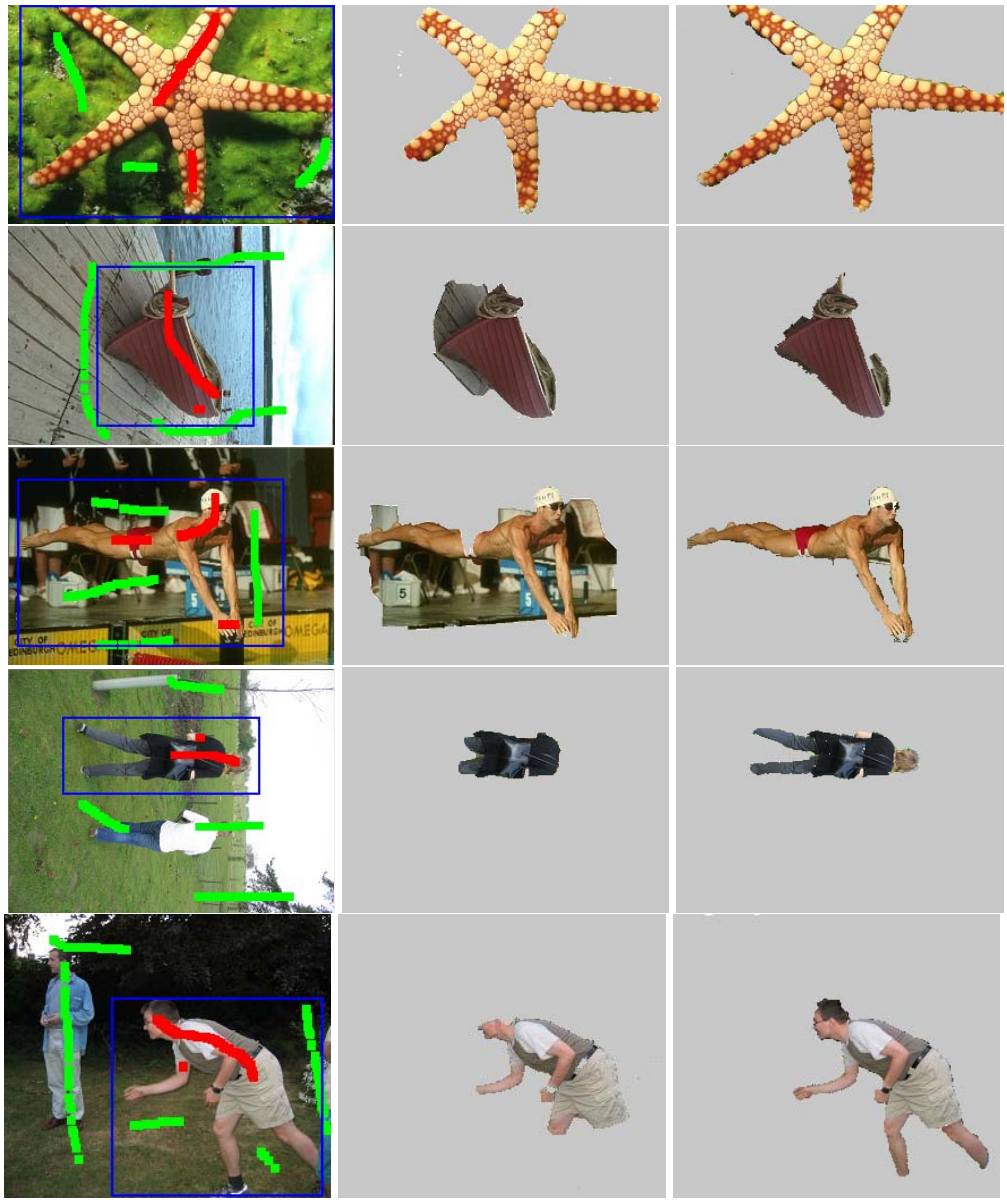


Figure 3.9 Segmentation results by *GrabCut* and the proposed method. The left column shows the original images with seeds. The blue rectangle is the interaction used in *GrabCut*, while the red and green strokes are the object and background seeds used in the proposed algorithm. The middle column shows the results of *GrabCut*. The right column shows results of *IRM-LGC*.

3.4.2 Comparison with GrabCut

Fig.**??** compares the results of *IRM-LGC* and *GrabCut*. The left column shows the original images with the seeds points. The middle column shows the segmentation results of *GrabCut*. Implementation of *GrabCut* uses 5 GMMs to model RGB color data and parameter λ is set to be 50. The right column is results of *IRM-LGC*. When the objects to be segmented contain similar colors with the background, *GrabCut* might fail to correctly segment them. Although our algorithm uses more user interaction than *GrabCut*, this tradeoff leads to more precise segmentation results.

3.4.3 Quantitative Evaluation

To better evaluate our algorithm, a quantitative evaluation of the segmentations is given by comparing with ground truth labels in the database. The qualities of segmentation are calculated by using four measures: the true-positive fraction (TPF), false-positive fraction (FPF), true-negative fraction (TNF) and false-negative fraction (FNF):

$$TPF = \frac{|A_A \cap A_G|}{|A_G|}, FPF = \frac{|A_A - A_G|}{|\overline{A_G}|}$$

$$TNF = \frac{|\overline{A_A \cup A_G}|}{|\overline{A_G}|}, FNF = \frac{|A_G - A_A|}{|A_G|}$$

where A_G represents the area of the ground truth of foreground and its complement is $\overline{A_G}$; A_A represents the area of segmented foreground by the tested segmentation method. Table **??** lists the results of TPF, FNF, TNF and FPF by the three methods over the 50 test images. We see the proposed method achieves the best TPF, FNF, TNF and FPF results.

Table 3.3 The TNF, TPF, FNF and FPF results by different methods.

Algorithms	TPF(%)	FNF(%)	TNF(%)	FPF(%)
GrabCut	83.65	16.35	96.59	3.41
GC_p	82.72	17.2	92.37	7.63
GC_r	88.01	11.99	93.78	6.22
IRM-LGC	91.29	8.71	97.75	2.25

As mentioned, the proposed *IRM-LGC* image segmentation method uses a modified watershed algorithm for initial segmentation. The median filtering of the gradient image controls the watershed segmentation output. To examine how the initial segments affect the final result of *IRM-LGC*, we applied the algorithm to different initial segmentation with different granularities, i.e. different numbers and sizes of regions in the initial segmentation. This can be done by changing the filtering times and using different sizes of filter windows. Fig.**??** shows an example. The first row shows three initial segmentations by the modified watershed algorithm, where the number of regions is 203, 372 and 1296 respectively. The second row shows the final segmentation results. We can see that segmentation quality is not sensitive to the initial segmentation. Fig.**??** compares the segmentation quality of the same image with 42 different initial segmentations, from which we can clearly see that the segmentation results are not influenced much by the initialization.

We use the max-flow algorithm [**?**] to implement the proposed *IRM-LGC* method. The worst case running time complexity for this algorithm is $O(mn^2|C|)$, where n is the number of nodes, m is the number of edges and $|C|$ is the cost of the

Figure 3.10 Initial segmentation of an image with different numbers of regions. In the first row, from the left to the right, there are 203, 372 and 1296 regions in the initial segmentation respectively. The second row shows the final segmentation results.

