

IMAGE SEGMENTATION WITH IMPROVED REGION MODELING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OZAN ERSOY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2004

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet Erkmen

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. A. Aydın Alatan

Supervisor

Examining Committee Members

Prof. Dr. Mete Severcan	(METU, EEE) _____
Assoc. Prof. Dr. A. Aydın Alatan	(METU, EEE) _____
Assoc. Prof. Dr. Gözde Bozdağı Akar	(METU, EEE) _____
Asst. Prof. Dr. Cüneyt Bazlamaççı	(METU, EEE) _____
Ersin Esen, MS	(BİLTEM) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name :

Signature :

ABSTRACT

IMAGE SEGMENTATION WITH IMPROVED REGION MODELING

Ersoy, Ozan

M.S., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. A. Aydin Alatan

December 2004, 76 pages

Image segmentation is an important research area in digital image processing with several applications in vision-guided autonomous robotics, product quality inspection, medical diagnosis, the analysis of remotely sensed images, etc. The aim of image segmentation can be defined as partitioning an image into homogeneous regions in terms of the features of pixels extracted from the image.

Image segmentation methods can be classified into four main categories: 1) clustering methods, 2) region-based methods, 3) hybrid methods, and 4) bayesian

methods. In this thesis, major image segmentation methods belonging to first three categories are examined and tested on typical images. Moreover, improvements are also proposed to well-known Recursive Shortest-Spanning Tree (RSST) algorithm. The improvements aim to better model each region during merging stage. Namely, grayscale histogram, joint histogram and homogeneous texture are used for better region modeling.

Keywords: Image Segmentation, Clustering, Region-Based, RSST, Recursive Shortest-Spanning Tree

ÖZ

GELİŞTİRİLMİŞ BÖLGE MODELLEMESİYLE RESİM BÖLÜTLEME

Ersoy, Ozan

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Assoc. Prof. Dr. A. Aydin Alatan

Aralık 2004, 76 sayfa

Resim bölütleme, dijital resim işlemenin görsel-yönlendirimli otonom robotik, ürün kalite denetimi, tıbbi teşhis, uzaktan algılanan resimler gibi alanlarda uygulamaları olan önemli bir araştırma konusudur. Resim bölütlemenin amacı, bir resmi, resimden çıkarılan belirleyici niteliklere göre türdeş bölgelerine ayırmak olarak tanımlanabilir.

Resim bölütleme metodları dört kategori içinde sınıflandırılabilir: 1) kümeleme metodları, 2) bölge-tabanlı metodlar, 3) melez metodlar, ve 4) bayesgil metodlar. Bu tezde, ilk üç kategoriye ait başlıca resim bölütleme metodları incelenmekte ve tipik resimler üzerinde test edilmektedir. Ayrıca, iyi

bilinen özyinelemeli-en-kısa-ağaç yöntemi (RSST) üzerine geliştirmeler önerilmiştir. Geliştirmeler, birleştirme aşamasında her bölgeyi daha iyi modellemeyi amaçlamaktadır. Daha iyi bölge modellemesi için gri-ölçek histogram, birleşik histogram ve homojen örgü kullanılmıştır.

Anahtar Kelimeler: Resim Bölümleme, Kümelendirme, Bölge-Tabanlı, RSST, Özyinelemeli-En-Kısa-Ağaç

ACKNOWLEDGEMENTS

Firstly, I would like to express my gratitude to my advisor Assoc. Prof. Dr. A. Aydın Alatan for his guidance, suggestions and support throughout this research. Special thanks to Gülsen Yıldırım for her precious support during the writing of this thesis. Also I would like to thank to Yiğit Akkök for his technical assistance, Melike Kuş for her friendship, my parents and my sister for their supports.

TABLE OF CONTENTS

PLAGIARISM	iii
ABSTRACT	iv
ÖZ	vi
ACKNOWLEDGEMENTS	viii
TABLE OF CONTENTS.....	ix
LIST OF FIGURES	xii
LIST OF TABLES	xvii
LIST OF ABBREVIATIONS	xviii
CHAPTER	
1. INTRODUCTION.....	1
1.1 Definition of the Image Segmentation Problem	1
1.2 Scope of the Thesis	2
1.3 Organization of the Thesis	2
2. OVERVIEW OF IMAGE SEGMENTATION METHODS	4
2.1 Image Segmentation Problem	4
2.2 Main Approaches to Image Segmentation	5
2.2.1 Clustering Methods	6
2.2.1.1 Hierarchical Clustering	7

2.2.1.2 Partitional Clustering	7
2.2.2 Region-Based Methods	9
2.2.2.1 Seeded Region Growing	9
2.2.2.2 Split-and-Merge	10
2.2.3 Hybrid Methods	12
2.2.4 Bayesian Methods	13
3. FUNDAMENTAL METHODS IN IMAGE SEGMENTATION	17
3.1 Motivation	17
3.2 K-Means Algorithm	23
3.2.1 Algorithm	23
3.2.2 Results	26
3.3 Fuzzy C-Means Algorithm.....	28
3.3.1 Algorithm	28
3.3.2 Results	30
3.4 Seeded Region Growing	33
3.4.1 Algorithm	33
3.4.2 Results	34
3.5 RSST	37
3.5.1 Fundamental Definitions in Graph Theory	37
3.5.2 Algorithm	38
3.5.3 Results	41
3.6 KMC.....	43
3.6.1 Algorithm	43

3.6.2 Results	47
3.7 Comparative Analysis	49
4. RSST WITH IMPROVED REGION MODELING	53
4.1 Introduction.....	53
4.2 Region Intensity Modeling by Histogram.....	59
4.2.1 Proposed Method	59
4.2.2 Simulation Results	60
4.3 Region Intensity Modeling by Joint Histogram.....	63
4.3.1 Proposed Method	63
4.3.2 Simulation Results	64
4.4 Region Intensity Modeling by Texture	67
4.4.1 Proposed Method	67
4.4.2 Simulation Results	68
4.5 Comparative Analysis	71
5. CONCLUSIONS.....	73
REFERENCES.....	75

LIST OF FIGURES

FIGURE

2.1	Split-and-merge segmentation	11
2.2	Clique examples	15
3.1	A screenshot of the utilized software during simulations.	19
3.2	A screenshot of the utilized seeded region growing software.....	19
3.3	Test image 1	20
3.4	Test image 2	20
3.5	Test image 3	20
3.6	Test image 4	20
3.7	Test image 5	20
3.8	Ground truth image of test image 1 (K=2).....	21
3.9	Ground truth image of test image 2 (K=3).....	21
3.10	Ground truth image of test image 3 (K=4).....	21
3.11	Ground truth image of test image 4 (K=3).....	21
3.12	Ground truth image of test image 5 (K=3).....	21
3.13	The flowchart of K-Means Algorithm	24
3.14	Segmentation of test image 1 with K-Means	26
3.15	Segmentation of test image 2 with K-Means	26

3.16	Segmentation of test image 3 with K-Means	26
3.17	Segmentation of test image 4 with K-Means	26
3.18	Segmentation of test image 5 with K-Means	26
3.19	Segmentation of test image 1 with Fuzzy C-Means.....	31
3.20	Segmentation of test image 2 with Fuzzy C-Means.....	31
3.21	Segmentation of test image 3 with Fuzzy C-Means.....	31
3.22	Segmentation of test image 4 with Fuzzy C-Means.....	31
3.23	Segmentation of test image 5 with Fuzzy C-Means.....	31
3.24	Segmentation of test image 1 with SRG	35
3.25	Segmentation of test image 2 with SRG	35
3.26	Segmentation of test image 3 with SRG	35
3.27	Segmentation of test image 4 with SRG	35
3.28	Segmentation of test image 5 with SRG	35
3.29	Merging process of RSST	38
3.30	The flowchart of the RSST algorithm.....	39
3.31	Segmentation of test image 1 with RSST	41
3.32	Segmentation of test image 2 with RSST	42
3.33	Segmentation of test image 3 with RSST	43
3.34	Segmentation of test image 4 with RSST	44
3.35	Segmentation of test image 5 with RSST	45
3.36	Segmentation of test image 1 with KMC	47
3.37	Segmentation of test image 2 with KMC	47
3.38	Segmentation of test image 3 with KMC	47

3.39	Segmentation of test image 4 with KMC	47
3.40	Segmentation of test image 5 with KMC	47
3.41	Chart showing Normalized Variances of algorithms during simulations ...	49
3.42	Chart showing Ground Truth Errors of algorithms during simulations	50
3.43	Chart showing Elapsed Times of algorithms during simulations	51
4.1	Test image 6	54
4.2	Test image 7	54
4.3	Test Image 2 segmented into 256 regions with RSST	55
4.4	Test Image 3 segmented into 256 regions with RSST	55
4.5	Test Image 5 segmented into 256 regions with RSST	55
4.6	Test Image 6 segmented into 256 regions with RSST	55
4.7	Test Image 7 segmented into 256 regions with RSST	55
4.8	Ground truth image of test image 2 (K=4).....	56
4.9	Ground truth image of test image 3 (K=4).....	56
4.10	Ground truth image of test image 5 (K=3).....	56
4.11	Ground truth image of test image 6 (K=7).....	56
4.12	Ground truth image of test image 7 (K=8).....	56
4.13	Segmentation of test image 2 with RSST	58
4.14	Segmentation of test image 3 with RSST	58
4.15	Segmentation of test image 5 with RSST	58
4.16	Segmentation of test image 6 with RSST	58
4.17	Segmentation of test image 7 with RSST	58
4.18	Histograms belonging to different regions.....	60

4.19	Segmentation of test image 2 with RSST with Region Intensity Modeling by histogram.....	62
4.20	Segmentation of test image 3 with RSST with Region Intensity Modeling by histogram.....	62
4.21	Segmentation of test image 5 with RSST with Region Intensity Modeling by histogram.....	62
4.22	Segmentation of test image 6 with RSST with Region Intensity Modeling by histogram.....	62
4.23	Segmentation of test image 7 with RSST with Region Intensity Modeling by histogram.....	62
4.24	Illustration of creation of joint histograms.....	63
4.25	Joint histograms belonging to different regions.....	64
4.26	Segmentation of test image 2 with RSST with Region Intensity Modeling by joint histogram	66
4.27	Segmentation of test image 3 with RSST with Region Intensity Modeling by joint histogram	66
4.28	Segmentation of test image 5 with RSST with Region Intensity Modeling by joint histogram	66
4.29	Segmentation of test image 6 with RSST with Region Intensity Modeling by joint histogram	66
4.30	Segmentation of test image 7 with RSST with Region Intensity Modeling by joint histogram	66
4.31	Feature layout for texture feature extraction.....	68

4.32 Segmentation of test image 2 with RSST with Region Intensity Modeling by texture.....	70
4.33 Segmentation of test image 3 with RSST with Region Intensity Modeling by texture.....	70
4.34 Segmentation of test image 5 with RSST with Region Intensity Modeling by texture.....	70
4.35 Segmentation of test image 6 with RSST with Region Intensity Modeling by texture.....	70
4.36 Segmentation of test image 7 with RSST with Region Intensity Modeling by texture.....	70
4.37 Chart showing Ground Truth Errors of proposed methods during simulations ..	72
4.38 Chart showing Elapsed Times of proposed methods during simulations ..	72

LIST OF TABLES

TABLE

3.1	Numerical results of K-Means Algorithm for test images	26
3.2	Numerical results of Fuzzy C-Means Algorithm for test images	31
3.3	Numerical results of SRG Algorithm for test images	35
3.4	Numerical results of RSST Algorithm for test images	41
3.5	Numerical results of KMC Algorithm for test images.....	47
3.6	Summary of the simulation results.....	52
4.1	Numerical results of classical RSST algorithm for test images.....	57
4.2	Numerical results of RSST algorithm with Region Intensity Modeling by histogram for test images	61
4.3	Numerical results of RSST algorithm with Region Intensity Modeling by joint histogram for test images	65
4.4	Numerical results of RSST algorithm with Region Intensity Modeling by texture for test images	69

LIST OF ABBREVIATIONS

KMC: K-Means with Connectivity Constraint

RSST: Recursive Shortest-Spanning Tree

SRG: Seeded Region-Growing

SSL: Sequentially Sorted List

SST: Shortest-Spanning Tree

CHAPTER 1

INTRODUCTION

An important area of digital image processing is the segmentation of an image into homogeneous regions. Image segmentation is one of the most challenging problems of digital image processing and many different approaches and methods have been proposed in the literature. However, there is still not an exact solution that can be applied to all image types and obtains perfect results.

