Pakaket Wattuya

Combination of Multiple Image Segmentations

-2010-

Informatik

Combination of Multiple Image Segmentations

Inaugural-Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften im Fachbereich
Mathematik und Informatik
der Mathematisch-Naturwissenschaftlichen Fakultät
der Westfälischen Wilhelms-Universität Münster

vorgelegt von
Pakaket Wattuya
aus Bangkok, Thailand
- 2010 -

Dekanin/Dekan:	Prof. Dr. Christopher Deninger
Erster Gutachter:	Prof. Dr. Xiaoyi Jiang
Zweiter Gutachter:	Prof. Dr. Horst Bunke
Tag der mündlichen Prüfung(en):	16 July 2010
Tag der Promotion:	16 July 2010

Abstract

The main focus of this thesis concerns combination of multiple image segmentations in the fields of contour detection and region-based image segmentation. The goal of a multiple segmentation combination concept is to combine multiple *imperfect* segmentation results produced from multiple sources into a single *improved* segmentation result. In Part One the concept of multiple segmentation combination is applied to a contour averaging problem. The contour averaging problem is formally formulated within the framework of generalized median as an optimization problem. A new efficient algorithm based on dynamic programming to exactly compute the generalized median contour is presented, as well as the usefulness of the exact solution of generalized median contour in verifying the tightness of a lower bound for generalized median problems in metric space.

Part Two of this thesis focuses on the combination of region-based image segmentations. A novel algorithm for combining multiple segmentations to achieve a final improved segmentation is presented. In contrast to previous works we consider the most general class of segmentation combination, i.e. each input segmentation can have an arbitrary number of regions. Our approach is based on a random walker segmentation algorithm which is able to provide high-quality segmentation starting from manually specified seeds. We automatically generate such seeds from an input segmentation ensemble. A median concept based optimality criterion is proposed to automatically determine the final number of regions in a final combined result. In addition, the study of the interplay between accuracy and diversity of segmentation ensemble and its influence on final segmentation combination performance are carried out. Finally, we describe a number of real-world applications in computer vision that can be solved efficiently and reliably using our proposed combination algorithm. Experiments demonstrate the effectiveness of the proposed algorithm in a variety of imagery data and image segmentation methods.

In Part Three we focus on experimentally investigating a number of existing well-known segmentation evaluation measures. A metric property of these measures

is addressed and behavioral clustering frameworks for clustering them have been proposed. The results of this study are intended to be as a guideline for appropriately using and choosing the existing evaluation measures.

Acknowledgements

I would like to express my greatest thanks to my supervisor, Prof. Dr. Xiaoyi Jiang, for his great supervision and his the most generous encouragement and support through my success. My work has always been inspired and enhanced by his irrepressible intellectual inspiration and guidance. His point of view always broadens my thinking and I would like to thank him for all of those invaluable discussions and recommendations. I have learned so much during these years. Thank you for making me appreciate the great experience of research. And lastly, I would like to take this opportunity to thank him again for giving me the great opportunity to work with him in his research group.

I would like to express my sincere thank to Prof. Dr. Horst Bunke who has kindly reviewed my dissertation. I am very grateful for his interest in my work and his kindness. I would like to express my special thank to Prof. Dr. Klaus Hinrichs and Prof. Dr. Wolfram-M. Lippe for their kindness and being my dissertation committee.

I would like to thank Mrs. Hildegard Brunstering for her wonderful helpfulness and friendliness. She always keeps things well organized.

I would like to thank Kai Rothaus, whose comments and ideas have been very useful and influential in the development of this work. He has never disappointed me anytime I have discussed with him. And thank you for his willingness to help whenever I got into troubles. I would like to thank Dr. Da-Chuan Cheng for his very warm welcome, his constant helps, and being a great friend. Daniel Duarte Abdala and Lucas Franek, working with them has been an interesting, pleasure and enjoyable.

I would like to thank all the members of Computer Vision and Pattern Recognition Research Group, colleagues and professors in the Institute of Computer Science for their friendship and friendly environment. I have enjoyed entire course of my study here.

Finally, I would like to thank my family for supporting and encouraging me in all possible ways along these years.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Image segmentation is defined as the meaningful partitioning of images into non-overlapping homogeneous regions exhibiting similar features or image content. In general, image segmentation is a key step towards high level tasks such as image understanding, and serves in a variety of applications including object recognition, scene analysis or image/video indexing. Due to its importance, numerous approaches for image segmentation have been developed and proposed. Over the last 40 years, image segmentation has evolved very quickly and has undergone great change [149]. A comprehensive survey of image segmentation techniques presented thus far are discussed and summarized in [18, 47, 59, 86, 102, 149]. In spite of several decades of intensive research and a large extent of progress in general purpose image segmentation, image segmentation remains a challenging unsolved issue.

- *Instability of Segmentation Algorithm*: Image segmentation is known to be unstable, strongly affected by small image perturbations and feature choices [104]. A single segmentation algorithm with a single segmentation technique and a single feature set may (often) not be able to comprehensively capture the large degree of variability and complexity encountered in many real-world images. In fact different segmentation techniques, as well as different set of image features, may be able to capture different facets of true image structure. Ensemble combination provides a powerful means for combining such information. In this thesis, we study the question of how to best integrate such information from multiple segmentations of an image to improve the accuracy and robustness of segmentation result.

- *Parameter Selection Problem*: Image segmentation algorithms mostly have some parameters that define the behavior of their operations. As a consequence, the segmentation results depend heavily on the choices of initial parameter values. Different initial parameter values may yield to completely different results as illustrated in Figure 1.1. The granularity of the regions changes with the changes of parameter values. Thus, initial parameter values need to be set appropriately in order to obtain a quality segmentation result. However, a lack of both assumption about data distribution structures and prior information about statistical properties of the regions to be segmented presents a difficulty for handling the initial parameter values correctly. Moreover, adequate values of the algorithm parameters for one image may not be effective for others, and this may lead to an undesirable result as illustrated in Figure 1.2. No single setting of parameter has been found that performs adequately across a wide diversity of images. A high variation in input images, due to effects such as shading, highlights, non-uniform illumination or texture, involves additional difficulties in image segmentation problem.

  These difficulties arise the problem of algorithm parameter selection. The parameter selection problem has not received the due attention in the past. Researchers typically claim to have empirically determined the parameter values (in an ad-hoc manner). More systematically, the optimal parameter values can be trained in advance based on manual ground truth by exploring a subspace of the parameter space to find out the best parameter [8, 22, 97, 107]. In fact the parameter selection and/or parameter learning should be usually done on a large enough data set, so that it well enough represents the entire domain for building up a general model for segmentation. However, it is often not possible to obtain a large enough data set and, furthermore, ground truth segmentations for training procedure are often not available. Another class of methods assumes a segmentation quality measure, which is used to control a parameter optimization process [1, 105]. However, these approaches are typically restricted to a specific application or a specific domain of images they work with. (The problem of parameter selection is addressed at length further on in Chapter 7.)

  In fact for most image segmentation algorithms each image requires its own set of parameter values in order to obtain quality and satisfactory segmentation results. We encourage that image segmentation algorithms should possess adaptive behavior to adjust values of its own parameters according to the changes of image quality and image characteristics.

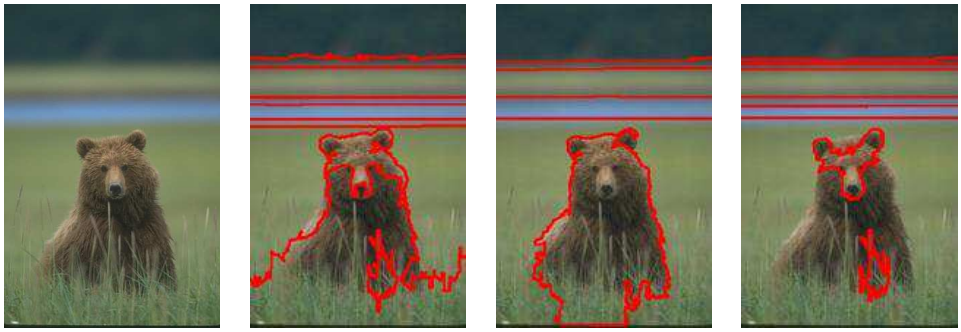


Figure 1.1. Illustration of the problem of segmentation algorithm instability. Segmentations of the same input image obtained by the FH algorithm [38] given the different set of parameter values (b) $\sigma = 0.6, k = 300, M = 1500$, (c) $\sigma = 0.6, k = 700, M = 1500$, and (d) $\sigma = 0.7, k = 700, M = 1500$. The details of segmentation algorithm and its parameters will be given in Chapter 2.

Figure 1.2. Illustration of the problem of segmentation algorithm parameter selection. Segmentations obtained by the FH algorithm [38] given the same set of parameter values ($\sigma = 0.9, k = 700, M = 1500$). The details of segmentation algorithm and its parameters will be given in Chapter 2.

In this thesis we address this problem of finding the optimal setting of algorithm parameters, preferably on a per-image basis, and propose the multiple segmentation combination strategy as a solution to the problem. The fundamental idea is not to explicitly determine the optimal parameter setting for a particular image. Instead, we compute a set of segmentations (ensemble) according to a subspace sampling of the parameter space and then try to reach an optimum out of the segmentation ensemble. The main advantage of our approach is that the parameter selection problem can be effectively solved without the need of ground truth and in a fully automatic manner.

- *Algorithm Selection Problem*: Although there has been a large extent of progress in general purpose image segmentation, ranging from simple statistical mod-

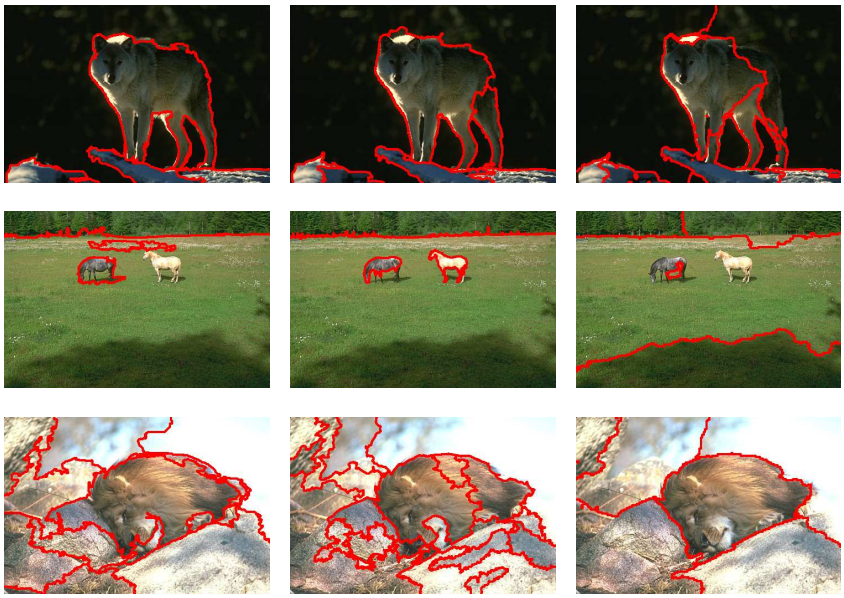
els [118], adaptive filters [124] to sophisticated methodologies such as color and texture analyzes, wavelets [82], fuzzy sets [19, 99], and neural networks [33], it remains an extremely difficult problem when facing with the challenging segmentations of complex pictures such as outdoor and natural images. Those algorithms all suffer from sensitivity to the properties of images, such as noise level, illumination condition, and the target size [145].

Image segmentation techniques are basically ad hoc and differ precisely in the way they emphasize one or more of the desired properties and in the way they balance and compromise one desired property against another [59]. Consequently, the results of different segmentation algorithms on a particular image differ greatly due to their objective constraints they try to satisfy as illustrated in Figure 1.3. The segmentations created by each algorithm exhibit different natures. More importantly, these underlying segmentation constraints often limit the use of the algorithm in the wide-range of images. In fact there is no single method which can be considered good for all images, nor are all methods equally good for a particular type of image [102].

As a matter of fact, many researchers [46, 51, 60, 145] have suggested an effective and straightforward solution by using different algorithms to segment different images. However, automated selection of an optimal algorithm for one particular image is not trivial task. Most recent approaches for selecting an optimal segmentation algorithm according to image characteristics have exploited machine learning techniques and learning-based system [93, 120, 144, 145, 150]. The main drawback of these approaches is their requirement of either the assumption of ground truth segmentations or the human intervention in a training process. (The problem of algorithm selection is addressed at length further on in Chapter 6)

To tackle the segmentation algorithm selection problem, we neither explicitly select the optimal segmentation algorithm for a particular image nor are interested in optimizing a segmentation algorithm for a given task. Instead, we propose to use the segmentation combination strategy to solve the problem. The rationale behind this idea is that while none of the segmentation algorithms is likely to segment an image correctly, we may benefit from combining the strengths of such multiple segmenters. The advantages of our approach are that it requires no assumption of ground truth segmentations and no human intervention in a framework operation.

Another potential challenging issue concerning the field of image segmentation is image segmentation evaluation. Performance evaluation is not only important for



(a) FH (b) MS (c) mNC

Figure 1.3. Illustration of the problem of segmentation algorithm selection. Segmented images are computed by FH, MS, and mNC segmentation algorithms (left/middle/right). The comparative performance of different segmentation algorithms can vary significantly across images. The details of segmentation algorithms will be given in Chapter 2.

evaluating and comparing the performance of individual image segmentation methods, but also useful for parameter tuning/learning [12, 97] and algorithm selection problem [51, 120, 147]. Despite of its importance, image segmentation evaluation has not received the due attention in the past. Moreover, most efforts spent on evaluation are just for designing new evaluation methods and only very few authors have attempted to characterize the different existing evaluation methods [148]. The most well known and cited by many authors in this area is the work of Zhang [147]. Zhang studies different segmentation evaluation methods proposed so far, and classified them into three groups: the analytical, the empirical goodness and the empirical discrepancy groups. A brief description of each method in every group and some comparative discussions about different method groups are carried out. A brief review of supervised evaluation methods can also be found in [10, 73, 132, 143].

In this thesis we experimentally investigate the existing supervised evaluation measures for image segmentation in two following frameworks:

- *Comparison of the metric property*: The well known segmentation evaluation measures commonly used in the computer vision literature are compared ac-

cording to their property of being a metric. Being a metric is the highly desired property for distance measures in pattern matching and visual applications in order to match the human intuition of similarity. There is essentially no literature for any kind of segmentation evaluation measure which investigate the metric property. An experimental comparison is performed to provide a rank of how likely these evaluation measures are metric. We hope that this study would be helpful for an appropriate use of existing evaluation methods, where the property of a metric is expected, for example, in this work the computation of generalized median.

- *Clustering of existing evaluation measures*: For last decades many different segmentation evaluation measures have been proposed in the literature. These measures are typically endowed with different standard for measuring the quality of the segmentation. As a result, evaluating results vary significantly between different evaluation measures. In particular, it is difficult for the users to choose an appropriate measure when they are faced with such a variety of possibilities.

  The segmentation evaluation measures under consideration are clustered into groups based on their behaviors in evaluating the same series of segmented images. The evaluation measures' behavioral characteristics are captured through the use of selecting and ranking strategies. The basic idea is that the evaluation measures with similar behavioral characteristics will select or rank the segmentation results in a similar manner and will be clustered into the same group. We hope that this behavioral clustering study could be useful for users as a guideline in choosing different appropriate evaluation measures, especially from different clusters, in order to fairly report the performance of the proposed algorithm.

We hope that these two analytical studies will give pioneer frameworks for comparing and clustering other evaluation measures existing in literatures.

1.2 Objectives

The main objectives of this thesis are summarized as follows:

- To propose an algorithm for combining multiple image segmentations to achieve a final *improved* segmentation.

- To investigate three different potential application scenarios to demonstrate the usefulness of segmentation combination: 1) Exploring parameter space without ground truth, which addresses the problem of parameter selection, 2) Multiple segmentation algorithm combination, which addresses the problem of optimal algorithm selection, and 3) Segmentation algorithm instability problem.

Along with the main objectives, there are important relevant issues that are integral parts of our approach and need to be considered in this work.

- To propose the new optimality criterion for automatically determining the final number of regions in a combination result.
- To propose two novel frameworks for experimentally investigating a number of existing (supervised) evaluation measures for assessing the quality of image segmentations: Comparing the metric property of evaluation measures and behavioral clustering of evaluation measures.

1.3 Thesis Organization

The remainder of the thesis is organized as follows.

In Chapter 2 we give an overview of some fundamental concepts and algorithms that are required for understanding and building our algorithm framework and will be used throughout the thesis, such as the generalized median concept, three baseline image segmentation algorithms, fourteen commonly used evaluation measures, and the segmentation evaluation methodology.

In Chapter 3 we present our general framework of segmentation combination. Its components, features and goals are discussed, as well as examples of its possible applications. This general framework will be applied for both multiple contour combination and multiple region-based image segmentation combination.

The remainder of the thesis is organized in three parts.

The first part composing of only one chapter (Chapter 4) focuses on the problem of contour averaging. Contour averaging has found several applications in computer vision including prototype formation and computational atlases. A contour averaging problem is formal formulated within the framework of generalized median as an optimization problem. A special class of contours, which frequently occurs in

many applications of image analysis, is considered. We propose an efficient algorithm based on dynamic programming to exactly compute the generalized median contour in this domain. Experimental results will be reported on two scenarios to demonstrate the usefulness of the concept of generalized median contours: Exploring the parameter space of a (segmentation) algorithm and verification of optimal lower bound for generalized median problems in metric space.

The second part composing of Chapter 5–Chapter 8 is devoted to the problem of multiple region-based image segmentation combination and its potential applications .

In Chapter 5 the problem of multiple region-based image segmentation is discussed. We propose a novel algorithm for combining multiple segmentations to achieve a final improved segmentation result. The proposed algorithm is based on a random walker algorithm for image segmentation. In contrast to previous works we consider the most general class of segmentation combination, i.e. each input segmentation can have an arbitrary number of regions. A new optimization method based on the generalized median concept for automatically estimating the number of regions in a final combined result is also proposed. We investigate the effectiveness of this generalized median-based criterion by comparing it with three existing different criteria.

In Chapter 6 a variety of segmentation ensemble generation approaches is presented to verify the effectiveness of our segmentation combination algorithm in various situations. The study of the interplay between accuracy and diversity of such segmentation ensemble and its influence on final segmentation combination performance are carried out. In addition, the problem of optimal algorithm selection is also exhaustively addressed in this chapter.

In Chapter 7 the proposed segmentation combination algorithm is applied to solve the potential problem of parameter selection. The efficacy of our combination approach is compared to three training approaches, ranging from simple traditional approach to a more adaptive approach such as cased-based reasoning. Extensive experimental comparisons are conducted on both natural image and real range image data sets.

In Chapter 8 we demonstrate another usefulness of our segmentation combination for solving the problem of instability of image segmentation algorithm. The instability of the segmentation algorithm caused by parameter variation and noise is investigated. We compare the ability of our segmentation combination in dealing with this problem to the set median concept approach.

In third part composing of Chapter 9 and Chapter 10 is about the image segmentation evaluation.

In Chapter 9 the metric property of evaluation measures is addressed and fourteen well-known (supervised) evaluation measures are compared in terms of this property. This study would hopefully be helpful for appropriate use of the existing evaluation methods, where the property of a metric is expected.

In Chapter 10 the same set of evaluation measures considered in Chapter 9 is clustered into groups based on their evaluating behaviors on the same set of test images. This study is intended to provide a guideline for a user in choosing appropriate evaluation measures, especially from different clusters, in order to fairly report the performance of the proposed algorithm.

Contributions of our work in summary and conclusions on this thesis are given in Chapter 11.

Chapter 2

Fundamental Concepts

Before proceeding to present our multiple segmentation combination framework and algorithms, there are some fundamental concepts and algorithms that are required for understanding and building of our algorithm framework. This chapter gives an overview of these necessary backgrounds that will be used throughout the thesis. The first section provides a general overview of *median concept*. Median concept plays an important role in the contour combination and the estimation of the number of regions in region-based segmentation combination in subsequent chapters. The second section gives a short methodological review of well-known image segmentation algorithms that will be used as baseline segmentation algorithms in ensemble generation procedure. The third section gives a brief review of existing evaluation measures for evaluating quality of segmentation result. Some of these measures are used as measures for quantitatively evaluating the quality of the resulting segmentations. The comparison and clustering analysis of these measures will also be conducted and reported in the last part of our thesis. The last section discusses a method to objectively evaluate the segmentation performance by comparing the machine segmentation result against its corresponding ground truth (human segmentation). The human segmentation data set that is used in most of our experiments throughout the thesis is also detailed.

2.1 Median Concept

The general concept of average, or mean, has turned out to be useful in numerous contexts of science and engineering. In general, we are given a set of noisy samples of the same object and want to infer a representative model. One powerful tool for

this purpose is provided by the generalized median concept.

Assume that we are given a set S of objects in some representation space U and a distance function $d(p, q)$ to measure the dissimilarity between any two objects $p, q \in U$. The essential information of the given set of objects is captured by an object $\overline{p} \in U$ that minimizes the sum of distances to all objects from S, i.e.

$$\overline{p} \;=\; \arg\min_{p \in U} \sum_{q \in S} d(p, q) \tag{2.1}$$

Object $\overline{p}$ is called a *generalized median* of S. A related concept is the so-called *set median*, which results from constraining the search to the given set S

$$\hat{p} \;=\; \arg\min_{p \in S} \sum_{q \in S} d(p, q) \tag{2.2}$$

The set median may serve as an approximative solution for the generalized median. Note that neither the generalized median nor the set median is unique in general.

Independent of the object type and the underlying representation space we can always find the set median of N objects by means of $\frac{1}{2}N(N-1)$ pairwise distance computations (although more efficient algorithms have been reported as well). In contrast there is no general approach to computing generalized medians. The reason is that any such algorithm must be of constructive nature and the construction process crucially depends on the structure of the objects under consideration. Additional difficulty is caused by the fact that determining the generalized median is provably of high computational complexity in several cases.

2.2 Baseline Segmentation Algorithms

The purpose of this section is to give an overview of the well-known image segmentation techniques which will be used as the baseline segmentation algorithms for producing initial segmentations for our combination approach. We will describe their underlying principles and discuss the particular characteristics of each class of algorithms.

Three image segmentation algorithms are chosen from three different categories of image segmentation methods which are widely-used in the vision community.

- *Mean Shift-based Method*: Mean Shift image segmentation (MS) proposed by Comaniciu and Meer [23] is based on *feature space analysis* techniques. The versatility of the feature space analysis enables the design of algorithms in

which the user controls performance through a single parameter which is the resolution of the analysis (i.e., bandwidth of the kernel). Comaniciu and Meer applied the feature space analysis technique to a feature space that represents both an L*u*v* representation of the color image (range domain) and the spatial coordinates of a pixel (spatial domain). The multivariate kernel is defined as the product of two radially symmetric kernels and the Euclidean metric allows a single bandwidth parameter for each domain

$$K_{h_s,h_r}(x) = \frac{C}{h_s^2 h_r^2} k(\left\| \frac{x_s}{h_s} \right\|^2) k(\left\| \frac{x_r}{h_r} \right\|^2),$$

where x^s is the spatial, x_r is the range part of a feature vector, $k(x)$ the common profile used in both two domains, h_s and h_r the employed kernel bandwidths, and C the corresponding normalization constant. In practice, a normal kernel always provides satisfactory performance, so that the user only has to set the bandwidth parameter $rmh = (h_s, h_r)$, which determines the resolution of the mode detection.

The mean shift technique for image segmentation is comprised of two basic steps:

- *Mean Shift Filtering*: this step consists of finding the modes of the probability density function underlying the image data in feature space which correspond to the locations with highest data density. In terms of a segmentation, it is intuitive that the data points close to these high density points (modes) should be clustered together.
- *Mean Shift Segmentation*: After mean shift filtering, each data point in the feature space has been replaced by its corresponding mode. Clustering proposed in [23] is described as a simple post-processing step in which any modes that are less than one kernel radius apart are grouped together and their basins of attraction are merged.

Mean shift image segmentation is able to produce segmentations that correspond well to human perception. However, this algorithm is quite sensitive to its parameters, especially h_r. Slight variations in h_r can cause large changes in the granularity of the segmentation. This algorithm is used in a graphical interface EDISON system which is publicly available at [52].

- *Graph-based methods*: They treat an image as a connected graph $G = (V, E)$ where each node $v_i \in V$ corresponds to a pixel in the image, and the edges in E connect certain pairs of neighboring pixels. The weight of an edge is some

measure of the dissimilarity between the two pixels connected by that edge (e.g., the difference in intensity, color, motion, location or some other local attribute).

Felzenszwalb and Huttenlocher [38] proposed the *efficient graph-based image segmentation algorithm* (FH) for general purpose image segmentation. In contrast to MS, this algorithm works directly on the data points in feature space, without first performing a filtering step. The underlying principle of FH algorithm is based on the idea that the image should be partitioned into regions such that for any pair of regions, the variation across regions should be larger than the variation within the region. They develop a simple algorithm which computes segmentations according to this idea by defining two measures:

- the *internal difference*, $Int(C)$, which measures the dissimilarity among neighboring elements within a component $C \subseteq V$, is defined to be the largest weight in the minimum spanning tree of the component, $MST(C, E)$:

$$Int(C) = \max_{e \in MST(C,E)} w(e)$$

- the *external difference*, $Dif(C_1, C_2)$, which measures the dissimilarity between elements along the boundary of the two components $C_1, C_2 \subseteq V$ to be the minimum weight edge connecting the two components:

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j).$$

The algorithm start with a segmentation S^0, where each vertex v_i is in its own component. Then it iteratively merges disjoint components where the external variation between them is small with regard to their respective internal variations,

$$Dif(C_1, C_2) > MInt(C_1, C_2)$$

and

$$MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)).$$

The threshold function $\tau(C) = k/\left|C\right|$ controls the degree to which the difference between two components must be greater than their internal differences, where $|C|$ denotes the size of C, and k is some constant parameter.

The key success of this method is that, unlike the classical methods, this technique adaptively adjusts the segmentation criterion based on the degree of variability in neighboring regions of the image. This results in a method that, while making greedy decisions, can be shown to obey certain non-obvious global properties. However, this algorithm suffers somewhat from sensitivity to a parameter k.

This segmentation algorithm is attractive due to its competitive segmentation performance and high computational efficiency. In fact, the running time is nearly linear in the number of graph edges and very fast in practice. Felzenszwalb and Huttenlocher have made available an implementation of their algorithm at [37].

- *Spectral Methods*: Spectral segmentation methods also model images as connected graphs. Similar to the graph-based methods defined above, the weight w_{ij} of the edge connecting two vertices i, j measures the similarity between two image elements and can be stored in an affinity matrix W. Spectral methods identify partitions via the eigenvectors of the affinity matrix (or other matrices derived from it) by using dominant eigenvectors of matrices to perform segmentation. These approaches are attractive in that they are based on simple eigen–decomposition algorithms whose stability is well understood. Nevertheless, the use of eigen–decompositions in the context of segmentation is far from well understood [142].

  Cour et al. [25] applied spectral analysis techniques to solve the image segmentation problem, called *multiscale Normalized Cuts* (mNC). The algorithm works on multiple scales of the image in parallel with the use of the Normalized Cut graph partitioning framework [121]. The algorithm solves a cross scale constraint matrix which processes the different spatial scales in parallel by forcing the system to seek an average segmentation across all scales. Let X be a multiscale partitioning matrix, where $X_s \in \{0,1\}^{N_s \times K}$ is the partitioning matrix at scale s, $X_s(i,k) = 1$ iff graph node i belongs to partition k. The algorithm segments an image by finding the graph cut that correspond to the constrained multiscale Normalize Cut:

$$\text{maximize } \epsilon(X) = \frac{1}{K} \sum_{l=1}^{K} \frac{X_l^T W X_l}{X_l^T D X_l}$$

$$\text{subject to } CX = 0, X \in \{0,1\}^{N^* \times K}, X 1_K = 1_{N^*},$$

  where C is a cross-scale constraint matrix and $CX = 0$ is a cross-scale segmentation constraint equation, $N^* = \sum_s N_s$ and D is a diagonal matrix,

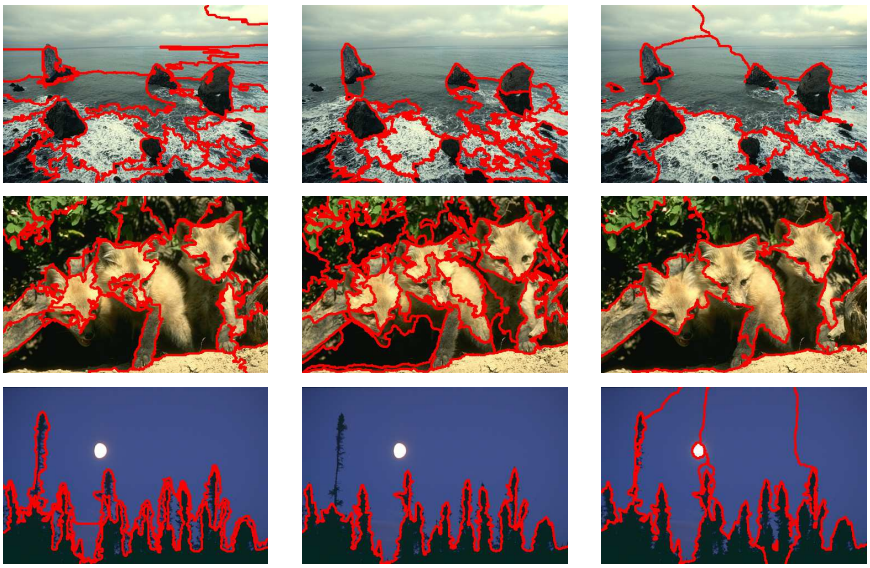
Table 2.1. Parameters and descriptions of baseline segmentation algorithms

Algo.	Parameter	Description
MS	h_s	a spatial bandwidth parameter of the kernel function, determining the resolution of the mode detection.
	h_r	a range bandwidth parameter of the kernel function, determining the resolution of the mode detection.
	M	specify a minimum size of regions in the result enforced by post-processing. The range parameter h_r and M control the number of regions in the segmented image.
FH	σ	a gaussian filter parameter which is used to smooth the image before computing in order to compensate for digitization artifacts.
	k	a parameter of a threshold function, τ, a larger k causes a preference for larger components in the result. Setting of k depends on the resolution of the image and the degree to which fine detail is important in the scene.
	M	specify a minimum size of regions in the result enforced by post-processing.
mNC	*scale*	specify a scale of input image to be segmented.
	nsegs	specify a number of segments in the segmented image.

$D(i,j) = \sum_j W(i,j)$, and 1_N is a vector of N ones. A graph weight W is defined based on two simple and effective local grouping cues, namely, intensity and contours.

The complexity of this algorithm is linear in the number of pixels and the number of segments requested, where the main computation bottleneck is in the eigenvector computation. We choose this algorithm because it is a general purpose approach and is well representative of spectral method in image segmentation. The implementation of this algorithm is publicly available at [24]. More review and discussion of different spectral clustering methods can be found in [142].

Algorithm parameters of each segmentation algorithm are summarized in Table 2.1. Sample segmentations produced by the three image segmentation algorithms are shown in Figure 2.1. It is worth noticing that the FH algorithm tends to produce long, thin regions along image edges while the MS algorithm produces reasonable segmentations at coarser levels. However, both algorithms provoke also noticeable



(a) FH (b) MS (c) mNC

Figure 2.1. Sample segmented images computed by FH, MS, and mNC segmentation algorithms. Parameter set for FH: $\sigma = 0.9, k = 300, M = 1500$; MS: $h_s = 8, h_r = 7, M = 1500$; and mNC: $scale = 0.8, nseg = 12$.

over-segmentation. Multiscale NCuts attempts to find global solution with larger segments that have a chance to be objects but often oversegmenting large homogeneous regions.

In the subsequent chapters, the word 'MS' refers to the mean shift-based segmentation method by Comaniciu and Meer [23], the word 'FH' refers to the efficient graph-based segmentation method by Felzenszwalb and Huttenlocher [38], and the word 'mNC' refers to the multiscale normalized cuts method by Cour et al. [25].

2.3 Review of Segmentation Evaluation Measures

In this section we review well-known evaluation measures that are used in this study. These measures are chosen because of their extensive use in the literature. The measures will be reviewed according to their categories. The first category involves the methods specifically derived for segmentation evaluation task, while the second category involves the methods developed in statistics for comparing clusterings but popularly used in the computer vision literature.

2.3.1 Measures for Comparing Segmentations

Region consistency

Martin et al. [90] proposed two measures of error that can be used to evaluate the consistency of a pair of segmentations. The measures are designed to be tolerant to refinement, that is, if one segment is a proper subset of the other, then the pixels lie in an area of refinement, and the local error should be zero. If there is no subset relationship, then the two regions overlap in an inconsistent manner. In this case, the local error should be non-zero. Let $R(S, p_i)$ be the set of pixels corresponding to the region in segmentation S that contains pixel p_i, the asymmetric local refinement error between two input segmentation S_1 and S_2 is defined as:

$$E(S_1, S_2, p_i) = \frac{|R(S_1, p_i) \backslash R(S_2, p_i)|}{|R(S_1, p_i)|} \tag{2.3}$$

where $\backslash$ denote set difference, and $|x|$ the cardinality of set x. The error measure evaluates to 0 if all the pixels in S_1 are also contained in S_2. This local refinement error encodes a measure of refinement in one direction only. Given this local refinement error in each direction at each pixel, there are two natural ways to combine the values into an error measure for the entire image. Let n be the number of pixels:

$$GCE(S_1, S_2) = \frac{1}{n} \min \left\{ \sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i) \right\} \tag{2.4}$$

$$LCE(S_1, S_2) = \frac{1}{n} \sum_i \min\{E(S_1, S_2, p_i), E(S_2, S_1, p_i)\} \tag{2.5}$$

Global Consistency Error (GCE) forces all local refinements to be in the same direction, while Local Consistency Error (LCE) allows refinement in different directions in different parts of the image. Since both measures are tolerant of refinement, there are two trivial segmentations that achieve zero error: One pixel per segment, and one segment for the entire image. The former is a refinement of any segmentation, and any segmentation is a refinement of the latter. Thus, Martin [89] proposed an alternative measure that does not tolerate refinement termed the Bidirectional Consistency Error (BCE). The measure penalized dissimilarity between segmentations proportional to the degree of region overlap by replacing the poxelwise minimum with a maximum, defined as:

$$BCE(S_1, S_2) = \frac{1}{n} \sum_i \max\{E(S_1, S_2, p_i), E(S_2, S_1, p_i)\} \tag{2.6}$$

The values of these three error measures lie in the range [0,1], with a value of 0 indicating no error and a value of 1 indicating maximum deviation between two segmentations to be compared.

Another region-based evaluation is proposed by Huang and Dom [65]. They introduced the concept of directional Hamming distance to quantitatively describe the degree of mismatch from one segmentation $S_1 = \{R_1^1, R_1^2, ..., R_1^m\}$ to another segmentation $S_2 = \{R_2^1, R_2^2, ..., R_1^n\}$. They associate each region R_2^i from S_2 with a region R_1^j from S_1 such that $R_2^i \cap R_1^j$ is maximal. Directional Hamming distance from S_1 to S_2 is defined as:

$$D_H(S_1 \Rightarrow S_2) = \sum_{R_2^i \in S_2} \sum_{R_1^k \neq R_1^j, R_1^k \cap R_2^i \neq \oslash} \left| R_2^i \cap R_1^k \right|$$

where $|\cdot|$ denotes the size of a set. Therefore, $D_H(S_1 \Rightarrow S_2)$ is the total area under the intersections between all $R_2^i \in S_2$ and their non-maximal intersected regions R_1^k from S_1. The reversed distance $D_H(S_2 \Rightarrow S_1)$ can be similarly computed. The overall performance measure based on normalized Hamming distance is defined as

$$p = 1 - \frac{D_H(S_1 \Rightarrow S_2) + D_H(S_2 \Rightarrow S_1)}{2\,|S|} \tag{2.7}$$

where $|S|$ is the image size and $p \in [0, 1]$. The smaller the degree of mismatch, the closer the p is to one.

Boundary Matching

F-measure is a boundary-based evaluation developed by Martin et al. [92]. It was proposed solving an approximation to a bipartite graph matching problem for matching segmentation boundaries and computing the percentage of matched edge elements. In this framework the two terms of measures for boundary detection, precision and recall, are computed. Precision (P) is the fraction of detections which are true positives, while recall (R) is the fraction of positives that are detected. The F-measure is an overall performance measure that captures the trade-off between these two quantities as the weighted harmonic mean of P and R, defined as:

$$F = PR/(\alpha R + (1 - \alpha)P) \tag{2.8}$$

This yields a value of F-measure between zero and one where a value of one indicates a perfect matching between two segmentations. A relative cost α between P and R quantities focuses attention at a specific point on the precision-recall curve. We set α to 0.5 in our experiments.

2.3.2 Measures for Comparing Clusterings

Considering image segmentation as a pixel clustering process, we can apply measures for comparing clusterings developed in statistics for the purpose of segmentation

evaluation.

A clustering $\mathcal{C}$ is a partition of a set of points, or data set D into mutually disjoint subsets $C_1, C_2, ..., C_K$ called clusters. Formally, $\mathcal{C} = \{C_1, C_2, ..., C_K\}$ such that $C_k \cap C_l = \emptyset$ and $\bigcup_{k=1}^{K} C_k = D$. Let the number of data points in D and in cluster C_k be n and n_k, respectively. We have, of course, that $n = \sum_{k=1}^{K} n_k$.

We also assume that $n_k > 0$, in other words, that K represents the number of non-empty clusters. Let a second clustering of the same data set D be $\mathcal{C}' = \{C'_1, C'_2, ..., C'_K\}$, with cluster sizes $n'_{k'}$. Note that the two clusterings may have different numbers of clusters.

Comparing clusterings by counting pairs

An important class of criteria for comparing clusterings is based in counting the pairs of points on which two clustering agree/disagree. A pair of points from D can fall under one of four cases described below.
N_{11} - number of point pairs that are in the same cluster under both $\mathcal{C}$ and $\mathcal{C}'$
N_{00} - number of point pairs in different clusters under both $\mathcal{C}$ and $\mathcal{C}'$
N_{10} - number of point pairs in the same cluster under $\mathcal{C}$ but not under $\mathcal{C}'$
N_{01} - number of point pairs in the same cluster under $\mathcal{C}'$ but not under $\mathcal{C}$
The four counts always satisfy $N_{11} + N_{00} + N_{10} + N_{01} = n(n-1)/2$.

Several comparing measures are based on these four counts. The Rand index introduced in [111] is the percentage of pairs for which there is an agreement and defined as:

$$\mathcal{R}(\mathcal{C}, \mathcal{C}') = \frac{N_{11} + N_{00}}{n(n-1)/2} \tag{2.9}$$

This gives a measure of similarity with values ranging over [0,1] interval. $\mathcal{R}$ is 1 for identical clusterings.

Hubert and Arabie [66] noticed that the Rand index is not correct for chance that is equal to zero for random partitions having the same number of objects on each class. They, therefore, introduced the adjusted version of the Rand index, whose expectation is equal to zero. The resulting adjusted Rand index has the expression

$$\mathcal{AR}(\mathcal{C}, \mathcal{C}') = \frac{\mathcal{R}(\mathcal{C}, \mathcal{C}') - E[\mathcal{R}]}{1 - E[\mathcal{R}]} \tag{2.10}$$

Thus, the adjusted Rand index can take on a wider range of values, ranging in the range [-1,1]. $\mathcal{AR}$ is 1 when the two partitions are identical.

Unnikrishnan et al. [132] proposed the modifications to the basic Rand index, termed the Probabilistic Rand (PR) index, that allows comparison of a test segmentation with multiple ground truth images $\mathcal{C}'_K$, defined as

$$\mathcal{PR}(\mathcal{C}, \mathcal{C}'_K) = \frac{1}{T} \sum_{i,j} [n_{ij} p_{ij} + (1 - n_{ij})(1 - p_{ij})] \tag{2.11}$$

where p_{ij} is the probability that pixels i and j have the same label. When the sample mean is used to estimate p_{ij}, PR index is simply an average value of Rand index among different ground truth segmentations in a set [4].

There are other criteria in the literature, to which this class of criteria applies. Wallace [134] proposed the two asymmetric criteria $\mathcal{W}_I$, $\mathcal{W}_{II}$ below:

$$\mathcal{W}_I(\mathcal{C}, \mathcal{C}') = \frac{N_{11}}{\sum_k n_k(n_k - 1)/2}, \quad \mathcal{W}_{II}(\mathcal{C}, \mathcal{C}') = \frac{N_{11}}{\sum_k n'_{k'}(n'_{k'} - 1)/2} \tag{2.12}$$

They represent the probability that a pair of points which are in the same cluster under $\mathcal{C}$ (respectively $\mathcal{C}'$) are also in the same cluster under the other clustering.

Fowlkes and Mallows [42] introduced a criterion which is symmetric, and is the geometric mean of $\mathcal{W}_I$, $\mathcal{W}_{II}$:

$$\mathcal{F}(\mathcal{C}, \mathcal{C}') = \sqrt{\mathcal{W}_I(\mathcal{C}, \mathcal{C}')\mathcal{W}_{II}(\mathcal{C}, \mathcal{C}')} \tag{2.13}$$

The Jacard index [7] is given by

$$\mathcal{J}(\mathcal{C}, \mathcal{C}') = \frac{N_{11}}{N_{11} + N_{01} + N_{10}} \tag{2.14}$$

The above two indices give a measure of similarity with a value domain [0,1]. The value is 1 when the two clusterings are identical.

The Mirkin [98] metric is another adjusted form of the Rand index and can be written as [94]:

$$\mathcal{M}(\mathcal{C}, \mathcal{C}') = \sum_k n_k^2 + \sum_{k'} n_{k'}'^2 - 2 \sum_k \sum_{k'} n_{kk'}^2 = 2(N_{01} + N_{10}). \tag{2.15}$$

$\mathcal{M}$ is 0 for identical clusterings and positive otherwise. In fact, this metric corresponds to the Hamming distance between certain binary vector representations of each partition [94].

Comparing clusterings by set matching

A second class of criteria is based on set cardinality alone and does not make any assumption about how the clusterings may have been generated. A symmetric cri-

terion that is also a metric was introduced by van Dongen [133]

$$\mathcal{D}(\mathcal{C},\mathcal{C}') = 2n - \sum_{k} \max_{k'} n_{kk'} - \sum_{k'} \max_{k} n_{kk'}. \tag{2.16}$$

Hence, $\mathcal{D}$ is 0 for identical clusterings and strictly smaller than $2n$ otherwise.

Information-theoretic clustering comparison

The last class of criteria is based on mutual information, a well-known concept in information theory. The mutual information between two clusterings measures how much information one clustering gives about the other. For more details about the information theoretical concepts, the reader is referred to [26].

Let the probability that a point being in cluster C_k equals $P(k) = \frac{n_k}{n}$. Thus, the random variables associated with the clusterings $\mathcal{C}$, $\mathcal{C}'$ denote by $P(k), k = 1, ..., K$ and $P'(k'), k' = 1, ..., K'$. Let $P(k,k')$ represent the probability that a point belongs to C_k in clustering $\mathcal{C}$ and to $C_{k'}$ in $\mathcal{C}'$, namely the joint distribution of the random variables associated with the two clusterings: $P(k,k') = \frac{|C_k \cap C_{k'}|}{n}$. The mutual information between the clustering $\mathcal{C}$ and $\mathcal{C}'$ is equal to the mutual information between the associated random variables.

$$I(\mathcal{C},\mathcal{C}') = \sum_{k=1}^{K}\sum_{k'=1}^{K'} P(k,k')\log\frac{P(k,k')}{P(k)P'(k')}. \tag{2.17}$$

The mutual information between two random variables is always non-negative and symmetric.

Strehl and Ghosh [126] proposed the normalized version of mutual information using geometric mean of $H(\mathcal{C})$ and $H(\mathcal{C}')$ as

$$NMI(\mathcal{C},\mathcal{C}') = \frac{I}{\sqrt{H(\mathcal{C})H(\mathcal{C}')}} \tag{2.18}$$

where $H(\mathcal{C})$ and $H(\mathcal{C}')$ denote the entropy associated with clustering $\mathcal{C}$ and $\mathcal{C}'$

$$H(\mathcal{C}) = -\sum_{k=1}^{K} P(k)\log P(k),\ H(\mathcal{C}') = -\sum_{k'=1}^{K'} P'(k')\log P'(k'). \tag{2.19}$$

Entropy is always non-negative. It takes a value of 0 only when there is no uncertainty, namely when there is only one cluster. Thus, in this case NMI is not defined. The value of NMI ranges in a range [0,1] and is 1 for identical clusterings.

Another normalized version of mutual information between two partitions was proposed by Fred and Jain [44]. They used arithmetic mean of $H(\mathcal{C})$ and $H(\mathcal{C}')$ in the normalizing term:

$$NMI_{\text{arith}}(\mathcal{C}, \mathcal{C}') = \frac{2 \cdot I}{H(\mathcal{C}) + H(\mathcal{C}')}. \tag{2.20}$$

Meila [94] suggests a further alternative called variation of information (VI) defined as:

$$VI(\mathcal{C}, \mathcal{C}') = H(\mathcal{C}) + H(\mathcal{C}') - 2I(\mathcal{C}, \mathcal{C}'). \tag{2.21}$$

Meila proved that the VI is a metric and bounded by $\log n$, however, if $\mathcal{C}$ and $\mathcal{C}'$ have at most K^* clusters, it is bounded by $2 \log K^*$. Thus, the VI metric takes a value of 0 when two clusterings are identical and positive otherwise.

2.4 Segmentation Evaluation

The various methods for performance evaluation, in general, can be categorized according to their taxonomy [73] as summarized in Figure 2.2. A theoretical evaluation is done by applying a mathematical analysis without the algorithms ever being implemented and applied to an image. The major limitations of theoretical approaches are the simplistic mathematical models and the difficulty in applying them to many of the more modern segmentation algorithms because of their complexity. An experimental (empirical) evaluation can be divided into feature-based and task-based. Within the former category, we can further distinguish between non-GT(ground truth)-based (also called unsupervised) and GT-based (also called supervised) approaches. The basic idea of GT-based approaches is to measure the difference between the machine segmentation result and the ground truth[1]. In contrast, non-GT-based methods compute performance measures directly by means of some desirable properties of the segmentation result. Task-based evaluation follows a very different philosophy. In this kind of methods, image segmentation is treated as part of a proposed solution to a larger vision system, for example, object recognition, and is indirectly evaluated based on the overall performance of the entire system. However, this strategy can quickly become unfair and, more seriously, inconsistent when evaluating algorithms that are tailored to different applications [132].