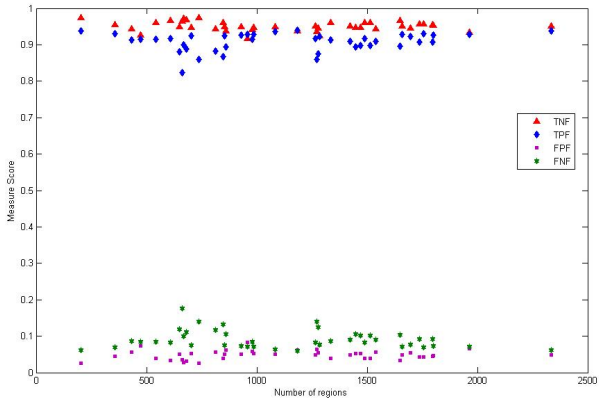


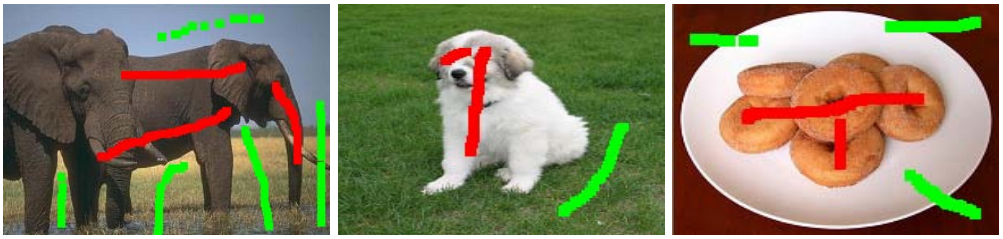
Figure 3.11 Segmentation qualities vs. initial segmentation in different granularities. For the original image used in Fig.**??**, 42 different initial segmentations are obtained and used in the proposed algorithm. The segmentation quality is measured by TPF, FPF, TNF and FNF scores.

minimum cut in the graph. In each iteration of *IRM-LGC*, the number of nodes and edges are largely reduced in comparison of the pixel based graph cuts algorithm. Our experiment is implemented on a PC with Intel Core 2 Duo 2.66 GHz CPU, 2GB memory. The running time to perform min-cut/max-flow algorithm on the whole graph which is based on image pixels is around 10-20ms, while the proposed *IRM-LGC* takes far less than 1ms. However, it should be noted that the majority of time for our algorithm is spent on constructing color models and updating the graph ($\sim$ 0.3s per iteration), thus the speedup on the min-cut/max-flow part would be relatively modest for the overall algorithm.

3.4.4 Discussion

In graph cuts based segmentation, parameter λ is used to weight the data and smoothness terms. In recent years, some literature [**?** **?**] has studied the parameter selection for graph cuts. There are two problems in graph cuts algorithm about the selection of λ. First, given different images, graph cuts with a fixed value of λ cannot lead to satisfactory segmentation. The appropriate λ values would vary largely among different images, so the user may have to spend a significant amount of time searching for it. Fortunately, the proposed *IRM-LGC* is not sensitive to the selection of λ across different images. This can be illustrated by the following experiments. In practice we found that the region based graph cuts (i.e. GC_r) has similar property to pixel based graph cuts(i.e. GC_p) in parameter selection. Sometimes, GC_r may not lead to satisfying segmentation result throughout the searching space of λ. Thus to study on a more general case, the GC_p is used in the following experiments. Fig.**??** shows some examples of the segmentation by GC_p and *IRM-*

LGC. For a comparable quality of the segmentation results by the two methods, the best value of parameter λ in GC_p varies a lot for different images (2nd row in Fig.**??**); however, a constant λ in *IRM-LGC* can lead to satisfying segmentations across different images (3rd row in Fig.**??**).

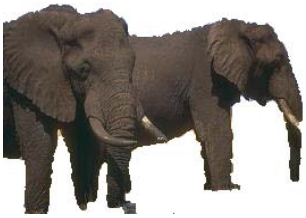


(a) Images with user input seeds

(b) GC_p,λ = 18

(c) GC_p,λ = 50

(d) GC_p,λ = 170



(e) *IRM-LGC*,λ = 50

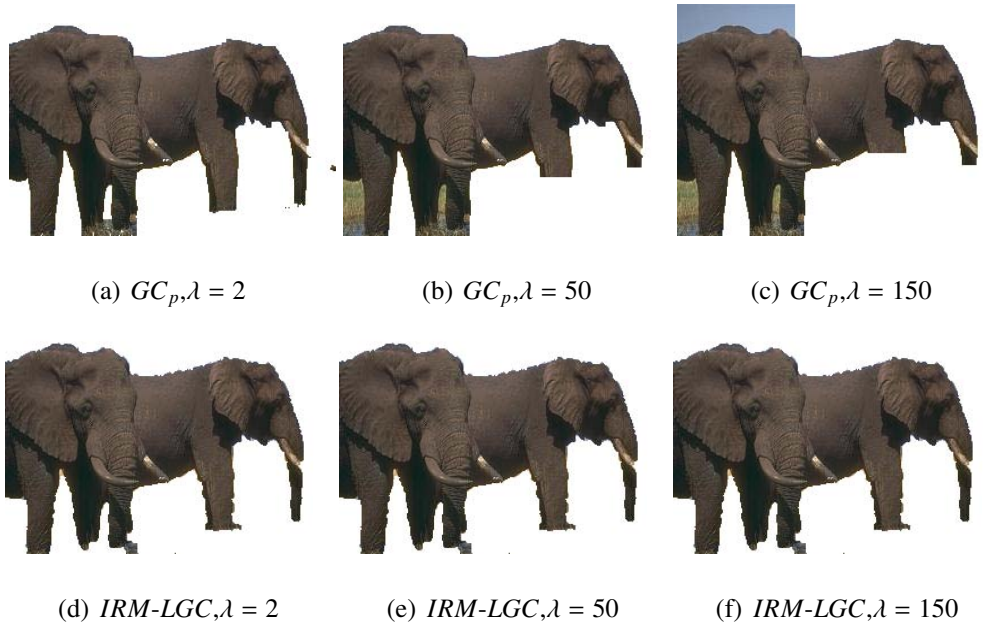
(f) *IRM-LGC*,λ = 50

(g) *IRM-LGC*,λ = 50

Figure 3.12 The values of parameter λ in GC_p and *IRM-LGC* for different images.

The second problem of standard graph cuts is that different values of λ will result in very different segmentation results for the same image. Fig.**??** compares GC_p and *IRM-LGC* by increasing the value of parameter λ. The original image with user

input seeds is in Fig.**??**. In Fig.**??**, GC_p produces a relatively good segmentation with $\lambda = 2$. In Fig.**??** and Fig.**??**, it produces big segmentation errors with $\lambda = 50$ and $\lambda = 150$, respectively. However, by using *IRM-LGC*, we can obtain similar and good segmentation results for a wide range of values: $\lambda = 2$, $\lambda = 50$ and $\lambda = 150$.