1.1 Definition of the Image Segmentation Problem

Image segmentation is a low-level image processing task that aims at partitioning an image into homogeneous regions in terms of the features of pixels extracted from the image [1]. The definition of a region homogeneity depends on the application. Examples for such homogeneity features are pixel gray level, pixel RGB color, range of the pixel from the camera, position of the pixel, local covariance matrix, etc. [2]

The results of image segmentation are often used as initial parameters to higher-level digital image processing tasks and have many important application areas such as vision-guided autonomous robotics, product quality inspection, medical diagnosis and the analysis of remotely sensed images.

1.2 Scope of the Thesis

This thesis aims comparative analysis of major image segmentation methods, as well as proposing some improvements. Major image segmentation algorithms, namely K-Means, Fuzzy C-Means, Seeded Region Growing, Recursive Shortest-Spanning Tree and K-Means with Connectivity Constraint, are analyzed and applied on test images. Moreover, improvements to region modeling of Recursive Shortest-Spanning Tree algorithm are proposed and tested. Namely, grayscale histogram, joint histogram and homogeneous texture are used for better region modeling.

1.3 Organization of the Thesis

In Chapter 2, the overview of image segmentation methods in the literature is presented.

In Chapter 3, major image segmentation algorithms are examined and experimental results on test images are given.

In Chapter 4, some improvements to region modeling of well-known Recursive Shortest-Spanning Tree algorithm are proposed.

In Chapter 5, some concluding remarks are stated on the performance of image segmentation algorithms.

CHAPTER 2

OVERVIEW OF IMAGE SEGMENTATION METHODS

2.1 Image Segmentation Problem

As already defined, image segmentation is a low-level image processing task that aims at partitioning an image into homogeneous regions in terms of the features of pixels extracted from the image [1].

A more formal definition of image segmentation can be given in the following way [3]:

Let I denote an image and let H define a certain “homogeneity predicate”; then the segmentation of I is a partition P of I into a set of N regions R_n , $n = 1, \dots, N$, such that:

$$1) \bigcup_{n=1}^N R_n = I \text{ with } R_n \cap R_m \neq \emptyset, n \neq m$$

$$2) H(R_n) = \text{TRUE } \forall n$$

3) $H(R_n \cup R_m) = \text{FALSE} \quad \forall R_n \text{ and } R_m \text{ adjacent.}$

Here, Condition 1 states that the partition has to cover the whole image, whereas Condition 2 indicates that each region has to be homogeneous with respect to the predicate H , and finally Condition 3 states that the two adjacent regions cannot be merged into a single region that satisfies the predicate H .

In a large number of applications in image processing and computer vision, segmentation plays a fundamental role as the first step before applying to images higher-level operations, such as recognition, semantic interpretation and representation. Image segmentation has important applications in vision-guided autonomous robotics, product quality inspection, medical diagnosis and the analysis of remotely sensed images [3].

2.2 Main Approaches to Image Segmentation

The problem of image segmentation is an important research field and many segmentation methods have been proposed in the literature. The methods can be divided into 4 main categories:

- 1) clustering methods,
- 2) region-based methods,

3) hybrid methods, and

4) Bayesian methods.

2.2.1 Clustering Methods

Clustering may be defined as the process of organizing objects into groups whose members are similar in some way. The following definitions may be functional [4]:

- i) A *cluster* is a set of entities, which are “alike”, while entities from different clusters are not “alike”.
- ii) A *cluster* is an aggregation of points in the test space such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it.
- iii) *Clusters* may be described as connected regions of a multi-dimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points.

Research into clustering algorithms has been useful in many applications, mainly in the field of pattern recognition and data mining. Clustering methods can be divided into two categories: hierarchical and partitional [5]. Within each of the

types there exists a wealth of subtypes and different algorithms for finding the clusters.

2.2.1.1 Hierarchical Clustering

Hierarchical clustering techniques are based on the use of a proximity matrix indicating the similarity between every pair of data points to be clustered. The end result is a tree of clusters, called a *dendrogram* representing the nested grouping of patterns and similarity levels at which groupings change. It proceeds successively by either merging smaller clusters into larger ones (agglomerative, bottom-up), or by splitting larger clusters (divisive, top-down). By cutting the *dendrogram* at a desired level, a clustering of data items into disjoint groups is obtained. The clustering methods differ in regards to the rules by which two small clusters are merged or a large cluster is split. Some of the hierarchical algorithms include Cobweb, Cure and Chameleon [5].

2.2.1.2 Partitional Clustering

Partitional clustering attempts to directly decompose the data set into a set of disjoint clusters. An objective function expresses the wellness of a representation, and then the clustering algorithm tries to minimize this function in order to obtain the best representation. Partitional algorithms are categorized into Partitioning

Relocation Algorithms and Density-Based Partitioning. Algorithms of the first type are further categorized into Probabilistic Clustering (SNOB), K-Medoids, and K-Means. The second type of partitional algorithms are called Density-Based Partitioning, they include algorithms such as Dbscan, Optics Dbclasd, Denclue, Gdbscan [5].

A hierarchical clustering is a nested sequence of partitions, whereas a partition clustering is a single partition. Thus, a hierarchical clustering is a special sequence of partitional clustering. At the end of hierarchical clustering process, one cluster tree is formed. Traveling down the branches, the subsequent merge steps can be seen. Using hierarchical clustering is only practical on small data sets. Hierarchical clustering methods are clearly not practical in image segmentation process [4].

Partitional clustering techniques such as, K-means clustering and ISODATA have an advantage over the hierarchical clustering techniques, where a partition of the data points which optimizes some criterion functions. In hierarchical clustering once a data point is assigned to a particular cluster, it cannot be altered. Therefore, if a data point is incorrectly assigned to a particular cluster at an early stage, there is no way to correct the error. However, there is also a disadvantage of the partitional clustering techniques on how to determine the number of clusters, K [5].

Clustering methods are global in the sense that they do not retain positional information. The major drawback is its invariance to spatial rearrangement of the pixels, which is an important aspect by the definition of segmentation. The resulting segments might not be connected and can be widely scattered. This also causes clustering methods be sensitive to noise and intensity inhomogeneities. This lack of spatial modeling, however, can provide significant advantages for fast computation [6] [7].

K-Means and Fuzzy C-Means [4] are well-known clustering-based image segmentation algorithms and they are examined in Chapter 3.

2.2.2 Region-Based Methods

The region-based methods try to isolate areas of images that are homogeneous according to given set of characteristics. Two classical region-based methods are seeded region growing and split-and-merge [6].

2.2.2.1 Seeded Region Growing

Seeded region growing is one of the most simple and popular region-based segmentation algorithms. It starts by choosing a (or some) starting point (or seed pixel). Then, the region grows by successively adding neighboring pixels that are

similar, according to a certain homogeneity criterion, increasing step by step the size of the region. This criterion can be, for example, to require that the variance of a feature does not exceed a threshold, or that the difference between the pixel and the average of the region is small. The growing process is continued until a pixel not sufficiently similar to be aggregated is obtained. It means that the pixel belongs to another object and the growing in this direction is complete. Monitoring the procedure gives an impression of regions in the interior of objects growing until boundaries correspond with the edges of the object. Important problems of seeded region growing are the selection of initial seeds that properly represent regions and the suitable homogeneity criterion to be used during the growing process [6].

2.2.2.2 Split-and-Merge

One of the basic properties of segmentation is the existence of a predicate P which measures the region homogeneity. If this predicate is not satisfied for some region, it means that that region is inhomogeneous and should be split into subregions. On the other hand, if the predicate is satisfied for the union of two adjacent regions, then these regions are collectively homogeneous and should be merged into a single region [1][2]. A method towards the satisfaction of these homogeneity criteria is the split-and-merge algorithm. This technique consists, as its name denotes, of two basic steps. First, the image is recursively split until all

the regions verify a homogeneity criterion. Next, in a second step, all adjacent regions are reassembled in a way that resulting regions satisfy the homogeneity criterion. The steps are shown in Fig. 2.1. [6]

The procedure can be summarized as follows [8][9]:

- i) If for any region R_i , $P(R_i) = \text{FALSE}$, then split R_i into four subquadrants.
- ii) If for any adjacent regions R_i and R_j , $P(R_i \cup R_j) = \text{TRUE}$, merge them.
- iii) If no further splitting or merging is possible, stop. Else go to step 1.

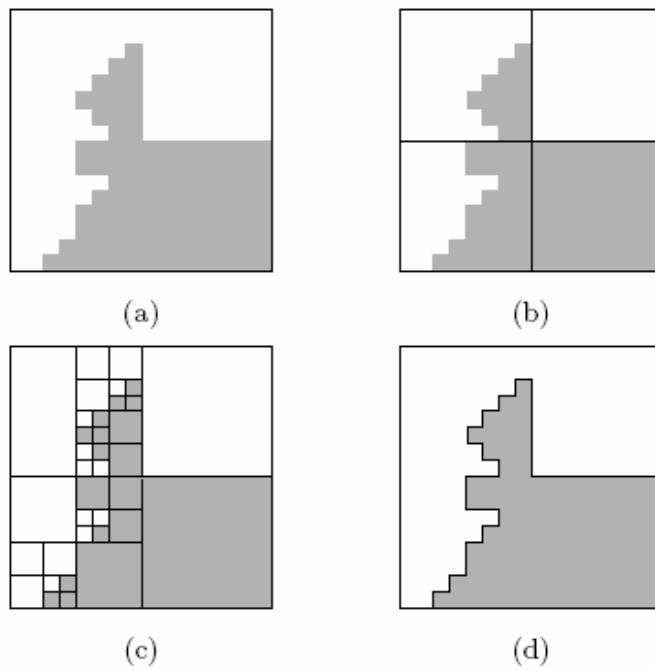


Figure 2.1: Split-and-merge segmentation, (a) original image, (b) initial split in four square blocks, (c) splitting of the image in homogeneous blocks and (d) final segmentation after the merging.

A quad-tree structure is often used to affect the step of splitting: it is based on the recursive decomposition of the regions that does not satisfy the homogeneity criterion into four squared subregions, starting from the whole image. Therefore, an inverse pyramidal structure is built. The merging step consists of merging the adjacent blocks which represent homogeneous regions but have been divided by the regular decomposition [6].

Split-and-merge method does not suffer from predetermination of number of regions, or any other constraints. However, the main drawback is the artificial blocking effects on the resulting region boundaries [9]. The main advantage offered by region-based methods is that the regions obtained are certainly spatially connected and rather compact.

Recursive Shortest-Spanning Tree algorithm [9], as will be explained in Chapter 3, is another popular region-based algorithm.

2.2.3 Hybrid Methods

Hybrid image segmentation methods combine the principles of two or more primitive image segmentation techniques, either in a hierarchical or parallel manner to segment the images [5]. As mentioned in previous sections, both clustering methods and region-based methods have advantages and disadvantages. Hybrid methods aim to get use of advantageous parts of different techniques. For

example, clustering methods usually suffer from unconnected and scattered regions. The appropriate use of a region-based algorithm may help to overcome this problem.

A typical example for hybrid methods is K-Means with Connectivity Constraint (KMC) algorithm [12], in which an initial K-Means clustering is further refined by considering the spatial coherence of neighboring pixels. KMC is examined in Chapter 3.