In this work we focus on the supervised evaluation method which is considered as a principled and powerful way to objectively assessing the performance of

[1] Ground truth is an expected ideal segmentation, which is in almost all cases specified manually.

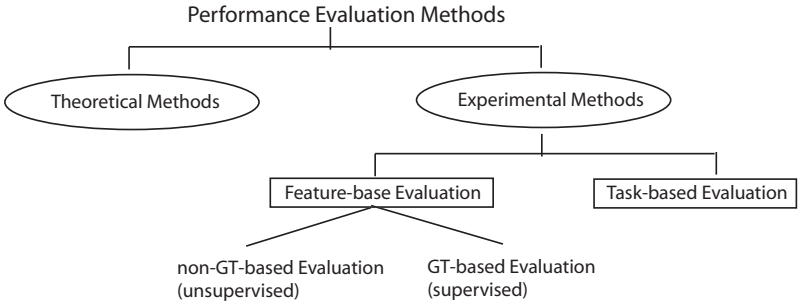


Figure 2.2. Performance evaluation method taxonomy.

segmentation algorithms [147]. Furthermore, they are relatively general which are applicable to comparing different kinds of segmentation algorithms. The purpose of supervised approaches is to measure the discrepancy between the machine segmentation obtained by an algorithm and the ground truth. A large discrepancy involves a large segmentation error and thus this indicates a low performance of the considered segmentation algorithm.

In the following, we overview the human segmentation data set and evaluation measures that will be used to quantitatively evaluate the quality of segmentation results in our experiments throughout the thesis.

2.4.1 The Berkeley Segmentation Dataset

The current public version of the Berkeley Segmentation Dataset (BSDS) [90] is composed of 300 natural images of size 481×321 pixels. The data set is divided into two sets: a training set containing 200 images that can be used to tune the parameters of a segmentation algorithm, and a testing set containing the remaining 100 images on which the final performance evaluations should be carried out. For each image a set of 4 to 9 human segmentations is provided.

Martin et al. [90] show that the human segmentations, though varying in detail, are consistent with one another in that regions segmented by one subject at a finer level of detail can be merged consistently to yield the regions extracted by a different subject at a coarser level of detail. They show regularities that can be exploited to design and evaluate segmentation algorithms. Figure 2.3 shows some example images from the data set and their five human segmentations segmented by different subjects.

Since each image contains more than one human segmentations, one segmenta-

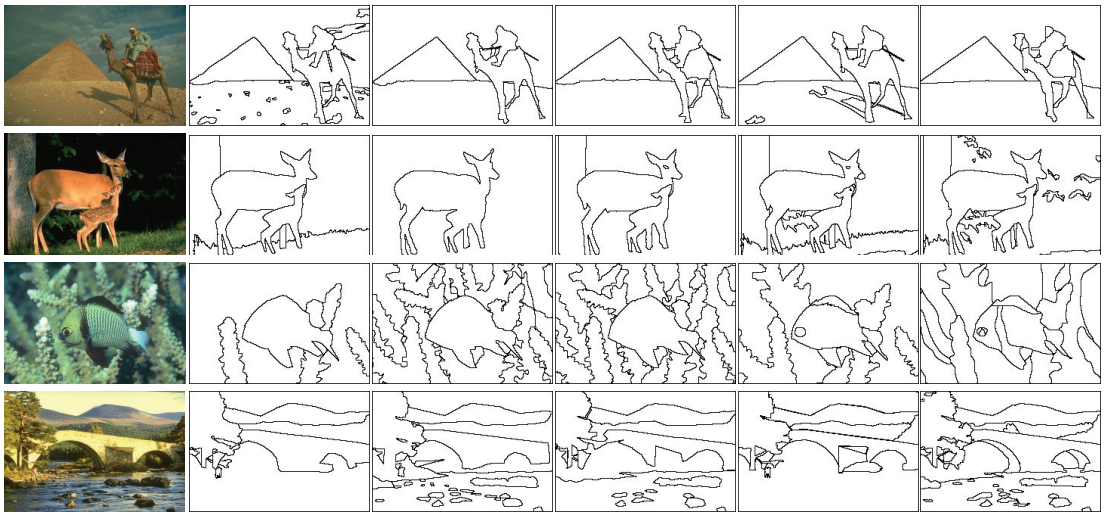


Figure 2.3. Sample of four images from the segmentation data set and their segmentations segmented by five different people. The examples illustrate that even though the human segmentations of the same image are not identical, they are differ only in the levels of granularity.

tion result is compared to all manual segmentations and the average performance value is reported. Additional details on the data set construction can be found in [90]. The data set can be obtained from [91].

2.4.2 Evaluation Measures

In this section we present the details of measures for assessing the quality of a machine segmentation against human ground truths. Both region-based and boundary-based are used in our framework.

- *Normalized Mutual Information index* (NMI): Mutual information is a well-known concept in information theory that measures the statistical information shared between two random variables. It has been used for assessing the consistency between clusterings in many works such as [6, 39, 44, 87, 126, 151]. In this thesis the normalization version of mutual information defined in (2.18) is used to assess the quality of a machine segmentation in the sense that a good machine segmentation should share the most information with a corresponding human ground truth. Since the BSDS data set provides multiple human segmentations for each image and a good machine segmentation of a particular image should be able to explain all of them, in all experiments

reported in this thesis one machine segmentation result is compared to all human segmentations and the average NMI (ANMI) value is used. ANMI value between a machine segmentation, $\hat{S}$, and a set of human segmentations, S_q.

$$\phi^{(\mathrm{ANMI})}(\hat{S}, \Lambda) \;=\; \frac{1}{N}\sum_{q=1}^{N} \phi^{(\mathrm{NMI})}(\hat{S}, S_q), \tag{2.22}$$

where N is the number of human segmentations. The higher the ANMI value, the better is the machine segmentation quality.

Unlike Rand index and other criteria such as conditional entropy [5] (that are biased toward large k), NMI provides a measure that is impartial with respect to k. It reaches its maximum value of one only when the two segmentations have a perfect one-to-one correspondence [126]. However, NMI index under some conditions is biased toward solutions that have the same number of clusters as there are classes [39].

- *F-measure*: Since the BSDS data set provides multiple human segmentations (binary boundary maps) for each image and simply unioning the humans boundary maps is not effective because of the localization errors present in the data set itself, Martin et al. [92] finesse this issue by corresponding the machine boundary map separately with each human map in turn. Only those machine boundary pixels that match no human boundary are counted as false positives. The hit rate is simply averaged over the different humans, so that to achieve perfect recall the machine boundary map must explain all of the human data. In order to apply F-measure (defined in (2.8)) in this work, it is needed to convert a labeled segmentation into a region boundary map. We compute a binary boundary map with 1 pixel wide boundaries, where boundary pixels are offset by 1/2 pixel towards the origin from the actual segment boundary.

  However, Martin et al. [92] note that computing the precision and recall of a single thresholded machine boundary map given a single human boundary map would not tolerate any localization error and would consequently over penalize algorithms that generate usable, though slightly mislocalized boundaries. Furthermore, for a given matching of edge elements between two images, it is possible to change the locations of the unmatched edges almost arbitrarily and retrain the same precision and recall score.

There are some situations that a boundary detection evaluation method is not appropriate for a region segmentation, e.g., a missing pixel in the boundary between two regions may not be reflected in the boundary benchmark, but can have substantial consequences for segmentation quality, namely, incorrectly merging two large regions. It can also be argued that the boundary benchmark favors contour detectors over segmentation methods, since the former are not burdened with the constraint of producing closed curves. However, F-measure has been provided with the benchmark dataset we used in our experiment. It is reasonable to report the results on this measure so that it is possible to render comparison to other segmentation algorithms and it does not ignore the principled design considerations used in the Berkeley evaluation. For this reason, both region-based (NMI) and boundary-based (F-measure) measures will be used to report the results.

Chapter 3

Segmentation Ensemble Framework

The concept of clustering ensemble combination is well known and widely accepted in the area of pattern classification [81] and prototype learning [74]. The main goal of clustering ensembles has been to improve the accuracy and robustness of a given classification or clustering. We expect the similar advantages of ensemble combination for the unsupervised image segmentation problem, namely, to combine multiple *imperfect* segmentations produced from multiple sources of segmentations into a single *improved* segmentation result. The solution achieved from combination of segmentation ensemble should go beyond what is typically achieved by a single segmentation algorithm in the following respects:

- *Novelty*: A combined solution should be unattainable by any single segmentation algorithm.
- *Accuracy*: The quality of combination solution should be superior to the initial segmentations or at least better than their average. Segmentation accuracy can be objectively assessed by the use of ground truth (manual segmentation).
- *Stability*: A combined solution should be stable to changes of segmentation algorithm parameters, especially, in a reasonable parameter subspace (i.e. a lower and upper bound for each algorithm parameter is assumed to be known.). Stability can be assessed from ensemble distribution.
- *Robustness*: A combined solution should be robust to small variations in an input image, for example, due to noise or transformations.

A segmentation algorithm that can yield these features will be a very useful and predictable preprocessing step in a larger high-level computer vision system (e.g. object recognition, image understanding, etc.).

3.1 Segmentation Ensemble Framework

Segmentation ensemble is a framework for building a robust segmentation from combining different segmentation results given by individual segmentation algorithms. Our segmentation ensemble combination framework is built in two steps:

1. *Segmentation Ensemble Generation Step*: This step is to generate initial different multiple segmentations of the same image for combination procedure. Two important aspects in building segmentation ensembles are *diversity* and *accuracy* of the ensemble. Fern and Brodley [39] showed that both the diversity and quality of a cluster ensemble significantly impact what can be achieved by combining the clusterings of the ensemble.

    - *Diversity of ensemble*: Diversity of the initial segmentations is one of the crucial factors to the success of segmentation ensemble combination, especially for improving segmentation quality [126]. Different segmenters[1] may produce significantly different segmentations of the same image that capture various distinct aspects of the data. Thus there could be a potential for greater gains when combining the strengths of many individual segmenters. On the other hand, different segmenters make different mistakes. The combination of them will compensate for their weaknesses. It is intuitive that a combination of relatively identical segmentation solutions would not achieve improved segmentation that outperforms the individual ensemble members. Many generative procedures have been proposed in order to achieve diversity in an ensemble, which will be described later in this section.
    - *Strength of ensemble components*: This raises questions of how to design the individual segmenters so that they form potentially an accurate ensemble, and how weak could each input component is to ensure a successful combination. From the supervised case, one can expect that using

[1] Segmenters may be versions of the same segmentation algorithm, or different segmentation algorithms, or other methods that yield different segmentation results of the same image.

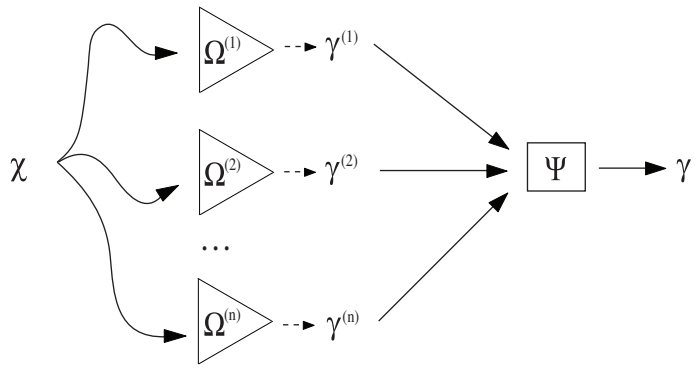


Figure 3.1. Segmentation ensemble combination architecture. Given multiple segmentations $\gamma^{(i)}$ of an image $\mathcal{X}$ produced by variety of sources $\Omega^{(i)}$. A goal is to compute a final segmentation result γ which is superior to the initial segmentations.

many simple, but computationally inexpensive components will be preferred to combining segmentations obtained by sophisticated, but computationally involved algorithms [128].

It is expected that the accuracy of the ensemble improves when a larger number of input segmentations is given, provided that the segmentations are diverse. However, studying diversity in segmentation ensembles, as well as clustering ensembles, is relatively new area of unsupervised ensembles. The impact of diversity and quality of the individual segmentation/clustering solutions on the final ensemble performance has not been fully understood. The preferred level of diversity (high, medium, or low) is under investigation by some researchers [39, 55, 129]. Topchy et al. [129] shows that a consensus solution is shown to converge to a true underlying clustering solution as the diversity in the ensemble increases, while Hadjitodorov et al. [55] shows that in some cases ensembles which exhibited a moderate level of diversity gave a more accurate clustering. However, none of the literature on image segmentation combination proposed thus far concerns this issue. In Chapter 6 we will study the interplay between accuracy and diversity of our segmentation ensemble and their influence on segmentation combination performance.

Note that segmentation ensemble generation can be implemented and executed in parallel to improve processing speed.

2. *Segmentation Ensemble Combination Step*: In order to find the final combined segmentation, we need a combination algorithm (for which some literature on pattern recognition and machine learning refer to as a consensus function)

for utilizing information provided by multiple initial segmentations (which are sometimes referred to as base partitions/clusterings). This step questions how to best combine multiple input segmentations of an image to achieve a final segmentation result which is superior to the initial segmentations. Similar to traditional clustering combination problem, there are two difficult tasks which are specific to the design of segmentation combination algorithm:

- *Label correspondence*: Due to unavailability of training data, there is no explicit correspondence between the labels delivered by different partitions. Different clusterings may produce incompatible data labeling, resulting in intractable correspondence problems, especially when the numbers of clusters are different. For example, two identical partitions might have permuted labels and be perceived as different. This problem must be solved to obtain the same labeling of clusters throughout the ensembles partitions. Some example approaches to solve the label correspondence problem are following.

  A *direct re-labeling approach* seeks correspondence between the cluster labels across the partitions and fuses the clusters of the same label. As an outcome of the re-labeling procedure, we can straightforwardly apply a voting scheme or standard clustering (combination) algorithms to obtain the final combined results. Topchy et al. [130] use the Hungarian algorithm for minimal weight bipartite matching problem in order to re-label the partitions and a final consensus clustering was obtained by standard clustering combination algorithms. Boulis and Ostendorf [9] use Linear Programming to discover a correspondence between the labels of the individual clusterings and those of an optimal meta-clustering.

  A *hypergraph approach* transforms multiple partitions into a hypergraph representation and uses methods for hypergraph partitioning to obtain the ensemble result [79, 126].

  A *feature-based approach* interprets a set of multiple partitions as a new set of categorical features which are further standardized and transformed to quantitative features regarding as *intermediate feature space*. Then, the solution of combination can be approached by traditional clustering algorithm (i.e., k-means) [128].

  A *co-association approach* sidesteps the label correspondence problem by mapping the clustering ensemble to a co-association matrix, where entries can be interpreted as vote ratios on the pairwise cooccurrences between all pairs of objects. A final consensus clustering can be extracted by ap-

plying linkage-based clustering algorithms [44] or clustering combination algorithm [140] on this matrix.

- *Number of clusters*: For most clustering problems there is little prior information (e.g., statistical models) available about the data. Thus, the desired number of clusters is not known in advance and is often specified by a human user. In case of clustering ensemble, we can obtain some information on how these objects (or pixels) should be clustered. Benefited from this information, the combination algorithm will be able to settle naturally the appropriate number of clusters underlying a clustering ensemble. In fact, the right number of clusters in a dataset often depends on the scale at which the data is inspected, and sometimes equally valid (but substantially different) answers can be obtained for the same data [126].

In order to optimally combine segmentation ensemble in a fully automatic and effective manner, we need to address this issue by formulating combination algorithm that avoid an explicit solution to the correspondence problem and include a mechanism to automatically determine the final number of regions in combined segmentation result. One possibility is to compute an average, or more formally *generalized median* [74] for a set of multiple segmentations.

Our segmentation combination framework can be summarized as shown in Figure 3.1.

The key feature of our framework is its *generality*. It is very important that the proposed framework is not restricted to specific features or segmentation methods. In our framework the combination procedure is designed to be independent from the generative procedure. This allows the users to freely select different choices of any image segmentation algorithm, even in very different method classes, without the change in combination step. Moreover, this enables the combination procedure to lend itself to a wide range of segmentation tasks, for example, regions in color or texture images, surface patches in range images, etc.

In Chapter 4 we apply this framework for the tasks of multiple contour combination. In this task a special class of contours is considered, which start from the top, pass each image row exactly once, and end in the last row of an image. Multiple contours can be obtained by using different parameter values of the same contour detection algorithm. Then, they will be combined by means of generalized median.

In Chapter 5 the framework is used to deal with the problem of multiple region-based image segmentation combination. Fusion of multiple segmentations is achieved by means of co-association approach. Multiple segmentations are then combined

using our proposed combination algorithm which is based on a random walker algorithm. In this task the generalized median concept is used as a tool for selecting the best combination segmentation results (from a set of segmentation results with different number of regions), which is an implicit way to automatically determine the final number of regions.

There are several generative procedures for generating multiple region-based segmentations of the same image: 1) perturbing the data, such as sampling techniques [79] and bagging [41], 2) employing different image features [61], 3) computing a segmentation algorithm on different random image sites [116], 4) merging superpixels by varying the number of segments and initializations of a multiple segmentation algorithm [62], 5) using different segmentation algorithms [3, 141] or 6) using the same segmentation algorithm but different parameter values [88, 117, 139]. The experiments reported in Chapter 6 demonstrate a variety of generative procedures, where a segmentation ensemble is generated by using different parameter values of the same segmentation algorithm, using different segmentation algorithms, and using the same segmentation algorithm with fixed parameter values on different transformations of an input image.

3.2 Means for Combining Segmentation Ensemble

In this section we present a brief overview of a variety of combination methods. Firstly, we propose the concept of generalized median as a tool for combining multiple segmentations. Then, some powerful combination methods proposed in machine learning and pattern recognition literatures are reviewed. The advantages and limitations of these methods for applying in segmentation combination problem are also discussed.

3.2.1 Combination by Median Concept

The concept of generalized median strings can be applied to compute average contours if contours are represented by strings [76]. This is useful for object prototype learning. In Chapter 4 a special class of contours is considered, which start from the top, pass each image row exactly once, and end in the last row of an image. Despite of their simplicity they frequently occur in many applications of image analysis. A dynamic programming algorithm with $O(Nmn)$ time and $O(mn)$ space is designed,

where N images of size $m \times n$ are assumed. Recently, this algorithm has been extended to one of expectation-maximization type for handling the case, where the input contours are subject to a varying (unknown) horizontal displacement [17].

Chapter 5 mentions the median segmentation optimization function which is used to select the best segmentation from a set of combination segmentations with different number k of regions. In some sense this approach can be regarded as an approximation of generalized median segmentation by investigating the subspace of U (all possible segmentations of an image), which consists of the combination segmentations for the considered range of k.

3.2.2 Clustering Ensemble Techniques

There are a number of existing techniques that manipulate the clustering ensemble. We may consider an image segmentation as a clustering of pixels and apply some clustering combination algorithm for the segmentation combination purpose. However, standard clustering combination algorithms each have significant theoretical and practical limitations that make them unsuitable for the purpose of segmentation combination.

A well-known clustering combination strategy is graph-based partitioning approach introduced by Strehl and Ghosh [126]. They proposed three efficient heuristic consensus algorithms: 1) the Cluster based Similarity Partitioning Algorithm (CSPA) which induces a graph from a coassociation matrix and clusters it using the METIS algorithm. 2) the Hypergraph Partitioning Algorithm (HGPA) which represents each cluster by a hyperedge in a graph where the nodes correspond to a given set of objects. Good hypergraph partitions are found using minimal cut algorithms such as HMETIS coupled with the proper objective functions, which also control partition size. 3) Hyperedge collapsing operations are considered in another hypergraph based Meta Clustering Algorithm (MCLA). These graph-based partitioning methods have been used for combining multiple image segmentations by many researchers [15, 79, 87]. Although these graph-based methods are successful in several cluster ensemble applications, it lacks the ability of clustering the data with highly unbalanced clusters [126], which sometimes encountered in image data.

Another graph-based method is Hybrid Bipartite Graph Formulation (HBGF) proposed by Fern and Brodley [40]. It constructs a bipartite graph from a set of partitions to be combined, modeling objects and clusters simultaneously as vertices, and later partitioning the graph by a traditional graph partitioning technique. The implementation of this method is quite complicated.

Another well-known clustering combination strategy is coassociation-based algorithms proposed by Fred and Jain [44]. It is based on the idea of evidence accumulation by considering each partition as an independent evidence of data organization. Individual data partitions are combined based on a voting mechanism to generate a new $n \times n$ similarity matrix for n patterns. The final data partition of the n patterns is obtained by applying a hierarchical agglomerative clustering algorithm on this matrix. The method has shown its power in combining clusters in real datasets. Unfortunately, its computational and storage complexity scale quadratically with the number of pixels. Increasing the image size would lead to computationally infeasible situation.

Another voting-based algorithm was proposed by Fischer and Buhmann [41] called *path-based clustering*. They introduced a cost function with an explicit bias for chained structures, where an agglomerative algorithm is used for optimization. Agglomerative optimization has a low running time, however, it is more sensitive to small fluctuations in the data. Thus, a bootstrap resampling method is proposed to compensate this effect such that a data clustering method can extract structures from data in a noise robust way. The quality of path-based clustering with resampling is evaluated through an image segmentation application. However, this voting consensus algorithm assumes that each partition of the ensemble has the same number of clusters, which is equal to the target number of clusters in the consensus clustering, resulting in a limitation of the applications of this algorithm on diverse cluster ensembles (i.e. clustering with randomly selected number of clusters).

In general, it is not suitable to mechanically apply the combination algorithms from general clustering domain to segmentation domain. General clustering methods are global and do not retain positional information. The major drawback of this is that it is invariant to spatial rearrangement of the pixels, which is an important aspect of what is meant by segmentation. Resulting segments can be widely scattered, resulting in the need of post-processing step. More detailed reviews of the clustering combination algorithms for clustering general data can be found in the introductory sections of several papers in this area [6, 40, 41, 44, 53].

3.3 Applications of Segmentation Ensemble Combination

Segmentation ensemble combination provides a general framework for dealing with a variety of segmentation problems in various settings. In this section we present

some of the main applications of segmentation ensemble combinations to alleviate some hard problems in image segmentation.

Exploring Parameter Space without Ground Truth: The most disadvantage of image segmentation algorithms is their sensitivity to parameter settings and their optimal setting is not a trivial task. In Chapter 7 we propose to apply the multiple segmentation combination for dealing with the difficult problem of parameter selection without ground truth segmentation. It is assumed that we know a reasonable subspace of the parameter space (i.e. a lower and upper bound for each parameter), which is sampled into a finite number $\mathcal{N}$ of parameter settings. Then, we run the segmentation procedure for all the $\mathcal{N}$ parameter settings and compute a final combined segmentation of the $\mathcal{N}$ segmentations. The rationale behind our approach is that this segmentation tends to be a good one within the explored parameter subspace, given the fact that we do not know the optimal parameter setting for a particular image in advance.

Multiple Segmenter Combination: Different segmentation algorithms have different performance and different shortcomings. Some algorithms might perform well in specific images but not in others. Furthermore, it is not easy to know the optimal algorithm for one particular image. We postulate: Instead of looking for the best segmenter which is hardly possible on a per-image basis, now we look for the best segmenter combiner. The rationale behind this idea is that while none of the segmentation algorithms is likely to segment an image correctly, we may benefit from combining the strengths of multiple segmenters. This idea has been utilized to enhance the quality of segmentation results in many works [3, 77, 141]. Similarly, we may compute a single representative from multiple manually specified ground truth segmentations [138]. The application of combining multiple segmenters is also illustrated in Chapter 6

Instability of Segmentation Algorithms: The region growing paradigm is one of the most widely used techniques for image segmentation. It is shown that within a small parameter range, which leads to good segmentation results in the majority of cases, remarkably bad segmentation results may occur. Franek and Jiang [43] have empirically analyzed the frequency of such instabilities on natural images of BSDS data set [90] and proposed to solve this stability problem by computing the *set median* of a set of segmentations within a specific parameter subspace of interest. In the majority of cases the computation of set median avoids outliers and achieves robustness. In Chapter 8 we propose the use of *generalized median* as an alternative way to solve this problem. The generalized median of a set of segmentations is computed by applying our segmentation combination algorithm.

Multiple Feature Set Integration: A single segmentation strategy with a single feature set often does not comprehensively capture the large degree of variability and complexity encountered in many application domains. Combination approach can overcome this problem by acquiring multiple-source information through multiple features extracted from multiple processes. Hayman and Eklundh [61] propose two techniques for fusing the output of multiple cues (i.e., motion, colour, contrast and prediction) to robustly and accurately segment foreground objects from the background on video sequences. The first method is based on Baysian approach where the likelihood of observations over all cues at each pixel is computed before assigning a membership to a pixel. The second method allows each cue to make a decision independent of each other before fusing their outputs using weighted voting scheme.

Part I

Contour Detection

Chapter 4

Multiple Contour Combination

The ability to find the average of a set of contours has several applications in computer vision including prototype formation and computational atlases. While contour averaging can be handled in an informal manner, the formal formulation within the framework of generalized median as an optimization problem is attractive. In this chapter we will follow this line. A special class of contours is considered, which start from the top, pass each image row exactly once, and end in the last row of an image. Despite of the simplicity they frequently occur in many applications of image analysis. We propose a dynamic programming approach to exactly compute the generalized median contour in this domain. Experimental results will be reported on two scenarios to demonstrate the usefulness of the concept of generalized median contours. In the first case we postulate a general approach to implicitly explore the parameter space of a (segmentation) algorithm. It is shown that using the generalized median contour, we are able to achieve contour detection results comparable to those from explicitly training the parameters based on known ground truth. As another application we apply the exact median contour to verify the tightness of a lower bound for generalized median problems in metric space.

4.1 Problem Definition

While contour averaging can be handled in an informal manner as done in [14, 119], the formal formulation within the framework of generalized median as an optimization problem is attractive. This concept has been successfully applied to strings [69, 85] and graphs [74] in structured pattern recognition. In this work a special class of contours is considered, which start from the top, pass each image

row exactly once, and end in the last row of an image. If a contour is coded by a string, then the same procedure can be adapted to averaging contours [69]. However, this general approach suffers from high computational complexity. It is proved in [27] that computing the generalized median string is NP-hard. Sim and Park [122] proved that the problem is NP-hard for finite alphabet and for a metric distance matrix. Another result comes from computational biology. The optimal evolutionary tree problem there turns out to be equivalent to the problem of computing generalized median strings if the tree structure is a star (a tree with $n+1$ nodes, n of them being leaves). In [137] it is proved that in this particular case the optimal evolutionary tree problem is NP-hard. The distance function used is problem dependent and does not even satisfy the triangle inequality. All these theoretical results indicate the inherent difficulty in finding generalized median strings, or equivalently the generalized median contours. Not surprisingly, researchers make use of domain-specific knowledge to reduce the complexity [85] or resort to approximate approaches [69].

4.2 Related Work

Chalana and Kim [14] used the average of the multiple observers' curves to establish a gold-standard contour for evaluating boundary detection algorithms on medical images. Their contour averaging procedure is based on establishing one-to-one correspondence between the points constituting two or more curves. A point on the average curve is given by the centroid of these corresponding points along the curve. Then, for each point on the average curve, a normal to the curve at that point is drawn and the intersection of this normal with each of the input curves is determined. These points of intersection define another set of correspondence between the input curves. This new correspondence is averaged again to give a new average curve. The process is iterated until the average curve does not change any more. However, the average distance between two curves, computed this way, is not a metric (it does not satisfy the triangle inequality).

Another approach of contour averaging was proposed by Sebastian and Kimia [119]. An average of a set of curves is computed by averaging the intrinsic properties (namely, length and curvature) of the corresponding curve subsegment. The optimal correspondence is found by an efficient dynamic-programming method for aligning pairs of curve segments.

In this work we consider a special class of contours for which the generalized

median can be found by an efficient algorithm based on dynamic programming. We first motivate our work by giving some background information about this class of contours in Section 4.3. Then, the algorithm for finding the exact solution is described in Section 4.4. In Section 4.5 and 4.6 we describe two applications of generalized median computation: exploring the parameter space of a contour detection algorithm and tightness evaluation of a lower bound of generalized median problems in metric space. Finally, some discussions conclude the chapter.

4.3 Class of Contours

The class of contours considered in this work is defined as follows:

Definition 4.1 *For a given $M \times N$ image a contour $C = p_1 p_2, \ldots, p_M$ is a sequence of points drawn from the top to the bottom,where p_i, $i = 1, \ldots, M$, is a point in the i-th row. The points p_i and p_{i+1}, $i = 1, \ldots, M-1$, of two successive rows are continuous.*

These contours start from the top, pass each image row exactly once, and end in the last row.

At the first glance the question may arise why such simple contours are of use in practice. Some thoughts, however, reveal that there do exist several situations, where we are directly or indirectly faced with this class of contours. In medical imaging it is typical for the user to specify some region of interest (ROI) and then to find some contours within the ROI. As an example, Figure 4.1 shows a ROI in a CCA (Common Carotid Artery) B-mode sonographic image. The task is to detect the layer of intima and adventitia for computing the intima-media thickness which is an important index in modern medicine. Details of this application and an algorithm for automatic layer detection can be found in [17]. Essential to the current work is the fact that both the intimal layer and the adventitial layer are examples of the contour class defined above (although we have to rotate the image by 90 degrees). This application reflects a typical situation in medical image analysis. The same fundamental principle can be extended to deal with closed contours. For this purpose we need a point p in the interior of the contour. Then, a polar transformation with p being the central point brings the original image into a matrix, in which a closed contour becomes a contour from top to bottom afterwards. Note that this technique works well for all star-shaped contours including convex contours as a special case. As an example, Figure 4.2 shows a problem of eye contour detection

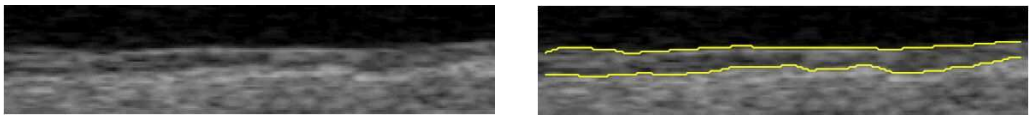


Figure 4.1. ROI in a CCA B-mode sonographic image (left) and detected layer of intima and adventitia (right).

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

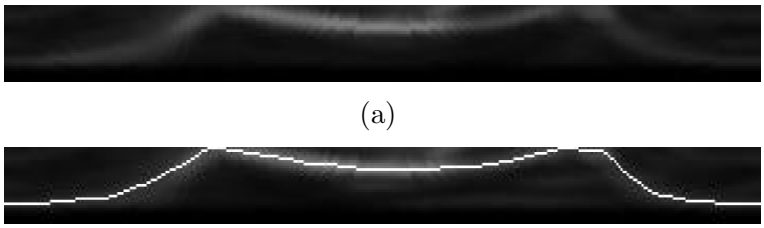
Figure 4.2. Detection of closed contour: (a) input image; (b) removal of iris; (c) detection of eye contour; (d) strabismus simulation.

taken from [75]. In the image after removal of iris, the eye contour is detected as a closed contour based on the interior reflection point. The polar space representation related to Figure 4.2(b) can be seen in Figure 4.3(a) where the intensity is replaced by a measure of edge magnitude. In this space we are faced with the same contour detection problem as in Figure 4.1. The result is shown in Figure 4.3(b) and Figure 4.2(c) after projecting back into the image space. The task in this application is then to simulate strabismus by replacing the iris. The eye contour serves to restrict the region, within which the newly positioned iris lies. For (almost) convex contours the selection of the origin of polar space is not critical. In the general case of star-shaped contours, however, it must be chosen within the area, in which the complete contour can be seen.

The two situations above and others appear in a variety of applications. They indicate the broad applicability of the class of contours considered in this paper and thus justify to investigate them in their own right.

The concept of generalized median in (2.1) can be easily adapted to our domain by specifying a distance function between two contours. Since each point p_i of a contour $P = p_1 p_2, \ldots, p_M$ has a constant y-coordinate i, we use p_i to represent its x-coordinate only in the following in order to simplify the notation. Given this convention, the distance between two contours P and Q can be defined by the k-th power of the Minkowski distance:

$$d(P, Q) \;=\; \sum_{i=1}^{M} |p_i - q_i|^k \tag{4.1}$$



(a)

(b)

Figure 4.3. Polar space for contour detection: (a) polar space; (b) optimal path.

In this case the representation space U contains all *continuous* contours from top to bottom of an input $M \times N$ image.

4.4 Algorithm: Computation of Generalized Median Contours

Given n contours $C_1, C_2, \ldots, C_n$, the task is to determine a contour $\overline{C}$ such that the sum of distances between $\overline{C}$ and all input contours is minimized. It is important to notice that we cannot solve this problem of generalized median contours by computing the optimal value for each of the M rows *independently*, which could be done, for instance, by enumerating all possibilities between the leftmost and rightmost point in the row. Doing it this way, we encounter the trouble of generating a discontinuous resultant contour.

Our proposed method is formulated as a problem of finding an optimal path in a graph based on dynamic programming. We first generate a two-dimensional $M \times N$ cost matrix of the same size as the image, in which every element corresponds to an image point. Each element is assigned a *Local_Goodness* value, which measures its suitability of being a candidate point on the generalized median contour we are looking for. According to the distance given in (4.1) the *Local_Goodness* value is simply:

$$Local\_Goodness(i,j) \;=\; \sum_{l=1}^{n} |x_{li} - j|^k\,, \quad 1 \le i \le M,\; 1 \le j \le N$$

where x_{li} represents the x-coordinate of the l-th contour C_l in i-th row. Generally, small *Local_Goodness* values indicate better candidates. As a matter of fact, the optimality of a candidate for $\overline{C}$ is measured by the sum of its *Local_Goodness* values over all image rows.

Dynamic programming is applied to search for an optimal path in a cumulative cost matrix CC. The cumulative cost of a node (i, j) is computed as:

$$CC(i,j) \;=\; \min_{l=-1,0,1}\{CC(i-1,j+l)\} + Local\_Goodness(i,j) \tag{4.2}$$

for $2 \leq i \leq M,\ 1 \leq j \leq N$. This means that a contour point (i, j) has three potential predecessors $(i-1, j-1),\ (i-1, j),\ (i-1, j+1)$ in the previous row. In addition, the choice of a transition from a point in i-th row to a predecessor in the $(i-1)$-th row is made based on the lowest cumulative cost of the predecessors. The computation of CC starts by initializing the first row by:

$$CC(1,j) \;=\; Local\_Goodness(1,j), \quad 1 \leq j \leq N$$

Then, the cumulative cost matrix CC is filled row by row from left to right by using (4.2).

The node in the last row of matrix CC with the lowest value gives us the last point of the optimum path. To determine this path, a matrix of pointers is created at the time of computing the matrix CC. The optimum path, which corresponds to the generalized median contour, is determined by starting at the last point and following the pointers back to the first row. Using this dynamic programming technique, we are able to compute the generalized median contour exactly. An overview of the proposed algorithm is shown in Figure 4.4.

The computational complexity of the algorithm amounts to $O(MNn)$ while $O(MN)$ space is required. Note that the search space of dynamic programming can be substantially reduced. For each row we only need to consider the range bounded by the leftmost and rightmost point from all input contours in that row. The size of this reduced search space depends on the variation of input data. The less variation of the input data, the more the reduction effect. Most likely, this reduction results in a computational complexity of $O(Mn)$ only. The proposed algorithm was implemented in Matlab on a Pentium IV 2.1 GHz PC. As an example, the computation time for 250 input contours of 105 points each with 0.00 standard deviation in the input data is 10 milliseconds. At an increased level of data variation of 81.74 standard deviation, 90 milliseconds were recorded. We can conclude that the dynamic programming approach delivers an efficient way of exactly computing the generalized median of contours.

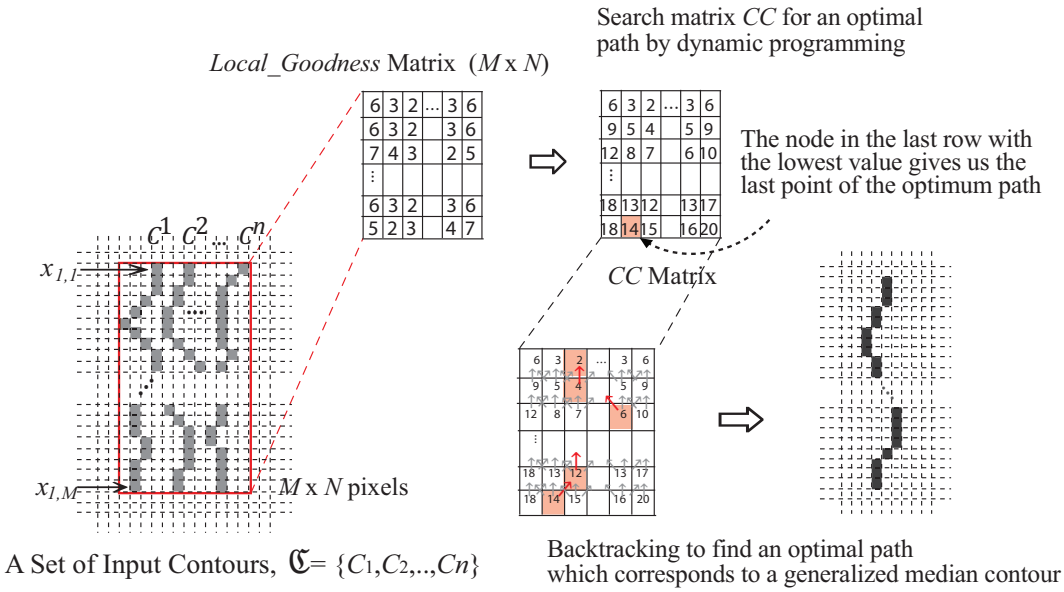


Figure 4.4. Overview of the proposed algorithm for computing the generalized median contours.

4.5 Application I: Parameter Selection Problem

4.5.1 Test images and contour data

In Section 4.5 and 4.6 we report some results to illustrate two applications of the concept of generalized median contours. The contour data used in both applications are based on CCA B-mode sonographic images [17]. An image dataset was established which consists of 23 such images of 105 columns each. They are actually ROI cut out of larger images. Each image contains two contours of interest: intima (y_1) and adventitia (y_2). Both contours run from left to right of an image. If we turn the images by 90 degrees, then we are faced with the problem of optimally masking the two contours of length 105 each from top to bottom.

Each image has its ground truth contours manually specified by an experienced physician. This information is used for an objective, quantitative comparison with automatic detection results. The similarity measure is simply the distance function in (4.1). In all our tests we have fixed k of the distance function to $k = 1$.

4.5.2 Exploring parameter space without ground truth

Segmentation algorithms mostly have some parameters and their optimal setting is not a trivial task. In recent years automatic parameter training has become popular. Typically, a training image set with (manual) ground truth segmentation is assumed to be available. Then, a subspace of the parameter space is explored to find out the best parameter setting. For each parameter setting candidate a performance measure is computed in the following way:

- Segment each image of the training set based on the parameter setting;
- Compute a performance measure by comparing the segmentation result and the corresponding ground truth;
- Compute the average performance measure over all images of the training set.

The optimal parameter setting is given by the one with the largest average performance measure. Since fully exploring the subspace can be very costly, space subsampling [97] or genetic search [22] has been proposed.

While this approach is reasonable and has been successfully practiced in several applications, its fundamental disadvantage is the assumption of ground truth segmentation. The manual generation of ground truth is always painful and thus a main barrier of wide use in many situations.

We propose to apply the concept of generalized median for implicitly exploring the parameter space without the need of ground truth segmentation. It is assumed that we know a reasonable subspace of the parameter space (i.e. a lower and upper bound for each parameter), which is sampled into a finite number $\mathcal{M}$ of parameter settings. Then, we run the segmentation procedure for all the $\mathcal{M}$ parameter settings and compute the generalized median of the $\mathcal{M}$ segmentation results. The rationale behind our approach is that the median segmentation tends to be a good one within the explored parameter subspace.

This idea has been verified on the database described above within the contour detection algorithm [17]. It has two parameters and a reasonable parameter subspace is divided into 250 samples. The database is partitioned into a training set of 10 images and a test set of 13 images. The training set is then used to find the optimal parameter setting among the 250 candidates, which is applied to the test set. The average performance measure over the 13 test images is listed in Table 4.1. Note that the testing procedure is repeated 5 times for different partitions of the

Table 4.1. Performance measures of parameter training and generalized median (GM) approaches on 5 test sets.

	y_1 (intima)		y_2 (adventitia)	
Test set	Parameter training	GM	Parameter training	GM
1	48.98	49.77	60.59	50.18
2	48.68	49.37	53.56	52.82
3	51.09	51.16	51.79	51.26
4	49.90	50.66	46.83	47.08
5	46.53	46.53	50.03	48.07
average	49.04	49.50	52.56	49.88

23 images into training and test set. On the other hand, the generalized median approach has no knowledge of the ground truth segmentation. It simply detects 250 contours and computes their generalized median. The average performance measure of the 13 generalized median contours in the test set as shown in Table 4.1 indicates that basically no real performance differences exist between these two approaches. Without using any ground truth information, the generalized median technique is able to produce contours of essentially identical quality as the training approach.

4.6 Application II: Verification of Optimal Lower Bound for Generalized Median Problems in Metric Space

The computation of generalized median patterns is typically an NP-complete task. Therefore, research efforts are focused on approximate approaches. One essential aspect in this context is the assessment of the quality of the computed approximate solutions. Since the true optimum is unknown, the quality assessment is not trivial in general. A recent work [72] presented the lower bound for this purpose.

Referring to the notation in (2.1), an approximate computation method gives us a solution $\tilde{C}$ such that

$$\mathrm{SOD}(\tilde{C}) \;=\; \sum_{i=1}^{n} d(\tilde{C}, C_i) \;\geq\; \sum_{i=1}^{n} d(\overline{C}, C_i) \;=\; \mathrm{SOD}(\overline{C})$$

where SOD stands for sum of distances and $\overline{C}$ represents the (unknown) true generalized median. The quality of $\tilde{C}$ can be measured by the difference $\mathrm{SOD}(\tilde{C}) - \mathrm{SOD}(\overline{C})$. Since $\overline{C}$ and thus $\mathrm{SOD}(\overline{C})$ are unknown in general, we resort to a lower bound

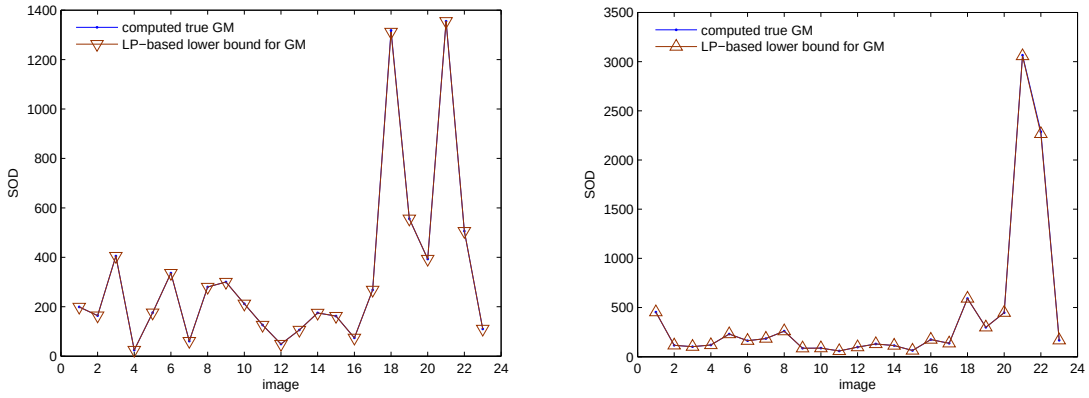


Figure 4.5. Tightness of lower bound Γ for 50 y_1 contours (intima, left) and 50 y_2 contours (adventitia, right) contours for all 23 images.

$\Gamma \leq \mathrm{SOD}(\overline{C})$ and measure the quality of $\tilde{C}$ by $\mathrm{SOD}(\tilde{C}) - \Gamma$. Note that the relationship

$$0 \;\leq\; \Gamma \;\leq\; SOD(\overline{C}) \;\leq\; SOD(\tilde{C})$$

holds. Obviously, $\Gamma = 0$ is a trivial, and also useless, lower bound. We require Γ to be as close to $\mathrm{SOD}(\overline{C})$ as possible. This tightness can be quantified by $\mathrm{SOD}(\overline{C}) - \Gamma$ with a value zero for the ideal case. In [72] the tightness of the lower bound has been tested in the domain of strings and graphs. Since the computation of generalized strings and graphs is exponential, only approximate solutions have been considered there.

Ideally, the tightness should be investigated in domains where we know the true generalized median. The current work provides us a means of validating the tightness under ideal conditions. For this purpose we sampled 50 parameter settings of the parameter subspace[1]. For each image, we thus compute 50 contours and afterwards their exact generalized median $\overline{C}$ by the dynamic programming technique proposed in this paper. In Figure 4.5 both the lower bound Γ and $\mathrm{SOD}(\overline{C})$ for all 23 images are plotted. Obviously, these two values are so similar that no difference is visible. This is clearly a sign of good tightness of the lower bound Γ. Although this statement is made for the particular case of contours, it builds a piece of the mosaic of validating the tightness in many problem spaces.

[1]The reason for selecting only 50 instead of 250 as in other experiments lies in the high computation time and space requirement of the lower bound computation which is based on linear programming.

4.7 Discussion and Conclusions

In this paper we have considered a special class of contours which start from the top, pass each image row exactly once, and end in the last row of an image. Despite of the simplicity they frequently occur in many applications of image analysis. We have proposed a dynamic programming approach to exactly compute the generalized median contour in this domain.

Experimental results have been reported on two scenarios, in which the concept of generalized median plays a very different role. In the first case we have postulated a general approach to implicitly explore the parameter space of a (segmentation) algorithm. It was shown that using the generalized median contour, we are able to achieve contour detection results comparable to those from explicitly training the parameters using a training set with known ground truth. This performance is remarkable and should be further investigated in other contexts.

Having a generalized median problem with exact solution is interesting in its own right for the specific problem domain. From a more general point of view, the exact solution gives us a means to verify the tightness of the lower bound for generalized median computation under ideal conditions. We have performed the verification which shows the high tightness. As part of our efforts in verifying the tightness of the lower bound using a variety of generalized median problems with exact solution, the current work represents a valuable contribution.