(a) GC_p,$\lambda = 2$ (b) GC_p,$\lambda = 50$ (c) GC_p,$\lambda = 150$

(d) *IRM-LGC*,$\lambda = 2$ (e) *IRM-LGC*,$\lambda = 50$ (f) *IRM-LGC*,$\lambda = 150$

Figure 3.13 Image segmentation with different parameter values. (a-c) show the segmented objects by GC_p and (d-f) show the segmented objects by *IRM-LGC*.

IRM-LGC can reduce greatly the search range of λ. On most of the test images in our database, λ is roughly between 50 and 100 for the proposed method, while for GC_p, the values vary from 10 to 200. An explanation for this is that if the data term in energy function can provide sufficient information for labeling, the graph node does not need a strong relationship with its neighbors. The proposed method gives good object/background models as iteration process goes on, thus the changes of λ for various image can be reduced. This brings much benefit for users in real

applications.

Although graph cuts algorithm has relaxed the user input compared with some other algorithms, such as livewire [**?**], the input seeds cannot always efficiently indicate the background regions, therefore when the connecting regions of the object and background have similar colors, they are still hard to be segmented correctly. It is empirically found that if the input seeds can cover the main features of the object and background, good segmentation result can be obtained. Some promising work [**? ?**] has exploited effective methods for arc weight estimation during the seeds marking process. Their work takes into account image attributes and object information in order to enhance the discontinuities between object and background, whereas a visual feedback can be provided to the user for the next action. We will investigate how to incorporate these methods into our work in the future. Fig.**??** shows a failure example. The regions circled in red only connect to object regions on the sub-graph, so they are easily assigned to the same label. Moreover, our method uses an initial segmentation to partition the image into regions, incorrect partition in initialization will also affect the final segmentation result.

IRM-LGC is independent of the initial segmentation step. However, under-segmented regions from the naive watershed algorithm cannot be re-partitioned due to the region-merging style of *IRM-LGC*. To reduce the over-segmentation and well keep the coherence of regions, more sophisticated pre-segmentation algorithms can be adopted for the initialization. For example, connected filters with morphological reconstruction operators can eliminate or merge connected components produced by watershed algorithm [**?**]. Hence they might be used as a more suitable tool for improving the initial segmentation quality than median filters.

As in traditional graph cuts algorithm, in the proposed *IRM-LGC* the user input information is also crucial for obtaining desirable segmentation. Since the newly added seeds in each iteration depend on the segmentation results in the previous iteration, the misclassified regions will probably destroy the rest part of segmentation. In the future work, other strategies of seeds selection will be taken into account. For example, the work in [**?**] does not use the seeds from previous delineation to re-compute the edge weights. It makes the well-segmented regions unchanged and therefore, the segmentation process becomes more traceable.

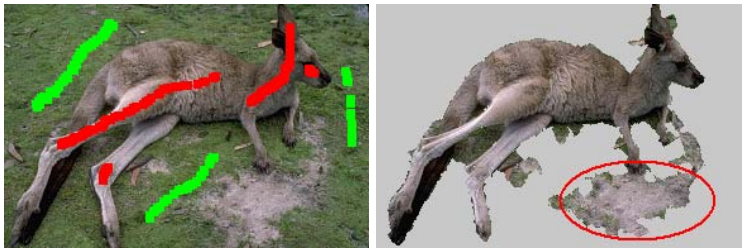


Figure 3.14 A failure example of the proposed method.

3.5 Conclusion

This chapter proposed an iterative region merging based image segmentation algorithm by using graph cuts for optimization. The proposed algorithm starts from the user labeled sub-graph and works iteratively to label the surrounding un-segmented regions. It can reduce the interference of unknown background regions far from the labeled regions so that more robust segmentation can be obtained. With the same amount of user input, our algorithm can achieve better segmentation results than the standard graph cuts, especially when extract the object from complex background.

Qualitative and quantitative comparisons with standard graph cuts and GrabCut show the efficiency of the proposed method. Moreover, the search space of parameter λ in graph cuts is also reduced greatly by the iterated region merging scheme.

Chapter 4

A Probabilistic Measure for Quantitative Evaluation of Image Segmentation

4.1 Introduction

Over the past decades, extensive research has been done in designing different algorithms for image segmentation. To place the existing algorithms on a solid scientific ground, it is very necessary to evaluate the segmentations produced by them. In the view of application-dependence, image segmentation acts as one step for the high-level vision tasks (e.g. object recognition). Hence it is claimed that evaluation should be based on the final performance of the entire system [**?**]. This strategy has its limitation for the fact that the general-purpose segmentation algorithms have drawn wider attention in the computer vision community due to their poten-

tials in the broader applications. For these algorithms, the consistency of performance is hard to maintain for different applications. Consequently, the application-independent strategy becomes more meaningful in the segmentation evaluation.

Much work in image segmentation has tried to localize the objects boundary according to the human-level interpretation of the image. This idea makes segmentation algorithms more intelligent, meanwhile leads to multiple acceptable solutions to the segmentation. The evaluation task is therefore taking a risk of being subjective towards the quality of segmentations. For example, in the most common evaluation method, one is required to visually compare the segmentation results, which often leads to inconsistent results among different observers.

As an alternative, several quantitative evaluation methods have been proposed to design the objective segmentation measures. The various objective evaluation methods, in general, can be categorized as the unsupervised and the supervised methods. In the first category, the empirical goodness measures [**? ? ? ?**] are proposed to capture the heuristic criteria in the desirable segmentations. The evaluation is based on the extent to which the segmentation matches these criteria. The ability of working without any manually segmented image allows unsupervised methods useful for online segmentation evaluation. However, the criteria are generalized from the common characteristics or semantic information of the objects (e.g. homogeneous regions, smooth boundaries, etc.), whereas are not accurate enough to describe the complex objects for natural images.

In supervised evaluation methods, the segmentation quality are based on the degree of similarity between the machine segmentation and the human-labeled ground truths. This comparison is more intuitive than the empirical goodness measures,

since the set of ground truths can well represent the human-level interpretation of the image. Many achievements in this area have benefited from the public availability of the standard set of images and the associated ground truths [**?**] from UC Berkeley, which provide a solid ground for the quantitative evaluation of image segmentation. Several measures are proposed to count the degree of overlapping between regions, with the strategies of being intolerant or tolerant of region refinement [**?** **?**]. In contrast to working on the regions, there are measures [**?** **?**] matching the boundaries between the segmentations. Some other measures [**?** **?** **?** **?**] use non-parametric tests, which work by counting the pairs of pixels that belong to the same region in different segmentations. Theoretically, the most reliable comparison result can only be obtained by operating on all the possible ground truths for an image. However, this assumption is not likely to be satisfied in the real application. Accumulating a wide range of human-labeled ground truths is time consuming and costly. This leads to some problems for the supervised segmentation evaluation:

- Limited representation for the data: the number of ground truths for an image is confined by the amount of subjects assigned to the labeling task. As a result, the set of ground truth does not contain all the possible segmentations.

- High risk of being subjective: integrating the high-level information in the ground truth might miss the details from the low-level perspective. And it tends to exhibit some unpredictable understandings on the complicated objects (e.g. translucent objects in the natural images).