2.2.4 Bayesian Methods

Bayesian methods use probability calculus to quantify the plausibility of a hypothesis. In the case of image segmentation, this hypothesis is about the existence of a particular “hidden field” (label field realization) along with the data. A *priori* knowledge, which can be exploited to improve the results, is used to regularize the inference of the hidden field, given the data. Formal optimization techniques are then used to work on the posterior inference.

The Bayes rule states that:

$$P(L | X) \propto P(X | L)P(L),$$

i.e. the posterior probability $P(L|X)$ of the label field given the data is proportional to the product of the model probability $P(X|L)$ and the prior probability of the

label realizations $P(L)$. $P(L)$ is defined using local information about the expected segmentation result (such as shape, etc.) and aims at encouraging spatial connectivity. [15]

The *prior* probability model for the segmentation label field is usually assumed to be a Gibbs random field (GRF), which expresses the expectations about the spatial properties of the segmentation, i.e., the GRF assigns higher probabilities to the segmentation fields having connected regions.

The feature image is explicitly assumed as the summation of two parts; one is a piecewise constant function, and the other is a Gaussian white noise with zero mean, μ , and variance, σ^2 . The segmentation is achieved by maximizing the *a posteriori* probability of the segmentation field, given the observed feature image.

The mathematical formulation is as follows:

The segmentation label field $Z(x,y)$ is modeled by

$$P(Z = z)\alpha \exp\{-U(z)\}$$

where $U(z)$ is the Gibbs potential and is defined by

$$U(z) = \sum_{c \in C} V_c(z).$$

Here C is the set of all cliques, and V_c is the individual clique potential whose value depends only on $z(x,y)$ where $(x,y) \in C$. A *clique* is a set of points, c , which are all neighbours of each other. Clique examples for first order and second

order neighbourhood systems are shown in Fig. 2.2 [17]. Spatial connectivity of regions can be imposed by assigning low values to $V_c(z)$, if $z(x,y)$ is constant for all $(x, y) \in C$, and high values otherwise.

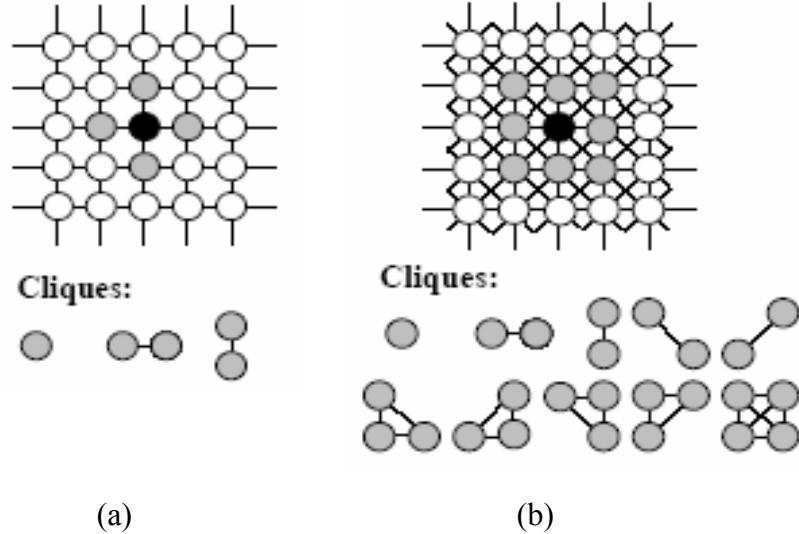


Figure 2.2: Clique examples for (a) first order neighbourhood system, (b) second order neighbourhood system

According to the above assumption about the formation of the image, the conditional probability of the observed feature image S , given Z is modeled by

$$P(S = s | Z = z) \propto \exp \left\{ \frac{-1}{2\sigma^2} \left[\sum_{i=1}^K \sum_{(x,y) \in R_i} \|s(x,y) - \mu_i\|^2 \right] \right\}.$$

The *a posteriori* probability can be manipulated using the Bayes rule:

$$P(Z = z | S = s) = \frac{P(S = s | Z = z)P(Z = z)}{P(S = s)}.$$

Then, maximizing $P(Z=z|S=s)$ is equivalent to minimizing

$$D' = \sum_{i=1}^K \sum_{(x,y) \in R_i} \|s(x,y) - \mu_i\|^2 + \lambda \sum_{c \in C} V_c(z)$$

with respect to the segmentation mask $z(x,y)$ [9].

CHAPTER 3

FUNDAMENTAL METHODS IN IMAGE SEGMENTATION

3.1 Motivation

In this chapter, some fundamental solutions to image segmentation problem are examined and tested to get some experience on their performances.

Following the classification of image segmentation methods in Chapter 2, K-Means and Fuzzy C-Means are chosen as typical examples of clustering-based algorithms. On the other hand, RSST and seeded region growing are tested among the methods in region-based approaches. Finally, as a typical example of hybrid methods, KMC algorithm is presented.

The screenshots of the programs for the algorithms can be seen in Figs. 3.1 and 3.2. The test images that are utilized during simulations are also shown in Figs. 3.3 to 3.7, while the ground truth images are presented in Figs. 3.8 to 3.12. The algorithms are implemented on Borland C++ Builder version 5.0 and run on an AMD Athlon XP 1800+ computer with 512 MB DDR ram.

In evaluation of the experimental results, two metrics, namely ground truth error and normalized variance, are used. Ground truth error value represents how much the segmentation result differs from the ideal segmentation mask.

On the other hand, normalized variance measures the compactness of the individual clusters. A natural measure of compactness is the average within-cluster variance. For a single cluster, the variance, $V(k)$, $k=1, \dots, K$, is given by

$$V(k) = \frac{1}{N_k} \sum_{s(x) \in C(k)} (s(x) - M(k))^2$$

where N_k is the number of pixels belonging to cluster k , $s(x)$ is the feature vector for pixel x , $C(k)$ is the cluster array and $M(k)$ is the cluster center for cluster k , $k=1, \dots, K$.

Averaging variances over K clusters gives

$$V = \sum_{k=1}^K N_k V(k) / N$$

where N is the total number of pixels in the data set.

It is useful to normalize this value by the overall variance, V_0 ,

$$V_0 = \sum_{x=1}^N (s(x) - M)^2 / N$$

where M is the mean of the all pixels.

So the normalized variance, V^* , is

$$V^* = \frac{V}{V_0} .$$

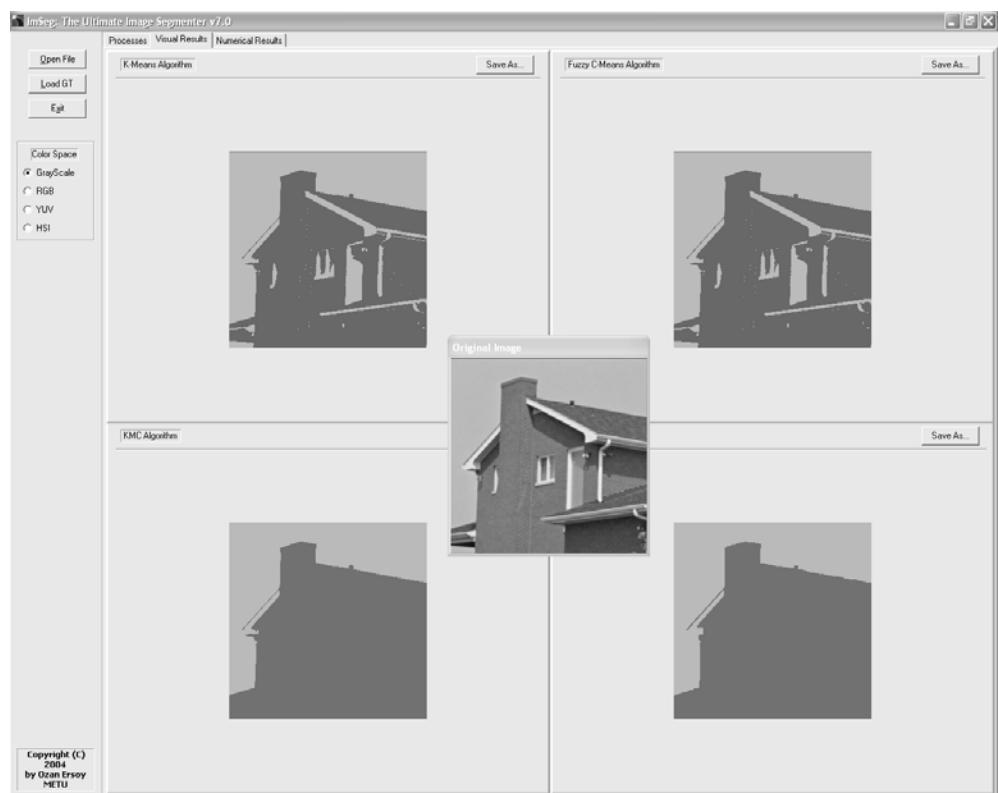


Figure 3.1: A screenshot of the utilized software during simulations.

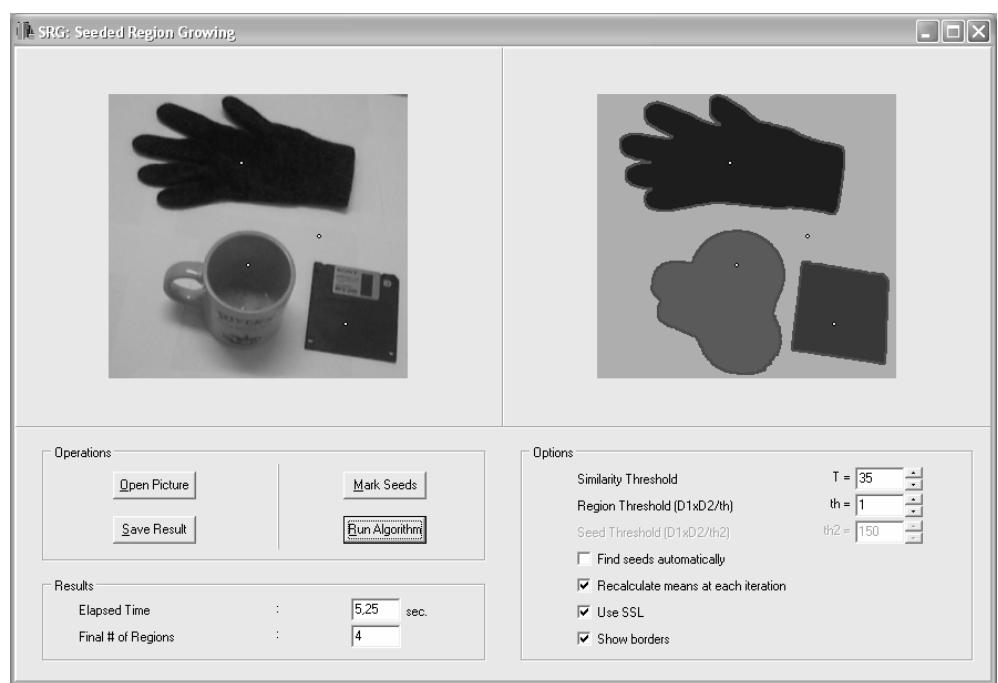


Figure 3.2: A screenshot of the utilized seeded region growing software.



Figure 3.3: Test image 1.

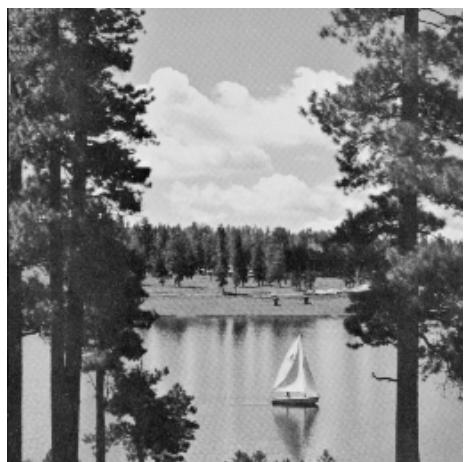


Figure 3.4: Test image 2.