Part II

Region-Based Image Segmentation

Chapter 5

Multiple Image Segmentation Combination

Image segmentation is known to be unstable, strongly affected by small image perturbations, feature choices, or different segmentation algorithms [104]. This instability has led towards combining multiple segmentations that take advantage of the complementary nature of several segmentations. In this chapter we present an algorithm for combining multiple region-based image segmentations to achieve a final improved segmentation. In contrast to previous works we consider the most general class of segmentation combination, i.e. each input segmentation can have an arbitrary number of regions. Our algorithm is based on a random walker segmentation algorithm which is able to provide high-quality segmentation starting from manually specified seeds. We automatically generate such seeds from an input segmentation ensemble.

In the previous chapter the generalized median concept has been used for computing the average of a set of contours. In this chapter it is used as a criterion for (indirectly) determining the number of regions in a final combined segmentation result. We demonstrate the effectiveness of this generalized median based criterion by comparing it with three alternative criteria for determining the number of regions. Extensive experiments with these criteria indicate that the generalized median concept is capable of selecting the optimal combined segmentation results.

5.1 Related Work

Unsupervised image segmentation is of essential relevance for many computer vision applications and remains a difficult task despite of decades of intensive research, for example, segmentation algorithms mostly have some parameters and theirs optimal setting is a non-trivial task. Moreover, there exists no universal segmentation algorithm that can successfully segment all images. It is not easy to know the optimal algorithm for one particular image. Recently, researchers start to investigate combination of multiple segmentations of the same image in order to improve segmentation accuracy over the individual input segmentations. Several works in medical image analysis consider segmenting an image into a *known* number of semantic labels [63, 115, 138]. Typically, such algorithms are based on local (i.e. pixel-wise) decision fusion schemes such as voting. Alternatively, a shape-based averaging is proposed in [114] to combine multiple segmentations.

The works [3, 15, 41, 77, 79, 87] deal with the general segmentation problem. They consider an image segmentation as a clustering of pixels and apply a standard clustering combination algorithm for the segmentation combination purpose. The authors of [15, 79, 87] applied the graph-based clustering combination algorithms proposed by Strehl and Ghosh [126] as a consensus function. The main difference between them lies in the way they generate the input segmentations: Keuchel and Küttel [79] used probabilistic sampling method to obtain a fast segmentation of the image by approximating the solution of the convex relaxation method, Chang et al. [15] used k-means algorithm with random initial cluster centroids, and Ma et al. [87] used spectral clustering with randomly selected value of kernel parameter in an appropriate range. A more recent work of [15] included texture information as another constraint on scale-invariant feature transformation [16]. In [77] a greedy algorithm finds the matching between the regions from the input segmentations which build the basis for the combination. Fischer and Buhmann [41] used bagging (or bootstrap aggregating) with path-based clustering to address the robustness issue. They proposed a direct re-labeling approach to obtain a consensus partition from clusterings of multiple bootstrap samples. They selected a relabeling out of all $k!$ permutations for a clustering, such that it maximizes the sum over the empirical cluster assignment probabilities estimated from previous mappings, over all objects of the new mapping configuration. Although this approach has demonstrated impressive results for image segmentation, an exhaustive experiment might not be feasible for large k. Another segmentation combination method, that is based on voting scheme, is proposed by Aljahdali and Zanaty [3]. This work is different from the above works in that they combined multiple segmentations produced from

different segmentation techniques (i.e. Histogram thresholding, Region growing, k-means, Fuzzy c-means, and Kernelized fuzzy c-means), while the above works used the same clustering methods to generate an ensemble.

Another purpose of exploiting the advantages of combination approach is to combine multiple sets of image features. Recent efforts in this direction include work by Hayman and Eklundh [61] and Haindl and Mikes [56]. The work [56] exploits the advantages of combination approach by combining several unsupervised segmenters of the same type but with different feature sets. Multiple segmentation results are combined by using the sum rule. The most recent version of this work presented in [57] with the modification of the sum rule which yields a significant improvement over their previous version. Hayman and Eklundh [61] presented two different methods: the voting and probabilistic fusion schemes, for combining segmentation results computed by multiple segmentation algorithms using each individual cue (i.e. motion, color, texture, and prediction).

However, these works still assume that all input segmentations contain the *same* number of regions, as well as in the combined segmentation. Moreover, these approaches are either restricted to specific base image segmentation methods or restricted to specific image domains. Our work is not limited to these restrictions and we consider the most general case (i.e. an arbitrary number of regions per segmentation, independent of base image segmentation methods, and independent of image domain).

Recently, several interesting works have made a clever use of multiple segmentations for achieving other various objectives. These works leverage the use of multiple segmentations as pre-processing step in high-level computer vision applications to avoid the risky commitment to a single segmentation which might be of rather poor quality. The key motivation of these works is that some segments appear to be fine in some segmentations and the synergy of many such segments (from different multiple segmentations) would compensate for their weakness. For example, Hoiem et al. [62] make use of multiple segmentations to obtain robust spatial support using in geometric class learning for recovering surface layout of a scene. The other works make use of multiple segmentations to obtain spatial support for objects, which is used as an additional features to improve the performance of many computer vision applications such as automatically discovering objects categories in image collections [49, 50, 109, 110, 117], image auto-annotation [127], and object recognition system [88, 104, 116]. Malisiewicz and Efros [88] demonstrated that multiple segmentations substantially improve spatial support estimation for objects compared to a single segmentation, and correct spatial support leads to substantially

better recognition performance. However, our approach differs from these ones in that these works treat multiple segmentations from an image as hypotheses for spatial/object support rather than a full segmentation combination of the image which is considered in this work.

Our proposed multiple segmentation combination algorithm is based on coassociation values and a random walker algorithm for image segmentation [54]. The coassociation values are one of the key successes of our algorithm. They provide the necessary guidelines for seed localization, which is used to bound a random walker, and provides the necessary information for biasing a random walker. In summary, the starting point of our algorithm is a graph $\mathcal{G}$, whose edge weights contain the coassociation values indicating how probably a pair of neighboring pixels x_i and x_j belong to the same image region. Once the graph $\mathcal{G}$ is defined, seed pixels required for establishing a random walker algorithm can be automatically located. Finally, given such graph $\mathcal{G}$ and seeds, the random walker algorithm is proceeded to achieve a quality final segmentation.

In the next section, we first briefly describe a random walker algorithm for image segmentation, which is a basis of our combination algorithm. We then present our novel multiple segmentation combination algorithm in Section 5.3, followed by some algorithm discussions in Section 5.4. In Section 5.5, we present the optimality criteria based on the generalized median concept for determining the final number of region in a combination result, together with other three alternative criteria. The experimental results on natural scene images to verify the proposed criteria are reported in Section 5.6, and finally, some discussions conclude the chapter.

5.2 Random Walker Based Segmentation Algorithm

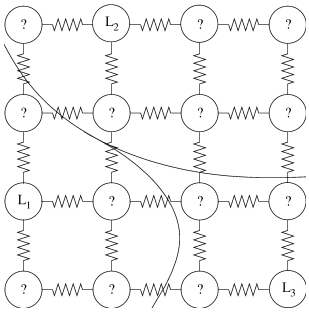
Our multiple segmentation combination algorithm was developed based on the random walker algorithm for image segmentation introduced by Grady [54]. There are a number of reasons for choosing this algorithm. Firstly, there exists a natural link between this algorithm and our problem (which will be described later in this section). Secondly, the algorithm requires a low computational time and memory which prevents us from scaling problem when the size of an ensemble and an image are increased. Lastly, the formulation of the algorithm is well-defined and can be easily modified. In the following, the formulation and the basic idea of the random walker algorithm for image segmentation are reviewed. The detail of our multiple

segmentation combination algorithm will be presented in the next section.

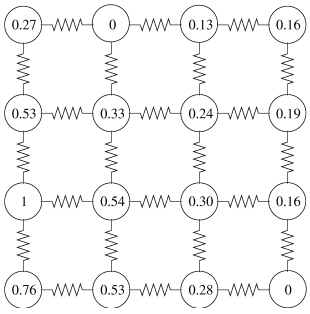
The random walker algorithm [54] is formulated in discrete space (i.e. on a graph) and developed along with the corresponding connections to discrete potential theory and electrical circuits. The algorithm functions by starting with k sets of pre-labeled pixels (called *seeds*) indicating k regions of the input image and then labeling an unseeded pixel by solving the question: Given a random walker starting at this unseeded pixel, what is the probability that it first reaches each of the k seed points? Finally, the label of the unseeded pixel is derived from these probabilities by selecting the most probable seed destination for a random walker. Connections between random walkes on graphs and discrete potential theory provide a simple, convenient method for exactly computing the desired random walker probabilities (without the simulation of a random walk) by simply solving a sparse, symmetric positive-definite system of linear equations that corresponds to a combinatorial analog of the Dirichlet problem. Figure 5.1 (taken from [54]) illustrates the approach to segmentation of a 4×4 graph with unit weights in the presence of three seeds representing three different labels (denoted L_1, L_2, L_3). The algorithm alternately fixes the potential of each label to unity (i.e. with a voltage source tied to ground) and set to zero (i.e. ground) the remaining nodes. The electric potentials calculated represent the probability that a random walker starting at each node first reaches the seed point currently set to unity. For illustration, all the weights (resistors) were set to unity. In the case of an image, these resistors would be a function of the intensity gradient.

The random walker algorithm for image segmentation [54] is formulated on an undirected weight graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, where a vertice $v_i \in \mathcal{V}$ corresponds to a pixel x_i in an image and an edge $e_{ij} \in \mathcal{E}$ connects a pair of neighboring pixels in 4-neighborhood. Associated with each e_{ij}, there is a weight $w_{ij} = w_{ij} > 0$ which indicates the similarity between two adjacent pixels x_i and x_j. A weight w_{ij} is fundamental to the random walker algorithm, corresponding to the likelihood that a random walker will move along an edge. For an example of intensity image, edge weights can be defined as a function that maps a change in image intensities and bias the random walker to avoid crossing sharp intensity gradients. Then, a quality segmentation that respects object boundaries is obtained. The term *pixel* and *node* will be used interchangeably throughout this chapter where *pixel* refer to a basic element of an image and *node* will be used in the context of graph.

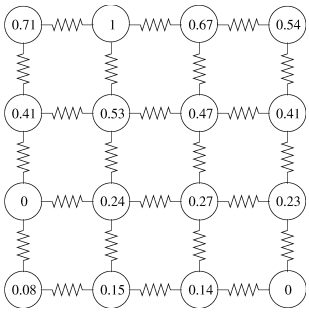
The random walker algorithm [54] begins with manually identifying k seeds (or pre-labeled pixels), indicating k regions of an input image. Seed can be a single pixel or a set of pixels. Seed of the same label can be placed on multiple locations of corresponding image region. Then, the algorithm labels unseeded pixels by resolving



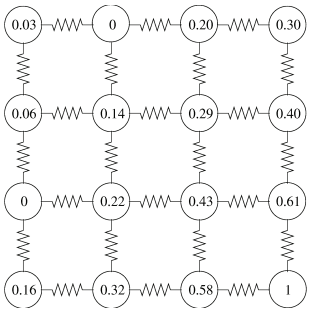
(a) Seed points with segmentation



(b) Probability that a random walker starting from each node first reaches seed L_1



(c) Probability that a random walker starting from each node first reaches seed L_2



(d) Probability that a random walker starting from each node first reaches seed

Figure 5.1. Illustrate of the approach to segmentation (taken from [54]). (a) The initial seed points and the segmentation resulting from assigning each node the label that corresponds to its greatest probability. (b)-(d) Probability that a random walker starting from each node first reaches seed L_1, L_2 and L_3, respectively.

the probability that a random walker starting from each unseeded pixel will first reach each of the k seed points. A final segmentation is derived by selecting for each pixel the most probable seed destination for the random walker. Note that the probabilities at each node sum to unity. The random walker algorithm for image segmentation is summarized in Algorithm 5.1.

In principle there exists a natural link of our problem at hand to the random walker based image segmentation. The consensus among the different initial segmentations provides strong hints about where to *automatically* place some seeds. Given such seed regions and an appropriate edge weight function, we are then faced with the same situation as image segmentation with manually specified seeds and can thus apply the random walker algorithm [54] to achieve a quality final segmentation.

Algorithm 5.1 A Random Walker Algorithm for Image Segmentation

Input: an image I
a set, V_M, of marked pixels (seeds) with k labels, specify k regions in the desired segmentation result.
Output:a segmentation of I into k regions, $S = \{s_1, s_2, ..., s_k\}$.

1. map the image intensities (or texture information, filter coefficients or other image features) to edge weights in the lattice $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$
2. perform a random walker algorithm
   2.1 assign a k-tuple vector to each pixel that specifies the probability that a random walker starting from each unseeded pixel will first reach each of the k seed points.
   2.2 calculate the random walker probabilities for each unseeded pixel by solving the Dirichlet problem (by means of a sparse, symmetric, positive-definite system of equations.
3. Obtain a final segmentation by assigning to each node, v_i, the label corresponding to the maximum probability from these k-tuples.

5.3 Multiple Segmentation Combination Algorithm

Let $\mathcal{N}$ initial segmentations be registered pixelwise on a four-connected lattice $\mathcal{G}$. Thus $\mathcal{N}$-tuples of labels are associated with each pixel. To develop the multiple segmentation combination algorithm based on the random walker we need three steps: (i) defining the weights of a graph $\mathcal{G}$, (ii) extracting seeds from $\mathcal{G}$, and (iii) computing a final combined segmentation by means of random walker algorithm. The steps of our segmentation combination algorithm are summarized in Algorithm 5.2.

5.3.1 Graph Weight Definition

A weight w_{ij} corresponds to the likelihood that a random walker will move along an edge. In the context of segmentation combination the edge weights should indicate how probably a pair of pixels x_i and x_j belong to the same image region. Hence we define the weight function as a coassociation value between two neighboring pixels x_i and x_j as:

$$w_{ij} = w(x_i, x_j) = \frac{n_{ij}}{\mathcal{N}} \tag{5.1}$$

where n_{ij} is the number of times a pair of pixels x_i and x_j is assigned to the same region among the $\mathcal{N}$ initial segmentations. The coassociation values have also been used as an effective mechanism to combine different partitions in [20, 44]. There are two reasons for choosing the coassociation values. Firstly, it is able to cope with the problems of different number of regions and label correspondence between input segmentations. Secondly, it is able to extract (to some degree) the information about homogeneous regions and region boundaries provided by the input segmentations. Homogeneous region information is necessary for seed initialization while boundary information is essential for biasing a random walker to avoid crossing region boundaries. We emphasize that the coassociation values are of essential part of our algorithm since the region information provided by the initial segmentations is embedded into these values, and, moreover, the random walker algorithm (for computing a final segmentation solution) is operated upon this representation of an ensemble. Thus, the efficacy of the proposed segmentation combination algorithm itself relies on the suitability of these values.

In order to visualize the coassociation values, a coarse measure [20] is applied. A coarse measure $c(x)$ is obtained by defining the scalar quantity with values between 0 and 255 for every pixel in a d-neighborhood system as $c(x) = \frac{255}{d} \cdot \sum_{i=1}^{d} w(x, x_i)$. An example of a gray level image of $c(x)$ is shown in Figure 5.2(a). A lighter pixel indicates a higher coassociation value. Note that the coassociation values can successfully extract some homogeneous regions (light pixels) and some nearly true region boundaries (dark pixels). The white areas indicate candidate locations for placing seeds.

In contrast to the coassociation matrix approach proposed by Fred and Jain [44], our approach requires only a small neighborhoods centered on a particular pixel (i.e. 4-connected neighborhood), and assumes that all pixels beyond this neighborhood are not linked to the pixel in question. This advantage results in a sparse affinity matrix, which is very helpful since it significantly reduces the amount of memory required to store the affinity matrix and facilitates the random walker computation for a final segmentation solution. In the approach of [44], its computational and storage complexity scale quadratically with the number of pixels. Increasing the image size would lead to computationally infeasible situation.

5.3.2 Seed Generation

Once the graph $\mathcal{G}$ is built, the next step consists in determining which subsets of nodes that correspond to homogeneous regions in the image. These nodes that cor-

respond to homogeneous regions will be regarded as *seeds* for establishing a random walker algorithm. The key principle here is that nodes that belong to the same region (or cluster) should be joined by edges with large weights, while nodes that are joined by weak edges are likely to belong to different regions. We describe a two-step strategy to automatically generate seed pixels as follows: (i) extracting candidate seeds from $\mathcal{G}$; (ii) grouping them to form final seeds to be used in the combination step. See Figure 5.3 for an illustration of seed generation procedure.

Step 1 *Extracting candidate seeds:* We build a new graph $\mathcal{G}^*$ by preserving those edges with cooccurrence probability $p(x_i, x_j) = 1$ only (i.e. x_i and x_j are assigned the same label in all $\mathcal{N}$ segmentations) and removing all other edges. This step basically retains those edges between two adjacent nodes which are most likely belong to the same region. Then, we detect all *connected subgraphs* in $\mathcal{G}^*$ and regard them as a set of initial seeds $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_m\}$ which are further reduced in the next step.

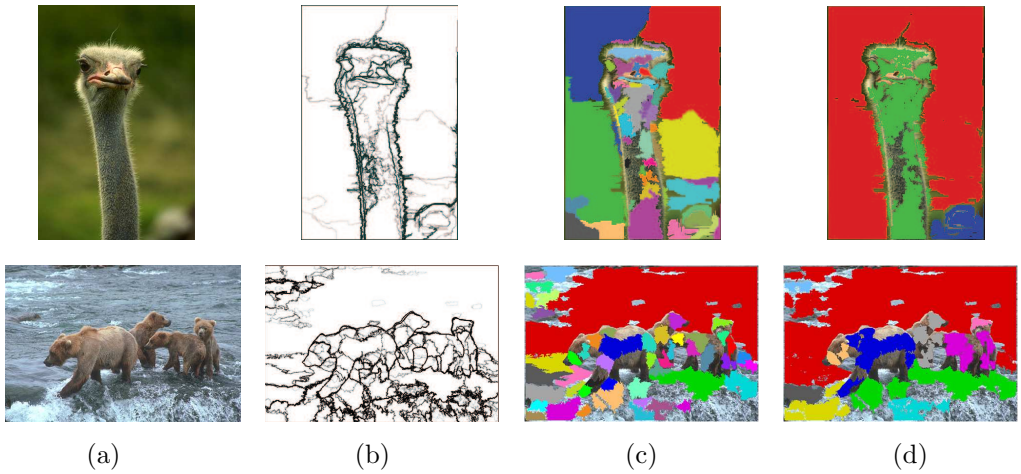
Step 2 *Grouping candidate seeds:* The number of candidate seeds from the first step is typically higher than a natural number of regions in an input image. Thus, a further reduction is performed by iteratively selecting the two candidate seeds with the highest similarity value and grouping them to build one single (possibly spatially disconnected) candidate seed. For this purpose we need to define an $m \times m$ symmetric affinity matrix A to store a pairwise similarity value among m candidate seeds, where an element $a_{ij} \in A$ contains a similarity value between two candidate seeds $\mathcal{C}_i$ and $\mathcal{C}_j$. The similarity between a pair of candidate seeds $\mathcal{C}_i$ and $\mathcal{C}_j$ is computed by averaging the coassociation values of all pair of pixels belonging to $\mathcal{C}_i$ and $\mathcal{C}_j$ as follows:

$$a_{ij} = \overline{\{w_{ij} \mid (x_i, x_j) \in \mathcal{C}_i \times \mathcal{C}_j,\ i \neq j\}} \tag{5.2}$$

where $\overline{B}$ denotes the average of the set B and $a_{ii} = 0$. The values in the affinity matrix satisfy $a_{ij} \in [0, 1]$, where the value of 1 represents perfect similarity between two candidate seeds, while 0 indicates that none of pixels in candidate seeds $\mathcal{C}_i$ and $\mathcal{C}_j$ is clustered together.

After grouping the first pair of candidate seeds $\mathcal{C}_i$ and $\mathcal{C}_j$ with the highest similarity value into a new single candidate seed $\mathcal{C}_q$, the similarity values between a new grouped seed $\mathcal{C}_q$ and all remaining candidate seeds are recomputed by averaging the similarity values of the two grouped candidate seeds scaled by their sizes as following:

$$a_{q,l} = \frac{a_{i,l}\,|\mathcal{C}_i| + a_{j,l}\,|\mathcal{C}_j|}{|\mathcal{C}_i| + |\mathcal{C}_j|}; l = 1, ..., m, l \neq i, l \neq j. \tag{5.3}$$



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

Figure 5.2. Examples of seed acquisition process. (a) Original image (b) Step 1: A graph $\mathcal{G}$ represented by a consensus image. 24 initial segmentations obtained by the FH algorithm [38] (above) and the MS algorithm [23] (below), (c) Step 2.1: Candidate seeds extracted from graph $\mathcal{G}$. Different candidate seeds indicated by different colors, and (d) Step 2.2: Final seeds after merging operation, which will be used in the combination step. Different colors indicate different seeds.

where $|\cdot|$ denotes the cardinality of a set.

There are two different approaches for stopping the merging operation. For the first approach, the merging operation is repeated until a stop condition is satisfied (see Section 5.5.2 and 5.5.3), and only one initial result is obtained. For the second approach, the merging operation is forced to generate a series of initial results with $k \in [k_{\min}, k_{\max}]$ seeds. Subsequently, each initial result is fed to the ensemble combination part of our algorithm (Step 3) to achieve a final segmentation result (for the first approach) or a total of $k_{\max} - k_{\min} + 1$ combination segmentations (for the second approach). For the second approach, we select an optimal one with respect to an objective segmentation criterion (see Section 5.5.1, 5.5.2 and 5.5.4) as the final combined segmentation.

5.3.3 Segmentation Ensemble Combination

Given the graph $\mathcal{G}$ and k seeds, the random walker algorithm performs the calculation by assigning to each pixel a k-tuple vector that specifies the probability that a random walker starting from each unseeded pixel will first reach each of the k seeds. A final segmentation is derived from these k-tuples by assigning each pixel the label of the largest probability. The computation of random walker probabilities can be

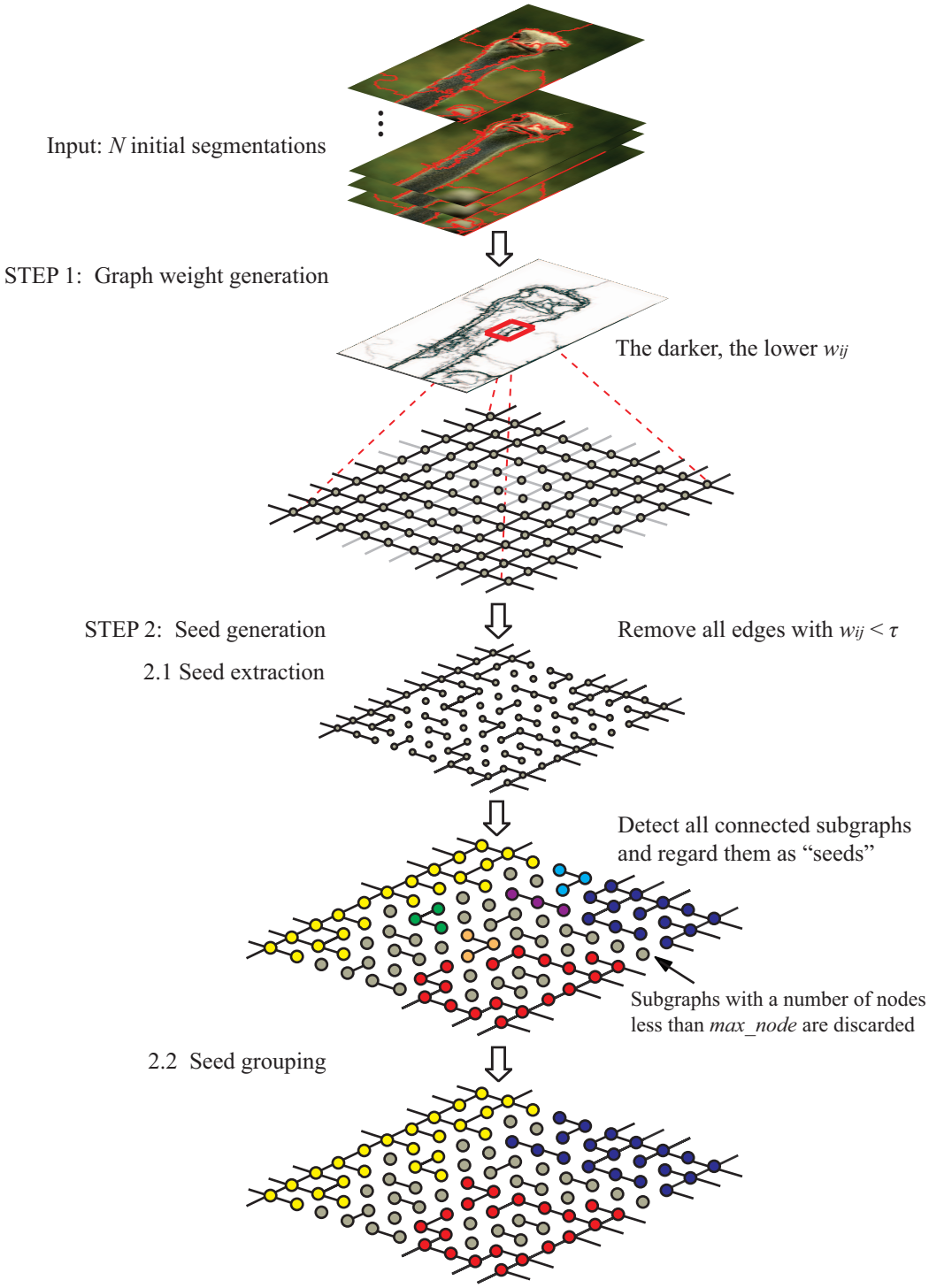


Figure 5.3. Overview of seed generation step.

Algorithm 5.2 Multiple Segmentation Combination Algorithm

Input: a set of $\mathcal{N}$ initial segmentations, $S_1, S_2, ..., S_{\mathcal{N}}$, to be combined.
Output: a combined segmentation S^*.

* *STEP1: Graph weight definition* *\

1. map $\mathcal{N}$ initial segmentations pixelwise on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$.
2. compute edge weight $wij = \frac{n_{ij}}{\mathcal{N}}$, for all $e_{ij} \in \mathcal{E}$.

* *STEP2: Seed generation* *\

3. extract candidate seeds
   - 3.1 build a new graph $\mathcal{G}^*$ by preserving those edges with $w_{ij} = 1$ and removing all other edges.
   - 3.2 detect all connected subgraphs in $\mathcal{G}^*$ and regard them as a set of initial seeds $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_m\}$.
4. group candidate seeds
   - 4.1 form an $m \times m$ similarity matrix A to store a pairwise similarity value among m candidate seeds, defined by (5.3).
   - 4.2 group the two candidate seeds, $\mathcal{C}_i, \mathcal{C}_j$, with the highest similarity value to build one single candidate seed, $\mathcal{C}_q$.
   - 4.3 update the similarity matrix A, the similarity values between a new grouped seed $\mathcal{C}_q$ and all remaining candidate seeds using (5.3).
   - 4.4 repeat 4.2 and 4.3 until the final desired number k of candidate seeds are reached.

* *STEP3: Segmentation ensemble generation* *\

5. given the graph $\mathcal{G}$ and a set of seeds $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_k\}$
   - 5.1 maximize the entropy of the edge weights using (5.4).
   - 5.3 run random walker algorithm in Algorithm 5.1 to compute the final segmentation, S^*.

exactly performed without the simulation of random walks, but by solving a sparse, symmetric, positive-definite system of equations.

In order to apply the random walker algorithm more efficiently to compute the final segmentation, a Gaussian weighting function (in accordance with [54]) is required for maximizing the entropy of the edge weights. The weights of the graph $\mathcal{G}$ is now recomputed by

$$gauss\_w_{ij} = \exp\left(-\beta \cdot (1 - w_{ij})\right) \tag{5.4}$$

where β is a free parameter of our algorithm (further discussion of β is given in Section 5.4).

5.4 Algorithm Discussion

In this section, we will talk about the general properties of the segmentation combination algorithm, investigate the stability of the proposed combination algorithm with respect to parameter β, introduce an alternative similarity measure between candidate seeds in the merging procedure, and discuss some faster practical techniques for extracting seed regions where the tradeoff of accuracy for speed does not degrade the performance of the algorithm.

5.4.1 Generality of the Combination Algorithm

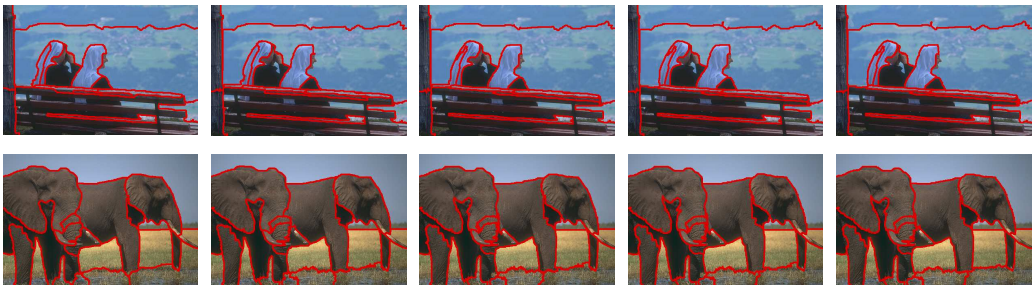
The proposed segmentation combination algorithm requires very few assumptions about the nature of the imaging process. As a result the algorithm is quite general. The generality of the algorithm can be summarized in the following aspects. Firstly, no assumptions are made about the equivalent number of regions among initial segmentations. Namely, the combination framework is able to combine initial segmentations that contain an arbitrary number of regions. This frees the user from providing a prior knowledge about the number of regions, and increases the diversity in the ensemble which is found to be beneficial in the clustering combination context [126]. Secondly, the combination algorithm is independent from the ensemble generation procedure. It takes only the results of the segmentation algorithms into account, so the way they are obtained is not important. Thus, it is possible to use any established segmentation methods for generating an input segmentation ensemble. Lastly, the combination algorithm is not restricted to specific image features. No assumption is needed at the moment about a prior knowledge about original

image features (e.g. color, texture), namely only the label feature delivered by segmentation algorithms is taken into account. This allows the combination procedure applicable for different imaging modalities (e.g. color, intensity, range, etc.).

5.4.2 Stability of the Combination Algorithm

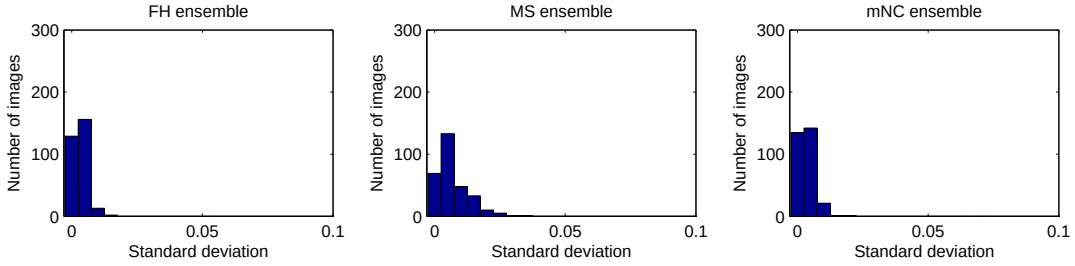
Algorithm stability is another important indication of an algorithm's usefulness. If an algorithm gives reasonably correct segmentations on average, but is wildly unpredictable on any given image or with any given parameter set, it will be useless as a preprocessing step for other algorithms, such as object recognition [103]. In this section we address the stability issue with respect to parameter choice. The algorithm, that has this stability, must give consistent results on the same image given different parameter inputs.

Referring to (5.4) in Section 5.3 there is a single parameter of our combination algorithm, β, which is an inverse temperature parameter for the Gaussian random field. The difference of combination results given different β values mostly occurs at very small boundary pixels along indistinct region boundaries. Some examples of segmentation results with different values of β are shown in Figure 5.4. We have systematically conducted a set of experiments that addresses the issue of stability with respect to parameter β. The experiment investigates the effect of β values over combination results by running the combination algorithm with $\beta = \{10, 30, 60, 90, 120\}$ for all 300 images in the BSDS dataset. The thresholding criterion (T_{merge}) is applied here for determining the number of regions in a final segmentation result since it is computationally inexpensive (This thresholding criterion will be presented later in Section 5.5.3). A segmentation ensemble is generated by varying the parameter values of a baseline segmentation algorithm. Three different experiments are conducted. In the first experiment, input segmentation ensembles are obtained by the FH algorithm. In the second experiment, input segmentation ensembles are obtained by the MS algorithm. In the third experiment, input segmentation ensembles are obtained by the mNC algorithm. The parameter subspace and sampled parameter values of each algorithm are summarized in Table 6.1. The average NMI (ANMI) index is applied for assessing a segmentation result against its corresponding ground truths. For each experiment, we compute a standard deviation of ANMI values of segmentation results of each input image computed using different values of β. A standard deviation histogram for each of three experiments are shown in Figure 5.5. For all cases, the histogram is skewed to the left which indicates that our combination algorithm has a rather small sensitivity to changes in β. We set β to 30 for all



(a) $\beta = 10$ (b) $\beta = 30$ (c) $\beta = 60$ (d) $\beta = 90$ (e) $\beta = 120$

Figure 5.4. Examples of combined segmentation results with different values of parameter β.



(a) FH ensemble (b) MS ensemble (c) mNC ensemble

Figure 5.5. Histograms of the standard deviations of ANMI values of segmentation results computed by different values of β. Results for the FH, MS and mNC segmentation ensembles are shown in columns (a), (b), and (c), respectively.

our experiments.

5.4.3 Random Walker Based Similarity Measure

The number of candidate seeds is typically higher than the true number of regions in an input image. Thus, a further reduction is needed and performed by iteratively merging the two closest candidate seed regions until some termination criterion is satisfied. For this purpose we need a similarity measure between two candidate seeds and a termination criterion. In Section 5.3.2, the similarity between two candidate seeds is computed on the basis of coassociation values (5.2) and (5.3). An alternate method to measure the similarity between two candidate seeds is based on random walker probability. This method has been presented in our previous work [141].

Recall that in the initial graph $\mathcal{G}$ the edge weights w_{kl} indicate how probably two pixels p_k and p_l belong to the same image region. This interpretation gives us a means to estimate how probably two candidate seed regions $\mathcal{C}_i$ and $\mathcal{C}_j$ belong to

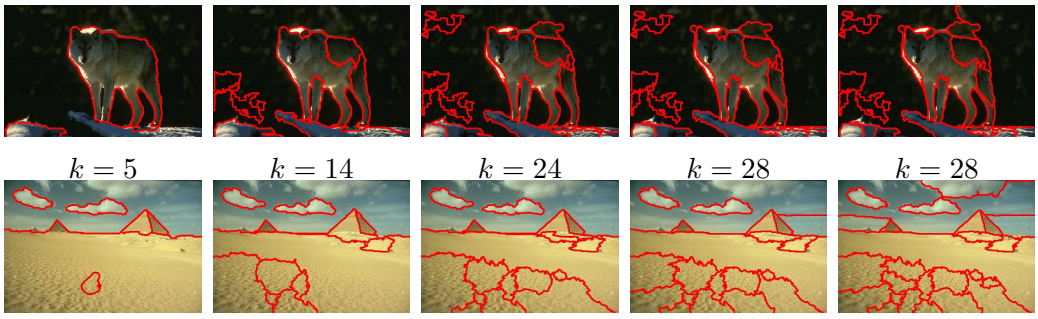


Figure 5.6. Examples of combined segmentation results using the random walker based similarity measure with different values of parameter β.

the same region. For a node $p_k \in \mathcal{C}_i$ we consider the probability $P(p_i, \mathcal{C}_j)$ that when starting from p_i, a random walk will reach any node in $\mathcal{C}_j$. Then, we define the similarity between $\mathcal{C}_i$ and $\mathcal{C}_j$ by:

$$Similarity(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{2}[\max_{p_k \in \mathcal{C}_i} P(p_k, \mathcal{C}_j) + \max_{p_l \in \mathcal{C}_j} P(p_l, \mathcal{C}_i)] \tag{5.5}$$

The probability $P(p_i, \mathcal{C}_j)$ can be efficiently computed by the baseline random walker algorithm [54] described in Section 5.2.

The computational cost of this similarity measure depends on the number of candidate seed regions. We need to run the random walker algorithm m times (being the number of candidate seed regions). We use only a small amount of pixels per candidate seed region by sampling them along the horizontal and vertical image grid by, for example, factor 5 in each direction. By doing this way, we can substantially reduce the time required by the random walker algorithm.

This random walker based similarity measure has shown its effectiveness to measure the similarity between two candidate seeds. However, it is relatively sensitive to the parameter β which is required for maximizing the entropy of the edge weights in the random walker algorithm. Figure 5.6 shows examples of combined segmentation results computed using the random walker based similarity measure (5.5) with different values of $\beta = \{10, 30, 60, 90, 120\}$.

5.4.4 Further Implementation Details

Speedup of Seed Generation

The main computational burden of our combination algorithm stems from computing the similarity matrix in (5.2). Instead of computing the similarity values between every pixel in one seed to every pixel in the rest seeds, we can cut down the number of similarity computation by randomly selecting for each seed a small set of pixels and use them in computation. This small fraction of pixels per seed, in practice, is sufficient to estimate the similarity between seeds. We have systematically investigated the influence of the number of pixels per seed used in similarity computation on the segmentation results. We have tested on 44 images in the BSDS data set. We firstly compute the combination results using both ten pixels per seed and all pixels per seed, then compute ANMI values between two combined results. The ANMI values indicates that there are no difference between segmentation results computed by using ten pixels per seed and using all pixels per seed (i.e. ANMI values between them are equal to 1). The only difference between them is the computational time. Thus, for all experiments reported in this paper we randomly select ten pixels per seed for the similarity computation.

Speedup of Candidate Seed Merging Procedure

In candidate seed region extraction step (in Section 5.3.2) the only connected subgraphs with $p(x_i, x_j) = 1$ will be regarded as seed regions. This criterion in some cases may create a very large number (e.g. more than 5,000) of candidate seed regions whose size is very small (e.g. smaller than 3 pixels per region) with respect to 321×481 image size. These very-small-sized candidate seeds mostly indicate either the same image regions as do the larger candidate seeds or noise regions. When they represent noise regions, they will not be merged into any meaningful seed regions, resulting in combination output with noisy specks (as shown in Figure 5.7(d)) or undesirable regions (as shown in Figure 5.7(b), small elongate regions along the region boundaries). Thus it would be more practical to disregard these very-small-size candidate seeds and take only the first $k_{\max}$ largest candidate seeds into account. In the case that the number of candidate seeds is smaller than $k_{\max}$ (which hardly ever occurs), the number of all candidate seeds will be used in place of $k_{\max}$.

For all combination results presented in this paper, connected subgraphs, whose size is larger than ten pixels, are considered as candidate seed regions and only the first $k_{\max} = 50$ largest candidate seed regions are used in the merging procedure.

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

Figure 5.7. Examples of combined segmentation results with different initial candidate seeds. (a) and (c) Only 50 largest candidate seed regions are used in merging procedure. (b) and (d) All candidate seed regions are used in merging procedure.

This number is experimentally determined to be large enough to cover all salient natural image segments as shown in Figure 5.7 (a) and (c).

By doing it this way, we can improve the computational performance of the algorithm without degrading the qualities of combination results. The proposed combination algorithm was implemented in MATLAB on an Intel Core 2 CPU. Seed region generation for an image of dimension 321×481 with 24 initial segmentations requires less than two seconds in average, and so does the random walker computation in the final step. Our algorithm is efficient enough to be capable of evaluating a series of possible combination results with different k values and selecting an optimal segmentation based on a median concept criterion. However, this stage could be parallelized to make the system more acceptable for real-time applications. Further reduction of computation time can be done in the $\mathcal{N}$-segmentation ensemble generation. One possibility is to obtain them in parallel.

5.5 Determination of the Final Number of Regions

The automatic identification of the appropriate number of clusters is a deep research problem that has attracted significant attention in data clustering community. Many approaches for dealing with this problem have been proposed in the literature. A comprehensive survey of methods for estimating the number of clusters is given in [31, 58, 95]. In this work, we investigate two different approaches for determining the number of regions in a final combined segmentation result: *optimization* approach and *thresholding* approach.

In optimization approach we first start with a series of n different segmentation results and then, for each segmentation result, we compute its cost according to the predefined objective function to be optimized. The segmentation result with

the minimum cost will be selected as the optimal segmentation solution. This algorithm follows from the general model selection approach to searching for the optimal partition of a data set, given the minimal and maximal number of clusters. Two methods of this category that are considered in this work are *generalized median concept based* and *MDL based* objective functions. We demonstrate that even though the performance of both methods is comparable, the strength of the generalized median concept method lies in the fact that no original image features (e.g. intensity, color, texture) of an input image are needed. This benefits in the situation when the original image features are not available and allows the method applicable for any kind of imagery/task without the need of modification.

The second set of approaches deals with the difficulty of establishing adequate stopping criteria in the candidate seed merging procedure. The iteration of the merging process which satisfies the criterion is chosen as the best iteration. Then, the selected segmentation level is the optimal segmentation. Two thresholding methods are investigated in this work. The first one selects the best segmentation iteration by taking the similarity values between two merging regions into account. The iteration is stopped if the merging similarity value falls below the predefined threshold value. The second thresholding method determines the best segmentation level by exploring the dendrogram computed from the merging procedure. We also show that by incorporating the optimization approach into the thresholding approach, we can achieve an approximated optimal segmentation solution with much lower computational cost.

5.5.1 Median Concept Criterion

The *generalized median concept* (see Section 2.1 for details) is a powerful tool for inferring a representative model of a given set of noisy samples of the same object and has found promising applications in several domains (e.g. graphs [74], prototype learning [74], and double contour detection [139]). In this work, we apply the generalized median concept to select the best (optimal) segmentation $S^{k_{\text{opt}}}$ from a set of combination segmentations with different number k of regions as the one with minimal sum of distances among all individual segmentation S_q in Λ:

$$S^{k_{\text{opt}}} = \arg\min_{\hat{S}} d(\hat{S}, \Lambda) \tag{5.6}$$

where $\hat{S}$ covers all possible $k \in [k_{\min}, k_{\max}]$ segmentations and $d(\cdot, \cdot)$ is a distance function.

If we replace $\hat{S}$ by a universe $\mathcal{U}$ of all possible segmentations of an image, then $S^{k_{\text{opt}}}$ would represent the optimal segmentation in accordance with the generalized median concept of the input ensemble [74]. Therefore, our approach can be regarded as an approximation of generalized median segmentation by investigating the subspace of $\mathcal{U}$ consisting of the combination segmentations for all possible $K \in [K_{\min}, K_{\max}]$ only.

Note that if we define the above median segmentation optimization function based on normalized mutual information (e.g. by using normalized mutual information as a distance function between two segmentations), this approach is equivalent to the concept of cluster ensemble framework presented by Strehl and Ghosh [126], in the sense that a good combined clustering should share as much information as possible with the given original clusterings.

5.5.2 MDL Criterion

In the absence of ground truth data, it is critical to have a criterion that enables the quality of a segmentation to be evaluated. A more sophisticated approach to deal with this problem is to use the *minimum description length* (MDL) principle. The MDL principle, originally developed by Rissanen [113], is a method for inductive inference that provides a generic solution to the model selection problem. The MDL principle defines the best fitted model as the one that produces the shortest code length of the data (e.g., the best encoding of the data). The MDL criterion was first used for the problem of image segmentation by Leclerc [83] and followed by many works such as [48, 78, 84, 112, 152]. The difference between them lies in the term they used to encode the image data (e.g. texture information, region boundary information, color information).

In order to apply the MDL principle to tackle the present problem, we first need to construct a code length expression to encode an image. In this work we follow the MDL-based objective segmentation criterion proposed by Rao et.al [112]. Rao et.al used the MDL principle to encode both the texture and boundary information of a natural image and defined the optimal segmentation of an image as the one that minimizes its total coding length. In the following we firstly describe how to encode the texture and boundary information of a natural image and then construct an objective segmentation criterion based on these coding length functions.

Adaptive Texture Encoding: Rao et.al construct texture vectors that represent homogeneous textures in image segments as follows. Let the w-neighborhood $\mathcal{W}_w(p)$

be the set of all pixels in a $w \times w$ window centered at pixel p. They construct a set of features X by taking the w-neighborhood around each pixel in an image I across the three color channels, and then stacking each window as a column vector:

$$X = \left\{ x_p \in \Re^{3w^2} : x_p = \mathcal{W}_w(p)^S \text{ for } p \in I \right\}.$$

For ease of computation, they reduce the dimensionality of these features by projecting the set of all features X onto their first D principal components. They denote the set of features with reduced dimensionality as $\hat{X}$ and choose to assign $D = 8$. Subsequently, the texture information is encoded using a Gaussian distribution. First Rao et.al consider a single region R with N pixels. For a fixed quantization error ϵ, the expected number of bits needed to code the set of N feature window $\hat{X}$ up to distortion ϵ^2 is given by:

$$L_{w,\epsilon}(R) = (\frac{D}{2} + \frac{N}{2w^2})\log_2\det(I + \frac{D}{\epsilon^2}\hat{\Sigma}_w) + \frac{D}{2}\log_2(1 + \frac{\|\hat{\mu}_w\|^2}{\epsilon^2}). \tag{5.7}$$

Adaptive Boundary Encoding: Rao et.al apply a well-known scheme, the *Freeman chain code*, for representing boundaries of image regions. In this coding scheme, the orientation of an edge is quantized along eight discrete directions. Let $\{o_t\}_{t=1}^T$ denote the orientations of the T boundary edges of R. Since each chain code can be encoded using three bits, the coding length of the boundary of R is

$$B(R) = 3\sum_{i=0}^{7} \sharp(o_t = i).$$

Given the prior distribution $P[\Delta o]$ of difference chain codes, $B(R)$ can be encoded more efficiently using a lossless Huffman coding scheme:

$$B(R) = -\sum_{i=0}^{7} \sharp(\Delta o_t = i)\log_2(P[\Delta o = i]). \tag{5.8}$$

Minimizing Coding Length: Suppose an image I can be segmented into non-overlapping regions $\mathcal{R} = R_1, ..., R_k, \cup_{i=1}^k R_i = I$. Based on the coding length functions developed in (5.7) and (5.8), the total coding length of the image I is

$$L_{w,\epsilon}^S(\mathcal{R}) = \sum_{i=1}^{k} L_{w,\epsilon}(R_i) + \frac{1}{2}B(R_i). \tag{5.9}$$

Note that the boundary term is scaled by a half because we only need to represent the boundary between any two regions once. The optimal segmentation of I is the one that minimizes (5.9).