- Localization errors: errors produced in the drawing process will be over exaggerated in the presence of small number of ground truths.

To objectively evaluate the quality of segmentations, some work [**?** **?** **?**] attempted to focus only on the variations in the ground truths and ignore the inherent limitation of them. And many of the other segmentation measures (see the survey paper [**?**]) conduct the comparison with the single ground truth image, i.e. the elements (e.g. pixels) from the segmentation are fully compared with those of the ground truth. As we discussed above, strict comparison to either the single ground truth or the multiple ground truths might lead to a certain bias on the results or even be far from the goal of objective evaluation.

In this chapter, we propose a new definition of "similarity" to measure the segmentation. The basic concept is illustrated in Fig.**??**. Fig.4.1(b) shows a possible segmentation of the sample image in Fig.4.1(a), which is not directly identical to any of the ground truths listed in Fig.4.1(d). However, one would agree that Fig.4.1(b) is similar to these ground truths in the sense that it is composed of similar local structures to them. We would like to employ this idea to design a segmentation measure which generalizes the configurations of the segmentation and preserves the structural consistency in the ground truths. We make an assumption that if a segmentation is "good", it can be constructed by some pieces of the ground truths. To this end, a composited ground truth is adaptively created to locally match the segmentation as much as possible. Note that in Fig.4.1(c), the shared regions between the segmentation and the ground truths are typically irregularly shaped, therefore the process of compositing the ground truth is data driven and can not be predefined. Also, the confidence of the selected structures for the composited ground truth will be examined during the process of comparison. Less reliance should be given on the ambiguous structures, even if they are very similar to the

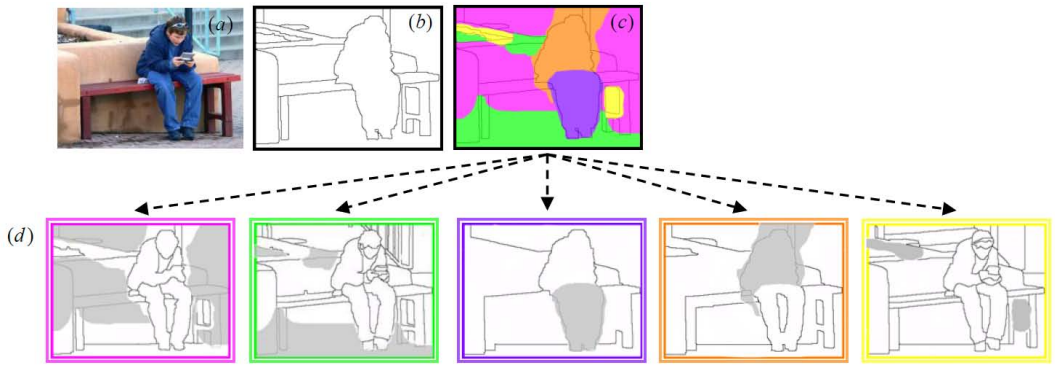


Figure 4.1 A composited interpretation between the segmentation and the ground truths - the basic concept. (a) A sample image. (b) A segmentation of the sample image in (a). (c) Different parts of the segmentation can be found to be very similar to one of the human-labeled ground truths in (d).

segmentation. In the proposed segmentation measure, we will integrate all these factors for the evaluation.

In the previous work, researchers [**?**] have found that human visual system (HVS) is highly adapted to extract structural information from natural scenes. As a consequence, a perceptually meaningful measure should be error-sensitive to the structures in the segmentations. It is also known that human observers may pay different attentions to different parts of the images [**?** **?**]. Ground truths of the same image therefore present various granularities in the object parts. This fact makes them rarely identical in the global view, while highly consistent in the local structures. For this reason, the evaluation of the segmentation should be performed more reasonably on the local structures, rather than only the global ones. The standard global similarity measures, such as Mutual Information [**?**], Mean Square Error (MES) [**?**], probabilistic Rand index [**?**] and Precision-Recall curves [**?**], do

not consider the geometric decomposition among multiple ground truths. They can only be loosely said to work with the average or the intersection of all ground truth segmentations. In most cases, the meaningful local structures are not guaranteed to be well captured, and thus these measures can not provide an intuitive comparison as the HVS does.

4.2 Image Segmentation Database

4.3 The Construction of G^*

4.3.1 Theoretical Framework

First of all, we define some terminologies that will be used throughout this chapter. Consider a set of ground truths $\boldsymbol{G} = \{G_1, G_2, \ldots, G_K\}$ of an image $X = \{x_1, x_2, \ldots, x_N\}$, where $G_i = \{g_1^i, g_2^i, \ldots, g_N^i\}$ denotes a labeling set of X, $i = 1, \ldots, K$, and N is the number of elements in the image (e.g., pixels, regions). Let $S = \{s_1, s_2, \ldots, s_N\}$ be a segmentation of X, where s_j is the label of x_j (e.g., boundary or non-boundary), $j = 1, \ldots, N$. To examine the similarity between S and $\boldsymbol{G}$, we compute the similarity between S and a new ground truth G^*, which is generated from $\boldsymbol{G}$. For this purpose, we denote by $G^* = \{g_1^*, g_2^*, \ldots, g_N^*\}$ the new composited ground truth, which is generated by putting together some pieces from $\boldsymbol{G}$, such that $g_j^* \in \{g_j^1, g_j^2, \ldots, g_j^K\}$. The principle to choose these pieces are: each one of them is most similar to its counterpart in S with the constraint of consistency. Then the similarity between S and $\boldsymbol{G}$ is dependent on the similarity between S and G^*.

As defined before, G^* is a geometric ensemble of some local pieces from $\boldsymbol{G}$,

i.e. $G^* = \{g_1^*, g_2^*, \ldots, g_N^*\}$, where $g_j^* \in \{g_j^1, g_j^2, \ldots, g_j^K\}$, $j = 1, \ldots, N$. We adopt an optimistic strategy to choose the elements of G^*, by which S will match $\boldsymbol{G}$ as much as possible. Therefore, G^* can be taken as a new segmentation of X by assigning a label g_j^* to each pixel. At the same time, g_i^* contains the information of the corresponding location in the K ground truths. Hence, as long as we find the correspondence between g_j^* and g_j^i ($i = 1, \ldots, K$), G^* can be constructed. For this purpose, we introduce a label l_{g_j} ($l = 1, \ldots, K$) to each g_j^* in G^*. The construction of G^* is illustrated by an example in Fig. **??**. This figure shows how to construct the new ground truth G^* for a part of the segmentation (in the red rectangle) in Fig.4.1(b). We can see that, given two ground truth images G_1 and G_2, G^* is found by firstly computing the optimal labeling $\{l_{G_1}, l_{G_2}\}$ for the ground truths. Then elements of G^* which are labeled as l_{G_1} (or l_{G_2}) will take their values from G_1 (or G_2). This leads to a maximum-similarity-matching between S and G^*.