Figure 3.5: Test image 3.



Figure 3.6: Test image 4.



Figure 3.7: Test image 5.



Figure 3.8: Ground truth image of test image 1 ($K=2$).



Figure 3.9: Ground truth image of test image 2 ($K=3$).

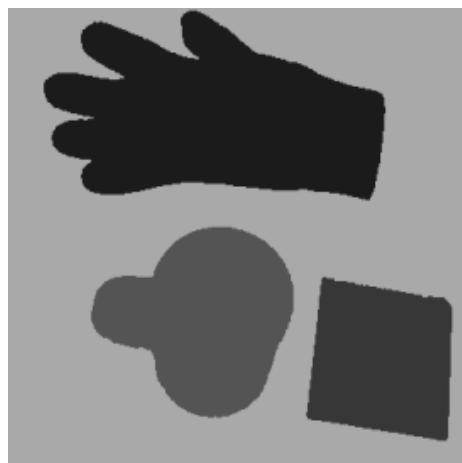


Figure 3.10: Ground truth image of test image 3 ($K=4$).



Figure 3.11: Ground truth image of test image 4 ($K=3$).



Figure 3.12: Ground truth image of test image 5 ($K=3$).

3.2 K-Means Algorithm

The basic idea of K-Means algorithm is to start with an initial partition and assign patterns to clusters in order to reduce the error between the intensities of the pixels of a cluster and its mean. The error tends to decrease as the number of iterations increases, as expected is minimized for a fixed number of clusters, K [4].

3.2.1 Algorithm

The flowchart of K-Means algorithm is given in Fig. 3.13 [13]. The algorithm is as follows [4]:

Step 1

Select a cluster number K . Choose the mean values for K initial cluster centers randomly, as $M_j(1)$, $M_j(2)$, ..., $M_j(K)$, where j is the iteration number. In this thesis initial centers are determined from the analysis of gray-level histograms.

Step 2

Assign each pixel x to one of the clusters according to the relation

$x \in C(k)$ if $\|s(x) - M_j(k)\| < \|s(x) - M_j(i)\|$ for all $i=1, \dots, K$ and $i \neq k$

where $s(k)$ is the feature vector for pixel x and $C(k)$ is the cluster array.

Step 3

Using new clusters $C(k)$, $k=1, \dots, K$, compute their cluster centers. The new cluster centers, $M_{j+1}(k)$, $k=1, \dots, K$, can be determined as the averages of the patterns in each cluster as follows:

$$M_{j+1} = \frac{1}{N_k} \sum_{x_i \in C(k)} s(x_i)$$

where N_k is the number of pixels in cluster k .

Step 4

Using the new cluster centers computed in the previous step, check whether the clustering is finished. Convergence is occurred, if none of the cluster centers are changed, that is

if $M_j(k) = M_{j+1}(k)$ for $k=1, \dots, K$

if $j > \text{MAXIT}$ then stop. If $j < \text{MAXIT}$ then increment j by 1 and return to step 2.

If convergence occurs clustering is complete and final K clusters should be obtained. Convergence could be computationally very expensive, and hence, as a safety margin, a maximum number of iterations called MAXIT can be assigned for stopping without convergence. In this thesis, MAXIT value is taken as 10.

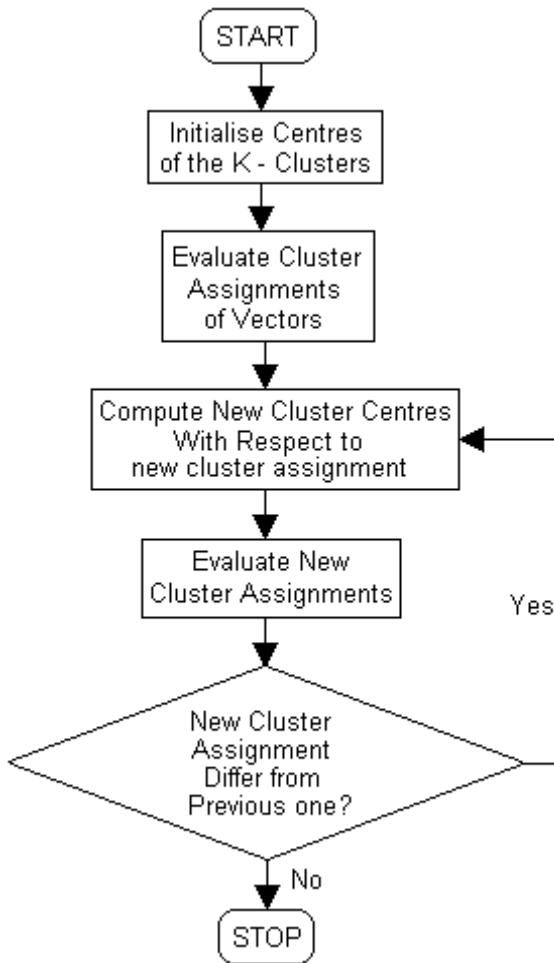


Figure 3.13: The flowchart of K-Means Algorithm.

3.2.2 Results

The test images of Figs. 3.3 to 3.7 are utilized to assess the performance of K-Means algorithm. During the simulations, cluster numbers are selected equal to the values which are used to define their respective ground truths in Figs. 3.8 to 3.12. Segmented images are displayed in Figs. 3.14 to 3.18 and the results are summarized in Table 3.1. Observing the resultant images, one can easily state that K parameter only fixes the number of different gray levels the image contains. Hence, in all these images there are many (more than K) connected regions (but only K gray levels). However, in many applications, one should only be interested in connected regions which have semantic meanings. Clustering based methods simply can not deliver this information.

Table 3.1: Numerical results of K-Means Algorithm for test images.

IMAGE	K	# of Iterations	Normalized Variance	Ground Truth Error	Elapsed Time (sec)
Test Image 1	2	4	0,2171	0,0844	0,062
Test Image 2	3	4	0,0756	0,3940	0,062
Test Image 3	4	10(MAXIT)	0,0219	0,2190	0,172
Test Image 4	3	6	0,1746	0,0485	0,109
Test Image 5	3	5	0,1240	0,2529	0,094



Figure 3.14: Segmentation of test image 1 with K-Means.



Figure 3.15: Segmentation of test image 2 with K-Means.



Figure 3.16: Segmentation of test image 3 with K-Means.



Figure 3.17: Segmentation of test image 4 with K-Means.



Figure 3.18: Segmentation of test image 5 with K-Means.

3.3 Fuzzy C-Means Algorithm

In contrast to K-Means algorithm in which the pixels can have only two states for their membership to a region, in fuzzy clustering all the continuous membership degrees between zero and one are used instead of crisp assignments of the data to clusters. Fuzzy C-Means algorithm assigns each pixel a fuzzy membership for every class. The fuzzy membership corresponds to the probability of a particular pixel belonging to a particular class. Therefore, every pixel belongs to every class at the same time with different probabilities [18].

3.3.1 Algorithm

The algorithm is as follows [4]:

Step 1

Select an initial partition with K clusters. Find random values for each fuzzy centroids V_k , for $k=1, \dots, K$. Subscript k on V_k shows the cluster number.

Fuzzy centroids are defined as

$$V_k = \frac{\sum_{i=1}^N \mu_k(x_i)^m x_i}{\sum_{i=1}^N \mu_k(x_i)^m}$$

where N is the total pixel number in the image and $\mu_k(x_i)$ is the fuzzy membership function. m is an arbitrary weighting exponent. The larger the chosen m the fuzzier is the assignment of membership to the cluster. $m=2$ gives the same segmentation results as K-Means algorithm. In this thesis, $m=10$ is chosen.

Step 2

Set $p = 1$ where p is the iteration index.

Step 3

Using the V_k , $k=1, \dots, K$, determined in step 1, compute the new fuzzy membership functions $\mu_k(x_i)$ as follows

$$\mu_k^p(x_i) = \frac{\left(\frac{1}{d^2(x_i, V_k^p)} \right)^{1/(m-1)}}{\sum_{k=1}^K \left(\frac{1}{d^2(x_i, V_k^p)} \right)^{1/(m-1)}}$$

where $i = 1, \dots, N$ and $k = 1, \dots, K$

Step 4

Compute the new fuzzy centroids V_k , for $k=1, \dots, K$ using

$$V_k^{p+1} = \frac{\sum_{i=1}^N \mu_k^p(x_i)^m x_i}{\sum_{i=1}^N \mu_k^p(x_i)^m}$$

Step 5

If $\mu^p = \mu^{p+1}$ stop. Else set $p = p+1$ and go to step 3.

Distance measure, $d(x, y)$, is the Euclidian distance [4] :

$$d(x, y) = \|x - y\| = \sqrt{(x - y)^T (x - y)} = \sum_{i=1}^n [(x_i - y_i)^2]^{1/2} .$$

3.3.2 Results

The test images of Figs. 3.7 to 3.11 are utilized to assess the performance of Fuzzy C-Means algorithm. During the simulations, cluster numbers are selected equal to the values which are used to define their respective ground truths in Figs. 3.8 to 3.12. Segmented images are displayed in Figs. 3.19 to 3.23 and the results

are summarized in Table 3.2. Comparing the resultant images of K-Means and Fuzzy C-Means, one can observe that they are quite similar. Moreover, there are no noticeable differences between normalized variances and ground truth errors of two algorithms. However, elapsed times of Fuzzy C-Means are significantly high compared to elapsed times of K-Means algorithm. Apparently, the fuzziness concept utilized by Fuzzy C-Means does not improve the segmentation performance of the algorithm. Due to ignorance of spatial information in clustering algorithms, their segmentation performances are limited to a degree and expecting lower ground truth errors is pointless. In this sense, normalized variance is a more realistic metric for performance measure of the clustering algorithms. Normalized variances obtained in K-Means algorithm are sufficiently low.

Table 3.2: Numerical results of Fuzzy C-Means Algorithm for test images.

IMAGE	K	# of Iterations	Normalized Variance	Ground Truth Error	Elapsed Time (sec)
Test Image 1	2	5	0,2171	0,0844	1,938
Test Image 2	3	6	0,0755	0,4045	3,593
Test Image 3	4	10	0,0244	0,2098	9,016
Test Image 4	3	10	0,1746	0,0527	6,953
Test Image 5	3	10	0,1240	0,2611	7,203



Figure 3.19: Segmentation of test image 1 with Fuzzy C-Means.



Figure 3.20: Segmentation of test image 2 with Fuzzy C-Means.

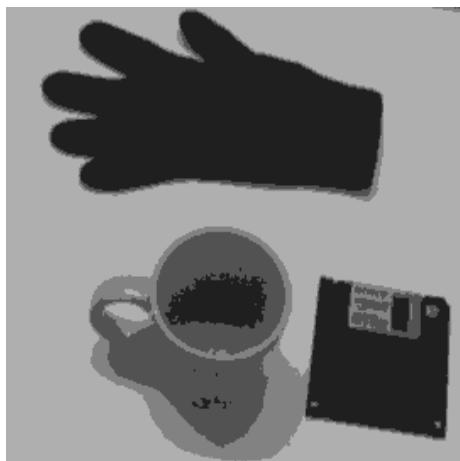


Figure 3.21: Segmentation of test image 3 with Fuzzy C-Means.



Figure 3.22: Segmentation of test image 4 with Fuzzy C-Means.



Figure 3.23: Segmentation of test image 5 with Fuzzy C-Means.

3.4 Seeded Region Growing

The Seeded Region Growing (SRG) approach to image segmentation is based on dividing an image into regions with respect to a set of n seed points. It starts by choosing a (or some) starting point or seed pixel. Then, the regions grow by successively adding neighboring pixels that are similar, according to a certain homogeneity criterion, increasing step by step the size of the region [6].