In this work an objective segmentation criterion in (5.9) is applied to determine the number of k in two ways: (i) selection strategy and (ii) merging strategy.

(i) Selection Strategy:

Given a sequence of combined segmentations for a range of values of $k \in [k_{\min}, k_{\max}]$, the best (optimal) combined segmentation $S^{k_{\text{opt}}}$ is the one that minimizes (5.9).

(ii) Greedy Merging Strategy:

To find the optimal segmentation we exactly follow an agglomerative process presented in [112]. We initialize the optimization process by utilizing a combined segmentation result with $k_{\max}$ as a superpixel. Given a superpixel of the image, at each iteration, we find the pair of adjacent regions R_i and R_j that will maximally decrease (5.9) if merged:

$$(R_i^*, R_j^*) = \arg \max_{R_i, R_j \in \mathcal{R}} \Delta L_{w,\epsilon}(R_i, R_j), \text{where}$$

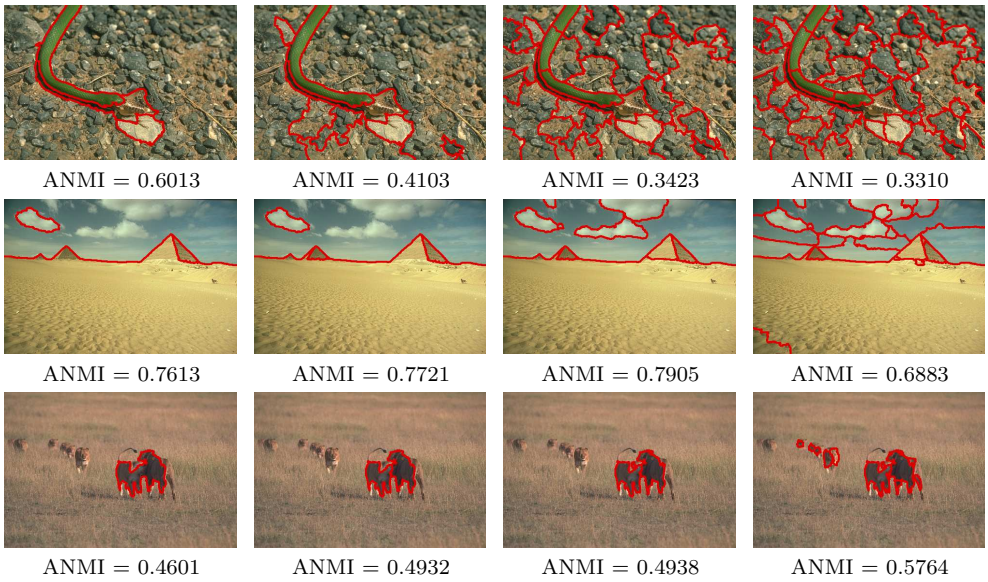
$$\Delta L_{w,\epsilon}(R_i, R_j) = L_{w,\epsilon}^S(\mathcal{R}) - L_{w,\epsilon}^S((\mathcal{R} \setminus \{R_i \cup R_j\}) \cup \{R_i \cup R_j\})$$

$$= L_{w,\epsilon}(R_i) + L_{w,\epsilon}(R_j) - L_{w,\epsilon}(R_i \cup R_j) + \frac{1}{2}(B(R_i) + B(R_j) - B(R_i \cup R_j)). \quad (5.10)$$

$L_{w,\epsilon}(R_i, R_j)$ essentially captures the difference in the lossy coding lengths of the texture regions R_i and R_j and their boundaries before and after the merging. If $\Delta L > 0$, we merge R_i^* and R_j^* into one region, and repeat this process until $L_{w,\epsilon}^S(\mathcal{R})$ cannot be further reduced.

5.5.3 Thresholding Criterion

Thresholding is the simplest criterion for determining the number of regions in segmentation. We define a threshold T_{merge} to indirectly control the number of regions k through a merging candidate seed operation. The merging operation is stopped if the highest similarity between two merging candidate seeds is below T_{merge}. Thus, the amount of detail (k) in the final segmentation can be influenced by changing the value of T_{merge} appropriately. Larger values of T_{merge} yield a larger number of seed regions, while smaller values of T_{merge} yield fewer number of seed regions. Figure 5.8 shows three examples of combined segmentation results of test images, where the segmentations obtained for different values of $T_{\text{merge}} = \{0.3, 0.5, 0.7, 0.9\}$ from (a) to (d), respectively. We can see that the value of T_{merge} relates heavily to the natural number of regions in an input image, for example, input image with fewer number of



(a) $T_{\text{merge}} = 0.3$ (b) $T_{\text{merge}} = 0.5$ (c) $T_{\text{merge}} = 0.7$ (d) $T_{\text{merge}} = 0.9$

Figure 5.8. Examples of combined segmentation results with different values of threshold T_{merge}.

regions prefers smaller value of T_{merge}. Thus, using a single value of T_{merge} throughout the data set cannot achieves the optimal segmentation result for all images. However, this approach is much faster than the above optimization approach.

Since setting the accurate value of T_{merge} so as to obtain a large enough number of seed regions to cover all salient natural image segments (i.e. not too coarser or too finer) is difficult, we can apply the above optimization approach to estimate the optimal values of T_{merge}. For example, we apply the median segmentation optimization criteria (5.6) defined in Section 5.5.1 to estimate the optimal values of T_{merge}. This can be done by replacing the subspace of $\mathcal{U}$ in (5.6) by a set of combined segmentations computed with all sampled values of $T_{\text{merge}} \in [0, 1]$. The optimal value of T_{merge}, denoted by $T^{\text{opt}}_{\text{merge}}$, is the one that minimizes the sum of distances among all individual segmentation S_q in Λ.

Notably, this strategy reduces a large amount of work required by the original optimization method, in order to achieve $S^{k_{\text{opt}}}$. The original optimization method has to consider a set of all possible $k \in [k_{\min}, k_{\max}]$ segmentations, while the optimization of T_{merge} considers only a set of segmentations computed by a small set of sampled values of T_{merge}, which is typically much smaller space than $[k_{\min}, k_{\max}]$. For example, if all possible k is set to [2, 50], and all sampled values of T_{merge} are set to $\{0.3, 0.4, ..., 0.9\}$. The amount of work needed by the optimization of T_{merge}

is 7 times less than the original optimization method. Based on our experience the meaningful range of T_{merge} values is [0.3, 0.9]. The experimental results reported in Section 5.6 demonstrate that the quality of segmentation results computed by the optimal T_{merge} is close to the quality of segmentation results computed by the median segmentation optimization criteria (5.6). Thus, the approach of optimization of T_{merge} is very useful when the need of computational time is more critical than the optimal solution. We would like to note that the MDL-based optimization criterion is able to apply to optimize T_{merge} as well, in a similar manner.

5.5.4 Lifetime Criterion

Another thresholding criterion considered in this work is called *lifetime criterion* proposed by Fred and Jain [44]. They used the *highest* lifetime partition criterion to decide the number of clusters in the combined partition. The $k - cluster$ lifetime is defined as the range of threshold values on the dendrogram that lead to the identification of k clusters. The results presented in their work are concerned with combined partitions extracted from the dendrogram produced by the single link and the average link methods. In order to apply this criterion in the present work, the dendrogram is computed from hierarchical merging procedure described in Section 5.3.2. For instance, Figure 5.9 shows the dendrogram produced by the merging procedure for the candidate seeds in Figure 5.2(c, above). Lifetimes of 2, 3, and 4-cluster partitions are represented in Figure 5.9 as l_2, l_3, and l_4, respectively. The lifetime of the 2-cluster solution, $l_2 = 0.0313$, is computed as the difference between minimum (0.9319) and the maximum (0.9632) threshold values that leads to the separation of patterns into two clusters. In this case the 3-cluster partition, $l_3 = 0.2241$, corresponds to the highest lifetime and is chosen as the optimal solution.

5.6 Experiments

The experiments presented in this section are intended to validate the effectiveness of the four criteria. The effectiveness of the segmentation combination algorithm will be presented in the next chapter. In these experiments the efficient graph-based image segmentation proposed by Felzenszwalb and Huttenlocher (FH) [38] is used as a baseline segmentation algorithm for producing a set of initial segmentations for combination. An input segmentation ensemble is generated by means of parameter subspace sampling (see Section 6.1 for more details). The parameter subspace of

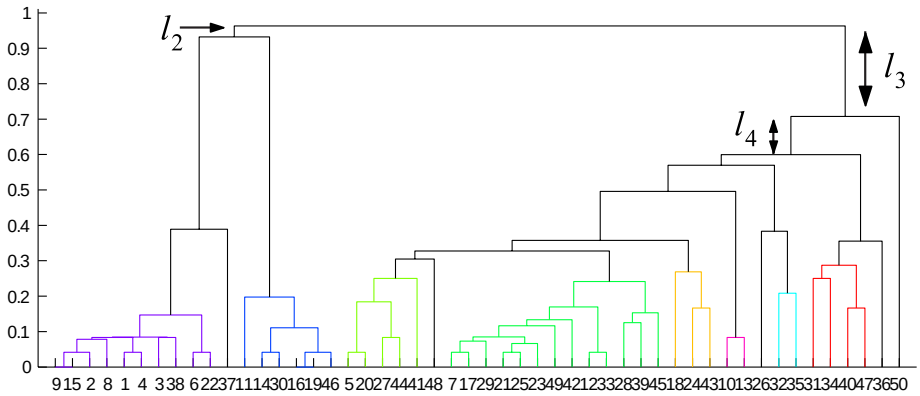


Figure 5.9. Dendrogram produced by the merging procedure described in Subsection 5.3.2 for the candidate seeds in Figure 5.2(c, above).

the FH segmentation algorithm is sampled into 24 combination of parameters (see Table 6.1). For each combination of parameters the segmentation algorithm is run over the complete set of 300 images from the BSDS data set. By doing this way, we obtain 300 segmentation ensembles (for each 300 images), and each ensemble consists of 24 initial segmentations. In the case of optimization approaches, we run the combination algorithm multiple times for each image, varying the region number k in an interval [2, 50], and then selecting the combination result in accordance with the criterion used for determining k. We apply both NMI and F-measure to quantitatively evaluate the segmentation quality against the ground truth. In the case of NMI index one segmentation result is compared to all manual segmentations and the average NMI (ANMI) is reported. Larger ANMI values indicate better combination results that share more information with the ground truths.

Figure 5.10 shows examples of the segmentation results for four images on natural scenes. From left to right, the six columns show segmentation results based on (a) the generalized median segmentation criterion, (b) MDL-based merging criterion, (c) MDL-based selection criterion, (d) Threshold T_{merge} criterion, (e) Optimal threshold T_{merge} criterion, and (f) Lifetime criterion, respectively. Quantitative comparison between these six different criteria is reported in Figure 5.11 in terms of both ANMI value (a) and F-measure (b). In each plot, we also include the average performance of segmentation results obtained by the baseline segmentation algorithm (the dot line) of all 300 images for each combination of parameters in comparison with the average performance of six different criteria for determining k. For both evaluation measure, the first five criteria (i.e. the generalized median criterion, MDL-based merging criterion, MDL-based selection criterion, threshold T_{merge} criterion, and

optimal threshold T_{merge} criterion) are able to achieve the average improved results over a single run of baseline segmentation algorithm, while the lifetime criterion is not.

For NMI index, the MDL-based merging criterion performs better than the generalized median criterion, whereas the generalized median criterion performs slightly better than the MDL-based merging criterion for F-measure index, and outperforms the MDL-based selection criterion for both index. However, it should be noted that even though the performance of MDL-based method and the generalized median method are comparable, the strength of the generalized median method lies in the fact that no original image features (e.g. intensity, color, texture) of an input image are needed. This benefits in the situation when the original image features are not available and allows the method applicable for any kind of imagery/task without the need of modification.

Empirical evidence also supports the idea that incorporation of the generalized median approach into the thresholding approach (T_{merge}) can produce an approximation of the optimal segmentation solution obtained by the generalized median approach, however, with much lower computational cost (than applying the generalized median approach alone). As shown in Figure 5.10 the performance of the optimal threshold T_{merge} criterion is relatively similar to the performance of the generalized median criterion for both evaluation measures. Thus, in the situation where the need of computational time is more critical than the optimal solution, we can apply the optimization version of thresholding approach instead of traditional optimization method.

5.7 Conclusion

A novel segmentation combination algorithm based on a random walker segmentation algorithm has been proposed. The combination algorithm uses coassociation values to encapsulate the cluster (region) information provided by an input segmentation ensemble, which is important not only for automatically generating seeds for a random walker algorithm, but also for biasing the random walker to avoid crossing the region boundaries. The combination algorithm has been designed in a general framework, which is not restricted to specific image features or segmentation methods. This enables the combination procedure to lend itself to a wide range of segmentation tasks (for example, regions in color or texture images, surface patches in range images, etc.) and a wide range of imagery data.

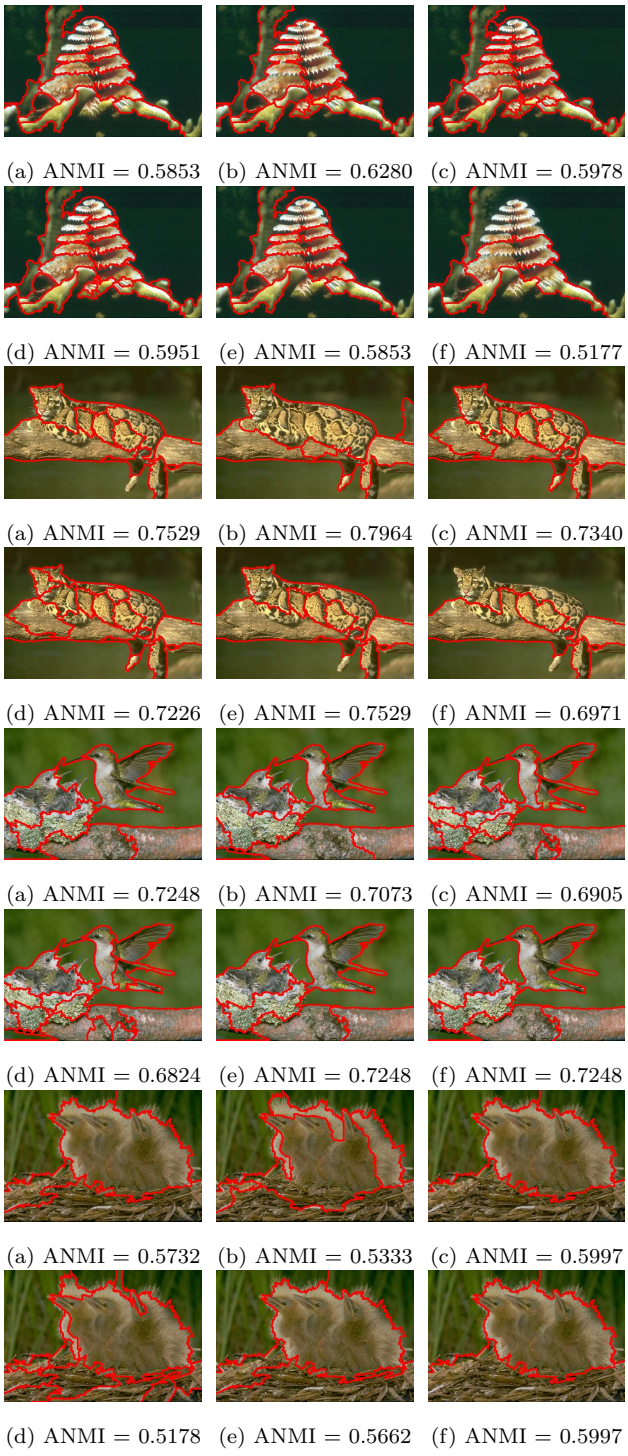
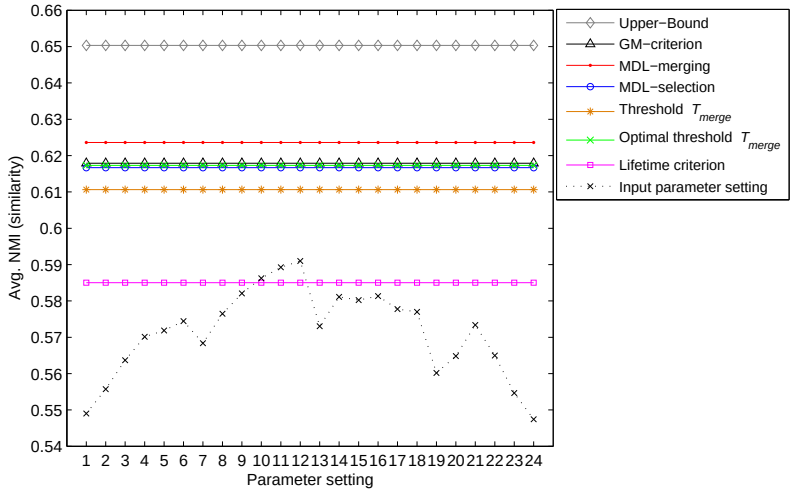
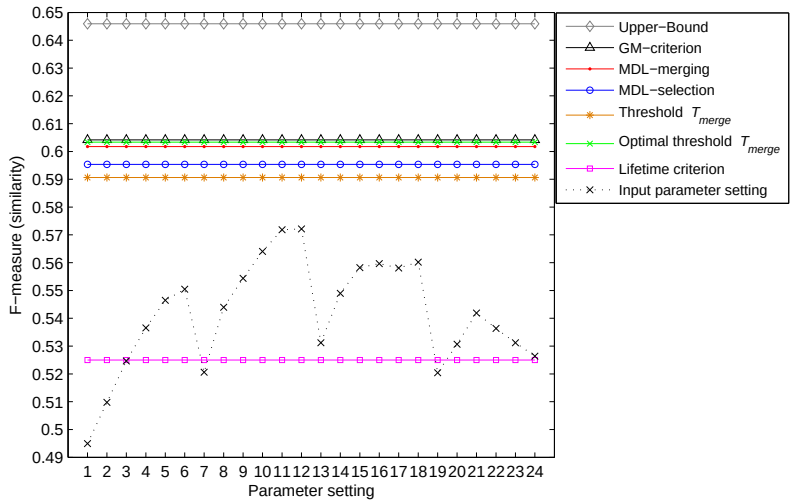


Figure 5.10. Examples of segmentation combination results computed using different criteria for determining the number of regions: (a) the generalized median segmentation criterion, (b) MDL-based merging criterion, (c) MDL-based selection criterion, (d) Threshold T_{merge} criterion, (e) Optimal threshold T_{merge} criterion, and (f) Lifetime criterion.



(a)



(b)

Figure 5.11. Average performance of combination results using different criteria for determining k over 300 images for each individual parameter setting.

We define the problem of determining the number of regions in a final combined segmentation as the optimization problem where, given a series of combined segmentation solutions, we want to find the segmentation that minimizes the sum of distances among all input segmentations in an ensemble, namely, an approximation of the generalized median segmentation. The effectiveness of the generalized median based criterion is demonstrated by comparing it with three alternative criteria for determining the number of regions. The experimental result shows that the performance of the generalized median criterion is superior to the thresholding criteria and is comparable to the MDL-based criteria.

While the presented results are very promising, there is still much more room for improvement. To illustrate this, we select the (best) optimal combined segmentation solution (from a series of combination results) with the highest ANMI values compared to its corresponding ground truths for each input image in the data set, and compute its average performance. This *ideal* performance indicates the upper-bound performance we can achieve from the segmentation combination algorithm. As shown in Figure 5.11 the average upper-bound performance line lies far above from the line of the best average performance we can obtain at the present. It could be concluded that the proposed optimality criteria for selecting the best combined segmentation (from a series of combination results) are not powerful enough to find the true optimal result. One direction to improve the performance of the current results towards the *ideal* performance here is to use/define new distance function with higher discrimination ability in distinguishing the difference between two segmentations (used in (5.6)) or constructing new, better representative coding length function (used in (5.9)).

Chapter 6

Ensemble Generation

In the previous chapter we presented a novel segmentation combination algorithm for combining multiple segmentations of the same image. The experiments[1] have been conducted to verify the capability of the four different criteria for determining the final number of regions in combination results. In this chapter a number of experiments are conducted to demonstrate the efficacy of the combination algorithm to produce the *improved* segmentation result over an input ensemble. We verify the efficacy of our segmentation combination algorithm in a variety of segmentation ensemble generation approaches:

- *Parameter subspace sampling approach*: This approach concerns with the problem of parameter selection, which is fully described in Chapter 7. A segmentation ensemble is obtained by varying the parameter values of the same segmentation algorithm in an appropriate range. This approach will be applied using three well-known segmentation algorithms, which are FH, MS and mNC segmentation algorithms.

- *Multiple segmentation algorithm approach*: This approach concerns with the problem of selecting the best segmentation algorithm for a particular image. Since the comparative performance of different segmentation algorithms can vary significantly across images, it is not easy to know the optimal algorithm for one particular image. In this approach, multiple segmentations of the same image are obtained by using different segmentation algorithms. The three well-known segmentation algorithms (i.e. FH, MS, and mNC) are used in the experiments.

[1]The experiments have been conducted on BSDS dataset, where multiple segmentations are generated by varying the parameter values of the FH segmentation algorithm.

- *Multiple image transformation approach*: This approach is different from the above approaches in that the variation in segmentation ensembles are created by varying the representations of an input image given the same segmenter, instead of varying the segmenters given the same input image. This approach is based on the fact that most segmentation algorithms existing in the literature are image dependent. Local variations of the image may change dramatically the segmentation results. A variety of image transformations, such as geometric transformations, affine transformations, and perspective transformations, are applied for generating multiple segmentations of the same image.

All experiments reported in this chapter will be conducted on BSDS dataset, where multiple segmentations are generated by the three above different approaches. The quality of segmentation result is quantitatively evaluated using NMI index and F-measure against the corresponding ground truth segmentations. In order to demonstrate the improvement of combination results over the input ensemble, the performance of combination approach is reported in comparison with the performance of the baseline segmentation algorithms.

Moreover, to gain insights into the performance improvement obtained by our segmentation combination method, we analyze the interplay between diversity and accuracy of the individual segmentation solutions in a segmentation ensemble and the influence of them on the final segmentation combination performance.

6.1 Parameter Subspace Sampling

In this experiment we adopt the ensemble combination principle to solve the parameter selection problem in image segmentation (The full detail of this problem is described in Chapter 7.). It explores the parameter space without the need of ground truth. It is assumed that we know a reasonable subspace of the parameter space (i.e. a lower and upper bound for each parameter), which is sampled into a finite number N of parameter settings. Then, we run the segmentation procedure for all the N parameter settings and compute a final combined segmentation of the N segmentations. The rationale behind our approach is that this segmentation tends to be a good one within the explored parameter subspace.

6.1.1 Segmentation Ensemble Generation

Multiple segmentations in an ensemble are obtained by varying the parameter values of the same segmentation algorithm in an appropriate range. The appropriate ranges of parameters are experimentally determined so that the resulting segmentations would have reasonable or acceptable quality (i.e. not overly under/over-segmentations). The sampled values of parameters within these ranges are chosen so as to yield segmentations with perceptible differences. These criteria are applied for all segmentation algorithms used in the experiments.

In this set of experiments, the three well-known segmentation algorithms: FH, MS, and mNC, are used as baseline segmentation algorithms for generating a set of initial segmentations to be combined. The detail of each algorithm is described in Chapter 2. The experiments are conducted using all three segmentation algorithms in order to demonstrate that our segmentation combination algorithm is able to work well with a variety of image segmentation methods. Ranges of the algorithm parameters and their sampled values for each segmentation algorithm used in the experiments are summarized in Table 6.1. The total number of parameter combinations for each algorithm is equal to 24 combinations.

For each combination of parameters the segmentation algorithms are run over the complete set of 300 images from the BSDS data set to form a set of initial segmentations (which will be called *a segmentation ensemble*) for a combination. In detail, for each segmentation algorithm we run the following procedure:

(1) For all 300 images in the BSDS data set:
  (1.1) For all 24 parameter settings:
    Run the segmentation algorithm on an input image.
  (1.2) Obtain a segmentation ensemble consisting of 24 segmentation solutions (according to 24 parameter settings)
(2) Obtain a set of 300 segmentation ensembles for all 300 images.

By this way, we achieve three different sets of 300 segmentation ensembles by running the three different segmentation algorithms. In the experimental report we refer a set of 300 segmentation ensembles produced by the FH algorithm as *FH ensembles*, a set of 300 segmentation ensembles produced by the MS algorithm as *MS ensembles*, and a set of 300 segmentation ensembles produced by the mNC algorithm as *mNC ensemble*.

Table 6.1. Parameters, descriptions and values of baseline segmentation algorithms.

Algo.	Parameter Value	Description
FH	$\sigma = \{0.4, 0.5, ..., 0.9\}$	A parameter of Gaussian filter
	$k = \{150, 300, 500, 700\}$	A parameter of a threshold function, larger k causes a preference for larger components in the result.
	$M = 1500$	We fix a minimum size of regions to be approximately 1% of input image area to avoid gross over-segmentation.
MS	$h_s = \{8, 16\}$	A spatial bandwidth parameter. The original paper of this algorithm [23] claimed that the algorithm is not very sensitive to the choice of h_s, and suggest to use $h_s = 8$ for 256×256 images and $h_s = 16$ for 512×512 images.
	$h_r = \{7, 11, 15\}$	A color bandwidth parameter.
	$M = \{100, 500, 1000, 1500\}$	The smallest region size. h_r and M control the number of regions in the segmented image. The more an image deviates from the assumed piecewise constant model (e.g. the heavily texture background), larger values have to be used for h_r and M to discard the effect of small local variations in the feature space (e.g. $h_r = 15$, $M = 1500$).
mNC	$scale = \{0.4, 0.8\}$	We set a scale of an input image less than one in order to produce a segmentation result within reasonable computation time.
	$nseg = \{4, 6, 8, ..., 26\}$	We set a number of regions in a segmented image varying in a reasonable range. Martin et al. [90] suggested that the number of things in each image between 2 and 20 should be reasonable for any of images in the BSDS data set.

6.1.2 Experimental results

In this set of experiments, we run our segmentation combination algorithm on all three segmentation ensembles (i.e. FH, MS, mNC). The generalized median segmentation optimization criterion (5.6) proposed in the previous chapter will be used to automatically determine the optimal combined segmentation result, since it is proved to be the most effective criterion among the four criteria (presented in the previous chapter). Thus, the only pre-specified parameter of our combination procedure is a range of possible k values, $[k_{\min}, k_{\max}]$. This parameter is, however, not difficult for nonexpert user to specify and can be specified without any knowledge of underlying combination algorithm. In the extreme case, the possible value for $k_{\min}$ is equal one and $k_{\max}$ is equal n^2, where n is a total number of pixels in an image. For all experiments reported in this work a range of k values is set to [2,50].

Another requirement for the generalized median segmentation optimization criterion (5.6) is a distance function used in the optimization. Since NMI index and F-measure are used for assessing the quality of image segmentations, it is reasonable to optimize the objective criterion based on the same measure. Thus, for each segmentation ensemble, the (final) optimal segmentation results are selected using both NMI distance based optimization criterion and F-measure distance based optimization criterion. The optimal segmentation solution selected based on NMI distance will be evaluated using NMI index. Similarly, the optimal segmentation solution selected based on F-measure distance will be evaluated using F-measure. The experimental results are reported separately for each set of segmentation ensembles (i.e. FH, MS, mNC). Since the values of NMI index and F-measure lie in the range [0,1], NMI distance can be computed by 1.0-NMI index and, similarly, F-measure distance can be computed by 1.0-F-measure.

Figure 6.1(d)– 6.3(d) show examples of combined segmentation results produced by our method on FH, MS, and mNC segmentation ensembles, respectively. For comparison purpose we also show the input segmentation with the worst, median and the best evaluation values (column (a)-(c)). For each image, the first row shows the combined segmentation result which is determined based on NMI distance, as well as the worst, median and the best input segmentations. Similarly, the second row shows the segmentation results which are determined based on F-measure distance. Generally, we can observe a substantial improvement of our combination compared to the median input segmentation. These results demonstrate that we can obtain an "average" segmentation which is superior to the - possibly vast - majority of the input ensemble. This fact can also be illustrated by the plots shown in Figure 6.4. Each plot shows a per-image performance of the 300 images in the data set, compared

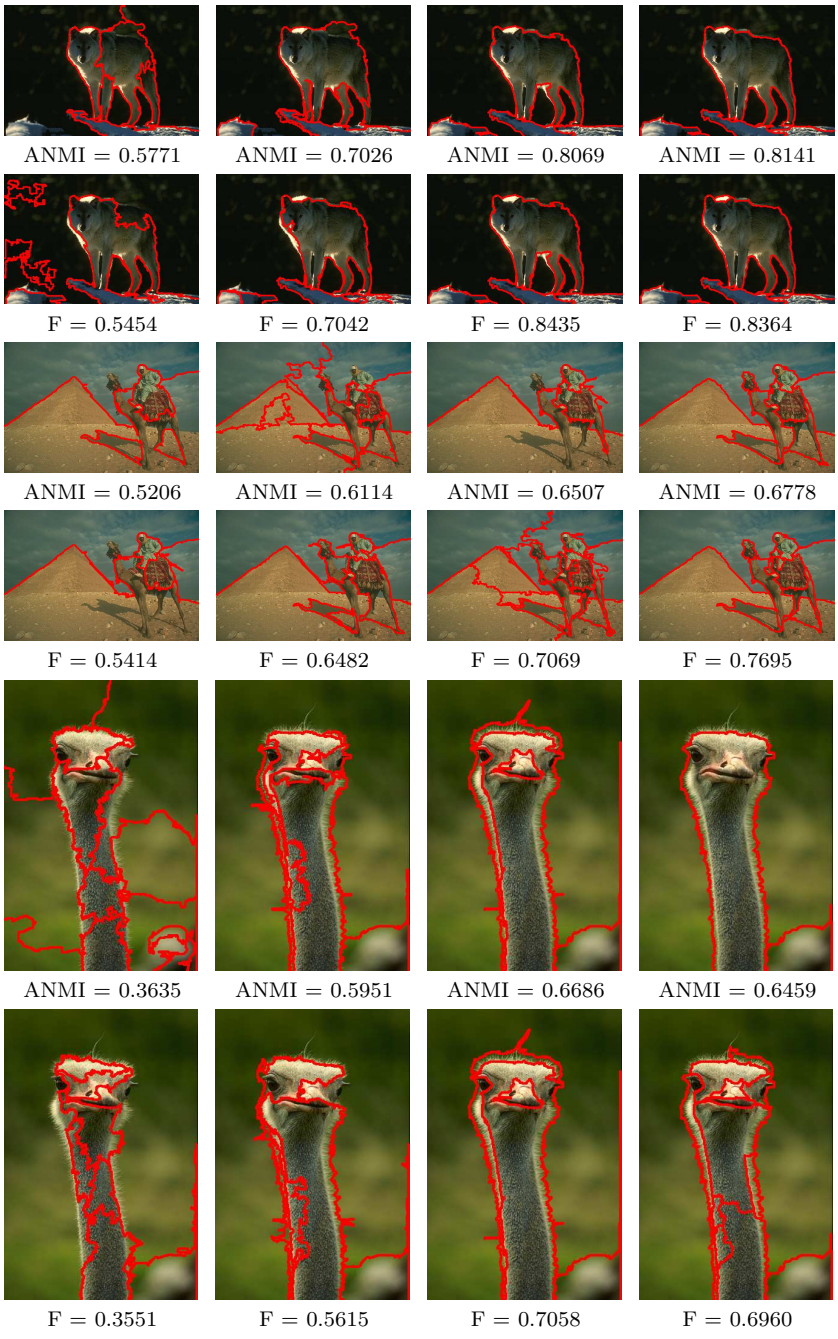


Figure 6.1. Parameter subspace sampling: combination segmentation results on FH ensembles. (a)-(c) Input segmentations with the worst, median and the best average NMI/F-measure values, respectively; (d) Combined segmentation.

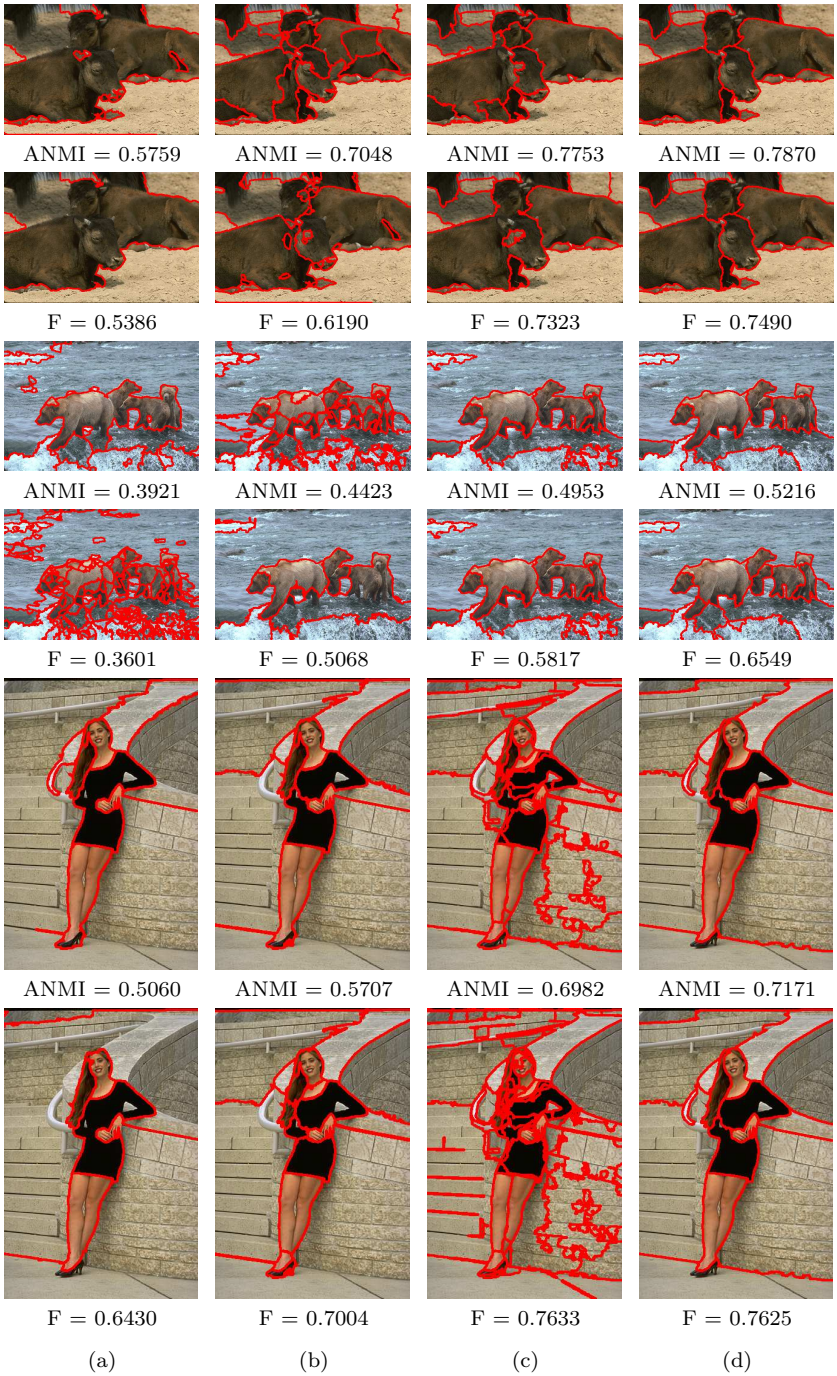


Figure 6.2. Parameter subspace sampling: combination segmentation results on MS ensembles. (a)-(c) Input segmentations with the worst, median and the best average NMI/F-measure values, respectively; (d) Combined segmentation.

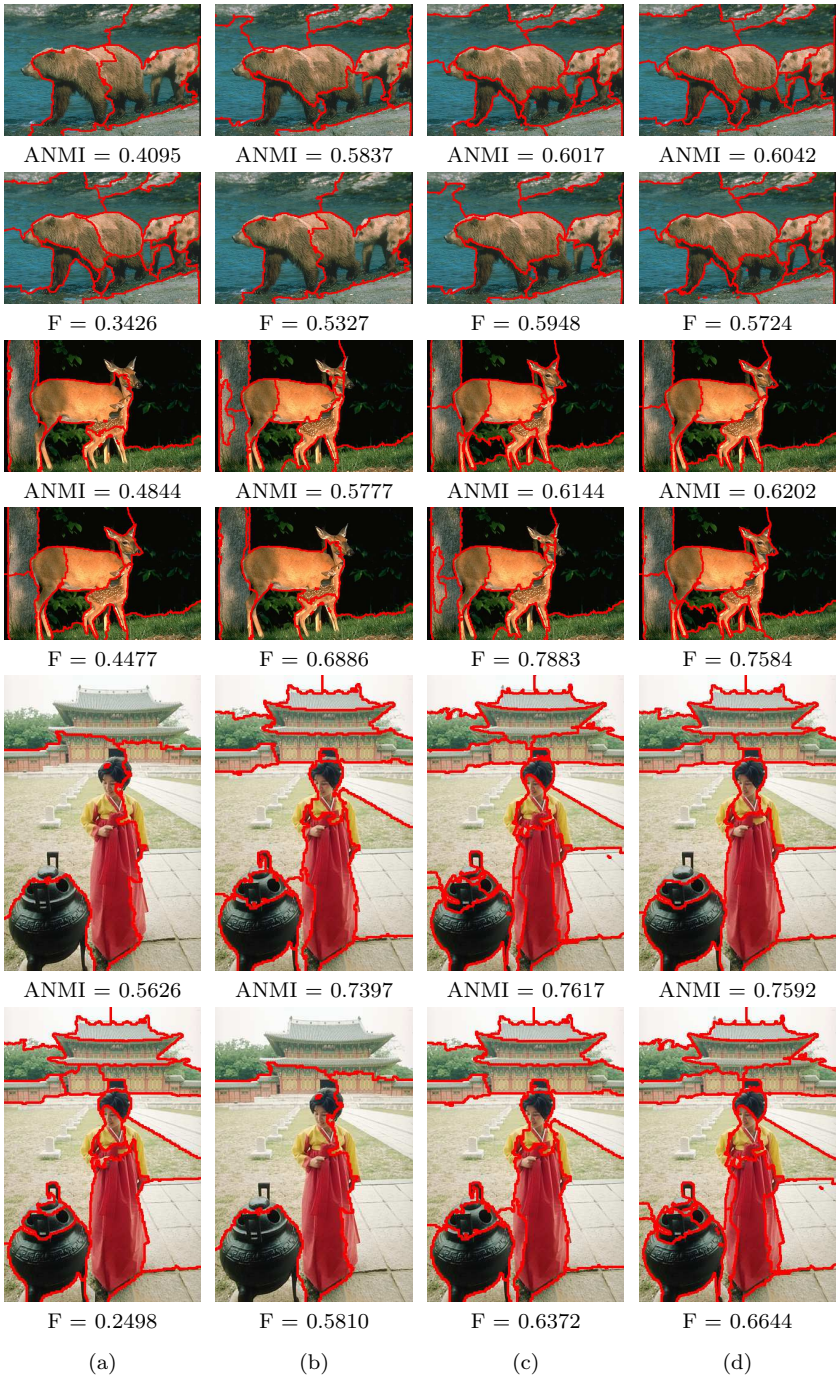
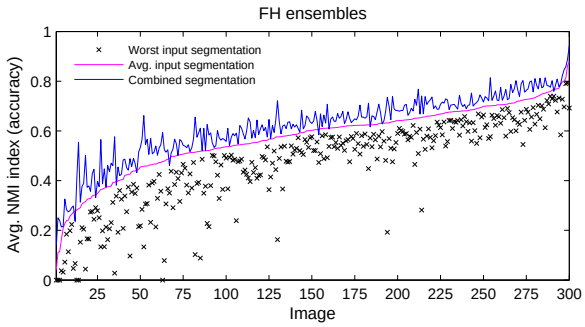
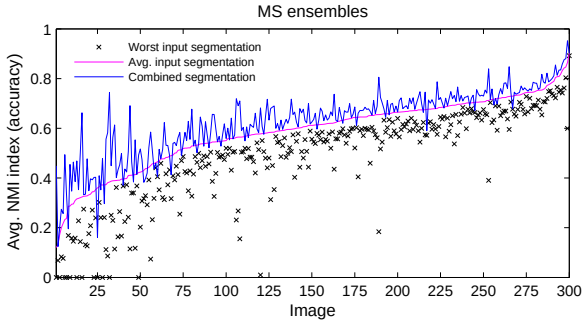


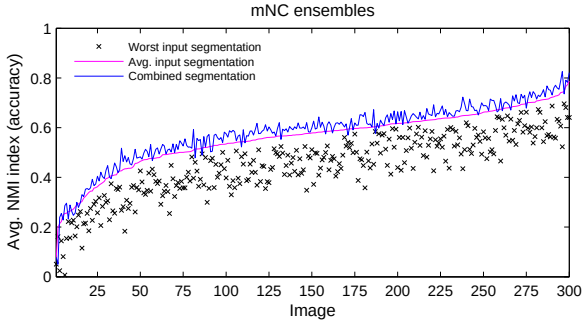
Figure 6.3. Parameter subspace sampling: combination segmentation results on mNC ensembles. (a)-(c) Input segmentations with the worst, median and the best average NMI/F-measure values, respectively; (d) Combined segmentation.



(a) FH ensembles

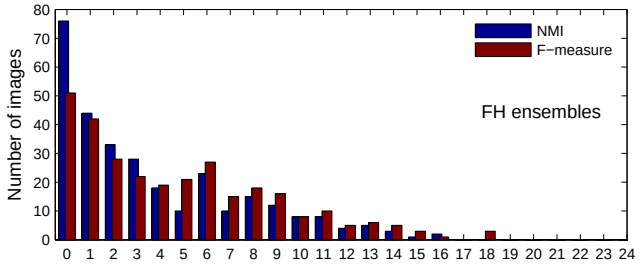


(b) MS ensembles

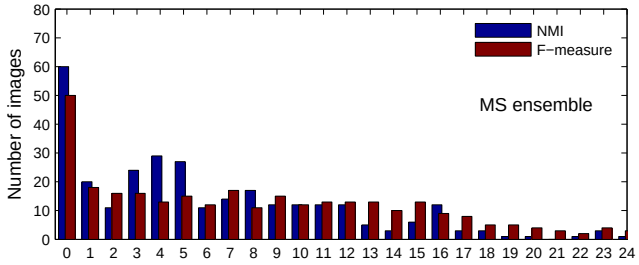


(c) mNC ensembles

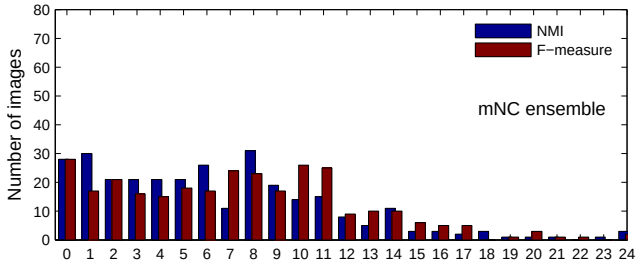
Figure 6.4. Comparison (per image): Average and worst input & combination result (in terms of average NMI values with respect to the ground truth).



(a) FH ensembles

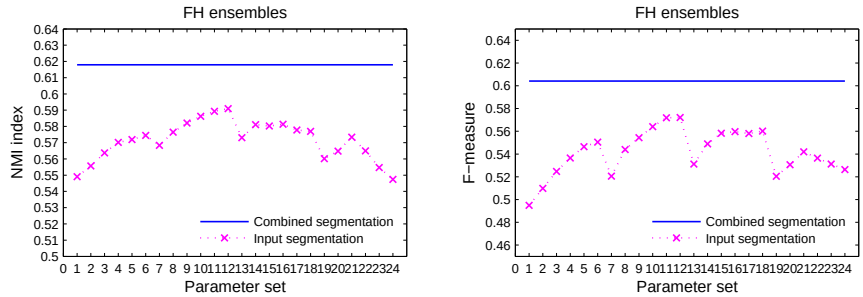


(b) MS ensembles

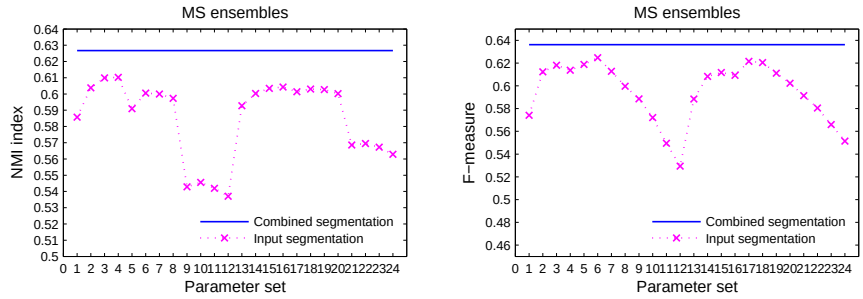


(c) mNC ensembles

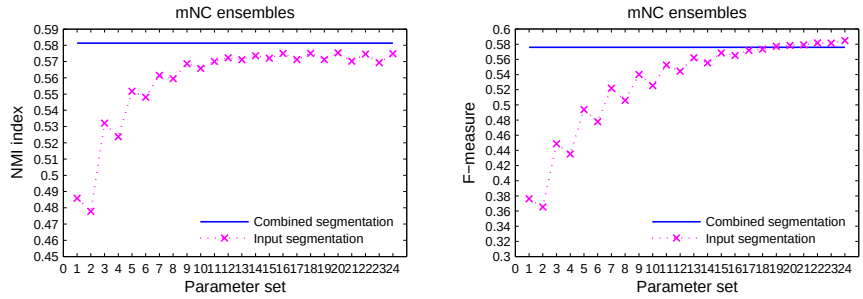
Figure 6.5. $f(n)$: Number of images for which the combination result is worse than the best N input segmentations.



(a) FH ensembles



(b) MS ensembles



(c) mNC ensembles

Figure 6.6. Average performance of combined results over 300 images for each individual parameter setting (in terms of the average NMI (left) and F-measure (right) values with respect to the ground truth).

Table 6.2. Segmentation combination versus base segmentation results over 300 images.