Many problems in computer vision can be taken as a labeling problem (e.g. image segmentation). Given a set of labels L and a set of sites G, a label $l_g \in L$ needs to be assigned to each of the sites $g \in G$. The label set could be an arbitrary finite set, i.e. $L = \{1, 2, \ldots, K\}$. Let $l = \{l_g | l_g \in L\}$ stand for a labeling, i.e. label assignments to all sites in G. We formulate the labeling problem in terms of energy minimization, i.e. one seeks to the labeling l that minimizes the energy E. In this work, we consider an energy function given by the Potts model [**?**]:

$$E(l) = \sum_{\{g_j, g_{j'}\} \in M} u_{\{g_j, g_{j'}\}} \cdot T(l_{g_j} \neq l_{g_{j'}}) + \lambda \cdot \sum_j D(l_j) \tag{4.1}$$

where M is a neighborhood system, the interaction penalty $u_{\{g_j, g_{j'}\}} \cdot T(l_{g_j} \neq l_{g_{j'}})$ indicates the cost of assigning different labels to the pair of sites in G^*. T is an

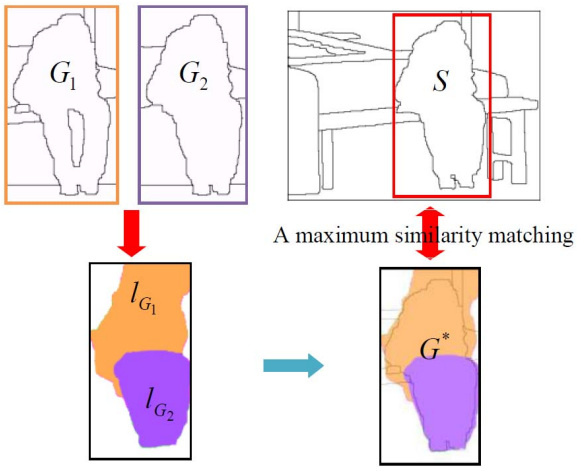


Figure 4.2 A labeling example of the ground truths leads to a new (partly) segmentation of S in Fig.4.1(b). G_1 and G_2 are two ground truths by human observers. The optimal labeling $\{l_{G_1}, l_{G_2}\}$ of G_1 and G_2 produces a composited ground truth G^*, which matches the S as much as possible.

indicator function:

$$T(l_{g_j} \neq l_{g_{j'}}) = \begin{cases} 1 & \text{if } l_{g_j} \neq l_{g_{j'}} \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

We call $u_{\{g_j,g_{j'}\}} \cdot T(l_{g_j} \neq l_{g_{j'}})$ the smoothness term in that it encourages labeling consisting of several regions where sites in the same region have the same labels. In the composition, we want the separations of regions to pay a higher cost on the sites which only emerge in few ground truths, while lower cost on those agreed by the majority of ground truths. Thus we define $u_{\{g_j,g_{j'}\}}$ in the expression that $u_{\{g_j,g_{j'}\}} = min\{\overline{\Delta d_j}, \overline{\Delta d_{j'}}\}$, where $\overline{\Delta d_j}$ is a normalized average distance between g_j^* and $\{g_j^1, g_j^2, \ldots, g_j^K\}$. $D(l_j)$ is called the data term, which penalizes the decision of assigning l_j to the site g_j. This requires to examining the distances $\Delta d(g_j, \boldsymbol{G})$ between the ground truths and the segmentation S, so we have: $D(l_j) = 1 - 1/W \cdot$

$\Delta d(g_j, \boldsymbol{G})$, where W is a normalization factor. In Eq.(**??**), the parameter λ is used to control the relative importance of the data term versus the smoothness term. If λ is very large, only the data term matters. In this case, the label of each site is independent of the other sites. If λ is very small, all the sites will have the same label.

Since Eq.(**??**) is a non-convex function, minimization of which will lead to an NP-hard problem. We use the effective expansion-moves/swap-moves algorithm described in [**?**] to solve the energy function. The algorithm aims to compute the minimum cost multi-way cuts on a defined graph. Nodes in the graph are connecting to their neighbors by n-links. Each n-link is assigned a weight $u_{\{g_j, g_{j'}\}}$ defined in the energy function Eq.(**??**). Suppose we have K ground truths, then there will be K virtual nodes in the graph, representing the K labels of these ground truths. Each graph node connects to the K virtual nodes by t-links. We weight the t-links as $D(l_j)$ to measure the similarity between the graph nodes and the virtual nodes. The K-way cuts will divide the graph into K parts, and bring a one-to-one correspondence to the labeling of the graph. In the example of Fig.**??**, red lines are the graph cuts computed by the expansion-moves/swap-moves algorithm. Labeling of the graph is accordingly obtained (shown in different colors). Finally, we copy the segmented pieces of regions from each ground truth to form the G^*.

4.3.2 Definition of the Distance Δd_j

In section **??**, the distance Δd_j is used to define the labeling energy function. Although many distance measures have been proposed in the existing literature, it is not a trivial work to perform the matching between the machine segmentation and

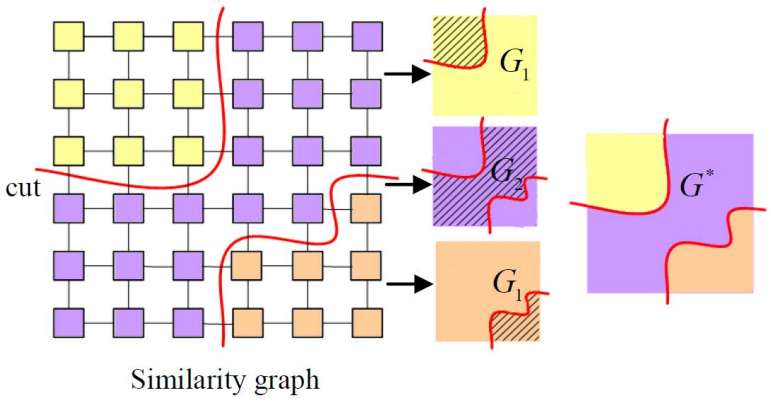


Figure 4.3 An example of the construction of G^*. It is produced by the copies of selected pieces (shadow areas) in the ground truths.

the human-labeled ground truth. Due to the location errors produced in drawing process, boundaries of the same object might not be fully overlapped. Generally, this is an inherent problem for human-labeled ground truths. In Fig.**??**, we show an example of boundary distortions in different ground truths. If simply match pixels between the segmentation S and ground truths $\boldsymbol{G}$, S will probably be over-penalized by the unstable and slightly mis-localized boundaries in $\boldsymbol{G}$. Moreover, different segmentation algorithms may produce the object boundaries in different widths. For example, if we take the border pixels in both sides of the adjacent regions, the boundaries will appear in a two-pixel width. To match the segmentation appropriately, the measure should tolerant some acceptable distortions between different segmentations. The previous work of David et al. [**?**] solves the problem by matching the boundaries under a defined threshold. However, their method is limited to performing the correspondence on the boundaries only. Here we adopt a simpler but effective method, which is similar to the work in [**?**] and works on every pixel in the image.