3.4.1 Algorithm

Each seed region is a connected component, comprising one or more points and is represented by a set A_i , where $i = 1, \dots, n$. Let T be the set of all unallocated pixels that border at least one of the A_i , i.e.

$$T = \{x \notin \bigcup_{i=1}^n A_i : N(x) \cap \bigcup_{i=1}^n A_i \neq \emptyset\},$$

where $N(x)$ represents the set of immediate neighbours of the pixel x . A single step of the algorithm involves examining the neighbours of each $x \in T$ in turn. If $N(x)$ intersects a region A_j then a measure $\delta(x)$, of the difference (similarity) between x and the intersected region is calculated. In the simplest case, $\delta(x)$ is defined:

$$\delta(x) = |g(x) - \text{mean}_{y \in A_j} \{g(y)\}|,$$

where $g(x)$ is the intensity (grey value) of the pixel x . If $N(x)$ intersects more than one region, then A_j is taken to be that region for which $\delta(x)$ is a minimum. In this way, a δ value is determined for each $x \in T$. Finally, the pixel $z \in T$ that satisfies

$$\delta(z) = \min_{x \in T} \{\delta(x)\}$$

is appended to the region corresponding to $\delta(z)$. The new state of the regions $\{A_i\}$ then constitutes the input to the next iteration. This process continues until all of the image pixels have been assimilated. [10]

While implementing the SRG algorithm, a data structure denoted as the sequentially sorted list (SSL), is utilized. SSL is a linked list of pixel addresses, ordered with respect to δ . A pixel can arbitrarily be inserted into the list in the position prescribed by its δ value. However, only the pixel with the smallest δ value can be removed from the SSL. Effectively, the SSL stores the points of the set T ordered according to δ [10].

3.4.2 Results

The test images of Figs. 3.7 to 3.11 are utilized to assess the performance of SRG algorithm. During the simulations, region numbers are selected equal to the values which are used to define their respective ground truths in Figs. 3.8 to 3.12. Segmented images are displayed in Figs. 3.24 to 3.28 and the results are

summarized in Table 3.3. Apparently, the ground truth errors of SRG are much better than ground truth errors of K-Means and Fuzzy C-Means algorithms. However, elapsed times of SRG are considerably high compared to K-Means. Nevertheless, one can still state that SRG is a more powerful segmentation algorithm in the sense of connected regions. A drawback of SRG is the assignment of initial seed points. The seed points can be automatically or manually selected. However, it is not always easy to find suitable initial seed points. During the simulations, seed points are assigned manually.

Table 3.3: Numerical results of SRG Algorithm for test images.

IMAGE	K	Normalized Variance	Ground Truth Error	Elapsed Time (sec)
Test Image 1	2	0,4635	0,0067	2,312
Test Image 2	3	0,3955	0,0142	0,765
Test Image 3	4	0,0751	0,0594	5,344
Test Image 4	3	0,3358	0,0344	8,531
Test Image 5	3	0,3922	0,1057	10,922

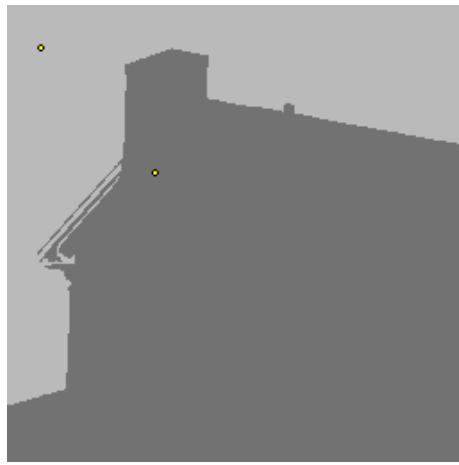


Figure 3.24: Segmentation of test image 1 with SRG.

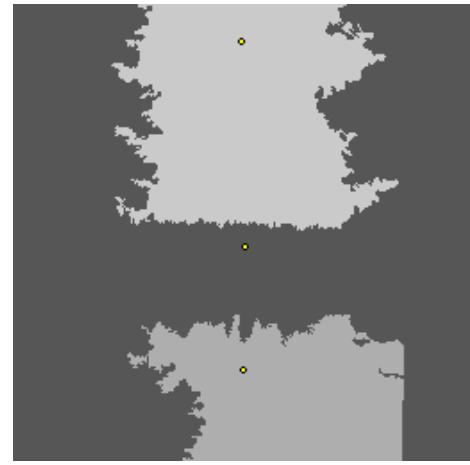


Figure 3.25: Segmentation of test image 2 with SRG.

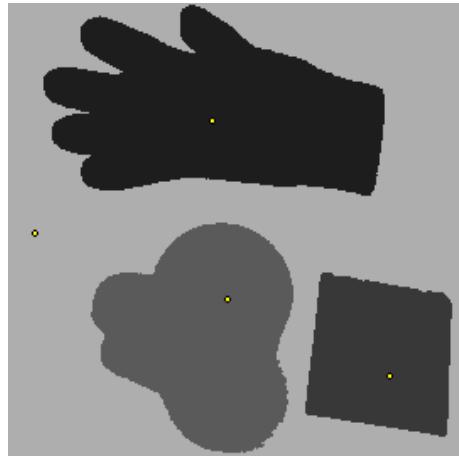


Figure 3.26: Segmentation of test image 3 with SRG.

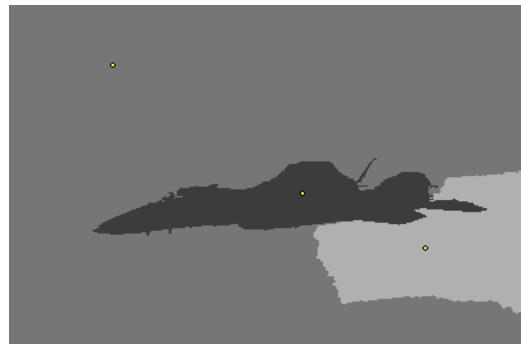


Figure 3.27: Segmentation of test image 4 with SRG.

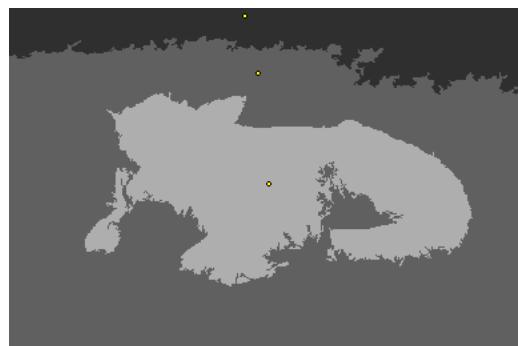


Figure 3.28: Segmentation of test image 5 with SRG.

3.5 RSST

The RSST method is a powerful image segmentation method in the sense that it is relatively fast and requires no initial segmentation masks and parameters. Furthermore, it is a hierarchical segmentation scheme that is yielding segmentation masks of various scales, from the finest to the coarsest, as the algorithm evolves from the finest level to the coarsest levels, it may be stopped, when the number of regions is reduced to the desired number, K , which should be specified externally [11].

3.5.1 Fundamental Definitions in Graph Theory

Graph Theory is the study of graphs and their applications. A graph $G = (V, E)$ is made up of a set of *vertices* V_i and V_j connected to each other by *links* $E_{i,j}$, for $i \neq j$, and where V_i and V_j are the terminal vertices that the link connects. In a weighted graph the vertices and links have weights associated with them, namely v_i and $e_{i,j}$, respectively. Each vertex is not necessarily linked to every other, but if the vertices are linked together then the graph is complete. A *partial graph* has the same number of vertices but only a subset of the links of the original graph. A *chain* is a list of successive vertices in which each vertex is connected to the next vertex by a link in the graph. A *cycle* is a chain whose end links meet at the same vertex. A *tree* is a connected set of chains such that there are no cycles. A

spanning tree is a tree, which is also a partial graph. The *shortest spanning tree* of a weighted graph is a spanning tree such that the sum of its link weights, or some other monotonic function of its link weights, is a minimum for any possible spanning tree [11].

3.5.2 Algorithm

The RSST algorithm consists of two functional blocks, namely, the *initialization stage* and the *linking process*. The flowchart of the RSST algorithm is given in Fig. 3.30. The RSST starts with a mapping of an image onto a weighted graph at the initialization stage. Each region or vertex initially contains only one pixel. The pixel intensity values of regions are used to evaluate vertex weights and link weights of the graph. A vertex weight (V_i) is defined as the average intensity value of the corresponding region, while a link weight (LW_i) is evaluated by a link weight function or a cost function, which is basically a function of the vertex weights and the sizes (N_i) of the connected regions, i.e.

$$LW_i = |V_i - V_j| \frac{N_i N_j}{N_i + N_j}$$

All links are then sorted in order according to their link weights, and stored in a heap. In entering the linking process, a link with the least link weight in the graph is chosen to be the next link of the shortest spanning tree (SST). The chosen link

is saved and the connecting regions are merged. The vertex weight of the newly merged region is updated, hence, all surrounding links need to be recalculated and all loop-forming links, also known as duplicated links, will be removed. Merging procedure is illustrated in Fig. 3.29. Subsequently, all remaining links are sorted. Thus, the number of regions is progressively reduced from $M \times N$ in an M pixel by N pixel image, down to just one if desired. Those saved links form a spanning tree representation of the image. By noting the order in which the links are saved, hierarchical representation of the original image can also be created [11].

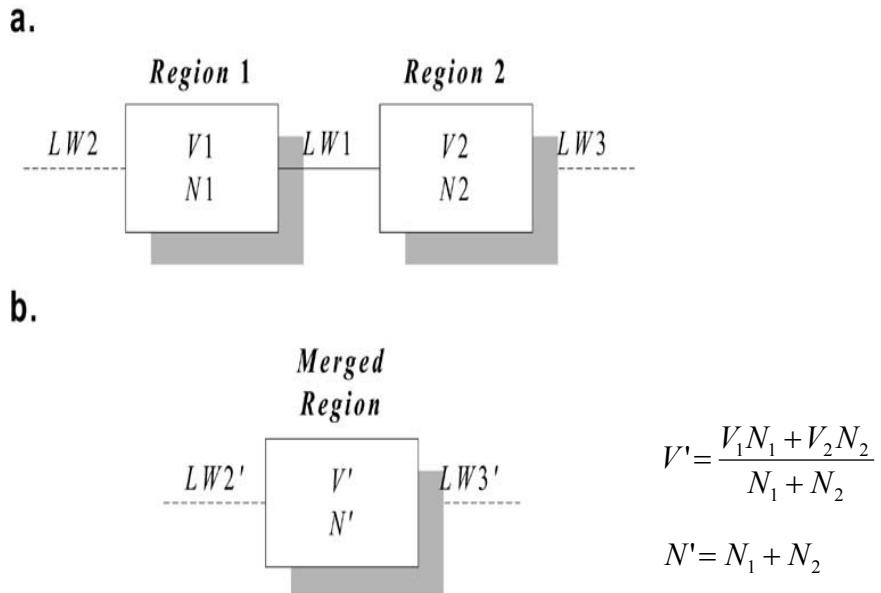


Figure 3.29: Merging procedure,

- (a) Regions before merging ($LW1$ is assumed to be the smallest link weight).
- (b) Merged region with new V' and N' .