Data set		FH	MS	mNC
NMI	Combination	0.6179 ± 0.1322	0.6267 ± 0.1344	0.5813 ± 0.1235
	Base segmentation	0.5714 ± 0.1371	0.5851 ± 0.1405	0.5580 ± 0.1173
F-measure	Combination	0.6042 ± 0.1227	0.6362 ± 0.1218	0.5760 ± 0.1102
	Base segmentation	0.5414 ± 0.1190	0.5948 ± 0.1329	0.5278 ± 0.1048

with the worst and average inputs. In order to make the plot simpler and easier to observe, the performance values are plotted in increasing order of performance value of average inputs.

Moreover, in some cases the combined segmentation even outperforms the entire input ensemble. This case is confirmed by Figure 6.5, which shows a statistic $f(n)$, indicating the number of images among the 300 test images, for which the combination segmentation is worse than the n best input segmentations. Remarkably, the combination segmentation outperforms all 24 input segmentations in $f(0) = 76$ cases for FH ensembles, $f(0) = 60$ cases for MS ensembles, and $f(0) = 28$ cases for mNC ensembles (on NMI index). In the case of FH ensembles, for 70% (210) of all 300 test images, the goodness of our solution is beaten by at most 5 input segmentations only. In the cases of MS and mNC ensembles, for 71% (213) and 70% (210) of 300 images, the goodness of our solution is beaten by at most 8 input segmentations, respectively. These statistics are a clear sign of combination quality of our approach.

To provide additional empirical justification of our method, Figure 6.6 shows the average performance of all 300 images with regard to each of the 24 individual configurations (parameter settings). We also draw the blue line for the average performance of our combination approach of all 300 images. This implies that for all 24 parameter settings the combination approach always achieved improved results in average. This is true for all three sets of segmentation ensembles, except for mNC ensembles on F-measure.

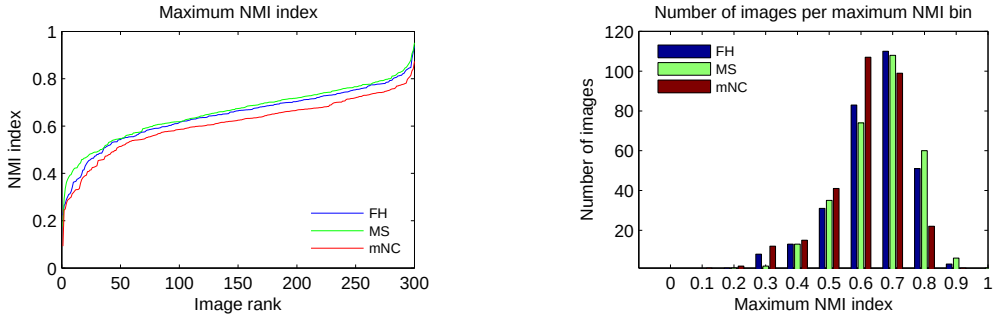
Table 6.2 summarizes the average performance of combination segmentations and baseline segmentations for all three segmentation ensembles. Among three sets of segmentation ensembles, it is obviously seen that the improvement of segmentation combination is the least for mNC ensembles. We conjecture that this is due to less diversity in the individual segmentations in the mNC ensemble. This conjecture will be examined in Section 6.1.4.

Another reason that explains the low improvement on mNC ensembles is a number of regions in the initial segmented results. The number of regions of segmentations in each mNC segmentation ensemble are forced to be $[4, 6, 8, ..., 26]$. Consequently, when selecting the optimal combined segmentation results using the generalized median segmentation criterion (5.6), which minimizes the sum of distances between the optimal segmentation to all segmentations in an ensemble, the number of regions in the optimal segmentation result mainly falls in the middle of the range $[4, 26]$, which often does not correspond well to the natural number of regions in a given input image. In contrast to the other two segmentation algorithms, FH and MS, allow each parameter configuration to determine its own number of regions in a resulting segmented image, which is more likely corresponding well to the natural number of regions.

It is also important to note that the choices of distance functions, used in the generalized median segmentation optimization criterion (5.6) for selecting the final optimal segmentation solution, is another key of success for our combination approach. The ability of distance functions in measuring similarity/dissimilarity between segmentations affects significantly the success of selecting the most optimal solution in a set of combination results. As shown in Figure 6.1– 6.3, NMI index and F-measure have its own preference to choosing the optimal segmentation solution, as well as the worst/median/best input segmentations. Even though many quantitative evaluation measures for image segmentations have been proposed over years, their behaviours and applicabilities on a variety of images remain unclear, and remain a potential problem in computer vision. In our work, we define the generalized median segmentation criterion (5.6) independent of the choices of distance functions, which provides the user opportunity to select a particular quantitative measure that best suits for a particular imagery data or a specific task of image segmentation.

6.1.3 Suitability of Parameter Ranges and Values

The ranges and values of baseline image segmentation parameters used in the experiments are empirically determined based on an intention of making as much as possible the correct segmentations within the chosen ranges. In this section we examine the suitability of our choices of parameter ranges and values of each segmentation algorithm that have been used in the experiments. We verify the suitability of chosen parameter ranges by examining the highest quality of segmentation results that we can achieve from each of segmentation algorithm for a given set of 24



(a) (b)

Figure 6.7. Maximum NMI value of each image obtained by each segmentation algorithm given the set of parameters. (a) The highest NMI values for each input image. (b) The number of images per maximum NMI index bin.

segmentation parameter combinations (see Table 6.1). The highest segmentation quality is computed by selecting the segmentation results (among 24 results) with the highest ANMI values comparing with the ground truth. The left plot in Figure 6.7 shows the maximum ANMI value on each segmentation algorithm. Note that the performance values are plotted in increasing order for each algorithm. Thus, the image rank on the x-axis may not represent the same image across algorithms. The plot shows that all of the algorithms have roughly equal ability to produce correct segmentations with the parameter setting chosen. The mNC algorithm has slightly lower performance than the other two. A histogram in the right plot of Figure 6.7 shows the number of images per maximum ANMI value bin, summarizing the same information in the left plot. Most segmentation results have ANMI values centered around 0.6 (for mNC) and 0.7 (for FH and MS) which demonstrates that all of the algorithms almost always have the potential to produce useful segmentation results. Thus, we can conclude that our choices of parameter ranges and values for each algorithm are reasonable.

6.1.4 Analysis of Diversity vs. Accuracy

In this section we study the impact of diversity and quality of the individual segmentation solutions on the final combined segmentation performance. The objective of this study is to show that diversity and quality of the base segmentations have proven to be a key element in increasing segmentation combination performance. In this study we firstly examine the diversity and accuracy of the base segmentation ensembles and, then, examine the influence of diversity and accuracy of the base segmentation ensembles on the performance of segmentation combination.

Diversity and Accuracy of Ensembles

We perform the ensemble diversity analysis following the approach taken by Fern and Brodley [39]. The diversity of an ensemble can be measured by calculating the NMI distance (e.g. 1.0-NMI) between each pair of segmentation solutions in the ensemble. To obtain a single accuracy measure for each pair, we average their NMI values as computed between each of the two segmentation solutions and the ground truth segmentations. In detail, for each set of segmentation ensembles (i.e. FH, MS, and mNC) we ran the following procedure:

(1) Repeat the following steps 300 times for all 300 ensembles of 300 images in the BSDS data set

(2) For all $i, j = \{1, 2, ..., 24\}$ and $i \neq j$

(2.1) Compute the pairwise diversity measures between each pair of segmentations in an ensemble:

$$D_{NMI} = 1 - \phi^{(\mathrm{NMI})}(S_i, S_j)$$

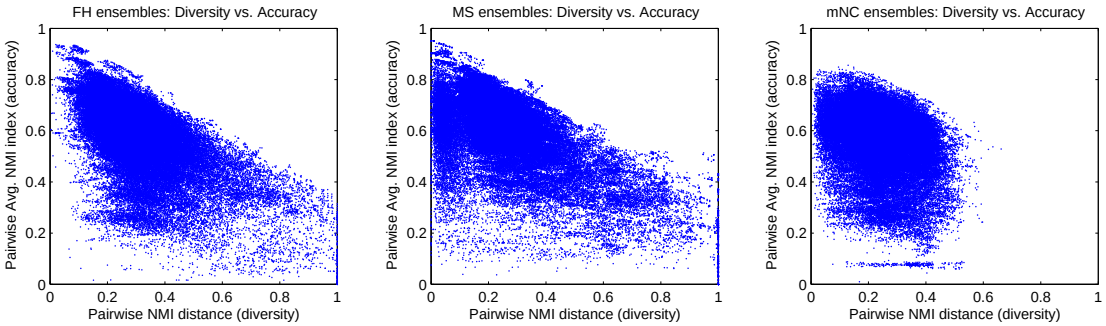
(2.2) Compute the average accuracy for each pair of segmentations against a set of ground truth, $\mathcal{S}$:

$$Acc_{NMI} = \frac{1}{2}[\phi^{(\mathrm{ANMI})}(S_i, \mathcal{S}) + \phi^{(\mathrm{ANMI})}(S_j, \mathcal{S})]$$

The graphs plotted the diversity (D_{NMI}) versus accuracy (Acc_{NMI}) for each pair of initial segmentations of all 300 ensembles for each of FH, MS and mNC algorithms are shown in Figure 6.8. In the diversity-accuracy diagram, the diversity is maximized when the D_{NMI} value between two solutions (shown on the x-axis) is one, as well as the accuracy which is maximized when maximizing the Acc_{NMI} values. Fern and Brodley [39] suggest that higher diversity among ensemble members tends to produce higher performance gain. Thus, a desirable location of our points is close to the right-hand top corner of a graph which has high both accuracy and diversity.

In Figure 6.8 each of the three ensemble datasets shows different behavior. The first two graphs show that FH and MS form a set of segmentation ensembles with a wide range of quality and diversity, where FH ensemble has slightly lower quality than MS ensemble. In contrast, mNC ensemble (right) has much lower diversity and lower quality than FH and MS.

In all cases, it is shown that the accuracy of the ensemble decreases as the diversity increases. This can be explained by the nature of image segmentation problem.



(a) FH ensembles (b) MS ensembles (c) mNC ensembles

Figure 6.8. The diversity-accuracy diagram for three segmentation ensembles.

The good segmentations of the same image are alike, while the bad segmentations are arbitrarily bad in its own way. Thus, it is quite difficult to obtain the ensemble with high accuracy and high diversity.

Influence of Diversity and Accuracy on Combination

In order to study the impact of diversity and accuracy of ensemble on the combination results, we consider the accuracy of a combination result and a segmentation ensemble with respect to the diversity of the ensemble. For each segmentation ensemble, the single diversity measure is computed by averaging the pairwise diversity measures of all pair of segmentations in the ensemble:

$$D_{NMI}^{\text{avg}} = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (1 - \phi^{(\text{NMI})}(S_i, S_j)), \text{ for } N = 24,$$

and the single accuracy measure is computed by averaging a pairwise accuracy between each segmentation solution in an ensemble with its corresponding ground truth, $\mathcal{S}$:

$$Acc_{NMI}^{\text{avg}} = \frac{1}{24} \sum_{i=1}^{24} \phi^{(\text{ANMI})}(S_i, \mathcal{S}).$$

For each segmentation algorithm (i.e. FH, MS, and mNC), we compute D_{NMI}^{avg} and Acc_{NMI}^{avg} 300 times for all 300 ensembles of 300 images in the BSDS data set. We then plot D_{NMI}^{avg} and Acc_{NMI}^{avg} of 300 ensembles for each segmentation algorithm as shown in the diversity-accuracy graphs in Figure 6.9 as a magenta line with cross mark. In each graph we also show the accuracy (in terms of average NMI comparing

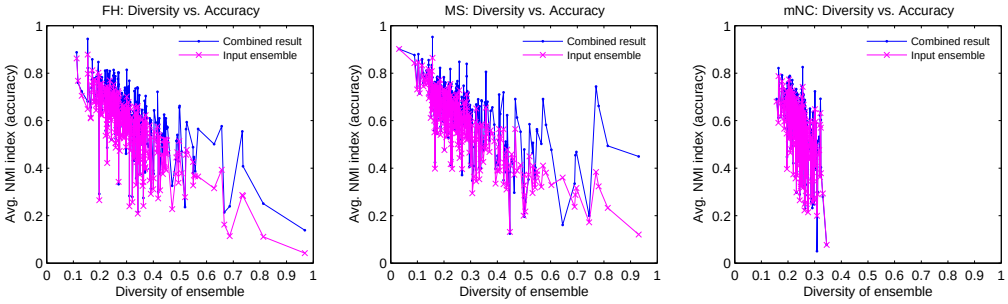
with the ground truth) for the combination solutions for each corresponding ensemble (a blue line with dot mark) as reported in Figure 6.4. The axes for each kind of plot have been kept constant so plots can be compared easily.

We see evidence that high diversity leads to greater improvements in the quality of combination results over an input ensemble (i.e. the further the blue line far away above from the magenta line, the higher the quality of the combination result over the quality of the input ensemble). Specifically, we see the least improvement of the combination result over an input ensemble for the mNC data set, which has significantly lower diversity than the other two. The average percentage of improvement at each diversity level is summarized in the histogram in Figure 6.10. In all cases, the higher the diversity of an ensemble, the greater gains the improvement of the combination result. These results suggest that the ensemble combination performance is strongly influenced by the diversity of the individual segmentation solutions. If the individual segmentation solutions have little diversity, then not much leverage can be obtained by combining them.

However, the quality of the individual segmentation solutions limits the performance of ensemble combination. From Figure 6.9, we compute the average accuracy of combination results at different levels of diversity and draw the histograms as shown in Figure 6.11. We see that the accuracy of combination results decreases as the diversity increases, even though the percentage of improvement increases as the diversity increases. This is because when the diversity of ensemble increases, the accuracy of ensemble decreases (as shown in Figure 6.8 and 6.9). However, note that in the case of MS, the accuracy of combination results does not monotonically decrease like in the other two cases. The average accuracy increases from diversity level 0.5 and 0.7 to diversity level 0.6 and 0.8, respectively. One possible reason is that the quality of the MS ensembles is higher than the quality of FH ensembles at high level of diversity. This may enhance the combination performance on MS ensembles at high diversity level. These results suggest that the ensemble performance is strongly influenced by both the quality and the diversity of the individual segmentation solutions.

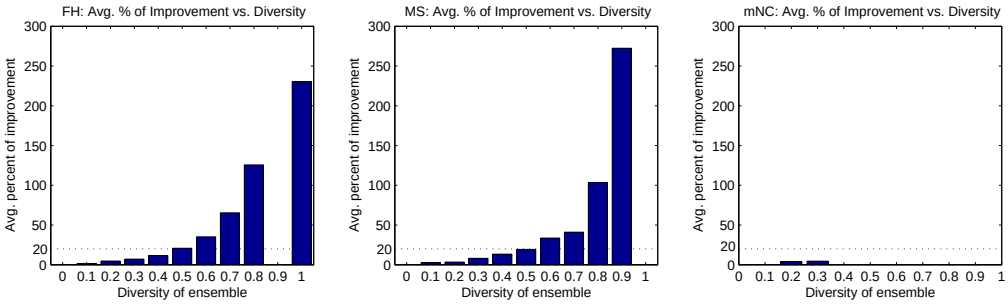
6.2 Multiple Segmentation Algorithm Combination

Despite the large number of segmentation techniques presently available [45, 47, 86, 102, 149], no general methods have been found that perform adequately across a



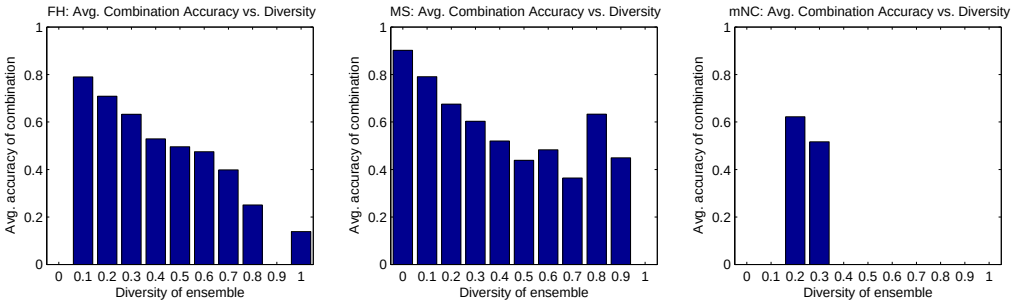
(a) FH ensembles (b) MS ensembles (c) mNC ensembles

Figure 6.9. The diversity-accuracy diagram for three data sets.



(a) FH ensembles (b) MS ensembles (c) mNC ensembles

Figure 6.10. Average percentage of improvements at different levels of diversity.



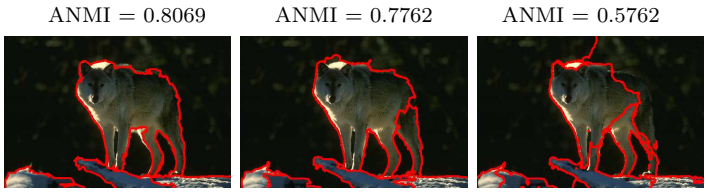
(a) FH ensembles (b) MS ensembles (c) mNC ensembles

Figure 6.11. Average accuracy of combination solutions at different levels of diversity.

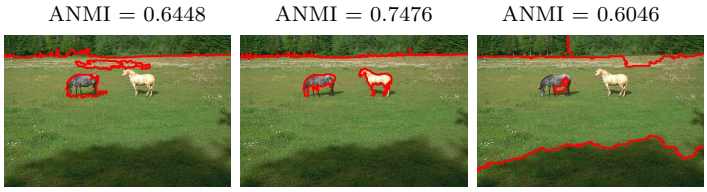
diverse set of imagery [8]. This situation is firstly due to the high variations of the input images, whose characteristics, such as contrast, noise and illumination, etc., may change greatly. Secondly, due to the ill-posed nature of image segmentation problem, defining meaningful constraints or an objective function for classifying pixels into regions is typically specific to the application domain. Consequently, the properties/behaviors of different segmentation algorithms differ due to the objectives they try to satisfy. Figure 6.12 illustrates the different behaviours of state-of-the-art segmentation algorithms. Some algorithms might perform well in specific images but not in others. Each column shows the best segmentation results of a given image for each of FH, MS and mNC algorithms. The best segmentation result of each given input image for each algorithm is selected from 24 segmentation results (according to 24 parameter settings defined in the previous section) with the highest ANMI value (as compared to the ground truth). Homogeneous regions and smooth boundaries in segmented images are constructed with respect to specific constraints used in each particular algorithm. Segmentation results show differences in terms of the number of segmented regions, sensibility to low local variation (a) and sensibility to small structures (b). The FH algorithm performs well for the first input image, while none of its 24 parameter settings can yield a good result for the rest input images (see Figure 6.12(a)). Similar for the MS algorithm, it performs well for the second input image but none of its 24 parameter settings can yield a good result for the rest (see Figure 6.12(b)), whereas the mNC algorithm performs well for the last input image (see Figure 6.12(c)).

Hundreds of segmentation techniques are present in the literature, but there is no single method which can be considered good for all images, nor are all methods equally good for a particular type of image [102]. In particular, the potential problem is that it is not easy to know the optimal algorithm for one particular image. Automated selection of an optimal algorithm according to image characteristics and/or the application need is a real challenge. Zhang and Luo [150] have attempted to construct an automated algorithm selection system by using the heuristic knowledge (which is obtained by objective evaluation of available segmentation algorithms in a number of situations) and feedback of segmentation evaluation. In other words, the algorithm selection is made according to the properties of the segmentation algorithms and images to be segmented.

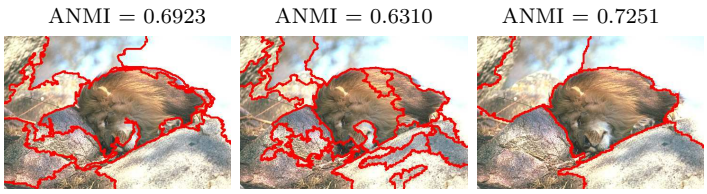
Yong et al. [144, 145] proposed a framework of algorithm selection system based on learning scheme. During training, both the performance ranks of candidate algorithms on every image and image features are used to train a predictor. Then, the performance ranks of all candidates will be predicted according to image features.



(a) Parameter setting for FH:$\sigma = 0.6, k = 700, M = 1500$; MS:$h_s = 8, h_r = 11, M = 1500$; mNC:$scale = 0.8, nseg = 10$



(b) Parameter setting for FH:$\sigma = 0.6, k = 700, M = 1500$; MS:$h_s = 16, h_r = 15, M = 500$; mNC:$scale = 0.4, nseg = 4$



(c) Parameter setting for FH:$\sigma = 0.9, k = 500, M = 1500$; MS:$h_s = 8, h_r = 11, M = 1500$; mNC:$scale = 0.4, nseg = 6$

Figure 6.12. Illustration of the problem of segmentation algorithm selection. Segmented images are computed by FH, MS, and mNC segmentation algorithms (left/middle/right). The comparative performance of different segmentation algorithms can vary significantly across images.

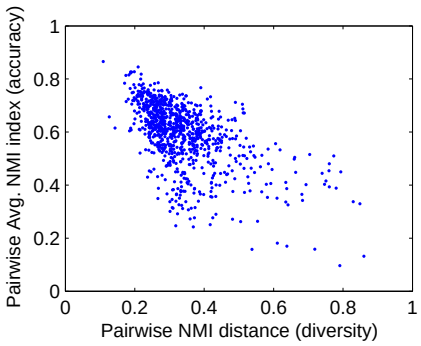


Figure 6.13. The diversity-accuracy diagram of 300 segmentation ensemble.

Finally, the algorithm with the highest rank will be regarded as optimal and applied to the image. In this framework, the histogram is used as image feature, the number of misclassified pixels and computation expenses are used to facilitate interactive segmentation evaluation, and Principle Components Analysis and Support Vector Machine are used to construct the predictor. Recently, a similar but more sophisticated learning-based framework is proposed by Shah [120]. Shah has proposed a probabilistic framework based on Bayesian theory for the performance prediction and selection of an optimal segmentation algorithm. Within the developed framework, the knowledge about each candidate algorithm's capability on input image features is learnt from a limited sample of images representing the context variety and a measure of candidate algorithms' performance. When this knowledge is put to use, features extracted from each new input image are used by the predictor and the performance of all algorithms on that image is predicted without actually running any of the candidate algorithms. The algorithm corresponding to the best performance is selected as optimal and applied to that image. The framework proposed by Shah differs from the framework proposed by Yong et al. in that in Yong's framework [144, 145], the interactive segmentation evaluation of segmentation results produced by candidate segmentation algorithms must be done by a user.

Martin and Thonnat [93] proposed a unified framework for learning of adaptive image segmentation methods which illustrates how a knowledge-based framework can be augmented with learning capabilities. In this framework the learning process involves three stages: extracting optimal parameters for each image of the training dataset, ranking algorithms to construct a case base, and training a neural network to select algorithms and their parameters for novel images. The basic concept of this framework is different from the first two works in that it provides a mechanism for tuning the parameters of candidate segmentation algorithms (in the first step of learning process).

Even though these approaches based on machine learning techniques and learning-based system have shown impressive results on a particular application/image domain, these methods require either the assumption of ground truth segmentations or the human intervention in a training process.

In order to tackle this segmentation algorithm selection problem, we neither explicitly select the optimal segmentation algorithm for a particular image nor are interested in optimizing a segmentation algorithm for a given task. Instead, we attempt to effectively utilize the existing (efficient) segmentation algorithms by postulating that "Instead of looking for the best segmenter which is hardly possible on a per-image basis, now we look for the best segmenter combiner". The rationale

behind this idea is that while none of the segmentation algorithms is likely to segment an image correctly, we may benefit from combining the strengths of multiple segmenters. For this purpose we may apply various segmentation methods (each perhaps run with multiple parameter sets) to build a segmentation ensemble. The advantages of our approach over above mentioned approaches are that our approach requires no assumption of ground truth segmentations and no human intervention in a framework operation.

6.2.1 Segmentation Ensemble Generation

In this experiment, a segmentation ensemble is generated using three different segmentation methods, namely FH, MS and mNC segmentation algorithms. A parameter setting for each segmentation algorithm is specified by choosing the best one from the 24 sets of parameter values described in Table 6.1, namely, a parameter setting with the highest average performance for all images in the BSDS data set. For the FH algorithm: $\sigma = 0.9, k = 300, M = 1500$, for the MS algorithm: $h_s = 8, h_r = 7, M = 1500$, and for the mNC algorithm: $scale = 0.8, nseg = 22$.

We run these three segmentation algorithms with their best parameter setting on each of 300 images in the BSDS data set to form a segmentation ensemble for each image. Each segmentation ensemble consists of three segmentation results. The diversity and accuracy of all 300 segmentation ensembles is shown in Figure 6.13. The three segmentation algorithms form a set of segmentation ensembles with moderate diversity and relatively high quality.

6.2.2 Experimental Results

Our combination algorithm is used to combine multiple segmentations of three different segmenters: FH, MS and mNC. The combination algorithm is performed on all 300 segmentation ensembles of 300 images in the dataset. Once again, the generalized median segmentation optimization criterion (5.6) is applied to choose the optimal segmentation result from a set of combined segmentations with the different number of $k \in [2, 50]$

Visual samples of segmentation combination results are shown in Figure 6.14(a), while, for a comparison purpose, Figure 6.14(b)-(d) show all three baseline input segmentations produced by FH, MS and mNC algorithms, respectively. It is obvious that the segmentations given by each baseline algorithm have different natures, depending on the specific underlying segmentation criterion it used. Particularly,

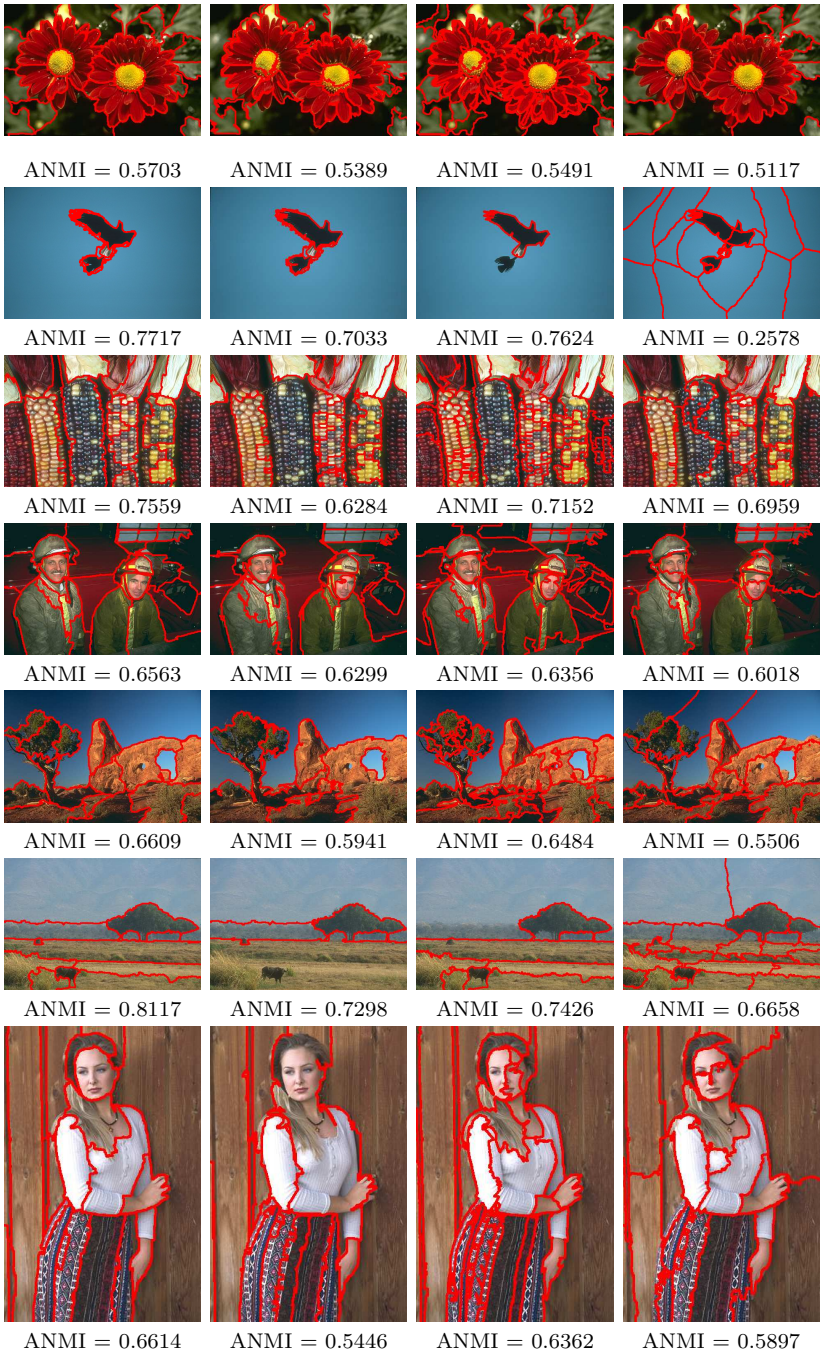


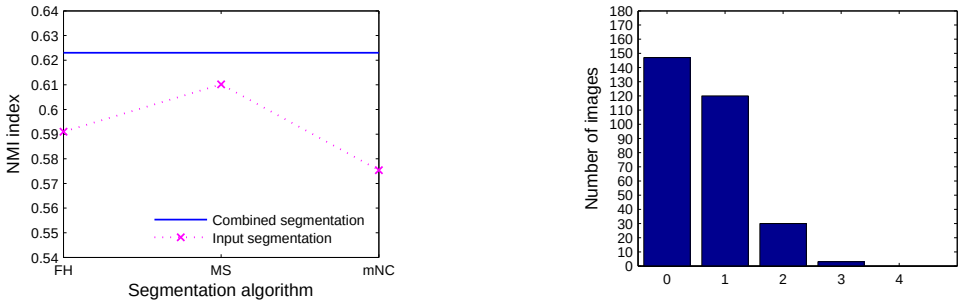
Figure 6.14. Segmenter Combination: (a) Segmenter combinations (b)-(d) Three input segmentations computed by running combination algorithm on FH, MS, and mNC ensembles, respectively.

an individual run of these baseline algorithms often produces less satisfactory results. On inspecting these results, we observe that our combination algorithm is able to uncover some parts of the true natural structure in the input image, even though these parts are not present in the segmentation ensemble. An obvious example of this argument can be seen in the last segmentation result. Our combination algorithm can successfully extract the face of a woman, even though none of the baseline segmentation algorithms does. This situation can also be observed in different parts of the image, as well as in other sampled segmentation results. These results clearly support our assumption on that we may benefit from combining the strengths of such multiple segmentation algorithms, even though none of them is likely to segment an image correctly.

Another key success of our combination approach is the use of *generalized median concept* to determine an optimal segmentation solution from a set of combination results, where an optimal segmentation solution is the one that minimizes the sum of distances to all segmentations in an ensemble. Thus, when the quality of the majority of segmentations in an ensemble is relatively good, we always achieve an improved combined segmentation solution. An obvious example of this situation can be seen in the second and the sixth rows of Figure 6.14. The combination results are more similar to the majority of segmentation solutions in an ensemble (i.e. the segmentations produced by the FH and the MS algorithms) and less affected by the outlier segmentations[1] (i.e. the segmentation produced by the mNC algorithm).

In addition, the experimental results also demonstrate that our combination algorithm is able to gain an improvement of segmentation results, even when the size of segmentation ensemble is small (i.e. 3 segmentations per ensemble). The improvement of our combination approach can be confirmed by the plots shown in Figure 6.15. Figure 6.15(a) shows the average performance of all 300 images for each baseline segmentation algorithm in comparison with the average performance of our approach. This plot implies that for all three baseline segmentation algorithms the combination approach always achieved improved results in average. A histogram shown in Figure 6.15(b) shows a statistic $f(n)$, indicating the number of images among the 300 test images, for which the segmenter combination segmentation is worse than the n best input segmentations. Remarkably, the segmentation combination approach outperforms all three input segmentations in f(0) = 147 cases (49%). For 89% (267) of all 300 test images, the goodness of our segmenter combination approach is beaten by the one best input segmentation only.

[1]Outlier segmentation is a segmentation that is far away from the majority of segmentations in a set, commonly with large-scale measurement error.



(a) (b)

Figure 6.15. (a) Average performance of combined results over 300 images for each baseline segmentation algorithm (in terms of the average NMI values with respect to the ground truth). (b) $f(n)$: Number of images for which the segmenter combination result is worse than the best N input segmentations computed by running combination algorithm on FH, MS, and mNC ensembles.

Given the fact that we do not know the optimal segmentation algorithm for a particular image in advance (see Figure 6.12), the comparative performance of our approach is remarkable and reveals its potential in dealing with the difficult problem of optimal algorithm selection even without ground truth. In fact our combination approach is even superior to conventional algorithm selection approaches, since in many cases it can provide better quality segmentations beyond what can be provided by the best segmenter in an ensemble.

6.3 Multiple Image Transformations

In this section we propose to improve the quality of image segmentations by making use of image transformations. This approach is different from the approaches presented so far in that the variation in segmentation ensembles are created by varying the representations of an input image given the same segmenter, instead of varying the segmenters (e.g. varying the algorithm parameters or applying multiple segmentation algorithms) given the same input image. This approach is based on the fact that most segmentation algorithms existing in the literature are image dependent. Local variations of the image may change dramatically the segmentation results. We have a conjecture that a combination of such different segmentation solutions resulting from segmenting different transformations of an input image will be able to improve the segmentation performance over the performance of a single segmentation solution of the original input image.

(a) Rotation (b) Shearing (c) Perspective

Figure 6.16. Examples of different image transformations.

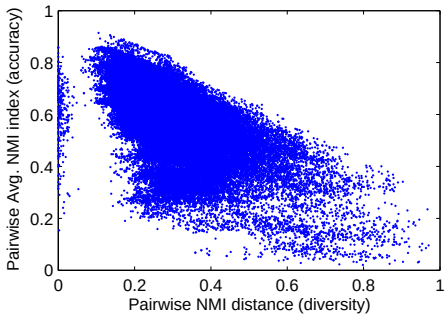


Figure 6.17. The diversity-accuracy diagram of 300 segmentation ensembles. A segmentation ensemble is formed by running FH segmentation algorithm on multiple transformations of the original input image.

6.3.1 Segmentation Ensemble Generation

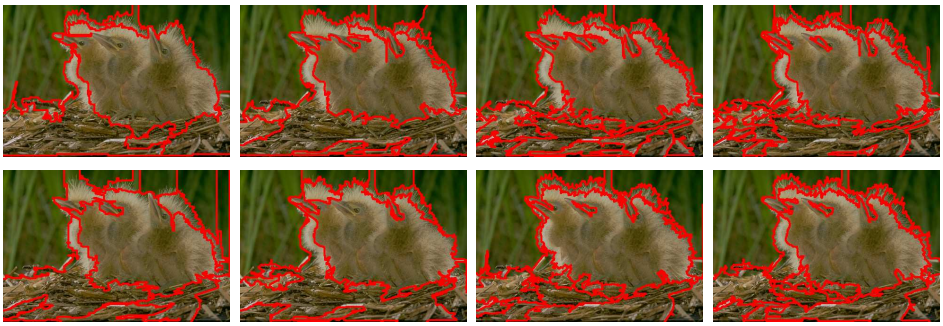
In this experiment, a variety of image transformations, such as geometric transformations, affine transformations, and perspective transformations, are applied for creating diversity in a segmentation ensemble.

- Geometric transformation - the transformation that includes rotation and scaling.
- Affine transformation - the transformation that includes shearing. Straight lines remain straight, and parallel lines remain parallel, but rectangles might become parallelograms.
- Perspective transformation - transformation in which straight lines remain straight but parallel lines converge toward vanishing points.

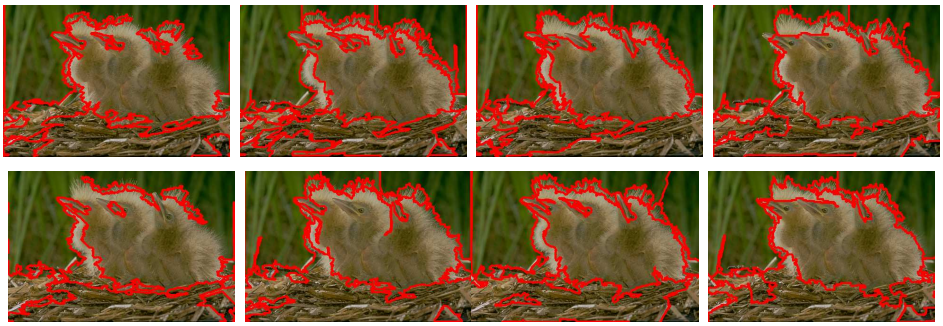
Figure 6.16 shows examples of different image transformations.

(a) Original image

(b) Rotation



(b) Scaling



(c) Shearing

(d) Perspective

Figure 6.18. Example of 25 segmentations in a segmentation ensemble resulting from segmenting different transformed images.

Table 6.3. Summary of 24 image transformations.

Transformation	Description
Rotation	270° rotation with Nearest-neighbor interpolation and
(4 transformations)	45° rotation with Nearest-neighbor, Bilinear, and Bicubic interpolations
Scale	Transformation parameter $S = [s_x, s_y]$, where
(8 transformations)	s_x specifies the scale factor along the x axis,
	s_y specifies the scale factor along the y axis.
	$S_1 = [1, 0.33]$, $S_2 = [1, 0.67]$, $S_3 = [1, 1.33]$, $S_4 = [1, 1.67]$
	$S_5 = [0.33, 1]$, $S_6 = [0.67, 1]$, $S_7 = [1.33, 1]$, $S_8 = [1.67, 1]$
Shear	Transformation parameter $A = [s_x, s_y, sh_x, sh_y]$, where
(8 transformations)	sh_x specifies the shear factor along the x axis,
	sh_y specifies the shear factor along the y axis.
	$A_1 = [1, 1, 0.5, 0]$, $A_2 = [1, 1, 1, 0]$, $A_3 = [1, 1, -0.5, 0]$,
	$A_4 = [1, 1, -1, 0]$, $A_5 = [2, 0.5, -0.5, 0]$, $A_6 = [2, 0.5, 0.5, 0]$,
	$A_7 = [0.5, 2, -0.5, 0]$, $A_8 = [0.5, 2, 0.5, 0]$
Perspective	Set an input coordinate system so that the input image fills the unit square with vertices (0,0), (1,0), (1,1), (0,1) and then transform the image into the quadrilateral with a set of vertices P.
(4 transformations)	$P_1 = [(0.2, 0), (-1, 1), (0.8, 0), (2, 1)]$,
	$P_2 = [(-1, 0), (0.2, 1), (2, 0), (0.8, 1)]$,
	$P_3 = [(0, 0.3), (0, 0.7), (1, 0), (1, 1)]$,
	$P_4 = [(0, 0), (0, 1), (1, 0.3), (1, 0.7)]$

A segmentation ensemble consists of 25 segmentation results: 24 segmentations resulting from segmenting 24 transformations of the input images plus one segmentation resulting from segmenting the original input image. 24 transformations include 4 rotations, 8 scaling, 8 shearing and 4 perspective transformations. The details of 24 transformantions are listed in Table 6.3. The FH segmentation algorithm with a parameter setting $\sigma = 0.8, k = 300, M = 500$ is used to segment the images. This parameter setting is chosen based on its highest average performance over all images in the dataset. Examples of different segmentations of different transformed images are presented in Figure 6.18.

Surprisingly, multiple transformations of original input image are able to form a set of segmentation ensembles with moderate diversity, which is much more diverse than segmentation ensembles that are generated by the mNC segmentation algorithm with multiple parameter values (see Figure 6.8(c)). The diversity and accuracy of all 300 segmentation ensembles is shown in Figure 6.17.

6.3.2 Experimental Results

We run our proposed combination algorithm on all 300 segmentation ensembles of 300 images in the dataset. The generalized median segmentation optimization criterion (5.6) based on NMI distance is applied for selecting the final optimal segmentation result from a set of combined segmentations with different number of $k \in [2, 50]$. Then, NMI index is used for quantitatively assessing the quality of segmentation results against the ground truth.

The performance of our segmentation combination approach is reported in comparison with performance of single segmentation of the original image. Some samples of combination results comparing with single results of the original image are shown in Figure 6.19. Based on visual judgment, combination results seem to have better quality than segmentations of the original image, even though in some cases the quantitative evaluating values of the combination results are equivalent or little worse than the segmentations of the original images. It is obviously seen that combined segmentations have smoother region boundaries and have no small elongate regions along the region boundaries.

In most cases, we can achieve the improvement of segmentation results. Figure 6.20 shows the performance of combination segmentations over the performance of segmentations of the original images. Again, to make the plot simpler, the average NMI values are plotted in increasing order of the average NMI values of the original input segmentations. In this plot we can observe a substantial improvement of our combination results compared to the original input segmentation (i.e. most of the blue markers lie above the magenta line). 86% of combination results (258 of 300 test images) obtain higher average NMI values than the segmentations of original input images. These results demonstrate the advantage of our combination approach to overcoming the imperfections of using a single segmentation algorithm with a single parameter.

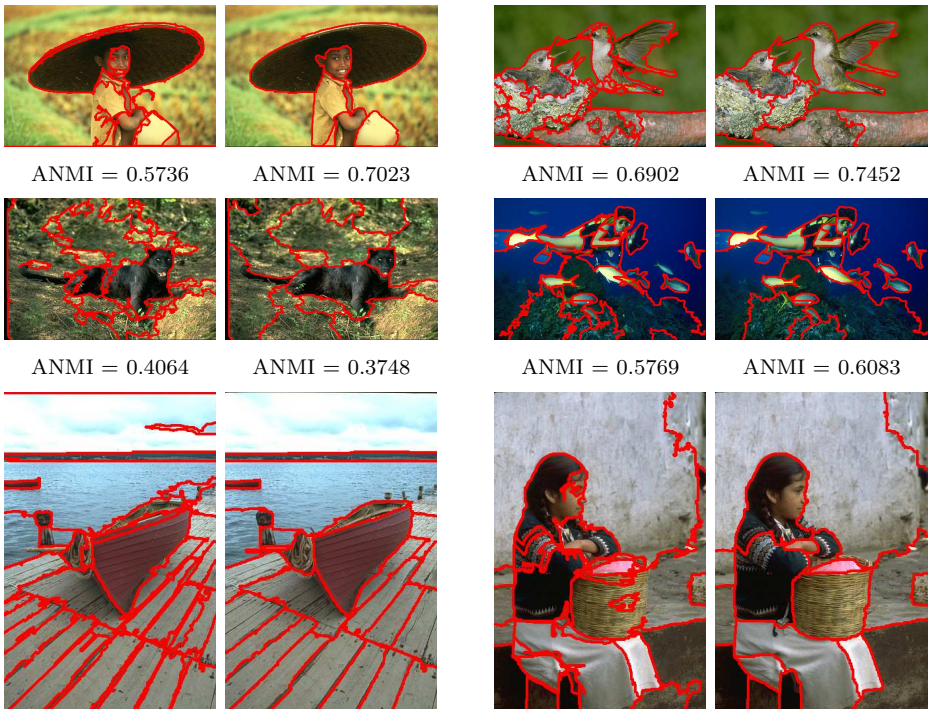


Figure 6.19. Multiple image transformation combination: (a) and (c) Segmentation results of the original input image computed by FH segmentation algorithm. (b) and (d) Combined segmentation results computed by our segmentation combination algorithm.

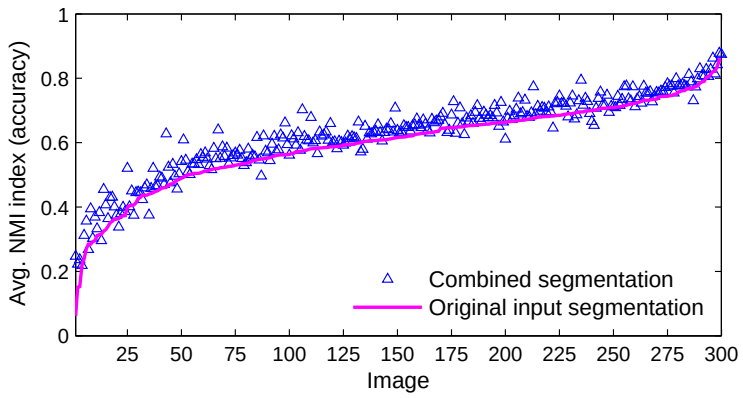


Figure 6.20. Average performance of combined results over 300 images comparing with the performance of segmentations of the original images. The ANMI values are plotted in increasing order according to the segmentation performance of the original images.

6.4 Discussion and Conclusion

On observing the experimental results of the three scenarios of segmentation ensemble generations, we can make the following conclusions. Given an input image, the segmentation results change with the different parameter values of a single segmentation algorithm, and different segmentation algorithms create different segmentations with different natures. However, such different segmentations may uncover some partial/different region structures which complement one another. In other words, each provides complementary sources of information about region membership. Combining such different segmentations can thereby lead to more accurate, robust and reliable of segmentation result.

The difficult image segmentation problem has various facets of fundamental complexity. In this chapter some of these segmentation problems have been addressed and carried out through the three segmentation ensemble generation scenarios: Parameter subspace sampling, Multiple segmentation algorithms, and Multiple image transformations.

The parameter subspace sampling approach concerns with the problem of optimal parameter selection, whereas the multiple segmentation algorithm approach concerns with the problem of selecting the optimal segmentation algorithm for a particular image. We have investigated the two approaches using three state of the art image segmentation algorithms as a baseline segmentation. In these frameworks we do not explicitly determine the optimal parameter setting/segmentation algorithm for a particular image. Instead, we try to reach an optimum output of the segmentation ensemble by means of generalized median concept. In all cases, we show that without knowing the optimal parameter setting/segmentation algorithm for a particular image in advance, the comparative performance of our approach is remarkable and reveals its potential in dealing with the difficult problem of parameter/segmentation algorithm selection without ground truth.

For the multiple image transformation approach, we propose an alternative way in dealing with the imperfections of segmentation algorithms by combining the principle of segmentation combination with image transformation techniques. This approach takes advantage of the disadvantage of a segmentation algorithm in that most segmentation algorithms are typically sensitive to the change in local variation in an input image. Transforming the input image may change greatly in a segmentation result. Surprisingly, the multiple transformation approach is able to create a moderate-diverse segmentation ensemble, and combining such an ensemble is able to improve the quality of segmentation result computed from a single run of

segmentation algorithm on an original input image (which might be of rather poor quality).

Even though in many cases the developed segmentation combination framework did not provide the superior segmentation quality over the best input segmentation, it did guarantee to produce the segmentation result with higher or equal quality to the average input segmentation. These results are indicative of the effectiveness of our combination framework to achieve statistically significant performance improvement over a segmentation ensemble.