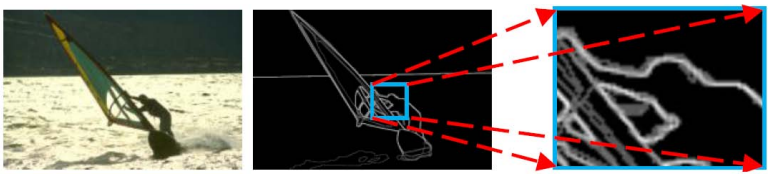


Figure 4.4 An example of distorted boundaries among different human-labeled ground truths. Ground truths are drawn into the same image, where the whiter boundaries indicate that more subjects marked as a boundary. (Taken from the Berkeley Dataset [**?**]).

In [**?**], Sampat et al. proposed a structural similarity index in the complex wavelet domain (*CW-SSIM*). The motivation of this index is based on the fact that the relative phase patterns of complex wavelet coefficients could well preserve the structural information of local image features and rigid translation of image structures will lead to constant phase shift. *CW-SSIM* overcomes the drawback of *SSIM* [**?**] in that it does not require the precise correspondences between pixels, thus becomes robust to the small geometric distortions. The following expression gives the definition of *CW-SSIM*:

$$S(c_x, c_y) = \frac{2\sum_{i=1}^{N} |c_{x,i}||c_{y,i}^*| + \alpha}{\sum_{i=1}^{N} |c_{x,i}|^2 + \sum_{i=1}^{N} |c_{y,i}|^2 + \alpha} \cdot \frac{2|\sum_{i=1}^{N} c_{x,i}c_{y,i}^*| + \beta}{2\sum_{i=1}^{N} |c_{x,i}c_{y,i}^*| + \beta} \tag{4.3}$$

where $|c_{x,i}|$ is the magnitude of wavelet coefficient c_x. c_x and c_y are extracted at the same spatial location in the same wavelet subbands of the two segmentations, respectively. c^* is the conjugate of c. α and β are small positive constants. It is easy to see that the maximum value of *CW-SSIM* is 0 if c_x and c_y are identical. In this work, we define a very similar measure as that defined in Eq. **??**, which uses the the Gabor spectrum coefficients instead of the wavelet coefficients. The coefficients are gotten by convolving the image with Gabor kernels, which contain 3 different

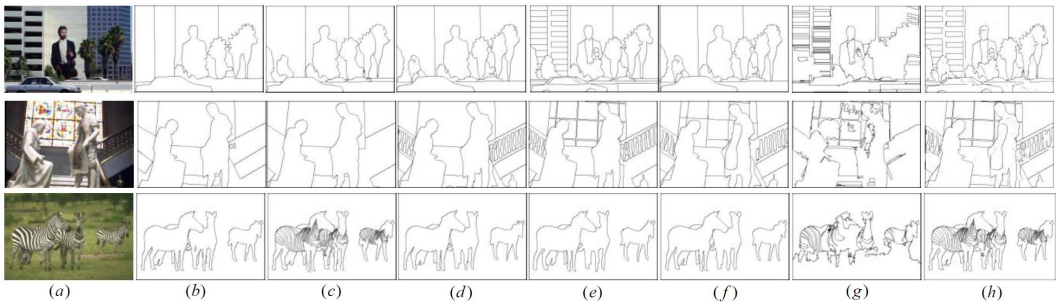


Figure 4.5 The composited ground truth G^* produced by the measure with pixel-based distance. (a) Original images. (b)-(f) Human labeled ground truths. (g) Segmentations of images by the mean-shift algorithm [**?**]. (h) The composited ground truth G^*.

scales and 8 different directions, respectively. In this way, the image information can be extracted without a down-sampling step as used in the *CW-SSIM*. Whereas, Δd can be written as:

$$\Delta d(c_x, c_y) = 1 - S(c_x, c_y) \tag{4.4}$$

CW-SSIM index provides an explicit description on the one-to-one similarity between pixels, hence it can also be used to compute the labeling problem in Eq.(**??**). In Fig.**??**, we show the composited ground truth G^* for three segmentation examples, which are produced by using the Δd defined in Eq.(**??**).

4.4 The Pixel Based Probabilistic Measure

4.4.1 Definition

In this section, the similarity measure between S and $\boldsymbol{G}$ is calculated with the similarity between S and G^* as well as the global confidence of G^*. The confidence

is examined for the principle that less reliance should be given on the ambiguous structures, even if they are very similar to the segmentation. Formally, the probabilistic measure on each element s_j of S is defined as:

$$M(s_j, \boldsymbol{G}) = P(s_j|H_G, H(g_j^*, \boldsymbol{G})) \tag{4.5}$$

where H_G is the hypothesis that S is generated from $\boldsymbol{G}$ via some geometric decomposition of $\boldsymbol{G}$. $H(g_j^*, \boldsymbol{G})$ is the hypothesis that g_j^* is selected from $\boldsymbol{G}$, which corresponds to the global confidence of g_j^* w.r.t. $\boldsymbol{G}$.

If there are K instances in $\boldsymbol{G}$ and all of them will contribute to the construction of G^*, at the same locations where G^* are selected from $\boldsymbol{G}$, we accordingly decompose S into K disjointed set $\{S_0, S_1, \ldots, S_K\}$. To compute the probabilistic measure on the S, we marginalize over all the elements of S to have:

$$M(S, \boldsymbol{G}) = \prod_{i=1}^{K} \prod_{s_j \in S_i} P(s_j|H_G, H(g_j^*, \boldsymbol{G})) \tag{4.6}$$

In this work, we assume that the random variables s_j are i.i.d. with the negative exponential distribution as:

$$P(s_j|H_G, H(g_j^*, \boldsymbol{G})) = \frac{1}{Z} exp\Big(\frac{-|\Delta d(s_j, g_j^*)|^2}{2\delta^2} \cdot R_{s_j}\Big) \tag{4.7}$$

where Z is a normalization factor, $\Delta d(s_j, g_j^*)$ is the distance between s_j and g_j^*, and R_{s_j} is the empirical value of the global confidence of g_j^* w.r.t. $\boldsymbol{G}$. For example, we can estimate it as the similarity between g_j^* and $\{g_j^1, g_j^2, \ldots, g_j^K\}$ and write it as:

$$R_{s_j} = 1 - \overline{\Delta d_j} \tag{4.8}$$

where $\overline{\Delta d_j}$ is a normalized average distance between g_j^* and $\{g_j^1, g_j^2, \ldots, g_j^K\}$. Using

Eq.(**??**), Eq.(**??**) can be rewritten as:

$$M(S,\boldsymbol{G}) = \frac{1}{Z}\prod_{i=1}^{K}\prod_{s_j\in S_i} exp\Big(\frac{-|\Delta d(s_j,g_j^*)|^2}{2\delta^2}\cdot R_{s_j}\Big)$$
$$= \frac{1}{Z}exp\Big(-U(\Delta d)\Big) \tag{4.9}$$

where $U(\Delta d)$ is the distance sum over all the elements in S:

$$U(\Delta d) = \sum_i\sum_j \frac{|\Delta d(s_j,g_j^*)|^2}{2\delta^2}\cdot R_{s_j} \tag{4.10}$$

We can see that the measure of segmentation S is the accumulated sum of the distance provided by the individual elements of S. The minimum value of $U(\Delta d)$ is zero, when segmentation S is completely identical to the ground truths $\boldsymbol{G}$, given that all the ground truths in $\boldsymbol{G}$ are identical. In this case, $M(S,\boldsymbol{G})$ achieves its maximum value. Note that $U(\Delta d)$ is decided by both of the distances $\Delta d(s_j,g_j^*)$(in Eq.(**??**) and $\overline{\Delta d_j}$ (in Eq.(**??**)). If S is only approaching to G^* without a high consistency among the ground truths data $\{g_j^1,g_j^2,\ldots,g_j^K\}$, we can not obtain a high value of $M(S,\boldsymbol{G})$. We would like to follow the principle that, if the ground truths are inherently ambiguous, a strong evidence of similarity provided by them is not reasonable to be accepted. This might be a common case for images with complex contents, where perceptual interpretation of the image contents is diverse. From this point of view, a segmentation database containing sufficiently enough ground truths will be necessary for the evaluation task.