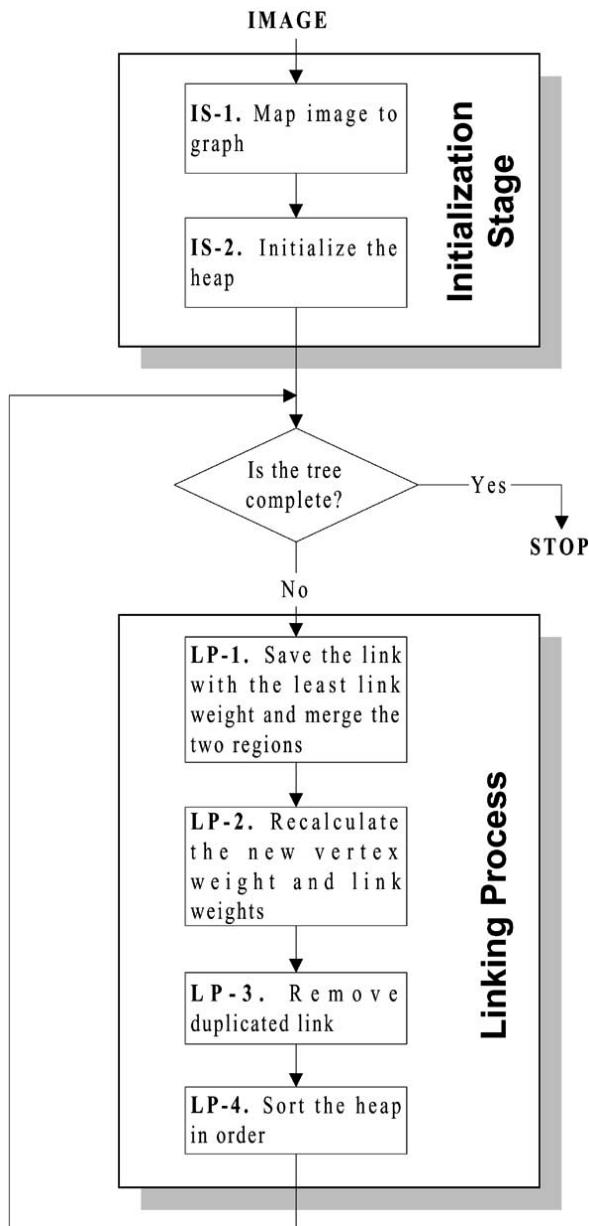


Figure 3.30: The flowchart of the RSST algorithm.

3.5.3 Results

The test images of Figs. 3.7 to 3.11 are utilized to assess the performance of RSST algorithm. During the simulations, region numbers are selected equal to the values which are used to define their respective ground truths in Figs. 3.8 to 3.12. Segmented images are displayed in Figs. 3.31 to 3.35 and the results are summarized in Table 3.4. The ground truth errors of RSST are slightly higher than the ground truth errors of SRG. On the other hand, elapsed times of RSST are significantly low compared to elapsed times of SRG. Moreover, RSST does not require initial parameters other than assignment of number of regions, K . Thus, RSST appears to be an effective image segmentation algorithm.

Table 3.4: Numerical results of RSST Algorithm for test images.

IMAGE	K	Normalized Variance	Ground Truth Error	Elapsed Time (sec)
Test Image 1	2	0,4522	0,0112	0,625
Test Image 2	3	0,3600	0,0613	0,563
Test Image 3	4	0,0871	0,0842	0,531
Test Image 4	3	0,2564	0,0427	0,859
Test Image 5	3	0,3644	0,1043	0,875



Figure 3.31: Segmentation of test image 1 with RSST.



Figure 3.32: Segmentation of test image 2 with RSST.

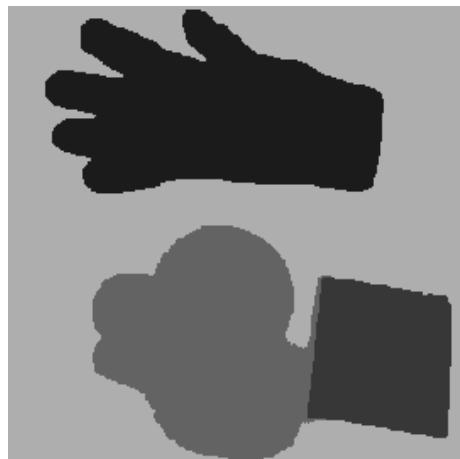


Figure 3.33: Segmentation of test image 3 with RSST.



Figure 3.34: Segmentation of test image 4 with RSST.



Figure 3.35: Segmentation of test image 5 with RSST.

3.6 KMC

Clustering based on the K-Means algorithm is a widely used region segmentation method which, however, tends to produce unconnected regions. This is due to the propensity of the classical K-Means algorithm to ignore spatial information about the intensity values in an image, since it only takes into account the global intensity or color information. In order to alleviate this problem, an extended version of K-Means algorithm, namely KMC, is proposed [12]. In this algorithm, the spatial proximity of each region is also taken into account by defining a new center for the K-Means algorithm and by integrating the K-Means with a component labeling procedure.

3.6.1 Algorithm

The algorithm is as follows [12]:

Step 1

The classical KM algorithm is performed for a small number of iterations. This result in K regions s_i , $i=1, \dots, K$, with intensity centers I_k , $k=1, \dots, K$. If M_k elements are assigned to s_k then

$$I_k = \frac{1}{M_k} \sum_{m=1}^{M_k} I(p_m^k)$$

where p_m^k , $m=1, \dots, M_k$, are the pixels belonging to region s_k ,

and spatial centers $S_k = (S_{k,x}, S_{k,y})$, $k=1, \dots, K$

$$S_{k,x} = \frac{1}{M_k} \sum_{m=1}^{M_k} p_{m,x}^k$$

$$S_{k,y} = \frac{1}{M_k} \sum_{m=1}^{M_k} p_{m,y}^k$$

where $p_k = (p_x^k, p_y^k)$.

The area of each region A_k is defined by $A_k = M_k$

and the mean area of all regions $A = \frac{1}{K} \sum_{k=1}^K A_k$.

Step 2

For every pixel $p = (x, y)$, the intensity differences are evaluated between center and pixel intensities as well as the distances between p and S . A generalized distance of a pixel p from a region s_k is defined as follows:

$$D(p, k) = \frac{\lambda_1}{\sigma_I^2} \|I(p) - I(k)\| + \frac{\lambda_2}{\sigma_S^2} A \frac{\|p - S_k\|}{A_k}$$

where $\|p - S_k\|$ is the Euclidian distance, σ_I , σ_S are the standard deviations of intensity and spatial distance, respectively, and λ_1 , λ_2 are regularization parameters. Normalization of the spatial distance $\|p - S_k\|$ with the area of each region A/A_k is necessary in order to allow the creation of large connected regions; otherwise pixels with similar intensity with those of a large region would be assigned to neighboring smaller regions. If $|D(p,i)| < |D(p,k)|$ for all $k \neq i$, $p = (x, y)$ is assigned to region s_i .

Step 3

Based on the subdivision, and eight connectivity component labeling algorithm is applied. This algorithm finds all connected components and assigns a unique value to all pixels in the same component. Regions whose area remains below a predefined threshold are not labeled as separate regions. The component labeling algorithm produces L connected regions. For these

connected regions, the intensity I_1 and the spatial S_1 are calculated.

Step 4

If the difference between the new and the old centers I_1 and S_1 is below a threshold,

$$\frac{1}{L} \sum_{l=1}^L \left(\frac{\lambda_1}{\sigma_I^2} \|I_l^i - I_l^{i-1}\| + \frac{\lambda_2}{\sigma_S^2} A \frac{\|S_k^i - S_k^{i-1}\|}{A_k} \right) \leq \text{threshold}$$

then stop, else goto *Step 2* with $K = L$ using the new intensity and spatial centers. Convergence could be computationally very expensive, and hence, as a safety margin, a maximum number of iterations called MAXIT can be assigned for stopping without convergence [4]. In this thesis, MAXIT value is taken as 10.

Through the use of this algorithm, the ambiguity in the selection of the number of regions, K , which is a weakness of the K-Means algorithm, is resolved. Starting from any K , the component labeling algorithm produces or rejects regions according to their compactness. In this way K is automatically adjusted during the segmentation procedure [12].

3.6.2 Results

The test images of Figs. 3.7 to 3.11 are utilized to assess the performance of KMC algorithm. During the simulations, initial cluster numbers are selected equal to the values which are used to define their respective ground truths in Figs. 3.8 to 3.12 and the other initial parameters of the algorithm are adjusted, so that final numbers of regions are equal to initial cluster numbers. Segmented images are displayed in Figs. 3.36 to 3.40 and the results are summarized in Table 3.5. The ground truth errors of KMC are slightly lower than the ground truth errors of RSST, but still higher than ground truth errors of SRG. However, elapsed times of KMC are significantly high compared to elapsed times of SRG and RSST, especially in textured images. Connected components labeling algorithm utilized in KMC enables to achieve connected regions, but increases the execution time.

Table 3.5: Numerical results of KMC Algorithm for test images.

IMAGE	K	# of Iterations	Normalized Variance	Ground Truth Error	Elapsed Time (sec)
Test Image 1	2	4(KM)+1	0,4487	0,0101	0,656
Test Image 2	3	4(KM)+2	0,3612	0,0454	6,141
Test Image 3	4	4(KM)+6	0,0641	0,0611	4,187
Test Image 4	3	4(KM)+2	0,3214	0,0406	3,11
Test Image 5	3	4(KM)+1	0,3862	0,0965	17,453



Figure 3.36: Segmentation of test image 1 with KMC.



Figure 3.37: Segmentation of test image 2 with KMC.



Figure 3.38: Segmentation of test image 3 with KMC.

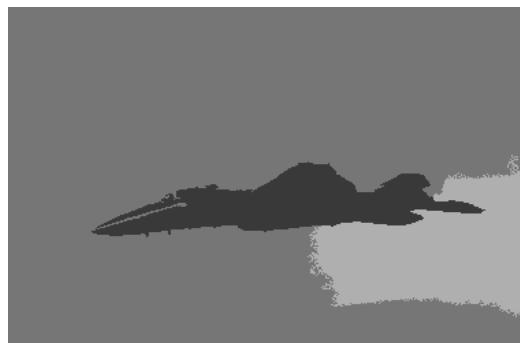


Figure 3.39: Segmentation of test image 4 with KMC.



Figure 3.40: Segmentation of test image 5 with KMC.

3.7 Comparative Analysis

Algorithms are applied on 5 test images. Normalized variance and ground truth error are utilized as performance metrics. Both metric values are bounded between 0 and 1 with smaller values indicating better segmentation. Normalized variance can be defined as the average within-cluster intensity variance. Ground truth error value represents how much the segmentation result differs from the ideal segmentation mask. The comparison of normalized variance, ground truth error and elapsed time values of algorithms can be seen in Figs. 3.41, 3.42 and 3.43, respectively.

The experimental results show that the normalized variances calculated for the segmentation results of clustering algorithms (K-Means and Fuzzy C-Means) are better than those of region-based algorithms. This is expected due to the fact that normalized variance only measures the amount of intensity values of pixels in clusters, differ from the cluster means in average. The fuzziness concept included in Fuzzy C-Means algorithm causes the computational burden of this algorithm to increase significantly, however it does not improve the segmentation quality much according to normalized variance values compared to K-Means algorithm.

On the other hand, ground truth error values are smaller for region-based algorithms (SRG and RSST) compared to clustering algorithms.

The KMC algorithm makes use of both intensity values and spatial information of the pixels. In this sense, the KMC is classified as a hybrid method. The connected-component labeling algorithm embedded in the KMC enables to get connected regions and also provides a dynamic determination of number of regions regardless of the initial K value. However, the computational complexity of the algorithm is significantly high. The results are summarized in Table 3.6.

Among the analyzed algorithms, the RSST has proven to be a good choice in the sense it is fast and in most cases give reasonable results.