We have also analyzed the interplay between diversity and accuracy of the individual segmentation solutions in a segmentation ensemble and the influence of them on the final segmentation combination performance. For all three segmentation ensemble generation approaches, it is revealed that (i) the accuracy of ensemble decreases as the diversity of ensemble increases. This relationship can be explained that the good segmentations of the same image are alike, while the bad segmentations are arbitrarily bad in its own way. Thus, the degree of diversity between bad segmentations is relatively higher than the degree of diversity between good ones. (ii) Both diversity and accuracy of the individual ensemble member are crucial factors to the success of segmentation ensemble combination, especially for improving segmentation quality. However, diversity alone may not consistently achieve high quality combination results. When the quality of the majority of the individual ensemble member is poor, a combination of such segmentations may not be able to overcome an error of this magnitude. Thus, a choice of heuristics for generating ensemble is as of important issue to the success of combination approach.

It is also interesting to note that our segmentation combination method is able to achieve improvement of segmentation results on different sizes of ensemble, from a very small size (e.g. 3 segmentations per ensemble, see Section 6.2) to medium size (e.g. 25 segmentations per ensemble, see Section 6.3). This may imply that for our framework the size of segmentation ensemble is not as much critical as the diversity and the accuracy of the ensemble.

Chapter 7

Application I: Parameter Selection Problem

Unsupervised image segmentation is of essential relevance for many computer vision applications and remains a difficult task despite of decades of intensive research. In particular, the parameter selection problem has not received the due attention in the past. In this work we adopt the ensemble combination principle to solve the parameter selection problem in image segmentation. The first scenario of comparison experiments is conducted on a natural color image data set (BSDS). We compare our combination approach to both a classical parameter training approach and a more sophisticated adaptive learning scheme, namely, a case-based reasoning approach. The second scenario of comparison experiments is conducted on two range image data sets. Our approach here is compared with an adaptive search algorithm for automated parameter training. The experimental results reveal that training approaches are not optimal and lack an adaptive behavior in dealing with a particular image, and demonstrate that our approach outperforms all of these three ground truth-based learning approaches.

7.1 Problem Definition

Segmentation algorithms mostly have some parameters and their optimal setting is not trivial since it controls the quality of segmentation results. Normally the correct setting of parameters is given by the algorithm developers. This setting is expected to give satisfactory segmentations for the images in the class used to tune the parameters, however, probably does not give satisfactory segmentations for other

classes of images. This is because most segmentation parameters are usually affected by the changes of the image characteristics such as contrast, noise and illumination. Variations between images may cause drastic changes in segmentation results. As a consequence, the values of segmentation parameters need be adjusted with respect to the changes of image characteristics in order to obtain satisfactory results. One fundamental problem is in fact to find suitable parameter values, preferably on a per-image basis. This need can be illustrated by the two pairs of images shown in Figure 7.1 and Figure 7.2. Each pair is segmented using the FH algorithm [38] based on exactly the same parameter set[1] However, while Figure 7.1(a) and 7.2(a) show a nearly perfect segmentation, we obtain a very bad segmentation in Figure 7.1(b) and 7.2(b). It is obvious that there is no single setting of parameters that will result in the best possible segmentation for any general image, and inappropriate choice of parameter settings result in unsatisfactory segmentations.

In fact, there are several factors that make the problem of parameter selection on a per-image basis rather difficult.

- *Size of Valid Parameter Space*: The size of the parameter search space in a particular segmentation algorithm can be prohibitively large, highly efficient methods may be needed in this case.

- *High Variations of Images*: Since variations between images cause changes in the segmentation results, the objective function that represents segmentation quality varies from image to image. The search technique used to optimize the objective function must be able to adapt to these variations between images [8].

- *Complex nature of the segmentation algorithms and the inherent parameter sets*: Complicated interaction between the segmentation parameters in a typical segmentation algorithm makes it fairly impossible to model the parameters' behavior in an algorithmic fashion. Thus, the multi-dimensional objective function defined using the various parameter combinations cannot generally be modeled in a mathematical way [8].

- *No Consensus on Objective Segmentation Evaluation*: Up to now, there is still no universally accepted method of objective evaluation of segmentation result, which makes evaluation-based algorithm selection hard to apply to real applications.

[1] see Section 2.2 in Chapter 2 for the meaning of these parameters.

(a) ANMI = 0.8435 (b) ANMI=0.5731

Figure 7.1. Illustration of the problem of segmentation algorithm parameter selection. Segmentations obtained by the FH algorithm [38] given the same set of parameter values ($\sigma = 0.9, k = 700, M = 1500$).

(a) ANMI=0.6501 (b) ANMI=0.1924

Figure 7.2. Illustration of the problem of segmentation algorithm parameter selection. Segmentations obtained by the FH algorithm [38] given the same set of parameter values ($\sigma = 0.5, k = 700, M = 1500$).

7.2 Related Works

Despite of its importance, the parameter selection problem has not received the due attention in the past. Researchers typically claim to have empirically determined the parameter values (in an ad-hoc manner). More systematically, the optimal parameter values can be trained in advance based on manual ground truth. A subspace of the parameter space is explored to find out the best parameter setting (with the largest average performance measure). Since fully exploring the subspace can be very costly, space subsampling [97] or genetic search [8, 22, 107] has been proposed. Min et al. [97] proposed an interesting multi-locus hill climbing scheme on a coarsely sampled parameter space for searching the optimal parameters for each segmentation algorithm. The algorithm in its concept does not guarantee to find

the global minima and thus it requires a larger number of initial points (parameter settings) to avoid local minima. More details of this adaptive searching algorithm are given in Section 7.5. A more complex approach for searching the parameter space was proposed by Bhanu and Ming [8]. They proposed the algorithm for tuning a color image segmentation algorithm by a genetic algorithm (GA), where a chromosome is formed by the program parameters. The GA is used to set the control parameters involved in a region-growing based intensity image segmentation using some qualitative evaluation of the segmentation results for guiding the genetic search. Following this work, Cinque et al. [22] used the same rationale for range image segmentation, however, independently from the specific segmenter. Some extensions of [22] are presented in [21, 107]. In [21], they improved the results given by the genetic search [22] by applying simulated annealing strategy. The output of the genetic search is used as a starting point for a simulated annealing process to obtain a more suitable solution at the cost of a relatively small increase of computation. While this approach is reasonable and has been successfully practiced in several applications, its fundamental disadvantage is the assumption of ground truth segmentation. The manual generation of ground truth is always painful and thus a main barrier of wide use in many situations.

Another class of methods assumes a segmentation quality measure, which is used to control a parameter optimization process. Abdul-Karim et al. [1] seek the optimal parameter setting of a vessel/neurite segmentation algorithm by means of a recursive random search algorithm. The search algorithm explores the parameter space driven by trading-off conciseness of the segmentation versus its coverage, which can be systematically defined based on the minimum description length principle. This tradeoff is controlled by external parameters, optionally specified by a user.

Recently, a different class of methods that assumes a segmentation quality measure has been proposed by Peng and Veksler [105]. They develop an algorithm for automatic parameter selection for graph cut based image segmentation. They approach the problem of segmentation quality as a binary classification problem (i.e. good segmentation versus bad segmentation), and train a classifier using the AdaBoost algorithm. Then they run the graph cut segmentation algorithm for different parameter values and choose the segmentation of highest quality according to our learnt measure. This approach has to re-run the graph cut algorithm for different parameter values. Hence, for practically computational reason, the parameter search space has to be low-dimensional. This approach does not assume the availability of the ground truth, however, human intervention is required for labeling the segmented image (as positive or negative example) for learning process.

Another class of methods that assumes a segmentation quality measure is an evaluation-based algorithm selection methods [34, 123]. Singh et al. [123] introduced a novel measurement of image segmentation quality and used this measures for automatic selecting the best segmentations from a set of segmentation results produced by different parameter settings of a segmentation algorithm. In this study, the measurement of image segmentation quality is based on region features from the segmented images. A similar methodology is also be found in the recent work of Espindola et al. [34], whose objective function is defined based on intrasegment homogeneity and intersegment separability. This objective function is used to decide which parameter settings generate the best segmentation result (i.e. the segmentation that maximized intrasegment homogeneity and intersegment heterogeneity). This method is robust as it utilizes the inherent characteristics of images: variance and spatial autocorrelation, which have not been considered in image segmentation evaluation before. Even though these methods show some promising results for some particular image segmentation tasks, it should be noted that the definition of an objective function itself can be a subject of debate because there are available no single, universally accepted measures of segmentation performance with which the quality of the segmented image can be uniquely defined [8].

In this work we propose a novel framework of parameter handling based on ensemble combination. No ground truth is assumed in our framework. The fundamental idea is not to explicitly determine the optimal parameter setting for a particular image. Instead, we compute a set of segmentations (ensemble) according to a subspace sampling of the parameter space and then try to reach an optimum out of the segmentation ensemble. One possibility is to compute an average, or more formally generalized median [74].

7.3 Traditional Parameter Training Approach

In recent years automated parameter training has become popular, mainly by probing a subspace of the parameter space by means of quantitatively comparing with a training image set with (manual) ground truth segmentation [22, 97]. Assume that a reasonable parameter subspace is specified and sampled into a finite number $\mathcal{N}$ of parameter settings. For each parameter setting candidate a performance measure is computed in the following way:

- Segment each image of the training set based on the parameter setting;

- Compute a performance measure by comparing the segmentation result with the corresponding ground truth;
- Compute the average performance measure over all training images.

The optimal parameter setting is given by the one with the largest average performance measure.

7.3.1 Experimental Results

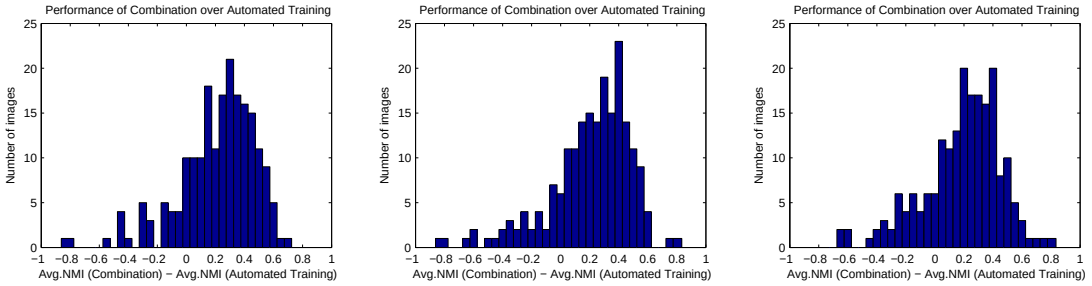
In this experiment series we investigate if our combination approach remains advantageous even if ground truth is available and the parameter training is thus possible. The FH segmentation algorithm is used as a baseline segmentation algorithm to be tuned, because of its competitive segmentation performance and high computational efficiency. The reasonable parameter subspace[1] of the FH algorithm is sampled into 24 parameter settings (see Table 6.1 for a list of parameter values). We apply a 3-fold cross validation in the training process described above. The BSDS data set is randomly partitioned into 3 groups (100 images each). One group forms a 100-images training set while the rest two groups form a 200-images test set. By this way we have 3 different training sets with their corresponding test sets. The training procedure is then run 3 times on each training set to find its optimal parameter setting among the 24 parameter setting candidates.

The average performance measure over 100 images of each training set and 200 images of each test set are listed in the second and third column of Table 7.1, respectively. The forth column of Table 7.1 shows the average performance of combination approach on each test set. The combination results shown in the table are taken from the experiment presented in Section 6.1. The average performance of the combination results is computed according to 200 images in each test set. Figure 7.3 details the summarized information in Table 7.1, which shows histograms of the 200 values of the difference of ANMI values between the two approaches. The positive differences indicate that the combination approach outperforms the automated training approach on each test image. For all three test sets, the distribution skews toward the higher values. The results clearly demonstrate that the combination approach is even superior to automated parameter training. Firstly, the combination approach needs no ground truth. Secondly, even in case of ground truth, the combination approach is able to produce segmentations (on test data) with higher average performance than those of the training approach. This is an indication that

[1] The same parameter subspace used in the experiments reported in Section 6.1 in Chapter 6

Table 7.1. Average performance measures of parameter training and combination approach on 3 test sets.

	Parameter training approach		Combination approach
Test set	Training data	Testing data	(Optimal k)
1	0.5716	0.5936	0.6252
2	0.5887	0.5921	0.6208
3	0.6144	0.5793	0.6078
average	0.5916	0.5883	0.6179



(a) Test set 1 (b) Test set 2 (c) Test set 3

Figure 7.3. Distribution of the difference of ANMI values between the combination approach and the automated training approach for each test set. The positive difference indicates that the combination approach outperforms the training approach.

the trained parameters based on manual ground truth lack an adaptive ability for dealing with the variation of an input image. Figure 7.4 shows the comparison of segmentation results of four images, produced by (a)-(b) trained parameters and (c) our combination algorithm.

The experimental results show that the combined segmentation outperforms the majority of the input segmentations and is in many cases even superior to the best input segmentation (see Figure 6.5). Given the fact that the optimal parameter setting may substantially vary among different images, our framework intends to achieve the highly desired adaptive behavior in dealing with a particular image (see Figure 6.6).

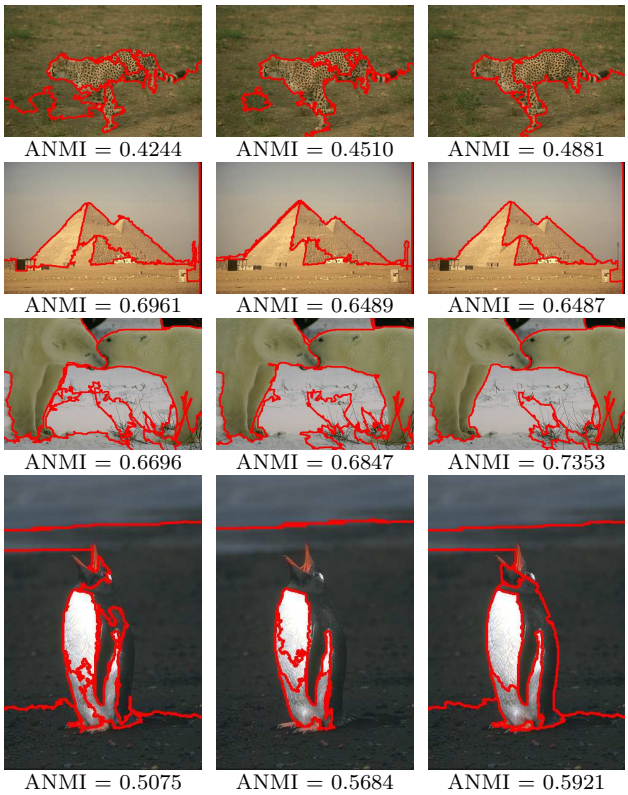


Figure 7.4. Comparison of segmentation results between traditional training approach and the combination approach. (a) and (b) computed by using two optimal parameter sets obtained by training approach: (a) $\sigma = 0.7, k = 300, M = 1500$; (b) $\sigma = 0.9, k = 300, M = 1500$, and (c) computed by our combination algorithm with the generalized median optimality criterion.

7.4 Case-based Reasoning for Image Segmentation

The parameter selection and/or parameter learning should be usually done on a large enough data set, so that it well enough represents the entire domain for building up a general model for segmentation. However, it is often not possible to obtain a large enough data set. Furthermore, a general model guarantees an average best fit over the entire set of images rather than the best segmentation for each image. Therefore, to obtain optimal segmentation on each particular image, the segmentation parameter values need to be adapted according to the changes of image quality and image characteristics.

Frucci et al. [46] and Perner [106] proposed to use case-based reasoning (CBR) for

automatically selecting the segmentation parameter values according to the current image characteristics. Their hypothesis is based on the assumption that images having similar characteristics will show similar good segmentation results when the same segmentation parameters are applied to these images. In the case base, a case consists of a description of the prototype of a class of similar images, coupled with the best solution to its segmentation (i.e. the values of the parameters producing the best result). Then, given an input image, they use CBR to identify in the case base the most similar prototype and the solution associated to the selected prototype is used to run the segmentation algorithm on the input image.

In [46] the description of the prototype is given in terms of the statistical features characterizing the whole image (see Table 7.2). These features are defined for a gray-level image. The first order histogram $H(g)$ is equal to $N(g)/S$, where g is the gray-level, $N(g)$ is the number of pixels with gray-level g and S is the total number of pixels. In our experiment, we adopt these features to handle a color image by applying them separately for each of three color channel. By doing this way, the total number of features for color image becomes three times as large in the number of features for gray-level image. These features are used for indexing the case-base and for retrieval of a set of cases close to the current problem, based on a proper similarity measure. Image similarity has a crucial role for both to build the case base (i.e. grouping similar images into cases) and to compare an input image to the prototypes of the cases in order to derive automatically the proper values for the segmentation parameters. The similarity between two images A and B in the original work [46] is computed on the basis of the statistical features (see Table 7.2) and defined as

$$dist_{AB} = \frac{1}{k}\sum_{i=1}^{k} w_i \left| \frac{C_{iA} - C_{i\min}}{C_{i\max} - C_{i\min}} - \frac{C_{iB} - C_{i\min}}{C_{i\max} - C_{i\min}} \right|, \tag{7.1}$$

where k is the number of features, C_{iA} and C_{iB} are the values of the ith feature of A and B, $C_{i\min}$ and $C_{i\max}$ are the minimum and the maximum value of the ith feature of all images in the database, and w_i weights the ith feature, with $w_1+w_2+\cdots+w_k = 1$. In this experiment the weights w_i assume equal values in accordance with the original work.

7.4.1 Building the Case Base for Image Segmentation

We build the case base following the original work, which proceeds as follows. The statistical features shown in Table 7.2 are used to describe the images. Then, clustering based on the normalized city-block metric (7.1) and the average linkage method

Table 7.2. Statistical features for gray-level image.

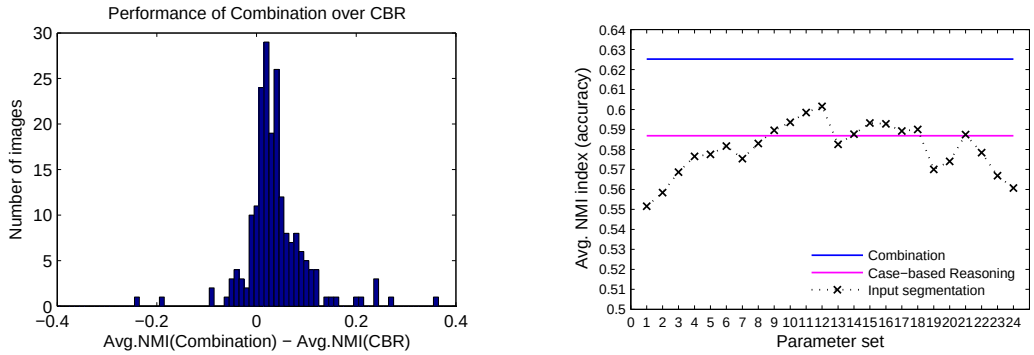
Feature name	Calculation	Feature name	Calculation
Mean	$\bar{g} = \Sigma_g g \cdot H(g)$	Variance	$\delta_g^2 = \Sigma_g (g - \bar{g})^2 H(g)$
Skewness	$g_s = \frac{1}{\delta_g^3} \Sigma_g (g - \bar{g})^3 H(g)$	Kurtosis	$g_k = \frac{1}{\delta_g^4} \Sigma_g (g - \bar{g})^4 H(g) - 3$
Variation Coefficient	$v = \frac{\delta}{\bar{g}}$	Entropy	$g_E = -\Sigma_g H(g) log_2 H(g)$
Centroid_x	$\bar{x} = \frac{\Sigma_x \Sigma_y x f(x,y)}{\Sigma_x \Sigma_y f(x,y)} = \frac{\Sigma_x \Sigma_y x f(x,y)}{\bar{g} S}$	Centroid_y	$\bar{y} = \frac{\Sigma_x \Sigma_y y f(x,y)}{\Sigma_x \Sigma_y f(x,y)} = \frac{\Sigma_x \Sigma_y y f(x,y)}{\bar{g} S}$

were applied to separate different cases and to form groups of similar cases. The expectation is that images, for which we got the best segmentation by using the same values of the parameters, would cluster into groups of similar images. When the values of the segmentation parameters experimentally found to produce the best segmentation results of all images in a cluster are identical, these values are selected as the solution and are recorded in the corresponding case together with the description of the prototype of the cluster. When different best values are found for the segmentation parameters of images in the same cluster, the solution is the set of values producing on the average the best segmentation results for the images in the cluster.

7.4.2 Experimental Results

We randomly divided 300 images in the BSDS dataset into two sets: 100 train images for building the case base and 200 test images for testing the performance of CBR. The 100 train images are clustered into 64 classes according to their statistical features on RGB color space.The FH segmentation algorithm is applied here. The best segmentation parameter for each class is determined by searching the FH parameter subspace (defined in Table 6.1) for the best parameter values. The best parameter setting is the one that produces the average best result for all images in a cluster. The quality of the resulting segmentation is assessed using NMI index by comparing with its corresponding ground truth segmentation. We would like to note that we have tested the CBR approach on a larger and finer parameter subspace (i.e. $\sigma = \{0.4, 0.5, ..., 0.9\}, k = \{300, 400, ..., 1000\}, M = \{100, 300, 600, 900, 1200, 1500\} = 288$ combinations of the segmentation parameter in total), but no significantly statistical improvement was obtained.

The combination results reported in this section are taken from the experiment presented in Section 6.1. Figure 7.5(a) shows a histogram of the 200 values of the difference of ANMI values between the two approaches. The positive differences



(a) (b)

Figure 7.5. (a) Distribution of the difference of ANMI values between the combination approach and the CBR approach. The positive difference indicates that the combination approach outperforms the CBR approach. (b) Average performance of combined results (blue line) and CBR results (magenta line) over 200 test images for each individual parameter setting.

indicate that the combination approach outperforms the CBR approach on each test images. In this case the histogram shows that 83.5% (167 of 200 images) of the combination results get higher performance than the CBR results.

Another perspective is given in Figure 7.5(b), showing the average performance of our approach (blue line) and CBR approach (magenta line) for all 200 test images with regard to the average performance of each of the 24 individual parameter settings (dot line). The blue line lies far above the dot line. This implies that for all 24 parameter settings the combination approach always achieved improved results in average. In contrast, the CBR approach cannot achieve improvement over all 24 individual parameter settings. Frucci et al. suggested that in order to reach the goal of solving segmentation parameter problem, a large case base should be available. However, the initial set of images, though large, does not generally include the prototypes of all possible classes of images. Thus, the segmentation model should be adjusted to fit new data by means of a suitable case base maintenance process (not yet included in our experiment, as well as in the original work). When the current image does not suitably match any image in the initial set, then the current image has to be added to the case base as a new case. To this purpose, the best segmentation parameter values have to be found experimentally for a new case.

Visual comparison of segmentation results of six sampled images are shown in Figure 7.6: (a) and (c) shows the segmentation obtained using the parameter setting selected by CBR, (b) and (d) shows the segmentation results by the combination algorithm. In most cases the combination approach can give more accurate segmen-

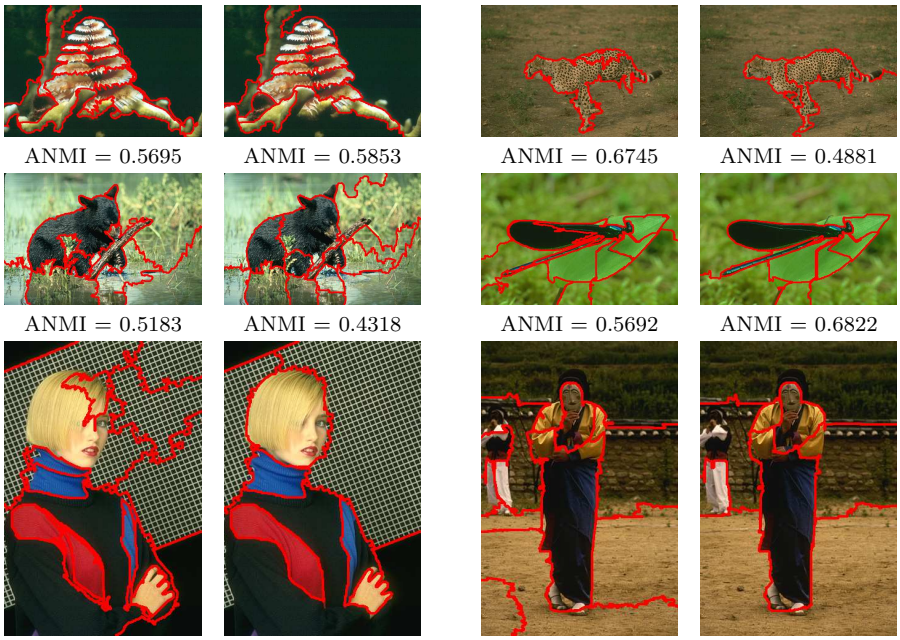


Figure 7.6. Samples of segmentation results: (a) and (c) Case-based reasoning approach. (b) and (d) Our combination approach.

tation results than the CBR approach.

The experimental results clearly demonstrate that the combination approach is even superior to the CBR approach. Firstly, the combination approach needs no ground truth. Secondly, even in case of ground truth, the combination approach is able to produce segmentations (on test data) with higher average performance than those of the CBR approach. Finally, the combination approach is able to operate without having any knowledge about the original features (e.g. intensity, color, etc.) of the input images.

7.5 Automated Training of Parameters on Range Image

Range images are colored according to the distance from the sensor that scans the image. Each pixel in a range image indicates the value of the distance from the sensor to the foreground object point. The range image segmentation algorithm aims at partitioning and labeling range images into surface patches that correspond to surfaces of 3D objects [107]. For decades several range image segmentation algo-

rithms have been proposed (We refer to [64, 67] for a survey of range segmentation algorithms). Each algorithm mostly contains a number of control parameters, whose default values are usually fixed beforehand by the developer of the algorithm. However, these parameter are generally affected by the type of surfaces (e.g. planar versus curved) and the nature of the acquisition system (e.g. laser range finders or structured light scanners). Thus, they need to be tuned according to the changes of image characteristics, in order to provide accurate results on a given class of images.

In this section we propose to approach this parameter selection problem in range image segmentations by our combination method. We compare our approach with automated tuning of parameter framework proposed by Min et al. [97]. Experimental results demonstrate the effectiveness of our approach.

7.5.1 Performance Evaluation on Range Image

A machine segmentation of an image can be compared to the ground truth specification for that image to count instances of correct segmentation, under-segmentation, over-segmentation, missed regions, and noise regions [64]. The definitions of these metrics are based on the degree of mutual overlap required between a region in the machine segmentation and a corresponding region in the ground truth. The meaningful range of required overlap is $50\% < T \leq 100\%$. Note that, currently, only correct segmentation instances metric is considered. A performance curve of the given metric can then be created for each overlap threshold T varies over its meaningful range, from which a quantitative performance value so–called area under the performance curve (AUC) [97] is scored. Performance curves can be normalized to a basis where the ideal curve has an area of one. Thus, the AUC becomes an index in the range of [0,1], representing the average performance of an algorithm over a range of values for the overlap threshold.

For experiments reported in this section, the AUC values are computed using a trapezoid rule with overlap threshold sampled at ten values: 0.51, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, and 0.95.

7.5.2 Automated Tuning of Parameters Framework

Automated tuning of parameters framework proposed in Min et al. [97] consists of three steps: The training step searches for the best parameter settings, the validation step decides how many of the segmenter's parameters should have their value learned through training versus left at the default value, and the test step determines

performance curves to be used in comparing different segmenters. The framework uses a validation step to avoid the over-training problem. After training on a given number of parameters, the parameter values for each training set are run on each validation set. If the area under the validation performance curves is statistically significantly improved in going from $N-1$ to N parameters, and additional parameters are available, then training is repeated using $N+1$ parameters. If there was no significant improvement in going to N parameters available, then the $(N-1)$–parameter training result is kept. If there are no additional parameters, then the N-parameter result is kept. The results of this step will yield parameter values to be used on the test step.

Search Algorithm for Automated Parameter Training Procedure

The adaptive search algorithm for automated parameter training procedure proposed in Min et al. [97] operates as follows. Assume that the number of parameters to be trained and the plausible range of each parameter are specified. The range of each parameter is sampled by five evenly-spaced points. If D parameters are trained, then there are 5^D initial parameter settings to be considered. The segmenter is run on each of the training images with each of these 5^D parameter settings. The segmentation results are evaluated against the ground truth using the AUC metric. The highest performing one percent of the 5^D initial parameter settings, as ranked by area under the curve (AUC), are selected for refinement in the next iteration. The refinement in the next iteration creates a $3 \times 3 \times \cdots \times 3$ sampling around each of the parameter settings carried forward. In this way, the resolution of the parameter settings becomes finer with each iteration, even as the total number of parameter settings considered is reduced in each iteration. The expanded set of points is then evaluated on the training set, and AUCs again computed. The top-performing points are again selected to be carried forward to the next iteration. Iteration continues until the improvement in the AUC drops below 5% between iterations. Then the current top-performing point is selected as the trained parameter setting. This search algorithm is a form of multi-locus hill climbing. The algorithm in its concept does not guarantee to find the global minima and that is why they set a larger number of initial points. This parameter space searching algorithm is summarized in Algorithm 7.3.

The whole training is a time-consuming process which depends on various factors such as the speed of segmenter, the number of images in the train set, the number of train sets, the number of parameters being tuned. For example, if we are about to tune 3 parameters of a segmenter on 10 train images, $5^3 \times 10$ $(= 1,250)$ executions

Algorithm 7.3 Adaptive Searching Algorithm for Parameter Training Procedure

Input: A set of training images,
D segmentation algorithm parameters to be trained and their plausible ranges.
Output: the trained parameter setting.

* *Initial step:* *\
1. Sample five evenly-spaced points from the range of each parameter to form 5^D initial parameter settings.
2. For all training images:
   2.1 Run the segmenter on each image with each of 5^D parameter settings.
   2.2 Compute AUC values for each segmentation results.
3. Select a set of parameter settings, Λ, with the highest performance one percent of the 5^D initial parameter settings for a refinement step.

* *Refinement step:* *\
4. Given a set of parameter settings, Λ:
   4.1 Creates $3 \times 3 \times \cdots \times 3$ sampling around each of the parameter setting in Λ to form a new set of initial parameter settings, Λ'.
   4.2 For all training images:
      4.2.1 Run the segmenter on each image with each parameter settings in Λ'.
      4.2.2 Compute AUC values for each segmentation results.
   4.3 If the improvement in the AUC drops below 5% go to step 5.
   4.4 Select a new set of parameter settings, Λ, with the highest performance one percent of the parameter settings in Λ' and go to step 4.1
5. Parameter setting with the highest AUC value is selected as the trained parameter setting.

of the segmenter are needed just in the initial step. In case of 4 parameters the number of initial segmenter executions goes up to 6,250.

7.5.3 Baseline Segmentation Algorithm and Range Image Dataset

In this experiment, we used the University of Bern (UB) range image segmentation algorithm proposed by Jiang and Bunke [71] as a baseline segmentation algorithm: the UB algorithm for planar–surface scenes will be the baseline segmenter on ABW images, and the UB algorithm for curved–surface scenes [68] will be the baseline segmenter on Cyberware images. All input images, their ground truths, and the

UB algorithms, as well as the package of automated parameter training presented in this experiment, are publicly available via [96]. The details of the algorithm and the range image data set are as follows.

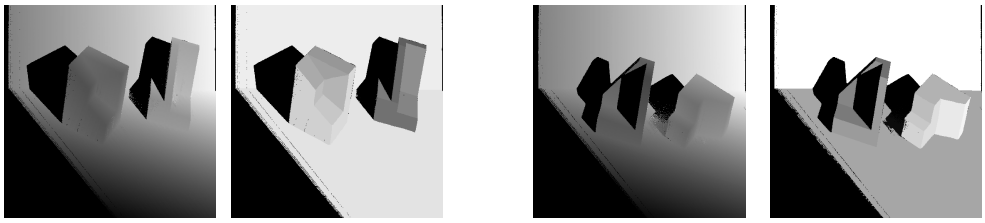
Range Image Dataset

We use the same set of 40 ABW range images for planar scenes [125] and 40 Cyberware range images for curved-surface scenes used in the original work [97]. ABW data set has been adopted by many authors to test their segmentation algorithms (such as [29, 80, 136]). For each image, they provide the ground truth image constructed by pixel-level manual specification. The ground truth for range image contains a region for each of surface patches (e.g. planar, cylindrical, spherical, conical, and toroidal), plus artifact regions for the areas that correspond to significant artifacts in the image (e.g. shadow regions). The average number of ground truth regions in an image is 16.5 for the ABW image set and 9.0 for the Cyberware image set. Figure 7.7 shows an example of ABW range images and its corresponding ground truth. The ABW scanner uses structured light to obtain range values, so *shadow* areas are possible. Pixels in shadow areas have a value of zero and appear black. The larger a depth value the brighter the pixel. An example of Cyberware range images and its corresponding ground truth are shown in Figure 7.8.

The UB Range Image Segmentation Algorithm

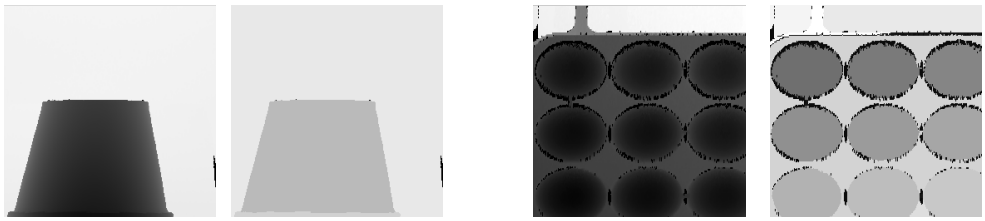
The UB algorithm for planar-surface scenes [71] uses a novel approach that exploits the scan line structure of the image. The segmenter is based on the fact that, in the ideal case, the points on a scan line that belong to a planar surface form a straight 3D line segment. On the other hand, all points on a straight 3D line segment surely belong to the same planar surface. Therefore, they first divide each scan line into straight line segments and subsequently perform a region growing process using the set of line segments instead of the individual pixels. The UB algorithm is considered as the most versatile range image segmentation algorithm in terms of its computational time and segmentation accuracy [64].

The UB algorithm for curved-surface scenes [68] assumes that a moderately well extracted binary edge map is given initially and subsequently refines such initial segmentation into regions by *direction-guided adaptive edge grouping algorithm*. The algorithms extracts closed contour by applying a process of hypotheses generation and verification. This algorithm is based on the consideration that any contour gap



ABW image ground truth (a) ABW image ground truth (b)

Figure 7.7. Example ABW range images and corresponding ground truth image.



CW image ground truth (a) CW image ground truth (b)

Figure 7.8. Example CW range images and corresponding ground truth image.

can be closed by dilating the input edge map. Thus, a single dilation operation followed by a region verification is applied until all regions are labelled. The geometry of contours is taken into account in order to apply the dilation–the dilation process is restricted to one direction. This algorithm is able to achieve appealing performance with respect to both segmentation quality and computation time.

The UB algorithm for planar-surface scenes has seven parameters, and for curved-surface scenes has ten parameters that control its operation, as listed in the order of significance in Table 7.3 and Table 7.4, respectively. These parameters are thresholds on various values in the segmentation algorithm.

7.5.4 Experiments

Automated Parameter Training Approach

Each set of 40 images is divided into a pool of 14 training images, 13 validation images, and 13 test images. Ten different training sets of six images each are created by random sampling from the pool of training images. Similarly, 10 validation sets

Table 7.3. Parameter ranges, their default values and sampling values for ensemble generation of UB algorithm for planar-surface scenes.

Parameter	Range	Default value	Sampling values for ensemble generation
T_1	[1.0, 3.0]	1.25	{1.0, 1.2, 1.4, 1.6}
T_2	[1.5, 3.5]	2.25	{1.5, 1.9, 2.3, 2.7}
t_1	[1.0, 8.0]	4.0	4.0
t_2	[0.05, 0.3]	0.1	0.1
t_3	[1.0, 8.0]	3.0	3.0
t_4	[0.05, 0.3]	0.1	0.1
t_5	[40, 300]	100	100

Table 7.4. Parameter ranges, their default values and sampling values for ensemble generation of UB algorithm for curved-surface scenes.

Parameter	Range	Default value	Sampling values for ensemble generation
T_1	[0.01, 1.0]	0.5	{0.01, 0.05, 0.1, 0.15, 0.2, 0.25}
T_2	[0.01, 5.0]	2.5	{0.01, 0.05, 1.0}
T_3	[10.0, 90.0]	45.0	10.0
t_4	[0.01, 1.0]	0.11	0.11
t_5	[0.01, 0.5]	0.09	0.09
t_6	[1, 10]	2	2
t_7	[1, 10]	3	3
t_8	[100, 500]	200	200
t_9	[0.01, 1.0]	0.11	0.11
t_{10}	[0.01, 0.5]	0.07	0.07

of six images each are created by sampling from the pool of validation images, and 10 test sets of six images each are created by sampling from the pool of test images. The training results and the trained parameter settings of the UB segmentation algorithm reported in the original work [97] are reproduced here (see Table 7.5 for ABW data set and Table 7.6 for Cyberware data set). Note that only the first parameter (T_1) of the UB planar-surface algorithm is trained, and the first three parameters (T_1, T_2, T_3) of the UB curved-surface algorithm are trained. The UB segmenter is then run on each of the test images with each of these trained parameter settings. The segmentation results are evaluated against the ground truth using the AUC metric. The average AUC values for 10 ABW test sets are reported in the second column of Table 7.7, as well as for 10 Cyberware test sets in the forth column.

Table 7.5. AUC values of 10 ABW training sets and their resulting trained parameter values. Only one parameter is trained for the UB algorithm for planer-surface segmentation.

the UB algorithm on planar-surface scenes										
Training set	1	2	3	4	5	6	7	8	9	10
AUC	.82	.79	.78	.80	.86	.78	.82	.77	.81	.79
T_1	1.6	1.2	1.4	1.2	1.0	1.2	1.2	1.4	1.4	1.2

Table 7.6. AUC values of 10 Cyberware training sets and their resulting trained parameter values. Three parameters are trained for the UB algorithm for curved-surface segmentation.

the UB algorithm on curved-surface scenes										
Training set	1	2	3	4	5	6	7	8	9	10
AUC	.71	.60	.61	.49	.67	.62	.71	.56	.53	.65
T_1	.109	.505	.208	.109	.2575	.0595	.208	.208	.109	.208
T_2	0.01	0.01	0.01	0.01	0.01	0.01	1.008	0.01	0.01	0.01
T_3	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0

Segmentation Combination Approach

A segmentation ensemble is generated by varying the first two parameter values of the UB segmentation algorithm. For the UB algorithm for planar-surface scenes, the first two parameters are considered to be the most critical. Thus, only values of the first two parameters are varied (see the last column in Table 7.3), and the five less important parameters were fixed at the same values as found by the manual training [64]. Similarly, only the first two parameters of the UB algorithm for curved-surface scenes are varied (see the last column in Table 7.4). These settings result in 16 combinations of the segmentation parameters for ABW dataset and 18 combinations of the segmentation parameters for Cyberware data set. For each data set, we run the UB segmentation algorithm on each image for all parameter combinations to form a segmentation ensemble for each of 13 test images.

We perform our combination algorithm on all 13 segmentation ensembles for each data set. The final number of regions in a resulting segmented image is determined based on the majority number of regions in segmentations in an ensemble. The variation of number of regions in a range image segmentation ensemble is substantially small, in contrast to an ensemble of natural scene image segmentations (e.g.

Table 7.7. Average AUC values of training approach and combination approach on 10 test sets of ABW and Cyberware data sets.

	ABW Data set		Cyberware Data set	
Test set	Automated Tuning	Combination	Automated Tuning	Combination
1	0.8140	0.8378	0.6138	0.4079
2	0.8563	0.8662	0.5160	0.6041
3	0.8452	0.8536	0.5054	0.5642
4	0.8048	0.8142	0.5299	0.6350
5	0.8557	0.8630	0.4520	0.3080
6	0.8426	0.8479	0.6344	0.4931
7	0.8327	0.8312	0.5249	0.6636
8	0.8552	0.8580	0.5838	0.6167
9	0.8344	0.8472	0.6682	0.3877
10	0.8356	0.8496	0.6597	0.6081
Average	0.8376	0.8469	0.5688	0.5288

the mean of standard deviation of number of regions for all 13 ABW range image segmentation ensembles is only 1.0188). Thus, we expect that the majority number of regions in segmentations in an ensemble would correspond well with the natural number of regions in a given input image.

Experimental Results

The average AUC values for combined segmentation results are reported in accordance with 10 test sets, which are listed in the third column of Table 7.7 for ABW test set and in the fifth column of Table 7.7 for Cyberware test set. For ABW dataset, the combination approach obtained almost always slightly better average AUC than the training approach does, while for Cyberware dataset, the combination approach obtained higher average AUC than the training approach does for only half of all test sets. It is possible that the UB segmentation algorithm for curved-surface scenes is more sensitive to its parameters than the UB segmentation algorithm for planar-surface scenes, and the parameter subspace of the UB curved-surface segmentation algorithm for generating CW segmentation ensemble is not large enough. However, given the fact that we do not need the ground truth segmentations for our operation, the comparative performance of our approach is remarkable and reveals its potential in dealing with the difficult problem of parameter selection without ground truth.

Figure 7.9 and 7.10 show examples of segmentation results on two ABW test images and two CW test images (see Figure 7.7 and 7.8 for input range images and their corresponding ground truths), respectively: (a)-(c) shows segmentation results computed by the UB segmentation algorithm with the trained parameter values, and (d) shows segmentation combination results. The combination approach is able to eliminate small noise segments presented in input segmentations, however, it inevitably removes small shadow areas from combined segmentation results.

Considering the results on ABW data set, we can see that the trained parameter value $T_1 = 1.4$ (in Figure 7.9(c)) yields the best result for the input image in Figure 7.7(b) but yields the worst result for the input image in Figure 7.7(a). On the other hand, the trained parameter value $T_1 = 1.0$ (in Figure 7.9(a)) performs best on the input image in Figure 7.7(a) but performs worst on the input image in Figure 7.7(b). This is an indication that the trained parameters based on manual ground truth lack an adaptive ability for dealing with the variation of an input image. This situation can also be observed in the results of Cyberware dataset (see Figure 7.10). Trained parameter setting $T_1 = 0.109$, $T_2 = 0.01$, $T_3 = 10.0$ (in Figure 7.10(a)) produces excellent segmentation results on the first range images, but it produces worst results for the second input range image.

7.6 Discussion and Conclusions

In this work we have taken a step towards solving the parameter selection problem in image segmentation. Since empirically fixing the parameter values or training in advance based on manual ground truth are not optimal and lack of adaptive behavior for dealing with the problem in a more general context, we have proposed to apply the concept of ensemble combination for exploring the (segmentation) parameter space without the need of ground truth. We verified our framework in a case study of segmentation combination. The experimental results confirm our expectation. Without using any ground truth information, our technique is able to produce segmentations with higher average quality than the training approach.

The focus of our current work is region-based image segmentation. It should be mentioned that our concept of ensemble combination is a general one. Given the demonstrated power we expect that it will be helpful towards solving the parameter selection problem in numerous other contexts. One such example is to explore the parameter space in a double contour detection problem [139]. We will consider further application scenarios in future.

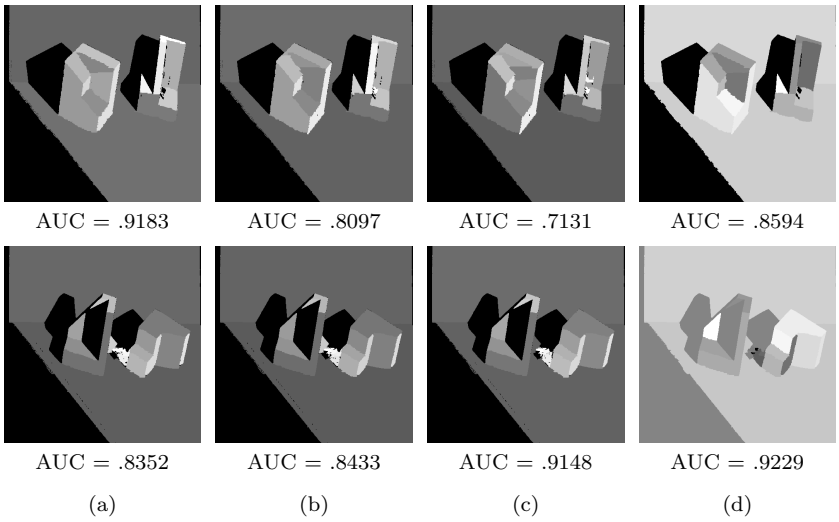


Figure 7.9. Comparison of segmentation results on ABW test images. (a)-(c) Segmentations produced by the UB planar-surface segmentation algorithm with trained parameters $T_1 = 1.0, T_1 = 1.2$ and $T_1 = 1.4$, respectively. (d) Combined segmentation results.

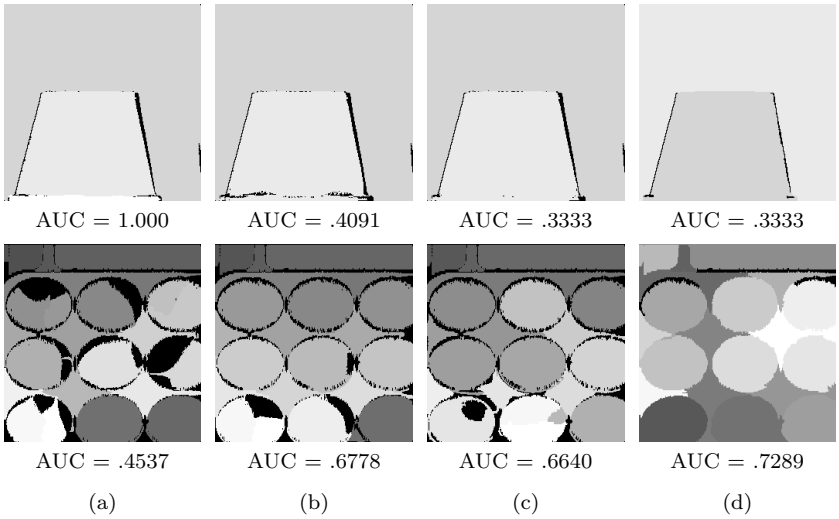


Figure 7.10. Comparison of segmentation results on Cyberware test images. (a)-(c) Segmentations produced by the UB curved-surface segmentation algorithm with three of trained parameter settings: (a) $T_1 = 0.109, T_2 = 0.01, T_3 = 10.0$; (b) $T_1 = 0.208, T_2 = 0.01, T_3 = 10.0$; and (c) $T_1 = 0.505, T_2 = 0.01, T_3 = 10.0$. (d) Combined segmentation results.