4.4.2 Experimental Results

4.5 The Boundary Based Measure

4.5.1 Definition

4.5.2 Experimental Results

Bibliography

[1] S. Abney. Understanding the yarowsky algorithm. *Computational Linguistics*, 30(3):365–395, 2004.

[2] A. A. Amini, T.E. Weymouth, and R.C. Jam. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12, 1990.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2294–2301, 2009.

[4] B. Ayed and A. Mitiche. A region merging prior for variational level set image segmentation. *IEEE Transactions on Image Processing*, 17(12):2301–2311, 2008.

[5] O. Veksler B. Peng. Parameter selection for graph cut based image segmentation. *British Machine Vision conference*, 2008.

[6] R. Bellman. Dynamic programming. In *Princeton University Press*, 1957.

[7] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, (48):259–302, 1986.

[8] S. Beucher and C. Lantuẽjoul. Use of watersheds in contour detection. *In-*

ternational workshop on image processing, real-time edge and motion detection, 1979.

[9] M. Bleyer and M. Gelautz. Graph-based surface reconstruction from stereo pairs using image segmentation. *SPIE Signal Processing and Target Recognition*, pages 288–299, 2005.

[10] S. Borra and S. Sarkar. A framework for performance characterization of intermediate-level grouping modules. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(11):1306–1312, 1997.

[11] M. Borsotti, P. Campadelli, and R. Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recognition Letter*, 19(8):741–747, 1998.

[12] Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary and region segmentation. *International Conference on Computer Vision*, 1:105–112, 2001.

[13] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. *International Conference on Computer Vision*, 1:26–33, 2003.

[14] Y. Boykov and G. Funka Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 69(2):109–131, 2006.

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

[16] F. Calderero and F. Marques. General region merging approaches based on information theory statistical measures. *The 15th IEEE International Con-*

ference on Image Processing, pages 3016–3019, 2008.

[17] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[18] H.D Cheng and Y. Sun. A hierarchical approach to color image segmentation using homogeneity. *IEEE Transactions on Image Processing*, 9:2071–2082, 2000.

[19] H. Christensen and P. Phillips. Empirical evaluation methods in computer vision. *World Scientific Publishing Company*, 2002.

[20] D. Comanicu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[21] C. Fowlkes D. Martin and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5), 2004.

[22] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *In IEEE conference on Computer Vision and Pattern Recognition*, 2010.

[23] F. Marques F. Calderero. Region merging techniques using information theory statistical measures. *IEEE Transactions on Image Processing*, 19:1567–1586, 2010.

[24] A.X. Falcão and F.P.G. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. on Medical Imaging*, 23:1100–1108, 2004.

[25] A.X. Falcão, J. Stolfi, and R.A. Lotufo. Ultrafast user-steered image seg-

mentation paradigm: live-wire-on-the-fly. *IEEE Transactions on Medical Imaging*, 19(1):55–62, 2000.

[26] A.X. Falcão, J. Stolfi, and R.A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:19–29, 2004.

[27] A.X. Falcão, J.K. Udupa, S. Samarasekara, and S. Sharma. User-steered image segmentation paradigms: Live wire and live lane. *Graphical Models and Image Processing*, 60:233–260, 1998.

[28] P. Favaro and S. Soatto. A variational approach to scene reconstruction and image segmentation from motion blur cues. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 631–637, 2004.

[29] M. Fernand. Sequential algorithms for cell segmentation: maximum efficiency? *International symposium on clinical cytometry and histometry, Scholoss Elmau*, 1986.

[30] M. Fernand. Hierarchies of partitions and morphological segmentation. *IEEE Workshop on scale-space and Morphology in computer vision*, 2001.

[31] D.A. Forsyth and J.Ponce. Computer vision: A modern approach. In *Prentice Hall*, 2002.

[32] E. Fowlkes and C. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[33] G. Funka-lea, Y. Boykov, C. Florin, M. Jolly, R. Moreau-gobard, R. Ramaraj, and D. Rinck. Automatic heart isolation for ct coronary visualization using graph-cuts. *In IEEE International Symposium on Biomedical Imaging*, pages

614–617, 2006.

[34] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[35] S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. *In A. M. Gleason (Ed.), Proceedings of the International Congress of Mathematicians*, pages 1496–1517, 1987.

[36] S. Geman and D. McClure. Bayesian image analysis: An application to single photon emission tomography. *In Proceedings of the Statistical Computing Section*, pages 12–18, 1985.

[37] R.C. Gonzalez and R.E. Woods. Digital image processing. Addison Wesley, Reading, MA, 1992.

[38] I.E. Gordon. Theories of visual perception. In *First ed. John Wiley and Sons*, 1989.

[39] L. Grady, Y. Sun, and J. Williams. Handbook of mathematical models in computer vision. pages 453–469. Springer, 2006.

[40] G. Haffari and A. Sarkar. Analysis of semi-supervised learning with the yarowsky algorithm. *23rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.

[41] K. Haris, S.N. Estradiadis, N. Maglaveras, and A.K Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing*, 7:1684–1699, 1998.

[42] Q. Huang and B. Dom. Quantitative methods of evaluating image segmentation. *IEEE Intl. Conf. on Image Processing*, pages 53–56, 1995.

[43] J. J. Liu and H.Y. Shum. Paint selection. *ACM Transactions on Graphics*, 2009.

[44] X. Jiang, C. Marti, C. Irniger, and H. Bunke. Distance measures for image segmentation evaluation. *EURASIP Journal on Applied Signal Processing*, 2006:209–209, 2006.

[45] M.P. Jolly, H. Xue, L. Grady, and J. Guehring. Combining registration and minimum surfaces for the segmentation of the left ventricle in cardiac cine mr images. *Proc. of Medical Image Computing and Computer-Assited Intervation*, pages 910–918, 2009.

[46] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 2:321–331, 1988.

[47] K. Koffka. Principles of gestalt psychology. In *New York: Harcourt, Brace and World*, 1935.

[48] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. *International Conference on Computer Vision*, 1:564–571, 2005.

[49] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bi-layer segmentation of binocular stereo video. *IEEE Conference of Computer Vision and Pattern Recognition*, pages 407–414, 2005.

[50] C. Lam and S. Yuen. An unbiased active contour algorithm for object tracking. *Pattern Recognition Letters*, 19(5-6):491–498, 1998.