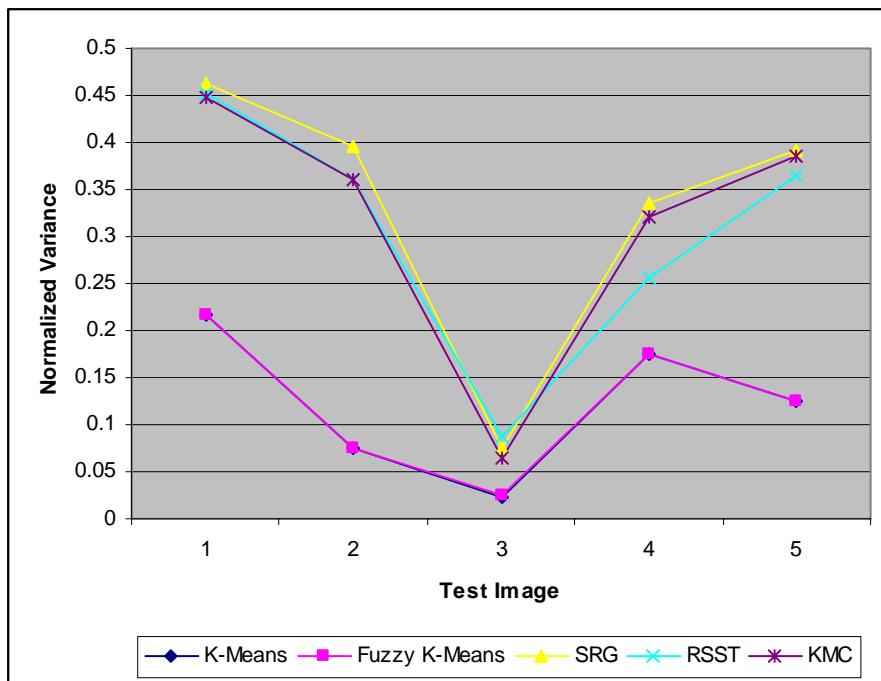


Figure 3.41: Chart showing Normalized Variances of algorithms during simulations.

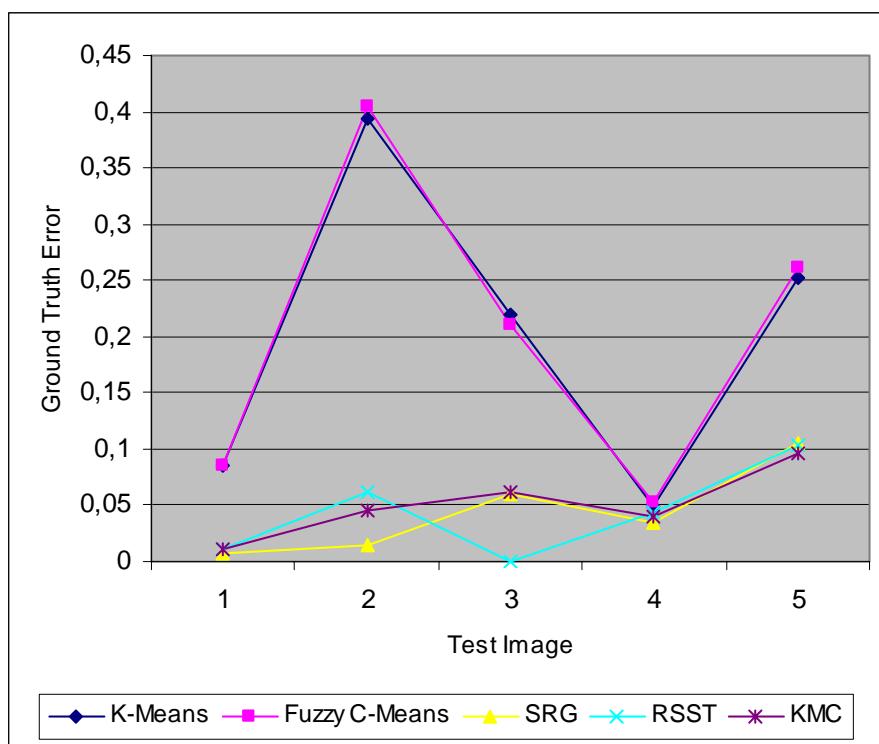


Figure 3.42: Chart showing Ground Truth Errors of algorithms during simulations.

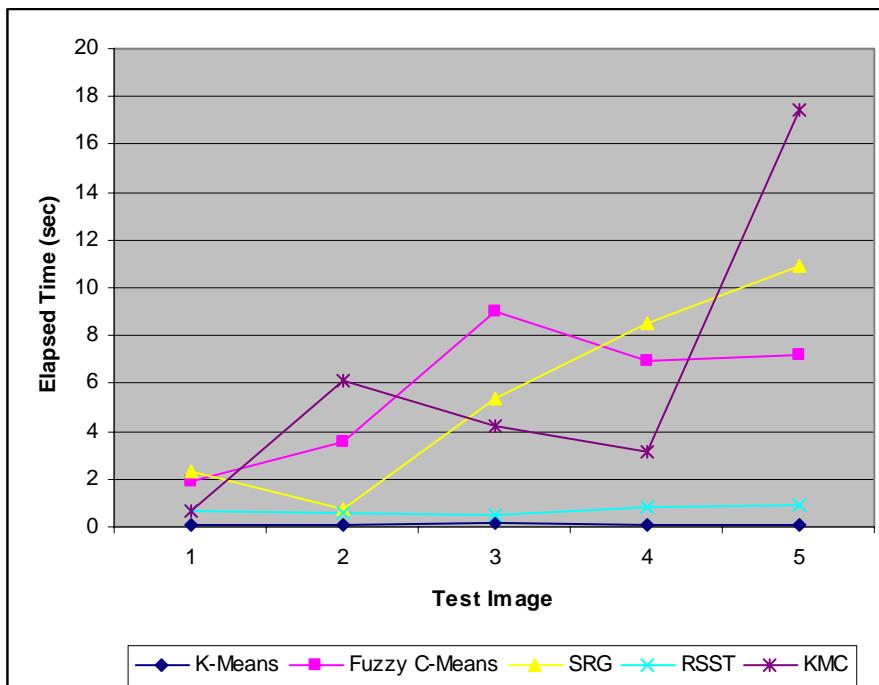


Figure 3.43: Chart showing Elapsed Times of algorithms during simulations.

Table 3.6: Summary of the simulation results.

Approach	Method	Normalized Variance	Ground Truth Error	Elapsed Time	Connected Regions
Clustering	K-Means	LOW	HIGH	LOW	NO
	Fuzzy C-Means	LOW	HIGH	AVERAGE	NO
Region-Based	SRG	HIGH	LOW	AVERAGE	YES
	RSST	HIGH	LOW	LOW	YES
Hybrid	KMC	HIGH	LOW	AVERAGE	YES

CHAPTER 4

RSST WITH IMPROVED REGION MODELING

4.1 Introduction

The experimental results have shown that RSST algorithm is a fast and powerful image segmentation algorithm. In RSST, each region is represented with its intensity mean. However, especially in textured parts of the images, the intensity mean is not adequate to explain the characteristics of a region. In this chapter, improvements to region modeling of RSST algorithm are examined. During the implementation of the proposed algorithms, first, test images are segmented into 256 regions using classical RSST algorithm. Next, the proposed algorithms are applied to merge these 256 regions into desired number of regions. Test images are also segmented into 25 regions using K-Means algorithm and the results are used while preparing the histograms in the algorithms. This is done to reduce the computational load of the algorithms. Test images segmented into 256 regions with RSST algorithm are shown in Figs. 4.3 to 4.7.

Test images 2, 3, 5 and two new test images 6 and 7 are utilized during simulations. Test images 6 and 7 are shown in Figs. 4.1 and 4.2. These images are

especially selected due to their textural content. Test image 6 is an artificial image, created by merging different textures together. On the other hand, test image 7 is a natural scene with significant amount of textured areas. The proposed algorithms are implemented on Matlab version 6.5 and run on an AMD Athlon XP 1800+ computer with 512 MB DDR ram.

In evaluation of the experimental results, ground truth error is used. The ground truth images of the test images utilized during the simulations are given in Figs. 4.8 to 4.12. Region numbers (K) are selected equal to the values which are used to define the respective ground truth images.

Segmentation results of test images with classical RSST algorithm are given in Figs. 4.13 to 4.17 and the results are summarized in Table 4.1. As it can be observed, classical RSST algorithm based on intensity means has difficulty in segmenting textured images. Small variances between the segmentation results of the RSST algorithm in Chapter 3 and Chapter 4 are due to implementation differences of the algorithm in Borland C++ Builder and Matlab.

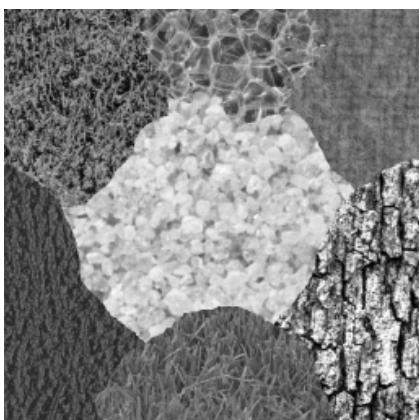


Figure 4.1: Test image 6.



Figure 4.2: Test image 7.

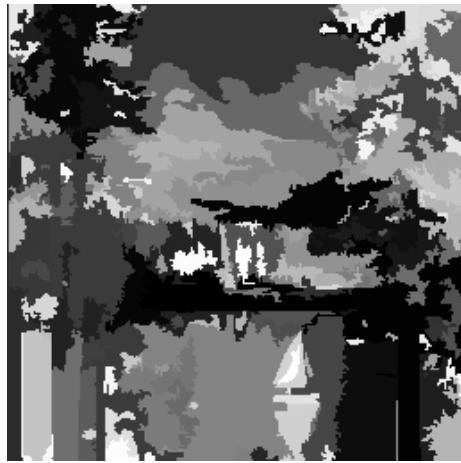


Figure 4.3: Test Image 2 segmented into 256 regions with RSST.



Figure 4.4: Test Image 3 segmented into 256 regions with RSST.



Figure 4.5: Test Image 5 segmented into 256 regions with RSST.

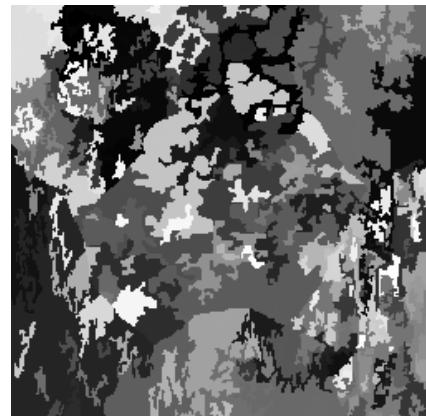


Figure 4.6: Test Image 6 segmented into 256 regions with RSST.



Figure 4.7: Test Image 2 segmented into 256 regions with RSST.



Figure 4.8: Ground truth image of test image 2 (K=4).



Figure 4.9: Ground truth image of test image 3 (K=4).



Figure 4.10: Ground truth image of test image 5 (K=3).



Figure 4.11: Ground truth image of test image 6 (K=7).



Figure 4.12: Ground truth image of test image 7 (K=8).

Table 4.1: Numerical results of classical RSST algorithm for test images.

IMAGE	K	Ground Truth Error	Elapsed Time (sec)
Test Image 2	4	0,2544	32,344
Test Image 3	4	0,0698	44,859
Test Image 5	3	0,1162	48,063
Test Image 6	7	0,4126	28,282
Test Image 7	8	0,5981	45,797



Figure 4.13: Segmentation of test image 2 with RSST.



Figure 4.14: Segmentation of test image 3 with RSST.



Figure 4.15: Segmentation of test image 5 with RSST.

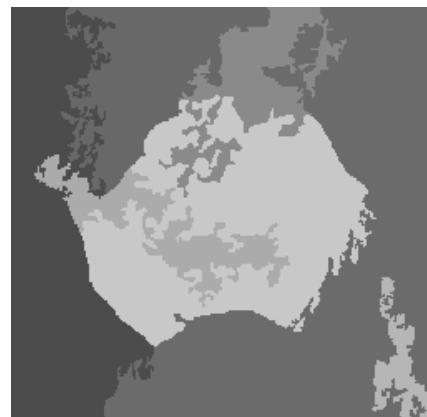


Figure 4.16: Segmentation of test image 6 with RSST.



Figure 4.17: Segmentation of test image 7 with RSST.