Chapter 8

Application II: Instability Problem of Image Segmentation Algorithms

In this chapter we show the other application of our segmentation combination approach. The instability of region growing based image segmentations algorithms is studied. The region growing paradigm is one of the most widely used techniques for image segmentation. It is shown that within a small parameter range, which leads to good segmentation results in the majority of cases, remarkably bad segmentation results may occur. The empirical study presented in [43] shown that instability is in fact a substantial problem of these algorithms. Franek and Jiang [43] also empirically analyzed the frequency of such stabilities on natural images of BSDS dataset [90] and proposed to solve this stability problem by computing the *set median* for a set of segmentations within a specific parameter subspace of interest. The experimental results reported in [43] concluded that adopting the concept of set median to region growing algorithms is reasonable to receive stability. In the majority of cases the computation of set median avoids outliers and achieves robustness.

We propose the use of *generalized median* as an alternative way to solve this problem. The generalized median of a set of segmentations is computed by applied our segmentation combination algorithm. The performance of generalized median comparing to the performance of set median is reported.

8.1 Problem Definition

The region growing paradigm is one of the most widely used techniques for image segmentation because of its competitive segmentation performance and high com-

putational efficiency. However, it is well known that region growing methods suffer from the chaining problem [108, 135]: Pixels of different intensity values can be joined into one region when there exists a chain of pairwise similar pixels which connects them. Furthermore, the direction, in which one region grows, is dependent on the order that pixels are examined. In each iteration region growing algorithms search the unlabeled pixel with the lowest intensity difference between the pixel and its neighboring region [2, 38]. Additionally, the features of each region are adaptively updated as the region growing proceeds. Suppose the input image changes a little, like in the case of image smoothing or noise. This change could cause a different sequence in the region growing and therefore slightly different input images may lead to different regions with different features.

Franek and Jiang [43] analysed two region growing algorithms extensively. It is shown that among a set of parameters which yield good segmentation results, there may be some parameters which yield remarkably bad segmentation results. They also perturb the input images with Gaussian noise and study how segmentations are influenced by noise.

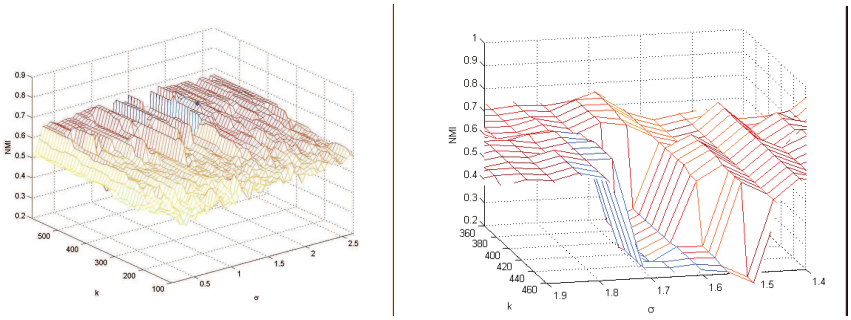
8.1.1 Instability Caused by Variation of Parameters

Franek and Jiang [43] explored the parameter space for each segmentation algorithm and 300 images of the BSDS dataset [90]. They used NMI index as performance measure and as distance measure in their proposed segmentation optimization method. Human segmentations from the BSDS dataset are used as ground truth images.

The first segmentation algorithm is the graph-based image segmentation algorithm[1] (FH) proposed by Felzenszwalb and Huttenlocher [38]. The algorithm has three parameters: a smoothing parameter (σ), a threshold function (k) and a minimum component size (M). A dense parameter grid with a total of 2,500 (50×50) parameter settings: $\sigma = 0.2, 0.25, ..., 2.65$ and $k = 100, 110, ..., 590$ are analyzed. A parameter M is fixed since empirical tests show that segmentation results are not sensitive to change of this parameter [43].

Figure 8.1(a) shows the resulting parameter space for the image in Figure 8.1(c) received by the FH algorithm. Furthermore, Figure 8.1(b) shows a detail of the parameter space. The best and worst results within the detailed parameter space are displayed in Figure 8.1(c) and 8.1(d), respectively. In this work our purpose is to

[1] The detail and the algorithm parameter descriptions of the FH segmentation algorithm is given in Section 2.2.



(a) A whole parameter space

(b) A Detailed view

(c) Best segmentation within the detailed view: NMI = 0.70, $k = 460, \sigma = 1.85$

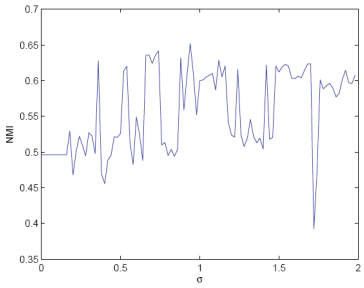
(d) Worst segmentation within the detailed view: NMI = 0.26, $k = 450, \sigma = 1.70$

Figure 8.1. Exploring parameter space for the FH algorithms (taken from [43]).

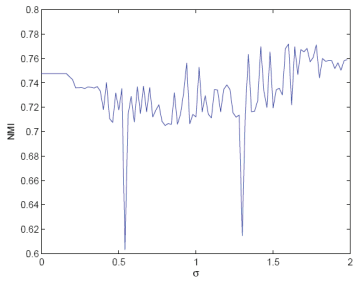
extract such parameter regions from the whole parameter space, where the majority of segmentations are good ones and some bad segmentations are observed.

For comparison purpose Franek and Jiang [43] also investigate the JSEG algorithm, proposed by Deng and Manjunath [28], which combines a color quantization approach with region growing paradigm. A Gaussian filter parameter σ is used in preprocessing. The rest parameters of the JSEG algorithm are set to default since the empirical tests show that segmentation results are not very sensitive to change of these parameters. Therefore, only a dense one-dimensional parameter space consisting of a total of 100 parameter settings: $\sigma = \{0.0, 0.02, \ldots, 1.98\}$ is explored. Two examples of resulting parameter spaces computed by the JSEG algorithm are shown in Figure 8.2(a) and 8.2(b). Furthermore, Figure 8.2(c)-8.2(f) demonstrates that the difference between the best and worst segmentation result within a small parameter range ($\sigma \in (1.6, 1.8)$ resp. $\sigma \in (1.2, 1.4)$) is significant.

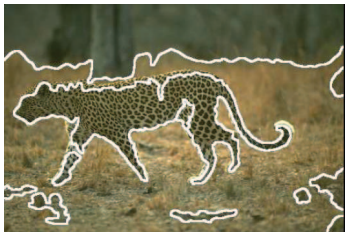
In both cases (FH and JSEG) the differences in segmentation performance is remarkable although the changes in parameters are only small. Often a small change in the smoothing parameter ($\Delta\sigma = 0.02$) leads to remarkable differences in segmen-



(a) Parameter space for image 8.2(c)



(b) Parameter space for image 8.2(e)



(c) Best $\sigma \in (1.6, 1.8)$: NMI = 0.62, $\sigma = 1.72$

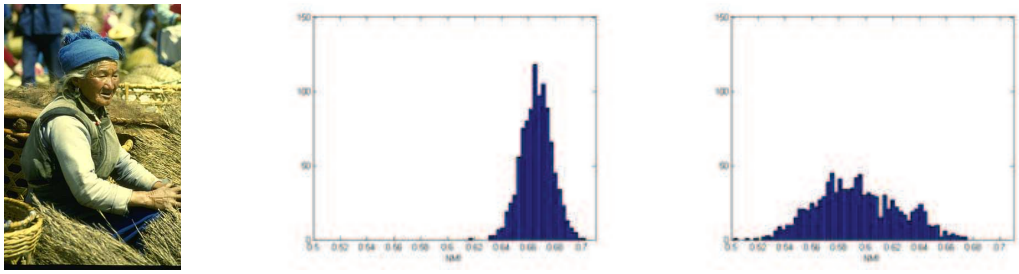
(d) Worst $\sigma \in (1.6, 1.8)$: NMI = 0.39, $\sigma = 1.74$

(e) Best $\sigma \in (1.2, 1.4)$: NMI = 0.76, $\sigma = 1.34$

(f) Worst $\sigma \in (1.2, 1.4)$: NMI = 0.61, $\sigma = 1.32$

Figure 8.2. Exploring parameter space for the JSEG algorithms (taken from [43])

.



(a) Input image (b) Segmented by the FH (c) Segmented by the JSEG

Figure 8.3. NMI-histogram: Segmentation performance of 1,000 noisy images generated by perturbing an input image (a) (taken from [43]).

tation quality. It must be emphasized that the peaks associated with bad results often are unexpected, as can be seen from the results. We conclude that region growing algorithms are very sensitive to Gaussian smoothing whereas the sensitivity to the other parameters (e.g. k) is not very significant. On the other hand, Gaussian filtering is often reasonable in the case of noisy images or to avoid small segments in the segmentation result. Note that image smoothing often is part of segmentation algorithms and can enhance segmentation results significantly, even if images are not noisy. For this reason in principle smoothing should not be avoided.

8.1.2 Instability Caused by Noise

In this section we study how segmentations are influenced by noise. The study is conducted by perturbing an input image with Gaussian noise. For every image of the BSDS dataset noisy images are generated by adding Gaussian noise with zero mean and standard deviation 10^{-3}. For instance, if an image is scaled in $[0, 255]$, the standard deviation corresponds to a deviation of about one grey level. To illustrate this study, we compute 1,000 noisy images from the input image shown in Figure 8.3(a) and segment them using both FH and JSEG algorithms. Then, the quality of all noisy images for each segmentation algorithms are plotted in the NMI-histograms shown in Figure 8.3(b) and 8.3(c), respectively. The NMI values form a Gaussian distribution with standard deviation of 0.01 and 0.03 for the FH and the JSEG algorithms, respectively. Similar results are also received for other images. This result demonstrates that region growing algorithms are not stable if Gaussian noise is added. A high standard deviation of NMI-histogram indicates an unstable algorithm. Suppose a couple of perturbed images are given. In this situation it is desirable to avoid the worst segmentation results and to match at least the mean segmentation result.

8.2 Experiments

We conduct the experimental comparison between our combination approach and the set median approach on 300 color natural images of size 481×321 from the BSDS dataset [90]. We apply the NMI index to quantitatively evaluate the segmentation quality against the ground truths. One segmentation result is compared to all manual segmentations and the average normalized mutual information (ANMI) is reported.

In [43] the set median (see (2.2) in Section 2.1) for each segmentation ensemble is determined by computing the SOD for each segmentation of the ensemble. The set median segmentation is the one that minimize SOD. Since NMI index is used as a performance evaluation measure of segmentations, it is reasonable to use it as a distance function (by 1.0-NMI) in a computation of set median. Let n denote the number of pixels in a segmentation and $|S_a|$ and $|S_b|$ denote the number of groups within labeling S_a and S_b, the computation of the set median of N segmentations has the complexity of $\mathcal{O}(|S_a|\,|S_b|\,nN)$.

In the case of combination approach, we use the generalized median criterion (5.6) for determining the final segmentation solution from a series of combination results with different $k \in [2, 50]$, and regards the final segmentation results as the generalized median segmentation.

8.2.1 Stability in Parameter Space

The parameter spaces of FH and JSEG defined in Section 8.1 are employed and summarized in Table 8.1. The purpose in this work is to examine small parameter regions (in the whole parameter space) that contains good segmentations and some outliers. The set median is then computed for each small parameter region to achieve the stability in such regions. Thus, for each image in the dataset segmentation ensembles consist of segmentations computed from small sets of neighboring parameter settings in the dense parameter space.

For the JSEG algorithm, the whole one-dimensional parameter space with the range of 100 parameter values is divided into ten equidistance parameter ranges, each consisting of ten parameter values. Thus, for each input image, ten segmentation ensembles are generated according to ten parameter subranges. Each segmentation ensemble consists of 10 segmentations (computed from 10 parameter values in a parameter subrange). The set median is then computed for each parameter subrange.

Table 8.1. Summary of the FH parameter subspace sampling and the JSEG parameter subspace sampling.

Algorithm	Parameter	Total parameter settings
FH	$\sigma = \{0.2, 0.25, 0.3, ..., 2.65\}$ $k = \{100, 110, 120, ..., 590\}$	2,500
JSEG	$\sigma = \{0, 10, 20, ..., 490\}$	100

For the FH algorithm, the whole two-dimensional parameter space of size 50×50 is examined. However, the only 5×5 parameter regions that yield good segmentations and some outliers are extracted from the whole parameter space. Note that not all regions in the whole parameter space is used since we are only interested in the area where instability occurs. Furthermore, the number of the extracted parameter regions for each image in the dataset are different, as well as the locations of the interesting regions, depending on the characteristic of that image. Thus, for each image in the dataset the number of generated segmentation ensembles is different. Each segmentation ensemble consists of 25 (5×5) segmentations (computed from 25 parameter settings in an extracted parameter region). The set median is then computed for each 5×5 parameter regions.

Some results of set median and generalized median are shown in Figure 8.4 for both FH and JSEG ensembles. For comparison purpose, the best, average and the worst input segmentations are also shown. Both approaches have relatively similar ANMI values. However, based on visual inspection, the results computed by the combination algorithm have less ragged region boundaries and less oversegmented than the results selected by the set median.

Similar qualities of the GM and SM results in this experiment are possibly due to a relatively small diversity of an input ensemble. An input ensemble is generated from a very narrow range of algorithm parameters. As a result, initial segmentations in an ensemble are quite similar to each other. As we mentioned earlier, a combination of relatively identical segmentation solutions would not achieve improved segmentation that outperforms the individual ensemble members. The combined segmentations are, consequently, relatively identical to the initial segmentations. However, the performance of the GM is slightly better than the performance of the SM. The percent improvements are 0.83 and 0.82 for FH and JSEG ensemble, respectively.

Figure 8.4. Examples of segmentation results on parameter subspace data. (a)-(c) The best, average and the worst input segmentations. (d) Set median segmentation. (e) Generalized median segmentation.

8.2.2 Stability Across Noisy Images

In this experiment we consider the influence of noise on regions growing algorithms. For this reason we fix parameters of the segmentation algorithms and investigate the segmentation performance on noisy images. For each image 100 noisy images are generated by adding Gaussian noise with zero mean and standard deviation 10^{-3} as described in Section 8.1. We postulate that in the situation of a couple of noisy images, it is desirable to avoid the worst segmentation and to match at least the segmentation with average ANMI *without knowing ground truth*. We propose to accomplish this by applying the median concept to compute an approximation of the mean segmentation result without knowing ground truth. Thus, the generalized median and the set median are computed from all 100 noisy images.

The performance of the generalized median and the set median is analyzed in three situations:

1. the segmentation whose ANMI is lower than (average ANMI - 0.1). This level corresponds to the situation where the segmentation is significantly worse than the average ANMI (i.e. the mean segmentation).

2. the segmentation whose ANMI is lower than (average ANMI - 0.05). This level indicates how close the segmentation to the mean segmentation.

3. the segmentation whose ANMI is larger than average ANMI. This level corresponds to the situation where the segmentation is better that the mean segmentation.

The barrier for classifying the segmentation results is chosen from experience. Table 8.2 shows the statistical performance of the computed median segmentations.

In the experiment of noisy images, the GM approach can handle noises in the data more effective than the SM approach. Visual comparison of segmentation results are presented in Figure 8.5. Based on visual inspection, it is clear that the GM approach is able to produce more meaningful segmented images and less affected by noises.

Figure 8.6 shows a histogram of the difference of 300 ANMI values between the GM and SM segmentations. For the FH algorithm, 73.67% of 300 GM segmentations (221 of 300 images) had a slightly higher ANMI values than SM segmentations and 66% of 300 GM segmentations (198 of 300 images) had a slightly higher ANMI values than SM segmentations for the JSEG algorithm. The percent improvement are 1.32 and 1.84 for FH and JSEG ensemble, respectively.

Table 8.2. Performance classification of the median results on noisy data.

Segmentation\Algorithm	FH	JSEG
Worst input < (average ANMI - 0.1)	15.33%	34.33%
Set median < (average ANMI - 0.1)	0.00%	0.67%
Set median < (average ANMI - 0.05)	0.67%	1.33%
Set median > average ANMI	82.33%	74.33%
Generalized median < (average ANMI - 0.1)	0.00%	1.00%
Generalized median < (average ANMI - 0.05)	2.00%	5.00%
Generalized median > average ANMI	88.33%	79.33%

8.3 Discussion and Conclusion

In this work we studied the instability of region growing segmentation algorithms, which is a substantial problem of such algorithms. Firstly, the frequency of instabilities caused by varying the smoothing parameter σ was empirically studied. The intention of this approach was to eliminate peaks associated with bad segmentation results. Experimental results demonstrated the performance of the median concept approach and proved that the median concept approach satisfies the intention well. At this place, we want to remark that computing generalized median segmentations (combination approach) is not the simplest and most efficient way for this particular application, since we only achieve slightly improved results at much higher computational costs. The set median solution is enough for this problem.

In the second application scenario we deal with the instability of the segmentation algorithm across noisy images. The generalized median and the set median of segmentations of noisy images are computed to avoid the worst segmentation results. In the presence of noise, the set median method shows a rather poor performance than the generalized median, mainly because the noise destroys the coherence of the image structures of interest. It is important to note that the set median segmentation is the segmentation selected from a segmentation ensemble, while the generalized median segmentation can go beyond what is typically achieved by a single segmentation in an ensemble. Thus, the generalized median is not directly affected by noise and able to yield improved segmentation results by combining the strength of each individual input segmentation in an ensemble.

Although the combination approach is not the most efficient way for solving this particular problem, the experimental results are mainly intended to show the broad applicability and usefulness of our algorithm in a variety of image segmentation problems.

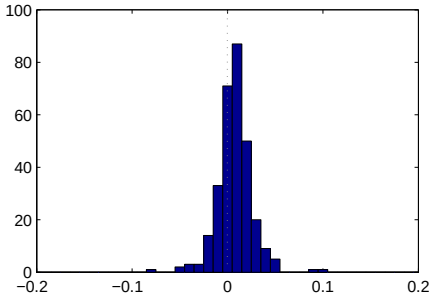
ANMI = 0.4765 ANMI = 0.5047 ANMI = 0.7470 ANMI = 0.4277

(a) Set median segmentation.

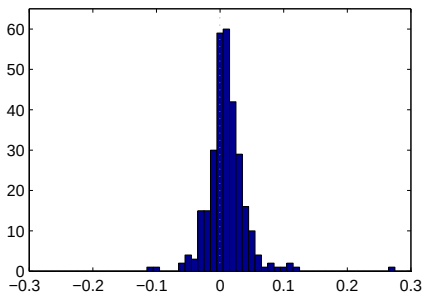
ANMI = 0.5221 ANMI = 0.5640 ANMI = 0.7608 ANMI = 0.4610

(b) Generalized median segmentation.

Figure 8.5. Examples of segmentation results on noisy data set. (a) Set median segmentation. (b) Generalized median segmentation.



(a) Ensembles produced by the FH



(b) Ensembles produced by the JSEG

Figure 8.6. Distribution of the difference of ANMI values between the generalized median and the set median results.

Part III

Evaluation Measures

Chapter 9

Comparison of Segmentation Evaluation Measures

Ideally, the distance function is desired to be a metric in order to match the human intuition of similarity. Distance functions with a metric property enable several advantages in many applications. For example, fast computation of the exact solution of set median strings [70], computation of optimal lower bound for generalized median problem using for assessing the quality of the computed approximated solutions of generalized median [72], and speedup the search in image retrieval system [35]. In this chapter we utilize the special property of a metric in the sense that distance functions, that are a metric, would give a more robust generalized median than using distance functions, that are not a metric. There is essentially no literature for any kind of segmentation evaluation measure which investigates the metric property. In contrast to the previous work, where comparisons between evaluation measures have done in terms of performance evaluation of segmentation results, our work is to compare the evaluation measures themselves. The evaluation measures considered in this work include both the methods specifically derived for segmentation evaluation task and the methods for comparing clusterings developed in statistics and the machine learning community for the purpose of segmentation evaluation.

9.1 Motivation

Recalling to Section 5.5.1 we regarded an approximation of generalized median segmentation as the optimal combined segmentation. The generalized median segmentation is determined by computing the sum of distances (SOD) to all combined

segmentations with different number of regions. The generalized median segmentation is the one that minimize SOD. Thus, the generalized median segmentation is explicitly characterized by a distance function. This raises the issue of how to define a measure of *distance* between segmentations. Ideally, the distance function is desired to be a *metric*, in order to match the human intuition of similarity.

Definition 9.1 (Distance metric) *A distance function d is called a metric distance, iff*

1. $\forall p, q : d(p, q) \geq 0$ (non-negativity)
2. $\forall p, q : d(p, q) = d(q, p)$ (symmetry)
3. $\forall p, q, r : d(p, q) + d(q, r) \geq d(p, r)$ (triangle inequality)

The triangle inequality is necessary since it excludes the undesirable case in which $d(p, r)$ and $d(r, q)$ are both very small, but $d(p, q)$ is very large. According to this special property of a metric, we have a conjecture that *using a metric distance function will give a more robust generalized median than using a non-metric distance function*. Thus, in this work we investigate the metric property of the existing measures.

Recently, there is an extensive literature about various ways to define distance between segmentations. These include methods specifically derived for segmentation evaluation task and the methods for comparing clusterings developed in statistics and the machine learning community but used for the purpose of segmentation evaluation. Among them there may exist functions that satisfy both the non-negativity and the symmetry, but not the triangle inequality. The work [10] extends the concept of metrics to so-called quasi-metrics with a relaxed triangle inequality, where the full power of the triangle inequality is not needed. Instead of the strict triangle inequality, the relation:

$$d(p, r) + d(r, q) \geq \frac{d(p, q)}{1 + \epsilon} \tag{9.1}$$

is required. Here ϵ is a small nonnegative constant. As long as ϵ is not very large, the relaxed triangle inequality still retains the human intuition of similarity. Note that the strict triangle inequality is a special case with $\epsilon = 0$. Thus, a desirable property of being a metric of a distance function is qualified by the relaxed triangle inequality: *The smaller the value of ϵ, the closer the distance function being a metric.*

There is essentially no literature about segmentation evaluation measures presented thus far that compares and investigates the metric property of the existing measures. The remainder of this chapter is organized as follows. In Section 9.2, we

firstly describe some basic requirements for a measure of segmentation evaluation. Then, experimental validation of the metric property of evaluation methods is reported in Section 9.3. Finally, Section 9.4 gives some discussions to conclude the chapter.

9.2 Requirements of Segmentation Evaluation Measures

In this work we are interested in the thirteen general-purpose evaluation measures defined in Chapter 2: GCE, LCE, BCE, p, F, $\mathcal{R}$, $\mathcal{AR}$, $\mathcal{F}$, $\mathcal{J}$, $\mathcal{M}$, $\mathcal{D}$, NMI, and VI. The following basic requirements for image segmentation evaluation measures are discussed in the light of these measures.

1. *Quantitative and Objective*: Quantitative study can provide precise results reflecting the exactness of evaluation while objective study will exempt the influence of human factor and provide consistency and no bias results [148]. Evaluation measures presented in Chapter 2 are normally quantitative as the values of quality measures can be numerically compute. The availability of the ground truth yields objective evaluation results.

2. *Tolerant to Different Segment Counts*: Tolerant to different segment counts is due to the complexity of the images [90]. Segmentation evaluation needs to be able to compare two segmentations when they have different numbers of segments and region size. This property is hold for all evaluation measures mentioned in Chapter 2, except for GCE and LCE. Since GCE and LCE are tolerant of refinement, there are two trivial segmentations that achieve zero error: One pixel per segment, and one segment for the entire image. The former is a refinement of any segmentation, and any segmentation is a refinement of the latter. Thus, these measures are meaningful only when comparing two segmentations with an approximately equal number of segments.

3. *Independent of the Coarseness of Pixelation*: In any situation where comparisons are not restricted to a single data set, a criterion that is not n-invariant would have little value without being accompanied by the corresponding n, where n is a number of pixels in an image. This property is hold for all evaluation measures defined earlier, except for Mirkin and Dongen metrics, which are strongly dependent on n (i.e., both metrics grow unboundedly with n) [94].

Meila denotes the n-invariant versions of $\mathcal{D}$, $\mathcal{M}$ by $\mathcal{D}_{\text{inv}}$, $\mathcal{M}_{\text{inv}}$:

$$\mathcal{D}_{\text{inv}}(\mathcal{C},\mathcal{C}') = \frac{\mathcal{D}(\mathcal{C},\mathcal{C}')}{2n}, \;\; \mathcal{M}_{\text{inv}}(\mathcal{C},\mathcal{C}') = \frac{\mathcal{M}(\mathcal{C},\mathcal{C}')}{n^2}.$$

4. *Tolerant to Refinement*: Refinement is the differences in the pixel-level granularity in the segmentations, of particular, the differences in granularity that are correlated with differences in the level of detail in the human segmentations [132]. Motivation for making segmentation error measures tolerant to refinement is that even if different human observers have the same perceptual organization of an image, they may choose to produce segmentations at varying levels of granularity. Martin et al. [90] argued that "If one segment is a proper subset of the other, then the pixels lies in an area of refinement, and the local error should be small or zero. If there is no subset relationship, then the two regions overlap in an inconsistent manner and the local error should be non-zero". GCE and LCE are completely tolerant to refinement while F-measure is not tolerant of refinement, it is possible for two segmentations that are perfect mutual refinements of each other to have very low precision and recall scores. Furthermore, for a given matching of edge elements between two images, it is possible to change the locations of the unmatched edges almost arbitrarily and retrain the same precision and recall score.

5. *Tolerant to Boundary Localization Error*: In many images even the ground truth data, pixel label assignments are ambiguous near segment boundaries. Hence, one desirable property of a good comparison measure is robustness to small shifts in the location of the boundaries between segments, if those shifts are represented in the manually labeled training set, even when the "true" locations of those boundaries are unknown [131]. F-measure with a single thresholded machine boundary map and a single human boundary map is not tolerate any localization error and would consequently overpenalize algorithms that generate usable, though slightly mislocalized boundaries.

6. *Nondegeneracy*: The measure does not have degenerate cases where input instances that are not well represented by the ground-truth segmentations give abnormally high values of similarity [131].

9.3 Validation of the Metric Property

We verify the metric property of the thirteen evaluation measures: GCE, $LCEL$, BCE, p, F, $\mathcal{R}$, $\mathcal{AR}$, $\mathcal{F}$, $\mathcal{J}$, $\mathcal{M}$, $\mathcal{D}$, NMI, and VI. However, the theme of this

chapter focuses on *distance* quantity rather than *similarity* quantity. Thus, the similarity measures under consideration are transformed to dissimilarity measures as follows.

$$p' = 1 - p,$$

$$F' = 1 - F,$$

$$\mathcal{R}' = 1 - \mathcal{R},$$

$$\mathcal{AR}' = (\mathcal{AR} + 1)/2,$$

$$\mathcal{F}' = 1 - \mathcal{F},$$

$$\mathcal{J}' = 1 - \mathcal{J},$$

$$NMI' = 1 - NMI.$$

Possible values of these dissimilarity measures lie in the range [0,1], where a value of 0 indicates identical segmentations and a value of 1 indicates no similarity between segmentations.

9.3.1 Experimental Setting

The triangle inequality property of these measures are verified by computing the values of ϵ in (9.1). A segmentation triple used in the test is constructed from human segmentations in the BSDS data set [90]. The data set consists of 214 landscape images and 86 portrait images, each having 4-9 human segmentations (resulting in a total of 1,633 human segmentations). We divided 300 images in the database into three sets since the total number of all possible triples for human segmentations of all 214 landscape images are extremely large, which is equal 774,407,868. Processing such size of segmentation triples would cost much the computational time and memory. The first two sets consist of 100 landscape images randomly selected from a pool of 214 landscape images (we do not repeatedly select the same image, thus, the images in both sets are distinct.), and the third set contains all of 86 portrait images from a pool of portrait images. Segmentation triples are then constructed from human segmentations corresponding to the images in each set. Note that segmentations in each triple may either be segmentations of the same image or be segmentations of the different images. Details on the three sets of segmentation triples are summarized in Table 9.1. We assume that all measures considered here are symmetric.

Table 9.1. Details on the three sets of segmentation triples.

Test set	Total number of images	Total number of segmentations	Total number of triples (N)
I	100 (landscape)	545	80,494,320
II	100 (landscape)	546	80,939,040
III	86 (portrait)	476	53,585,700

9.3.2 Experimental Results

For each set of segmentation triples $(t_1, t_2, ..., t_N)$, we obtain a set of ϵ values $(\epsilon_1, \epsilon_2, ..., \epsilon_N)$. Then, $\hat{\epsilon}$ is computed by $\hat{\epsilon} = max\{\epsilon_1, \epsilon_2, ..., \epsilon_N\}$ where $0 < \hat{\epsilon} < \infty$, so that

1. for all p, q, r in each triple in the set, $d(p, r) + d(r, q) \geq \frac{d(p,q)}{1+\hat{\epsilon}}$
2. there is no $\epsilon_i < \hat{\epsilon}$ so that 1) holds.

Statistical values of $\hat{\epsilon}$ for each set of triples are reported in Table 9.2–9.4, respectively. In each table, the results are reported in ascending ordered by the values of $\hat{\epsilon}$. The evaluation measures with smaller value of $\hat{\epsilon}$ exhibit more metric than the evaluation measures with larger value of $\hat{\epsilon}$. For all three sets of tested triples, it is not surprising that values of $\hat{\epsilon}$ of VI, $\mathcal{M}$, and $\mathcal{D}$ are less than or equal zeros, since these measures are proven to be a metric. Values of $\hat{\epsilon}$ of $\mathcal{J}'$, $\mathcal{R}'$, and p' measures are also less than zeros. Values of $\hat{\epsilon}$ of $\mathcal{AR}'$, NMI', F', and $\mathcal{F}'$ are relatively small, while values of $\hat{\epsilon}$ of BCE, GCE, and LCE are relatively large.

9.4 Discussion and Conclusion

In this chapter we tested the metric property of the thirteen evaluation measures. We have made a conjecture that in order to obtain a more robust generalized median segmentation, we want a measure to have the property of a metric. However, it is not necessary that a measure must satisfy the triangle inequality. It is sufficient for a measure to satisfy a relaxed triangle inequality, where a constant ϵ is not too large. An experiment is performed to rank how likely these evaluation measures are metric. We also hope that this study would be helpful for choosing appropriate measures in the situation where the property of a metric is required.

Table 9.2. Statistical values of $\hat{\epsilon}$ ascending sorted by the values of $\hat{\epsilon}$ for test set I.

Distance	$\hat{\epsilon}$	Mean of $(\epsilon_1, \epsilon_2, ..., \epsilon_N)$	Standard deviation
VI	-0.0093	-0.4907	0.1166
$\mathcal{J}'$	-0.0076	-0.4943	0.0907
$\mathcal{M}$	-0.0036	-0.4803	0.1691
$\mathcal{R}'$	-0.0036	-0.4803	0.1691
$\mathcal{D}$	-0.0021	-0.4891	0.1269
p'	-0.0021	-0.4891	0.1269
$\mathcal{AR}'$	0.1121	-0.4943	0.0910
NMI'	0.1717	-0.4939	0.0946
F'	0.2070	-0.4977	0.0577
$\mathcal{F}'$	0.2971	-0.4904	0.1184
BCE	141.2445	-0.3661	0.5406
GCE	175.1888	-0.3455	0.7695
LCE	244.3052	-0.3215	0.7848

Table 9.3. Statistical values of $\hat{\epsilon}$ ascending sorted by the values of $\hat{\epsilon}$ for test set II.

Distance	$\hat{\epsilon}$	Mean of $(\epsilon_1, \epsilon_2, ..., \epsilon_N)$	Standard deviation
VI	-0.0018	-0.4910	0.1146
$\mathcal{J}'$	-0.0013	-0.4945	0.0894
$\mathcal{M}$	-0.0006	-0.4810	0.1663
$\mathcal{R}'$	-0.0006	-0.4810	0.1663
$\mathcal{D}$	0	-0.4895	0.1247
p'	0	-0.4895	0.1247
NMI'	0.0905	-0.4940	0.0940
$\mathcal{AR}'$	0.1055	-0.4940	0.0939
F'	0.1707	-0.4975	0.0607
$\mathcal{F}'$	0.2634	-0.4904	0.1186
BCE	54.3801	-0.3474	0.4847
GCE	69.2655	-0.3202	0.9443
LCE	338.8778	-0.2629	1.5490

Table 9.4. Statistical values of $\hat{\epsilon}$ ascending sorted by the values of $\hat{\epsilon}$ for test set III.

Distance	$\hat{\epsilon}$	Mean of $(\epsilon_1, \epsilon_2, ..., \epsilon_N)$	Standard deviation
VI	-0.0118	-0.4933	0.0993
$\mathcal{J}'$	-0.0066	-0.4961	0.0754
$\mathcal{D}$	-0.0033	-0.4924	0.1056
p'	-0.0033	-0.4924	0.1056
$\mathcal{M}$	-0.0012	-0.4817	0.1627
$\mathcal{R}'$	-0.0012	-0.4817	0.1627
NMI'	0.0363	-0.4939	0.0940
$\mathcal{AR}'$	0.0539	-0.4946	0.0887
F'	0.1232	-0.4975	0.0602
$\mathcal{F}'$	0.1578	-0.4931	0.1004
BCE	59.7018	-0.2919	0.4926
GCE	66.0830	-0.2718	0.7844
LCE	193.3795	-0.2452	1.0894

Chapter 10

Clustering of Segmentation Evaluation Measures

Evaluation of image segmentation is as of indispensable for studying and improving the performance of image segmentation algorithms. Particularly, supervised segmentation evaluation methods are very useful in practice for quantitatively assessing and comparing the quality of resulting segmentations. While many different segmentation evaluation measures have been proposed in the literature, very few researchers have undertaken the task of analyzing existing measures.

Thus far, there is still no consensus on metrics to use for objectively evaluating of image segmentation [8, 144]. Most evaluation measures are generally endowed with different standard for measuring the quality of the segmentation. As a result, different evaluation measures may give significantly different evaluation results on the same set of segmented images. These situations present the difficulty for the users to choose a specific measure for a particular application when they are faced with such a variety of possibilities.

In this work we present an analytical framework for clustering the existing evaluation measures. These measures are clustered into groups according to their evaluating behaviors on the same set of segmented images. The measures with the same behavior will be grouped together. We expect that this study can provide some guidelines in choosing different appropriate evaluation measures, especially, from different clusters, in order to fairly report the performance of the proposed algorithm. There is essentially no literature for any kind of evaluation measures which attempt to cluster the existing evaluation measures according to their evaluating behaviors. Two state-of-the-art segmentation algorithms are involved in the

A B C D E

Figure 10.1. Example input images.

experiments in order to study the behaviors of evaluation measures under real condition. Thirteen evaluation measures under consideration are selected from different technique groups and are widely used in computer vision literature. The proposed clustering framework is general and would be valid for treating a wide range of evaluation measures and with any kind of segmentation methods.

10.1 Motivation

In this work we focus on the supervised evaluation methods. This kind of methods is considered as a principled and powerful way to objectively assess the performance of segmentation algorithms [147]. Moreover, most of them are relatively general and applicable to comparing different kinds of segmentation algorithms. For last decades, many supervised evaluation measures have been proposed in the literature. It is important to realize that each evaluation measure may have distinct standards for measuring the quality of the segmentation. Consequently, the evaluating results vary significantly between different evaluation measures. Particularly, if the segmentation algorithm to be evaluated has a bias in the same situations as the evaluation measure, then some biased results will be produced. In order to illustrate this situation, we apply two different image segmentation algorithms, FH[1] and MS[1], to segment five images in Figure 10.1. The resulting segmentations are shown in Figure 10.2(b) for the FH algorithm and 10.2(c) for the MS algorithm. Four different evaluation measures, $\mathcal{AR}$, NMI, BCE, and F (defined in Chapter 2), are taken to assess the quality of segmentation results against their corresponding ground truths (shown in Figure 10.2(a)). The quantitative performance of two algorithms are reported in Table 10.1. If one would like to claim that the overall performance of the FH algorithm is superior to the overall performance of the MS algorithm, one could choose to report the performance assessed by $\mathcal{AR}$ and NMI only. On the other hand, if one would like to claim that the overall performance of the MS algorithm is superior to the overall performance of the FH algorithm, one could choose to report

[1]Details of FH and MS segmentation algorithms have been described in Chapter 5.

(a) human segmentation (b) FH segmentation (c) MS segmentation

Figure 10.2. Examples of segmentation results produced by (b) the FH algorithm with $\sigma = 0.9, k = 300, M = 1500$ and (c) the MS algorithm with $h_s = 8, h_r = 15, M = 1000$ comparing with its ground truth segmentation in (a). See Table 6.1 for details of segmentation parameters.

the performance assessed by BCE and F instead.

This situation motivates our work. We present a novel framework for clustering different (supervised) evaluation measures proposed so far for segmentation evaluation in the context of region-based segmentation. The evaluation measures are clustered based on their behaviors on assessing the quality of segmented images against ground truth segmentations.

Normally, the raw numerical output of these measures is difficult to compare since they are neither measures of departure from a common baseline nor are they normalized to lie within certain fixed bounds (e.g., 0 and 1 or ± 1) [66]. In this study, the evaluation measures' behavior is captured through the use of selecting and ranking

Table 10.1. Evaluating values of segmentation results shown in Figure 10.2

	FH segmentations				MS segmentations			
Input Image	$\mathcal{AR}$[1]	NMI[1]	BCE[2]	F[1]	$\mathcal{AR}$	NMI	BCE	F
A	0.4199	0.5575	0.5804	0.3742	0.3572	0.5899	0.5374	0.5588
B	0.0717	0.2840	0.8112	0.1827	0.0064	0.2032	0.6551	0.2957
C	0.2959	0.5334	0.6511	0.4020	0.0961	0.2539	0.6444	0.3143
D	0.0438	0.2517	0.8125	0.2316	0.0896	0.1034	0.2300	0.2525
E	0.4618	0.6140	0.5387	0.3861	0.4270	0.6295	0.6421	0.4724
average	0.2586	0.4481	0.6788	0.3153	0.1953	0.3560	0.5418	0.3787

[1] $\mathcal{AR}$, NMI, F are similarity measures, the larger values indicate the better segmentation quality.

[2] BCE is a dissimilarity measure, the smaller values indicate the better segmentation quality.

strategies. *Selecting behavior* is the behavior on selecting k-best segmentations from a set of segmentations, and *ranking behavior* is the behavior on ranking the quality of segmentations in the set. These behaviors reflect directly the overall characteristics (e.g., refinement) and preferences (e.g., bias toward under-/oversegmentation) of evaluation measures. It is expected that the evaluation measures with similar characteristics and preferences will select or rank the segmentation results in a similar manner. Since there are so many choices for selecting a particular evaluation measure, we hope that this behavioral clustering study could be useful for users as a guideline in choosing different evaluation measures, especially in different clusters, in order to fairly report the performance of the evaluated algorithm.

There is essentially no literature for any kind of segmentation evaluation which attempts to cluster the existing evaluation measures according to their behavior under real conditions. In contrast to the previous work presented by Zhang [147] who broadly classified the existing evaluation methods proposed so far into three groups, namely, the analytical, the empirical goodness (unsupervised), and the empirical discrepancy (supervised) groups. Comparative discussion provided in [147] has been just done among the different groups of methods. The comparative study of different supervised evaluation measures can be found in [73, 101, 132]. However, the only properties of evaluation measures of interests (e.g. refinement) have been tested under the specific conditions (typically on synthetic images) separately. The empirical results concluded from these studies are difficult to summarize the overall behavioral similarity between different evaluation measures.

10.2 Behavior on Selecting the k-Best Segmentations

In this study scenario evaluation measures are asked for picking out the best segmentation from a set of segmentation results of the same image. For each segmented image in a given set of segmentation results, evaluation measures under consideration are computed to determine how close the machine segmented image is to the human segmentation. The segmentation result with the best evaluated value is the best segmentation. However, it is possible that the given set of segmentation results contains multiple segmentations with the similar best quality. In this situation choosing any one among them as the best segmentation would be equivalent. Thus, instead of considering only the one best segmentation of the set, we propose to consider a set of the k-best segmentations. Note that the former is a specific case of the latter where k equals one. A value of k indicates the degree of strictness in measuring the similarity between two evaluation measures. A larger value of k gives the higher chance that two evaluation measures will be similar to each other. Therefore, for meaningful clustering results, a value of k should be much smaller than a number of segmentations in a given set.

To cluster the evaluation measures according to their selecting behavior, we need a distance function for measuring the difference between two sets of the k-best segmentations produced by two evaluation measures. The lower distance values indicate the more similar behavior of the two evaluation measures. In other words, the evaluation measures with similar evaluating behavior should produce similar sets of the k-best segmentations and should be clustered into the same group. The distance between two sets of the k-best segmentations can be defined as follows. Let $\mathcal{S}$ be a set of n segmentations to be judged, $\mathcal{S} = \{S_1, S_2, ..., S_n\}$, each segmentation assigned a unique identifier $1, 2, ..., n$. Let π be a set of k-best segmentations that contains identifier of the selected best segmentations. We construct a binary indicator vector v whose length equals the number of segmentations in the set $\mathcal{S}$. v_i equals one if S_i is selected as the best segmentation and equals zero otherwise. For example, suppose that $k = 1$, $n = 5$, and $\pi = 3$, a binary indicator vector v is $[0, 0, 1, 0, 0]$. Then, we can apply any distance metric (e.g. the Minkowski distance) to evaluate the dissimilarity between two binary vectors. If two evaluation measures select the same k-best segmentations, distance between them is zero. Note that the order of segmentations in the set of the k-best segmentations is not important.

10.3 Behavior on Ranking Segmentation Qualities

The task of ranking a list of several alternatives based on one or more criteria is found useful in behavioral survey, such as social choice and voting, comparing genes using expression profiles, and search engine results. The task is relatively easy, and is simply a reflection of the judge's opinions and biases. In this study scenario we cluster different evaluation measures based on their ranking behavior. Our assumption is that the evaluation measures with similar behavior would rank the quality of segmentation results in a similar order. Thus, the similarity between two evaluation measures can be determined in terms of the similarity between two lists of ranking. Note that a set of the k-best segmentations (defined in previous section) is simply the k top segmentations in the ranking. However, the order of segmentations in the ranking list is important, while the order of segmentations in the set of the k-best is not.

In this study, two well known distance metrics for measuring the distance between two rankings are used: the Kendall's tau distance and the Spearman's footrule distance. Both distance functions are metrics and have been widely used in evaluating rankings and ranking aggregation problem in information retrieval [11, 30, 32].

1. *Kendall's tau distance*: Suppose that a set of different segmentations of the same image contains n segmentations. A ranking of n segmentations can be represented as a permutation of the integers $1, 2, \ldots, n$, $\sigma \in P_m$, where $\sigma(i)$ represents the place (rank) of the segmentation i in the ranking. The Kendall tau distance measures the distance between two rankings, σ and τ, by counting the number of pairwise disagreements between the two rankings, which can be formally defined as:

$$\mathcal{K}(\sigma, \tau) = |\{(i, j) | i < j, \sigma(i) < \sigma(j), \text{but } \tau(i) > \tau(j)\}| \,. \tag{10.1}$$

   A normalized version of the Kendall distance, which ranges between 0 and 1, can be obtained by dividing this number by the maximum possible value $\binom{m}{2}$. A smaller distance value implies stronger agreement between two evaluation measures on evaluating segmentations.

2. *Spearman's footrule distance*: Spearman's footrule distance is the sum over all elements $i \in S$, of the absolute difference between the rank of i according to the two lists. Formally, given two full lists σ and τ, the distance is simply the

distance induced by L_1 norm:

$$D(\sigma, \tau) = \sum_{i=1}^{n} |\sigma(i) - \tau(i)| \tag{10.2}$$

After dividing this number by the maximum value $|S|^2/2$, one can obtain a normalized value of the footrule distance, which is always between 0 and 1. Similarly, a smaller distance value implies stronger agreement between two evaluation measures on evaluating segmentations.

10.4 Experiments

Thirteen evaluation measures: GCE, LCE, BCE, p, F, $\mathcal{R}$, $\mathcal{AR}$, $\mathcal{F}$, $\mathcal{J}$, $\mathcal{M}$, $\mathcal{D}$, NMI, VI (defined in Chapter 2), are considered in the experiments. We investigate the behavior of evaluation measures on 300 natural images from the BSDS data set [90], since it provides human segmentations which are necessary for quantitative evaluation in our study. The BSDS provides multiple human segmentations for each image, and good segmentation should be able to explain all of them. Thus, one machine segmentation is compared to all human segmentations of the image, and the average evaluating values are used in the selecting and ranking procedures. The results of selecting and ranking procedure are then fed to a clustering procedure as input data.

To this end we firstly need a set of segmentations of the same image to be selected and ranked. For each image in the BSDS, we generate a set of different segmentations by varying the parameter values of the same segmentation algorithm. In order to make the study reliable, two state-of-the-art segmentation algorithms: the FH algorithm and the MS algorithm (defined in Chapter 2), are applied for segmenting images. The parameter descriptions and 24 sampled parameter values for the FH and the MS algorithms are summarized in Table 6.1. By doing this way, we have two different datasets obtained from the FH and the MS algorithms which are referred to *FH dataset* and *MS dataset*, respectively.