[51] S. LaValle and S.M. Hutchinson. Bayesian region merging probability for parametric image models. *Proc. IEEE Computer Soc.Conf. Computer Vision Pattern Recognition*, pages 778–779, 1993.

[52] S. Lee and M.M. Crawford. Unsupervised multistage image classification using hierarchical clustering with a bayesian similarity measure. *IEEE Transactions on Image Processing*, 14:312–320, 2005.

[53] Y. Li, J. Sun, C.K. Tang, and H.Y. Shum. Lazy snapping. *ACM Transactions on Graphics*, 23:303–308, 2004.

[54] H. Liu, Q. Guo, M. Xu, and I. Shen. Fast image segmentation using region merging with a k-nearest neighbor graph. *IEEE Conference on Cybernetics and Intelligent Systems*, pages 179–184, 2008.

[55] B. Marcotegui and S. Beucher. Fast implementation of waterfall based on graphs. *Computational Imaging and Vision*, 30:177–186, 2005.

[56] D. Martin. An empirical approach to grouping and segmentation. *Ph.D. dissertation U. C. Berkeley*, 2002.

[57] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *International Conference on Computer Vision*, 2001.

[58] P.A.V. Miranda and A.X. Falcão. Links between image segmentation based on optimum-path forest and minimum cut in graph. *Journal of Mathematical Imaging and Vision*, 35:128–14, 2009.

[59] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. Synergistic arc-weight estimation for interactive image segmentation using graphs. *Computer Vision and Image Understanding*, 114:85–99, 2009.

[60] A. Mishra, P. Fieguth, and D. Clausi. Accurate boundary localization using dynamic programming on snakes. *Proceedings of the Canadian Conference*

on Computer and Robot Vision, 2008.

[61] A. Moore and S. Prince. "lattice cut" - constructing superpixels using layer constraints. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2124, 2010.

[62] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[63] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. Scene shape priors for superpxiel segmentation. *International Conference on Computer Vision*, pages 771–778, 2009.

[64] F. Moscheni, S. Bhattacharjee, and M. Kunt. Spatio-temporal segmentation based on region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:897–915, 1998.

[65] R. Nock and F. Nielsen. Statistical region merging. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:1452–1458, 2004.

[66] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithm based on hamilton jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[67] S.J Osher and R.P Fedkiw. Level set methods and dynamic implicit surfaces. Springer Verlag, 2002.

[68] O.Veksler. Star shape prior for graph-cut image segmentation. *Proceedings of the 10th European Conference on Computer Vision*, pages 454–467, 2008.

[69] B. Paul, L. Zhang, and X. Wu. Canny edge detection enhancement by scale multiplication. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*,

27:1485–1490, 2005.

[70] V. Zavadsky Y. Boykov P.Das, O.Veksler. Semiautomatic segmentation with compact shape prior. *Image and Vision Computing*, 27.

[71] B. Peng, L. Zhang, and J. Yang. Iterated graph cuts for image segmentation. *Asian Conference on Computer Vision*, 2009.

[72] D.P. Huttenlocher P.F. Felzenszwalb. Image segmentation using local variations. *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 98–104, 1998.

[73] R. Potts. Some generalized order-disorder transformation. *Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952.

[74] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[75] C. Van Rijsbergen. Information retrieval. *second ed. Dept. of Computer Science, Univ. of Glasgow*, 1979.

[76] C. Rother, V. Kolmogorov, and A. Blake. Grabcut-interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, pages 309–314, 2004.

[77] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[78] M. Sampat, Z. Wang, S. Gupta, A. Bovik, and M. Markey. Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing*, 18(11):2385–2401, 2009.

[79] K.H. Seo, J.H. Shin, W. Kim, and J.J. Lee. Real-time object tracking and

segmentation using adaptive color snake model. *International Journal of Control, Automation, and Systems*, 4(2):236–246, 2006.

[80] J.A Sethian. Level set methods and fast marching methods. Cambridge University Press, 1999.

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

[82] Y. Shu, G.A, Bilodeau, and F. Cheriet. Segmentation of laparoscopic images: integrating graph-based segmentation and multistage region merging. *The 2nd Canadian Conference on Computer and Robot Vision*, pages 429–436, 2005.

[83] A. Trẽmeau and P. Colantoni. Regions adjacency graph applied to color image segmentation. *IEEE Transactions on Image Processing*, 9:735–744, 2000.

[84] R. Unnikrishnan and M. Hebert. Measures of similarity. *IEEE Workshop on Applications of Computer Vision*, pages 394–400, 2005.

[85] R. Unnikrishnan, C. Pantofaru, and M. Hebert. A measure for objective evaluation of image segmentation algorithms. *Proc. CVPR Workshop Empirical Evaluation Methods in Computer Vision*, 2005.

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

[87] C. Rother V. Kolmogorov, Y. Boykov. Applications of parametric maxflow in computer vision. *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[88] L. Vincent and P. Soiller. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

[89] P. Viola and W. Wells. Alignment by maximization of mutual information. 3:16–23, 1995.

[90] A. Wald. Sequential analysis. In *Wiley Publications in Statistics*, 1947.

[91] T. Wan, N. Canagarajah, and A. Achim. Statistical multiscale image segmentation via alpha-stable modeling of wavelet coefficients. *IEEE Transactions on Multimedia*, 11:624–633, 2009.

[92] S. Wang and J. M. Siskind. Image segmentation with ratio cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):675–690, 2003.

[93] Z. Wang and A. Bovik. Modern image quality assessment. 2006.

[94] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. on Image Process*, 13(4):600–612, 2004.

[95] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[96] M. Wertheimer. Laws of organization in perceptual forms (partial translation),a sourcebook of gestalt psycychology. pages 71–88. W.B.Ellis, ed.Harcourt, Brace, 1938.

[97] Z. Wu. Homogeneity testing for unlabeled data: A performance evaluation. 55:370–380, 1993.

[98] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering

theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.

[99] O. Veksler Y. Boykov and R. Zabih. Markov random fields with effcient approximations. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.

[100] V. Kolmogorov Y. Boykov. 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, 2004.

[101] H. Shum Y. Li, J. Sun. Video object cut and paste. *ACM Transactions on Graphics*, 24:595–600, 2005.

[102] W. Chen Q. Peng Y. Zeng, D. Samaras. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images. *Computer Vision and Image Understanding*, 112:81–90, 2008.

[103] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[104] S. Yu, R. Gross, and J. Shi. Concurrent object segmentation and recognition with graph partitioning. *Neural Information Processing Systems*, 2002.

[105] H. Zhang, J. Fritts, and S. Goldman. An entropy-based objective segmentation evaluation method for image segmentation. *SPIE Electronic Imaging Storage and Retrieval Methods and Applications for Multimedia*, pages 38–49, 2004.

[106] H. Zhang, Fritts J, and S. Goldman. A co-evaluation framework for improving segmentation evaluation. *SPIE Signal Processing and Target Recogni-*

tion, pages 420–430, 2005.

[107] L. Zhang and B. Paul. Edge detection by scale multiplication in wavelet domain. *Pattern Recognition Letters*, 23:1771–1784, 2002.