10.4.1 Clustering Results

We apply the average linkage method[1] based on the L_1 norm distance for clustering the selecting behavior, and the Kendall's tau distance and the Spearman's footrule

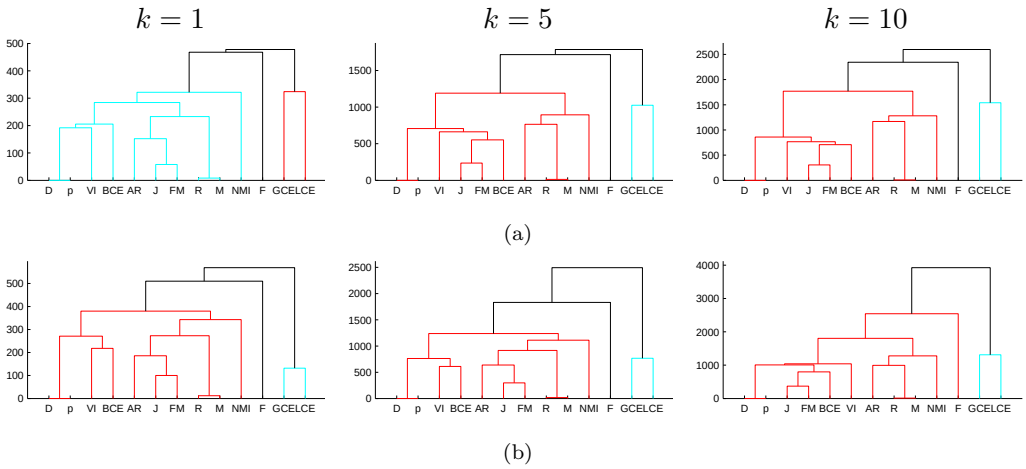
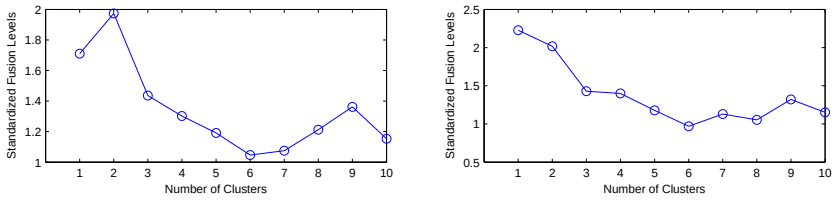


Figure 10.3. Dendrograms of clustering results on selecting behavior on (a) FH dataset and (b) MS dataset.

distance for clustering the ranking behavior. The clustering results of selecting behavior with $k = [1, 5, 10]$ on FH and MS datasets are reported by dendrograms in Figure 10.3(a) and (b), respectively. The clustering results of ranking behavior with the Kendall's tau distance and the Spearman's footrule distance on FH and MS datasets are reported by dendrograms in Figure 10.6(a) and (b), respectively. In the study of selecting behavior, the clustering results with small value of k on both datasets are not stable, i.e. $k < 5$ for FH dataset and $k < 10$ for MS dataset. This may be due to the number of similar best quality segmentations in a set (i.e. the difference between numerical evaluation values of those segmentations is less than 10^{-2}). The higher the number of such similar best quality segmentations, the more fluctuating the results of clustering. However, the clustering results become stable, when increasing a value of k. After clustering results remain stable (i.e. with $k \geq 5$ for FH dataset and with $k \geq 10$ for MS dataset), they show similar clustering results as produced on ranking behavior.

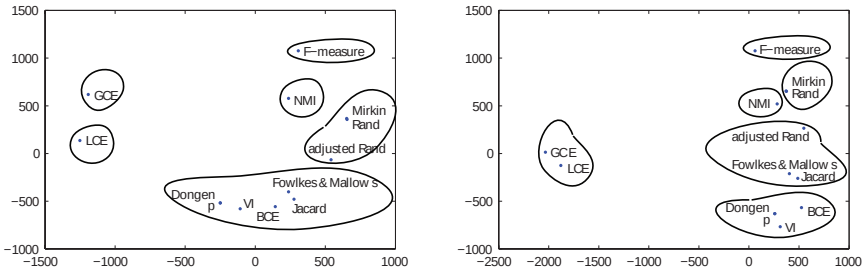
Due to its simple computation and intuitive formulation, the upper tail rule developed by Mojena [100] is applied to determine the appropriate number of clusters in hierarchical clustering. It uses the relative sizes of the different fusion levels in the hierarchy. We let the fusion levels $\alpha_0, \alpha_1, ..., \alpha_{n-1}$ correspond to the stages in the hierarchy with $n, n-1, ..., 1$ clusters. We also denote the average and standard deviation of the j previous fusion levels by $\overline{\alpha}$ and s_α. To apply this rule, we estimate the number of clusters as the first level at which we have $\alpha_{j+1} > \overline{\alpha} + cs_\alpha$, where c is a constant. Milligan and Cooper [95] suggest the value of c to be 1.25 based on

[1]In the clustering literature, the full name of this approach is the Unweighted Pair Group Method using Arithmetic Averages (UPGMA).



(a) (b)

Figure 10.4. The plots of the standardized fusion levels of the dendrograms in Figure 10.3 with $k = 5$ on (a) FH dataset and (b) MS dataset.



(a) (b)

Figure 10.5. 6-Clusters clustering results of selecting behavior with $k = 5$ on (a) FH dataset and (b) MS dataset.

their study on simulated data sets.

The plot of the standardized fusion levels of the dendrograms with $k = 5$ in Figure 10.3 for a maximum of 10 clusters is shown in Figure 10.4. In Figure 10.4(a) the 'elbow' in the curve indicates that 6 clusters are reasonable. In Figure 10.4(b) the 'elbow' in the curve indicates that 3 clusters are reasonable, however, some other 'elbows' at 6 and 8 might provide interesting clusters, too. In this case we choose the clustering results with 6 clusters for both FH and MS datasets as presented in Figure 10.5.

The plot of the standardized fusion levels of the dendrograms in Figure 10.6 for a maximum of 10 clusters is shown in Figure 10.7. In the left plot of Figure 10.4a and b, the 'elbow' in the curve indicates that 5 clusters are interesting. In the right plot of Figure 10.4(a) and (b), the 'elbow' in the curve indicates that 4 clusters are reasonable. In this case we choose the clustering results with 5 clusters for Kendall's tau distance and with 4 clusters for Spearman's footrule distance as presented in Figure 10.8.

It is surprising that the clustering results from all experiments are relatively consistent. Figure 10.8(b) shows the most coarse level of clustering with 4 clusters,

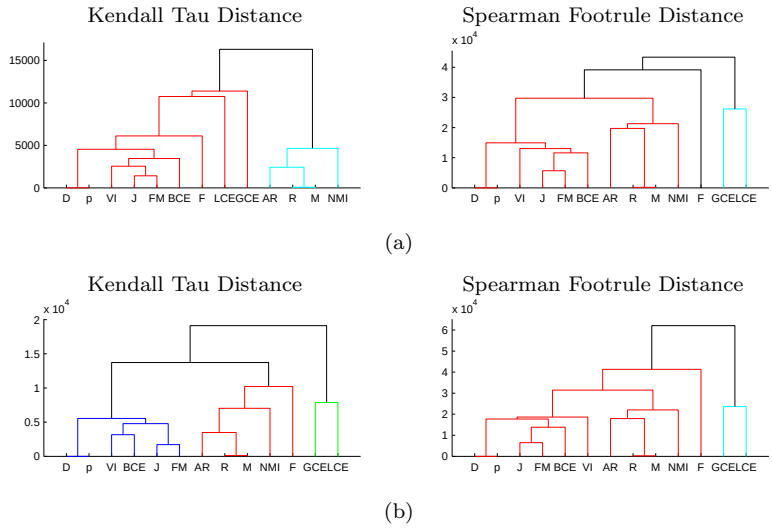


Figure 10.6. Dendrograms of clustering results on ranking behavior on (a) FH dataset and (b) MS dataset.

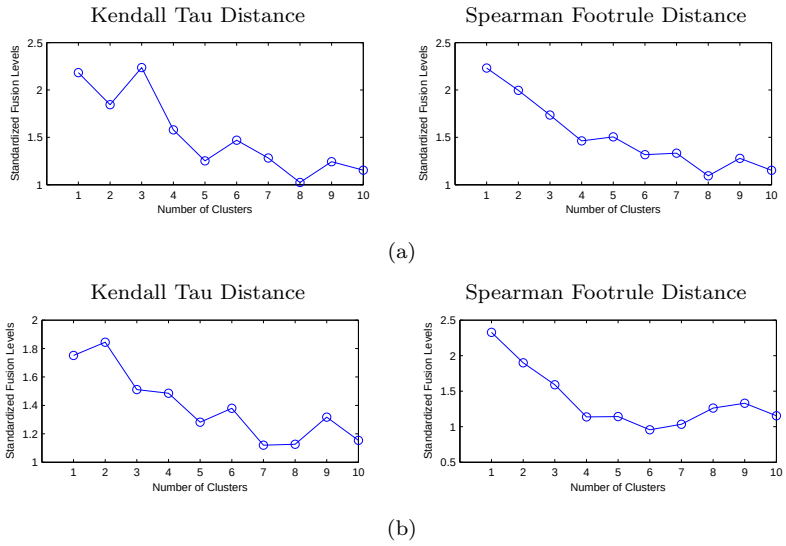


Figure 10.7. The plots of the standardized fusion levels of dendrogram in Figure 10.6 on (a) FH dataset and (b) MS dataset.

and follows by 5 clusters in Figure 10.8(a), while the finer level of clustering with 6 clusters is shown in Figure 10.5(a). It can be concluded that the evaluation measures can intrinsically be clustered. We can see that the clustering result in Figure 10.5 is relatively different from the others, however, it is getting similar to the others when $k \geq 10$. As mentioned earlier the clustering result of selecting behavior on MS dataset is not stable when $k < 10$.

Recalling to the example in Section 10.1, we applied $\mathcal{AR}$, NMI, BCE and F

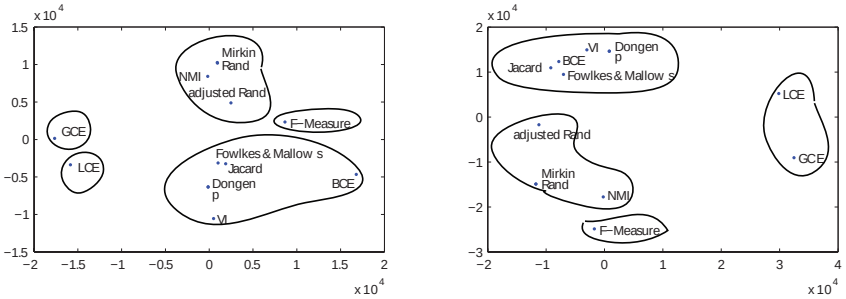


Figure 10.8. Ranking behavior clustering results on both FH and MS datasets: (a) 5-Clusters with Kendall's tau distance and (b) 4-Clusters with Spearman's footrule distance.

measures for evaluating the quality of the example images in Figure 10.1. We see that even though $\mathcal{AR}$ and NMI indices agree that the overall performance of the FH segmentation is superior to the performance of the MS segmentation, their numerical evaluation output for each individual segmented image shows conflict between their agreement (i.e. image A, D, and E). Similarly, BCE and F measures agree that the overall performance of the MS segmentation is superior to the performance of the FH segmentation, however, their numerical evaluation output for each individual segmented image shows some conflict (i.e. image C and E). These situations demonstrate the different evaluation behavior between them, which is consistent with our clustering results, namely, they are separated into different groups (see Figure 10.5). In addition, all clusterings agree to cluster BCE measure into the same group as $\mathcal{D}$, $\mathcal{F}$, $\mathcal{J}$ and VI measures. We can show that the evaluation values of these measures on each five example images correspond well to each other as reported in Table 10.2.

For all results, F-measure, GCE, and LCE are naturally separated from others. It is not surprising in these cases since these measures possess dominant characteristic that is not possessed by the rest measures. F-measure is the only one measure considered in this work that is a boundary based evaluation method. The criterion used for evaluating the segmentation boundary is different from region based evaluation methods. Generally, the former methods have no constraint of producing closed contours, like the latter methods. A missing pixel in the boundary between two regions may not be reflected in the boundary benchmark, but can have substantial consequences for segmentation quality, e.g., incorrectly merging two large regions. GCE and LCE are the only two measures that are tolerant of refinement and, therefore, are not sensible to over- and under-segmentation. However, the dendrograms show quite large difference between them. The reason is that, for any two segmentations, LCE $\leq$ GCE. It is clear that GCE is a tougher measure than LCE so that GCE would tolerate the simple refinement, while LCE would also tolerate

Table 10.2. Evaluating values of segmentation results shown in Figure 10.2

	FH segmentations					MS segmentations				
Image	BCE[1]	$\mathcal{D}$[1]	$\mathcal{F}$[2]	$\mathcal{J}$[2]	VI[1]	BCE	$\mathcal{D}$	$\mathcal{F}$	$\mathcal{J}$	VI
A	0.59	91551	0.54	0.36	2.60	0.54	69300	0.56	0.34	2.21
B	0.75	102786	0.39	0.23	2.98	0.66	93458	0.49	0.32	2.29
C	0.66	101589	0.48	0.29	2.59	0.64	70932	0.60	0.36	2.24
D	0.88	130962	0.27	0.10	3.74	0.23	22808	0.87	0.76	0.76
E	0.63	103043	0.48	0.32	2.61	0.64	105672	0.50	0.33	2.78
average	0.70	105986.20	0.43	0.26	2.90	0.54	72434	0.60	0.42	2.06

[1] BCE, $\mathcal{D}$, VI are distance measures, the smaller values indicate the better segmentation quality.

[2] $\mathcal{F}$ and $\mathcal{J}$ are similarity measures, the larger values indicate the better segmentation quality.

the mutual refinement.

The Dongen metric and p are always clustered into the same group since the Dongen metric is closely related to the performance measure p. The only difference is that the former is a distance measure, while the latter is a similarity measure. The two measures can be mapped to each other by a simple linear transformation $\mathcal{D}(\mathcal{C}, \mathcal{C}') = 2n(1 - p)$ [73]. This kind of relationship can also be found in a pair of the Mirkin metric and the Rand index. Similarly, the former is a distance measure, while the latter is a similarity measure. The two measures can be mapped to each other as $\mathcal{M}(\mathcal{C}, \mathcal{C}') = n(n-1)[1 - \mathcal{R}(\mathcal{C}, \mathcal{C}')]$ [94].

Jacard and FM indices are also closely related. Both similarity measures disregard the quantity N_{00} into account. The difference between them is just their normalizing term of N_{11} value. The former index uses geometric mean of $N_{11} + N_{01}$ and $N_{11} + N_{10}$, while the latter is based on the term $N_{11} + N_{01} + N_{10}$ (see Section 2.3.2).

10.4.2 Clustering Validation

In this section we discuss how appropriate the hierarchical clustering used in the experiments by applying the *Cophenetic Correlation Coefficient* (CPCC). The CPCC has been widely used in numerical phenetic studies, both as a measure of degree of fit of a classification to a set of data and as a criterion for evaluating the efficiency of various clustering techniques [36]. It assesses the results of a hierarchical clustering method by comparing the fusion level of observations with their distance. Values

Table 10.3. Cophenetic correlation coefficient for three hierarchical clustering techniques.

Clusterings	FH Dataset			MS Dataset		
	SL	CL	AL	SL	CL	AL
Selecting, $k = 1$	0.9383	0.9372	0.9514	0.9363	0.9523	0.9708
Selecting, $k = 5$	0.9049	0.9268	0.9314	0.9380	0.9388	0.9611
Selecting, $k = 10$	0.9053	0.8674	0.9247	0.9477	0.9193	0.9693
Ranking, Kendall's tau	0.7665	0.7699	0.8386	0.7178	0.8093	0.8231
Ranking, Spearman's footrule	0.9083	0.8722	0.9311	0.9398	0.9256	0.9677

close to one indicate a higher degree of correlation between the fusion levels and the distances. We use the CPCC to evaluate which type of the following hierarchical clusterings is the best fit for our data.

We calculated the CPCC for three hierarchical clusterings (i.e. single link, average link, and complete link methods). The higher the CPCC value, the better a hierarchical clustering fits the data. The values of CPCC shown in Table 10.3 suggest that the hierarchical clustering produced by the single link technique seems to fit the data less well than the clusterings produced by complete link and average link. The average link method best fits the data since it obtains the highest value of CPCC in all cases.

10.5 Discussion and Conclusion

In this chapter we present an analytical framework for clustering the existing evaluation measures. Thirteen well-known evaluation measures are clustered according to their evaluating behavior on the same set of segmented images. Their numerical outputs are captured through selecting and ranking strategies. The advantages of using these strategies in a study of judging behavior of evaluation measures are as follows. First, different evaluation measures with different range of values can be compared without normalization (into the same range of values, e.g. between 0 and 1). Second, even when the values of two evaluation measures are defined in the same range, the raw numerical evaluation values are also incomparable. Selecting and ranking provide an indirect way to compare two evaluation measures and avoid using their raw numerical outputs. Third, different evaluation measures defined in different philosophy (i.e. similarity/dissimilarity measures) can be directly compared and clustered without transformation.

A prospect of this behavioral clustering study is to give a guideline in choosing

different appropriate evaluation measures, especially from different clusters, in order to fairly report the performance of the proposed algorithm. In addition, we hope that this general clustering framework could be a pioneer framework for further comparing and clustering other evaluation measures existing in literatures. However, it should be noted that the experimental results reported here are preliminary results. More extensive experiments may be conducted to assure the results.

It is important to realize that the evaluation measures may be themselves biased in certain situations. Some research works [13, 73, 146] suggest that instead of using a single measure, we may take a collection of measures and define an overall performance measure. We believe that such combination approach will achieve a better behavior by avoiding the bias of the individual measures. The evaluation measures clustering presented in this chapter provides some useful information for this combination approach since we could select one representative measure from each cluster to build an overall evaluation measure.

Chapter 11

Conclusion

In this thesis, we have taken some steps towards a framework of multiple image segmentation combination. Segmentation ensemble combination has been approved to be a new and powerful means of improving the accuracy and the robustness of image segmentation. We have proposed two novel combination algorithms for combining both multiple contours and multiple region-based segmentations. Both algorithms are able to achieve appealing performance with respect to both segmentation quality and computation time. A problem of automatically determining a number of regions in a final segmentation result has also been carried out. Extensive experimental results verify the effectiveness of both our combination algorithms and our optimality criterion for determining a number of regions. It should be noted that the performance of the combination algorithm will be limited by the capabilities of the segmentation algorithm, but the results will be optimal for a given image based on our combination algorithm and optimality criterion. Beside image segmentation we have studied data analysis problems for segmentation evaluation. We have investigated and compared the metric property of the existing segmentation evaluation measures, as well as developed a clustering framework for clustering them into groups according to their evaluation behaviors.

To summarize, the main contributions of this thesis work are:

- *An algorithm for combining multiple contours.* We have considered a special class of contours which start from the top, pass each image row exactly once, and end in the last row of an image. Exploiting a dynamic programming technique, we are able to efficiently compute the exact solution of generalized median contour of such contours within quadratic computational complexity. Experimental results have been reported on two scenarios, in which the concept

of generalized median plays a very different role.

  - In the first case we have postulated a general approach to implicitly explore the parameter space of a (segmentation) algorithm. It was shown that using the generalized median contour, we are able to achieve contour detection results comparable to those from explicitly training the parameters using a training set with known ground truth.
  - In the second case the specific problem domain of generalized median concept has been considered. Having a generalized median problem with exact solution is interesting in its own right since it gives us a means to verify the tightness of the lower bound for generalized median computation under ideal conditions. As part of our efforts in verifying the tightness of the lower bound using a variety of generalized median problems with exact solution, the current work represents a valuable contribution.

- *An algorithm for combining multiple segmentations.* The algorithm is based on a random walker segmentation algorithm which is able to provide high-quality segmentation starting from manually specified seeds. We are successful in automatically generating such seeds from an input segmentation ensemble with the use of coassociation values. Our algorithm is superior to previous works in that we consider the most general class of segmentation combination, i.e. it is independent from the ensemble generation procedure where any (different) segmentation methods can be used concurrently, and each input segmentation can have an arbitrary number of regions. Extensive experimental results confirm the success of our algorithm in achieving the goal of computing a final segmentation result which is superior to the initial segmentations (in a statistical sense).

  The difficult image segmentation problem has various facets of fundamental complexity. A robust segmentation combination algorithm provides the basis for several ideas outlined in the introduction chapter to alleviate some hard problems in image segmentation. The current work represents a first step towards that development.

  - Solving the parameter selection problem in image segmentation: we have shown that without using any ground truth information, our technique is able to produce segmentations with higher average quality than the training approach. The focus of our current work is region-based image segmentation. It should be mentioned that our concept of ensemble combination is a general one. Given the demonstrated power we expect

that it will be helpful towards solving the parameter selection problem in numerous other contexts.

- Solving the algorithm selection problem in image segmentation: we have shown that even if we do not know the optimal segmentation algorithm for a particular image in advance, the comparative performance of our combination approach is remarkable and reveals its potential in dealing with the difficult problem of optimal algorithm selection even without ground truth. Moreover, our approach is even superior to conventional algorithm selection approaches since in many cases it can provide better quality segmentations beyond what can be achieved by the best segmenter in an ensemble.
- Solving the segmentation algorithm instability problem: the experimental results demonstrate that segmentation combination approach works well for the purpose, however, it is not the most efficient way for solving this particular application. The experiments are mainly intended to show the broad applicability and usefulness of our combination algorithm in a variety of image segmentation problems.

- *An optimality criterion for automatically determining the number of regions in a segmentation results.* We have shown that the number of regions is adequately estimated by adopting the concept of generalized median. In contrast to thresholding criteria, the generalized median based criterion is more adaptive in dealing with a variation in input images. In contrast to a more sophisticated MDL criterion, the advantage of the generalized median-based criterion is that it is not restricted to specific image features, namely only label feature delivered by segmentation algorithms is taken into account. It readily lends itself to applications with a wide range of different imaging modalities (color, intensity, range, etc.).

- *Comparison of evaluation measures.* We have investigated the metric property of evaluation measures by the use of relaxed triangle inequality. We verify our comparison method by taking into account both metric and non-metric evaluation measures in the investigation. The experiments show that metric evaluation measures satisfy very well the relaxed triangle inequality (i.e. $\epsilon \leq 0$) for all test sets, while some non-metric evaluation measures do not. In addition, the experimental results show that two non-metric evaluation measures we used in this work (i.e. NMI index and F-measure) satisfy well the relaxed triangle inequality. This comparison method is designed in a general

way, and we hope that it could be used to investigate the metric property of other existing evaluation measures.

- *Clustering of evaluation measures.* We have analysed the evaluation behavior of the existing evaluation measures through selecting and ranking strategy. Surprisingly, the clustering results of both strategies have shown their consistency, which indicates that the evaluation behaviors of these evaluation measures can be naturally grouped. We expect that this work provides the basis to design a general framework for analysing the existing evaluation measures and provides some useful guidelines for assisting the users in order to choose the evaluation measures to fairly report performance of their proposed segmentation algorithm.

While the preliminary results are very promising, several issues remain. On the proposed combination algorithm, there are some undeveloped ideas for improving the current performance, which need to be further implemented and analysed.

- *Parallel computing.* One efficient way to reduce computational time of our combination framework is to implement it in parallel.
- *New dissimilarity measurement between segmentations.* Since the optimal segmentation resulting from the generalized median based criterion is explicitly characterized by a distance function. A new distance function that better represents the human perceptions would yield more accurate results.
- *New criterion for determining the optimal segmentation combination result.* As we discussed earlier in Section 5, the current segmentation results selected by the proposed optimality criteria are still far away from the 'ideal' solution. There is much more room for improving the optimality criterion in order to obtain the final combination results as close as to that ideal solution.

In addition to future work on improving the proposed algorithm, we have considered some ideas of applying it to a wide variety of applications.

- *Applications.* We will consider further application scenarios for our ensemble combination concept. The proposed combination algorithm can be incorporated as a basic step in different computer vision applications such as medical applications, image retrieval, etc.
- *Extension to general data clustering.* We will consider an extension of the random walker based combination approach to other problem domains, such as clustering ensemble for general data.

Bibliography

[1] M.-A. Abdul-Karim, B. Roysam, N. Dowell-Mesfin, A. Jeromin, M. Yuksel, and S. Kalyanaraman, "Automatic selection of parameters for vessel/neurite segmentation algorithms," *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1338–1350, 2005.

[2] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[3] S. Aljahdali and E. A. Zanaty, "Combining multiple segmentation methods for improving the segmentation accuracy," in *Proceedings of the 13th IEEE Symposium on Computers and Communications*, 2008, pp. 649–653.

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

[5] R. B. Ash, *Information Theory*. Dover Publications, Inc., 1990.

[6] H. Ayad and M. S. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 160–173, 2008.

[7] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," in *Pacific Symposium on Biocomputing*, vol. 7, 2002, pp. 6–17.

[8] B. Bhanu, M. Lee, and J. Ming, "Adaptive image segmentation using a genetic algorithm," *IEEE Transactions on System, Man, and Cybernetics*, vol. 25, no. 12, pp. 1543–1567, 1995.

[9] C. Boulis and M. Ostendorf, "Combining multiple clustering systems," in *Proceedings of the 8th European Conference on Principles and Practice of*

Knowledge Discovery in Databases, ser. LNCS, J. F. B. et al., Ed., vol. 3202. Springer-Verlag, 2004, pp. 63–74.

[10] J. S. Cardoso and L. Corte-Real, "Toward a generic evaluation of image segmentation," *IEEE Transactions on Image Processing*, vol. 14, pp. 1773–1782, 2005.

[11] B. Carterette, "On rank correlation and the distance between rankings," in *Proceedings of the 32nd International Conference on Research and Development in Information Retrieval*. ACM, 2009, pp. 436–443.

[12] S. Chabrier, H. Laurent, and B. Emile, "Performance evaluation of image segmentation. application to parameters fitting," *European Signal Processing Conference*, 2005.

[13] S. Chabrier, C. Rosenberger, H. Laurent, and A. Rakotomamonjy, "Segmentation evaluation using a support vector machine," in *Proceedings of the 3th International Conference on Advances in Pattern Recognition*, ser. LNCS, S. S. et al., Ed., vol. 3686. Springer-Verlag, 2005, pp. 426–435.

[14] V. Chalana and Y. Kim, "A methodology for evaluation of boundary detection algorithms on medical images," *IEEE Transactions on Medical Imaging*, vol. 16, no. 5, pp. 642–652, 1997.

[15] Y. Chang, D. J. Lee, Y. Hong, and J. Archibald, "Unsupervised video shot detection using clustering ensemble with a color global scale-invariant feature transform descriptor," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[16] ——, "Unsupervised video shot segmentation using global color and texture information," in *Proceedings of the 4th International Symposium on Visual Computing, Part I*, ser. LNCS, G. B. et al., Ed., vol. 5358, 2008, pp. 460–467.

[17] D. Cheng, X. Jiang, A. Schmidt-Trucksäss, and K. Cheng, "Automatic intima-media thickness measurement of carotid artery wall in b-mode sonographic images," in *Proceedings of IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2006, pp. 912–915.

[18] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, "Color image segmentation: advances and prospects," *Pattern Recognition*, vol. 34, pp. 2259–2281, 2001.

[19] H. D. Cheng and J. Li, "Fuzzy homogeneity and scale-space approach to color image segmentation," *Pattern Recognition*, vol. 36, pp. 1545–1562, 2003.

[20] K. Cho and P. Meer, "Image segmentation from consensus information," *Computer Vision and Image Understanding*, vol. 68, no. 1, pp. 72–89, 1997.

[21] L. Cinque, F. Corzani, S. Levialdi, R. Cucchiara, and G. Pignalberi, "Improvement in range segmentation parameters tuning," in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 1, 2002, p. 10176.

[22] L. Cinque, S. Levialdi, G. Pignalberi, R. Cucchiara, and S. Martinz, "Optimal range segmentation parameters through genetic algorithms," in *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 1, 2000, pp. 474–477.

[23] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603–619, 2002.

[24] T. Cour, F. Benezit, and J. Shi, "Multiscale normalized cuts segmentation toolbox for MATLAB." [Online]. Available: http://www.seas.upenn.edu/ timothee/software/ncut_multiscale/ncut_multiscale.html

[25] T. Cour, F. Bénézit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Proceedings of the IEEE Computer Society Computer Vision and Pattern Recognition*, 2005, pp. 1124–1131.

[26] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Chichester, UK, 1991.

[27] C. de la Higuera and F. Casacuberta, "Topology of strings: median string is np-complete," *Theoretical Computer Science*, vol. 230, no. 1-2, pp. 39–48, 2000.

[28] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, 2001.

[29] Y. Ding, X. Ping, M. Hu, and D. Wang, "Range image segmentation based on randomized hough transform," *Pattern Recognition Letters*, vol. 26, no. 13, pp. 2033–2041, 2005.

[30] L. P. Dinu and F. Manea, "An efficient approach for the rank aggregation problem," *Theoretical Computer Science*, vol. 359, no. 1–3, pp. 455–461, 2006.

[31] R. C. Dubes, "Cluster analysis and related issues," *Handbook of Pattern Recognition and Computer Vision*, pp. 3–32, 1993.

[32] C. Dwork, R. Kumary, M. Naorz, and D. Sivakumarx, "Rank aggregation methods for the web," in *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 613–622.

[33] M. Egmont-Petersena, D. de Ridderb, and H. Handelsc, "Image processing with neural networks–a review," *Pattern Recognition*, vol. 35, pp. 2279–2301, 2002.

[34] G. M. Espindola, G. Camara, I. A. Reis, L. S. Bins, and A. M. Monteiro, "Parameter selection for region-growing image segmentation algorithms using spatial autocorrelation," *International Journal of Remote Sensing*, vol. 27, no. 14, pp. 3035–3040, 2006.

[35] R. Fagin and L. J. Stockmeyer, "Relaxing the triangle inequality in pattern matching," *International Journal of Computer Vision*, vol. 30, no. 3, pp. 219–231, 1998.

[36] J. S. Farris, "On the cophenetic correlation coefficient," *Systematic Zoology*, vol. 18, no. 3, pp. 279–285, 1969.

[37] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation c++ code." [Online]. Available: http://people.cs.uchicago.edu/ pff/segment/

[38] ——, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[39] X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," in *Proceedings of the International Conference on Machine Learning*, 2003, pp. 63–74.

[40] ——, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proceedings of the International Conference on Machine Learning*, 2004.

[41] B. Fischer and J. Buhmann, "Bagging for path-based clustering," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1411–1415, 2003.

[42] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American Statistical Association*, vol. 78, no. 383, pp. 553–569, 1983.

[43] L. Franek and X. Jiang, "An instability problem of region growing segmentation algorithms and its set median solution," in *Proceedings of the 5th International Symposium on Visual Computing*, ser. LNCS, G. B. et al, Ed., vol. 5876. Springer-Verlag, 2009, pp. 737–746.

[44] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, 2005.

[45] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration," in *Proceedings of the 7th European Conference on Computer Vision*, ser. LNCS, A. H. et al., Ed., vol. 2352. Springer-Verlag, 2002, pp. 408–422.

[46] M. Frucci, P. Perner, and G. Sanniti, "Case-based-reasoning for image segmentation," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 5, pp. 1–14, 2008.

[47] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, pp. 3–16, 1981.

[48] F. Galland, N. Bertaux, and P. Réfrégier, "Minimum description length synthetic aperture radar image segmentation," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 995–1006, 2003.

[49] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie, "Weakly supervised object recognition and localization with stable segmentations," in *Proceedings of European Conference on Computer Vision*, 2008.

[50] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[51] F. Ge, S. Wang, and T. Liu, "New benchmark for image segmentation evaluation," *Journal of Electronic Imaging*, vol. 16, no. 3, p. 033011, 2007.

[52] B. Georgescu and C. M. Christoudias, "The edge detection and image segmentation (EDISON) system." [Online]. Available: http://www.caip.rutgers.edu/riul/research/code.html

[53] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, p. 4, 2007.

[54] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[55] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova, "Moderate diversity for better cluster ensembles," *Information Fusion*, vol. 7, no. 3, pp. 264–275, 2006.

[56] M. Haindl and S. Mikes, "Unsupervised texture segmentation using multiple segmenters strategy," in *Proceedings of the 7th International Workshop on Multiple Classifier Systems*, ser. LNCS, M. H. et al., Ed., vol. 4472. Springer-Verlag, 2007, pp. 210–219.

[57] M. Haindl, S. Mikes, and P. Pudil, "Unsupervised hierarchical weighted multi-segmenter," in *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, ser. LNCS, vol. 5519. Springer-Verlag, 2009, pp. 272–282.

[58] J. Handl, J. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, vol. 21, no. 15, pp. 3201–3212, 2005.

[59] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 100–132, 1985.

[60] ——, *In Computer and Robot Vision.* Reading: Addison-Wesley, 1992, vol. 1.

[61] E. Hayman and J. O. Eklundh, "Probabilistic and voting approaches to cue integration for figure-ground segmentation," in *Proceedings of the 7th European Conference on Computer Vision, Part III*, ser. LNCS, A. H. et al., Ed., vol. 2352. Springer-Verlag, 2002, pp. 469–486.

[62] D. Hoiem, A. Efros, and M. Hebert, "Recovering surface layout from an image," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 151–172, 2007.

[63] K. Hollingsworth, K. W. Bowyer, and P. J. Flynn, "Image averaging for improved iris recognition," in *Proceedings of the 3rd International Conference on Biometrics*, ser. LNCS, M. Tistarelli and M. S. Nixon, Eds., vol. 5558. Springer-Verlag, 2009, pp. 1112–1121.

[64] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher, "An experimental comparison of range image segmentation algorithms," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 673–689, 1996.

[65] Q. Huang and B. Dom, "Quantitative methods of evaluating image segmentation," in *Proceedings of the International Conference on Image Processing*, vol. 3, 1995, pp. 53–56.

[66] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[67] X. Jiang, "Recent advances in range image segmentation," in *Proceedings of the International Workshop on Sensor Based Intelligent Robots*, ser. LNCS, H. I. C. et al., Ed., vol. 1724. Springer-Verlag, 1999, pp. 272–286.

[68] ——, "An adaptive contour closure algorithm and its experimental evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1252–1265, 2000.

[69] X. Jiang, K. Abegglen, H. Bunke, and J. Csirik, "Dynamic computation of generalized median strings," *Pattern Analysis and Applications*, vol. 6, no. 3, pp. 185–193, 2003.

[70] ——, "Median strings: A review," in *Data Mining in Time Series Databases*, M. Last, A. Kandel, and H. Bunke, Eds. World Scientific, 2004, pp. 173–192.

[71] X. Jiang and H. Bunke, "Fast segmentation of range images into planar regions by scan line grouping," *Machine Vision and Applications*, vol. 7, no. 2, pp. 115–122, 1994.

[72] ——, "Optimal lower bound for generalized median problems in metric space," in *Structural, Syntactic, and Statistical Pattern Recognition*, ser. LNCS, T. et al., Ed. Springer-Verlag, 2002, pp. 143–151.

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

[74] X. Jiang, A. Münger, and H. Bunke, "On median graphs: Properties, algorithms, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1144–1151, 2001.

[75] X. Jiang, S. Rothaus, K. Rothaus, and D. Mojon, "Synthesizing face images by iris replacement: Strabismus simulation," in *Proceedings of International Conference on Computer Vision Theory and Applications*, 2006, pp. 41–47.

[76] X. Jiang, L. Schiffmann, and H. Bunke, "Computation of median shapes," in *Proceedings of the 4th Asian Conference on Computer Vision*, 2000, pp. 300–305.

[77] Y. Jiang and Z.-H. Zhou, "SOM ensemble-based image segmentation," *Neural Processing Letters*, vol. 20, no. 3, pp. 171–178, 2004.

[78] T. Kanungo, B. Dom, W. Niblack, and D. Steele, "A fast algorithm for mdl-based multi-band image segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 94, 1994, pp. 609–616.

[79] J. Keuchel and D. Küttel, "Efficient combination of probabilistic sampling approximations for robust image segmentation," in *Pattern Recognition, 28th DAGM Symposium*, ser. LNCS, K. F. et al., Ed., vol. 4174. Springer-Verlag, 2006, pp. 41–50.

[80] K. Koster and M. Spann, "MIR: An approach to robust clustering application to range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 430–444, 2000.

[81] L. Kuncheva, *Combining Pattern Classifiers*. Wiley-Interscience, 2004.

[82] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, 1993.

[83] Y. G. Leclerc, "Constructing simple stable descriptions for image partitioning," *International Journal of Computer Vision*, vol. 3, pp. 73–102, 1989.

[84] T. C. M. Lee, "A minimum description length-based image segmentation procedure, and its comparison with a cross-validation-based segmentation procedure," *Journal of the American Statistical Assocociation*, vol. 95, no. 449, pp. 259–270, 2000.

[85] D. Lopresti and J. Zhou, "Using consensus sequence voting to correct OCR errors," *Computer Vision and Image Understanding*, vol. 67, no. 1, pp. 39–47, 1997.

[86] L. Lucchese and S. K. Mitra, "Color image segmentation: a state-of-the-art survey," in *Image Processing, Vision, and Pattern Recognition, Proc. Indian Nat. Sci. Acad.*, vol. 67, no. 2, 2001, pp. 207–221.

[87] X. Ma, W. Wan, and L. Jiao, "Spectral clustering ensemble for image segmentation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, L. X. et al., Ed., 2009, pp. 415–420.

[88] T. Malisiewicz and A. Efros, "Improving spatial support for objects via multiple segmentations," in *Proceedings of British Machine Vision Conference*, 2007.

[89] D. Martin, "An empirical approach to grouping and segmentation," Ph.D. dissertation, University of California, Berkeley, 2002.

[90] D. R. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th International Conference on Computer Vision*, 2001, pp. 416–425.

[91] ——, "Berkeley segmentation and boundary detection benchmark and dataset," 2003. [Online]. Available: http://www.cs.berkeley.edu/projects/vision/grouping/segbench

[92] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[93] V. Martin and M. Thonnat, "Scene reconstruction, pose estimation and tracking," *International Journal of Advanced Robotic Systems*, p. 530, 2007.

[94] M. Meila, "Comparing clusterings–an information based distance," *Journal of Multivariate Analysis*, vol. 98, no. 5, pp. 873–895, 2007.

[95] G. Milligan and M. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.

[96] J. Min, M. W. Powell, and K. W. Bowyer, "Package of evaluation framework for range image segmentation algorithms." [Online]. Available: http://marathon.csee.usf.edu/range/seg-comp/package.html

[97] J. Min, M. Powell, and K. Bowyer, "Automated performance evaluation of range image segmentation algorithms," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 1, pp. 263–271, 2004.

[98] B. G. Mirkin, *Mathematical Classification and Clustering*. Kluwer Academic Press, Dordrecht, 1996.

[99] A. Moghaddamzadew and N. Bourbakis, "A fuzzy region growing approach for segmentation of color images," *Pattern Recognition*, vol. 30, no. 6, pp. 867–881, 1997.

[100] R. Mojena, "Hierarchical grouping methods and stopping rules: An evaluation," *Computer Journal*, vol. 20, no. 4, pp. 359–363, 1977.

[101] F. C. Monteiro and A. C. Campilho, "Performance evaluation of image segmentation," in *Proceedings of the 3rd International Conference on Image Analysis and Recognition*, ser. LNCS, A. C. Campilho and M. S. Kamel, Eds., vol. 4141. Springer-Verlag, 2006, pp. 248–259.

[102] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[103] C. Pantofaru and M. Hebert, "A comparison of image segmentation algorithms," in *Technical report, Robotics Institute, Carnegie Mellon University*, 2005.

[104] C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," in *Proceedings of 10th European Conference on Computer Vision, Part III*, ser. LNCS, D. F. et al., Ed., vol. 5304. Springer-Verlag, 2008, pp. 481–494.

[105] B. Peng and O. Veksler, "Parameter selection for graph cut based image segmentation," in *Proceedings of the British Machine Vision Conference*, 2008.

[106] P. Perner, "An architecture for a cbr image segmentation system," *Journal on Engineering Applications in Artificial Intelligence*, vol. 12, no. 6, pp. 749–759, 1999.

[107] G. Pignalberi, R. Cucchiara, L. Cinque, and S. Levialdi, "Tuning range image segmentation by genetic algorithm," *EURASIP Journal on Applied Signal Processing*, vol. 8, pp. 780–790, 2003.

[108] L. Priese and V. Rehrmann, "A fast hybrid color segmentation method," in *Mustererkennung 1993, Mustererkennung im Dienste der Gesundheit, 15. DAGM-Symposium*, ser. Informatik Aktuell, S. J. Pöppl and H. Handels, Eds. Springer-Verlag, 1993, pp. 297–304.

[109] A. Rabinovich, A. Vedaldi, and S. Belongie, "Does image segmentation improve object categorization?, Tech. Rep. CS2007-090, 2007.

[110] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *Proceedings of IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[111] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[112] S. Rao, H. Mobahi, A. Yang, S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," in *Proceedings of the 9th Asian Conference on Computer Vision*, 2009, pp. 135–146.

[113] J. Rissanen, "Modeling by the shortest data description," *Automation*, vol. 14, pp. 465–471, 1978.

[114] T. Rohlfing and C. R. Maurer Jr., "Shape-based averaging," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 153–161, 2007.

[115] T. Rohlfing, D. B. Russakoff, and C. R. Maurer Jr., "Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation," *IEEE Transactions on Medical Imaging*, vol. 23, no. 8, pp. 983–994, 2004.

[116] V. Roth and B. Ommer, "Exploiting low-level image segmentation for object recognition," in *Pattern Recognition, 28th DAGM Symposium*, ser. LNCS, K. F. et al., Ed., vol. 4174. Springer-Verlag, 2006, pp. 11–20.

[117] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1605–1614.

[118] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics, and Image Processing*, vol. 41, pp. 233–260, 1988.

[119] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "On aligning curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 116–125, 2003.

[120] S. K. Shah, "Performance modeling and algorithm characterization for robust image segmentation," *International Journal of Computer Vision*, vol. 80, pp. 92–103, 2008.

[121] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[122] J. S. Sim and K. Park, "The consensus string problem for a metric is np-complete," *Journal of Discrete Algorithms*, vol. 1, no. 1, pp. 111–117, 2003.

[123] M. Singh, S. Singh, and D. Partridge, "Parameter optimization for image segmentation algorithms: A systematic approach," in *Proceedings of the 3rd International Conference on Advances in Pattern Recognition*, ser. LNCS, vol. 3687. Springer-Verlag, 2005, pp. 11–19.

[124] M. Spann and A. Nieminen, "Adaptive gaussian weighted filtering for image segmentation," *Pattern Recognition Letters*, pp. 251–255, 1988.

[125] T. G. Stahs and F. M. Wahl, "Fast and robust range data acquisition in a low-cost environment," in *Proceedings of SPIE#1395: Close-Range Photogrammetry Meets Machine Vision*, 1990, pp. 496–503.

[126] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *Journal on Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[127] J. Tang and P. H. Lewis, "Using multiple segmentations for image auto-annotation," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, N. Sebe and M. Worring, Eds., 2007, pp. 581–586.

[128] A. P. Topchy, A. K. Jain, and W. F. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Transactions on Pattern Analalysis and Machine Intelligence*, vol. 27, no. 12, pp. 1866–1881, 2005.

[129] A. P. Topchy, M. H. C. Law, A. K. Jain, and A. L. N. Fred, "Analysis of consensus partition in cluster ensemble," in *Proceedings of the 4th IEEE International Conference on Data Mining*, 2004, pp. 225–232.

[130] A. P. Topchy, B. Minaei-Bidgoli, A. K. Jain, and W. F. Punch, "Adaptive clustering ensembles," in *Proceedings of the International Conference on Pattern Recognition*, 2004, pp. 272–275.

[131] R. Unnikrishnan and M. Hebert, "Measures of similarity," in *7th IEEE Workshop on Applications of Computer Vision/IEEE Workshop on Motion and Video Computing*, 2005, p. 394.

[132] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929–944, 2007.

[133] S. van Dongen, "Performance criteria for graph clustering and markov cluster experiments," Centrum voor Wiskunde en Informatica, Tech. Rep. Technical report INS-R0012, 2000.

[134] D. L. Wallace, "A method for comparing two hierarchical clusterings: Comment," *Journal of the American Statistical Association*, vol. 78, no. 383, pp. 569–576, 1983.

[135] S. Y. Wan and W. E. Higgins, "Symmetric region growing," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1007–1015, 2003.

[136] H. Wang and D. Suter, "MDPE: A very robust estimator for model fitting and range image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 139–166, 2004.

[137] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment," *Journal of Computational Biology*, vol. 1, no. 4, pp. 337–348, 1994.

[138] S. K. Warfield, K. H. Zou, W. M. Wells, and M. William, "Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation," *IEEE Transactions on Medical Imaging*, vol. 23, no. 7, pp. 903–921, 2004.

[139] P. Wattuya and X. Jiang, "A class of generalized median contour problem with exact solution," in *Structural, Syntactic, and Statistical Pattern Recognition*, ser. LNCS, D. Y. Y. et al., Ed., vol. 4109. Springer-Verlag, 2006, pp. 109–117.

[140] P. Wattuya, X. Jiang, S. Praßni, and K. Rothaus, "A random walker based approach to combining multiple segmentations," in *Proceedings of the 19th International Conference on Pattern Recognition*, 2008.

[141] P. Wattuya, X. Jiang, and K. Rothaus, "Combination of multiple segmentations by a random walker approach," in *Pattern Recognition, DAGM Symposium*, ser. LNCS, G. Rigoll, Ed., vol. 5096. Springer-Verlag, 2008, pp. 214–223.

[142] Y. Weiss, "Segmentation using eigenvectors: a unifying view," in *Proceedings of the IEEE International Conference on Computer Vision*, 1999, pp. 975–982.

[143] L. Yang, F. Albregtsen, T. Lønnestad, and P. Grøttum, "A supervised approach to the evaluation of image segmentation methods," in *Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns*. Springer-Verlag, 1995, pp. 759–765.

[144] X. Yong, D. Feng, and Z. Rongchun, "Optimal selection of image segmentation algorithms based on performance prediction," in *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing*, 2004, pp. 105–108.

[145] X. Yong, D. Feng, Z. Rongchun, and M. Petrou, "Learning-based algorithm selection for image segmentation," *Pattern Recognition Letters*, vol. 26, pp. 1059–1068, 2005.

[146] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation:a survey of unsupervised methods," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 260–280, 2008.

[147] Y. J. Zhang, "Segmentation evaluation and comparison: a study of various algorithms," in *Visual Communications and Image Processing*, B. G. Haskell and H.-M. Hang, Eds., vol. 2094, 1993, pp. 801–812.

[148] ——, "A survey on evaluation methods for image segmentation," *Pattern Recognition*, vol. 29, no. 8, pp. 1335–1346, 1996.

[149] ——, *Advances in Image and Video Segmentation*. IRM Press, 2006.

[150] Y. J. Zhang and H. Luo, "Optimal selection of segmentation algorithms based on performance evaluation," *Optical Engineering*, vol. 39, no. 6, pp. 1450–1456, 2000.

[151] Z. H. Zhou and W. Tang, "Clusterer ensemble," *Knowledge-Based Systems*, vol. 19, no. 1, pp. 77–83, 2006.

[152] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation," *IEEE Transactions on Pattern Anaysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, 1996.