4.2 Region Intensity Modeling by Histogram

4.2.1 Proposed Method

In classical RSST algorithm, intensity means are used while comparing the similarity of regions. In the proposed method, grayscale histogram of each region is used for region intensity modeling instead of means. The grayscale histogram contains more information about the grayscale characteristics of a region. As it can be seen in Fig. 4.18, the grayscale histograms of similar regions show similar characteristics. The symmetric Kullbach-Leibler (KL) distance [16] is used to measure the difference between histograms. The symmetric Kullbach-Leibler distance can be defined as:

$$D(P \parallel Q) = \sum_{x \in S} (P(x) - Q(x)) \log \frac{P(x)}{Q(x)}$$

where $P(x)$ and $Q(x)$ are the compared probability distributions over set S with members x . These values are estimated by normalized histogram values of each region.

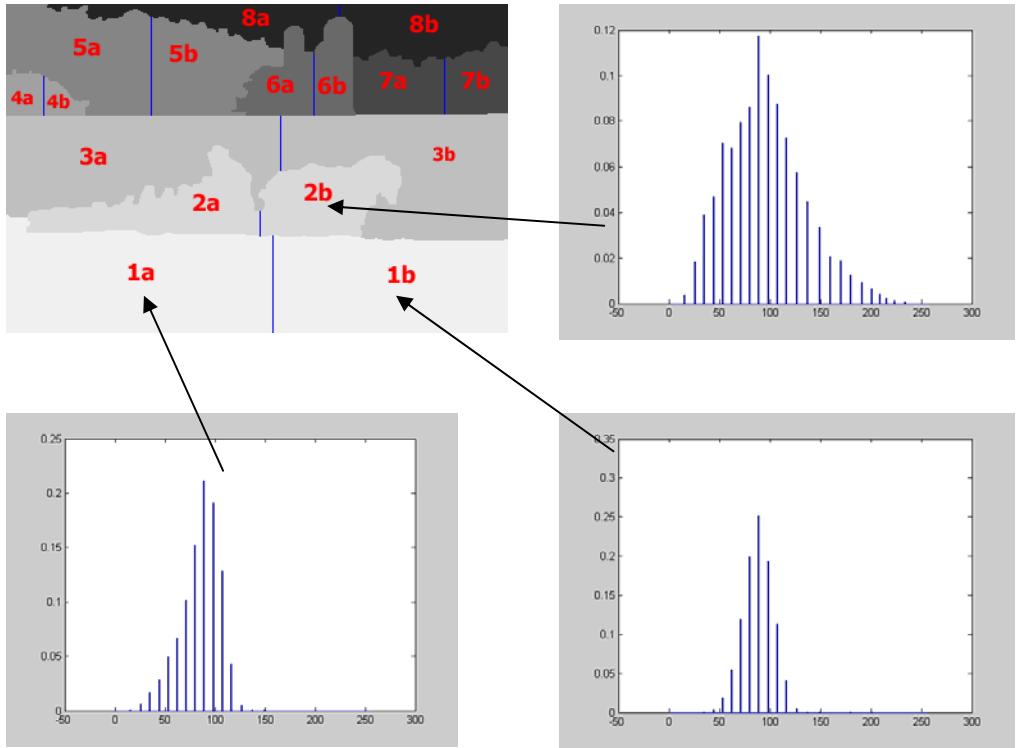


Figure 4.18: Histograms belonging to artificially segmented different regions for test image 7.

4.2.2 Simulation Results

The test images of Figs. 3.4, 3.5, 3.7, 4.1 and 4.2 are utilized to assess the performance of proposed algorithms. Segmented images are displayed in Figs. 4.19 to 4.23 and results are summarized in Table 4.2. It can be observed that there is a noticeable improvement in the segmentation performances compared to classical RSST, especially in textured images, test image 5, 6 and 7. However, there are still undesired artifacts, like the leakage of up-middle region into the middle region of test image 6. Ground truth errors are better than those of classical

RSST algorithms, whereas elapsed times are slightly worse, which is actually expected due to higher computational load of the proposed algorithm.

Table 4.2: Numerical results of RSST algorithm with Region Intensity Modeling by histogram for test images.

IMAGE	K	Ground Truth Error	Elapsed Time (sec)
Test Image 2	4	0,0707	42,032
Test Image 3	4	0,0535	57,328
Test Image 5	3	0,0883	61,421
Test Image 6	7	0,1083	37,703
Test Image 7	8	0,5039	55,578



Figure 4.19: Segmentation of test image 2 with RSST with Region Intensity Modeling by histogram.



Figure 4.20: Segmentation of test image 3 with RSST with Region Intensity Modeling by histogram.



Figure 4.21: Segmentation of test image 5 with RSST with Region Intensity Modeling by histogram.



Figure 4.22: Segmentation of test image 6 with RSST with Region Intensity Modeling by histogram.



Figure 4.23: Segmentation of test image 7 with RSST with Region Intensity Modeling by histogram.

4.3 Region Intensity Modeling by Joint Histogram

4.3.1 Proposed Method

The comparison of intensities of neighboring pixels gives information about the intensity distribution of the pixels in that region. In the proposed method, the joint histograms which show the intensities of neighboring pixels in regions are used for region modeling. The creation of joint histograms is as illustrated in Fig. 4.24. As it can be seen in Fig. 4.25, the joint histograms of similar regions are also similar.

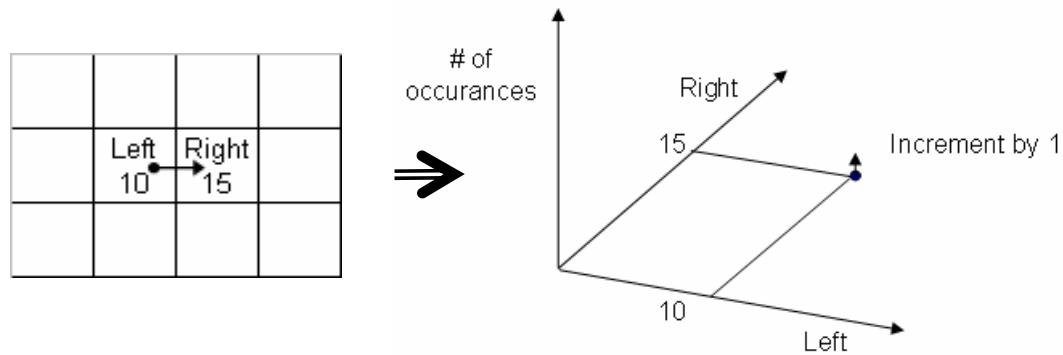


Figure 4.24: Illustration of creation of joint histograms.

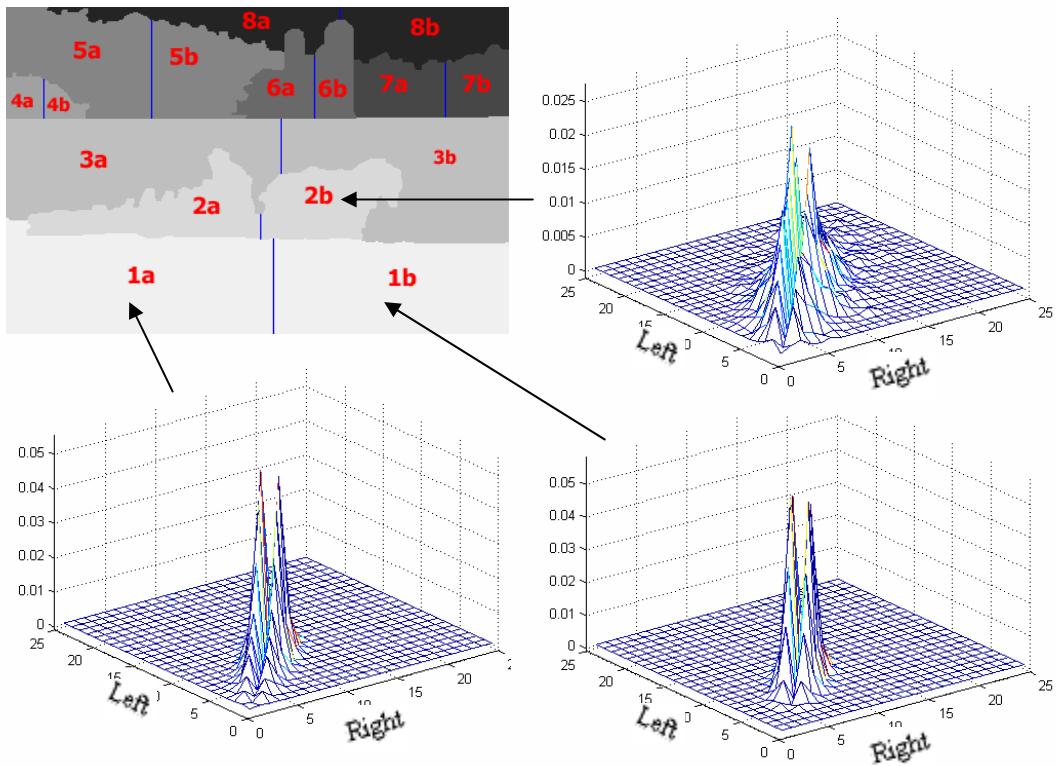


Figure 4.25: Joint histograms belonging to artificially segmented different regions for test image 7.

4.3.2 Simulation Results

The test images of Figs. 3.4, 3.5, 3.7, 4.1 and 4.2 are utilized to assess the performance of the proposed algorithm. Segmented images are displayed in Figs. 4.26 to 4.30 and results are summarized in Table 4.3. The segmentation performances are generally worse than classical RSST algorithm. Joint histogram does not appear to be a good choice for region modeling of RSST. Furthermore, the initial segmentation performed by classical RSST algorithm to segment the test image into 256 regions results in loss of some of the texture information in the

test images, which also has a role in failure of this method. Ground truth errors of the proposed algorithm are worse than those of classical RSST algorithm, except for test image 6. Besides, elapsed times of this method are quite high compared to classical RSST and RSST with histogram-based region modeling.

Table 4.3: Numerical results of RSST algorithm with Region Intensity Modeling by joint histogram for test images.

IMAGE	K	Ground Truth Error	Elapsed Time (sec)
Test Image 2	4	0,409	173,578
Test Image 3	4	0,4362	247,079
Test Image 5	3	0,4604	230,187
Test Image 6	7	0,2738	169,922
Test Image 7	8	0,616	227,562



Figure 4.26: Segmentation of test image 2 with RSST with Region Intensity Modeling by joint histogram.



Figure 4.27: Segmentation of test image 3 with RSST with Region Intensity Modeling by joint histogram.



Figure 4.28: Segmentation of test image 5 with RSST with Region Intensity Modeling by joint histogram.



Figure 4.29: Segmentation of test image 6 with RSST with Region Intensity Modeling by joint histogram.



Figure 4.30: Segmentation of test image 7 with RSST with Region Intensity Modeling by joint histogram.

4.4 Region Intensity Modeling by Texture

The region modeling based on region intensity means of classical RSST algorithm is not sufficient to segment the texture areas in the images successfully. To overcome this problem, the use of a texture feature, namely homogeneous texture descriptor, is proposed in this section.

4.4.1 Proposed Method

In this method, homogeneous texture feature [14] is used for region modeling. This descriptor characterizes the region texture by using the energy and energy deviations in a set of frequency channels which is shown in Fig. 4.31. The frequency space from which the texture features in the image are extracted is partitioned with equal angles of 30 degrees in the angular direction and with an octave division in the radial direction, as shown in Fig. 4.31. The partitions in the frequency domain are denoted as the feature channels. On top of the feature channels, a 2D Gabor function is applied. The standard deviations of the applied Gabor function are determined by processing the Gabor function with its neighbour functions at half maximum in both radial and angular directions. The descriptor consists of the mean and standard deviation of the image intensity, the energies and the energy deviations of the feature channels. Finally these values are

used to decide which regions are to be merged. In the simulations, the mean and energy values are used to determine the similarity of corresponding regions.

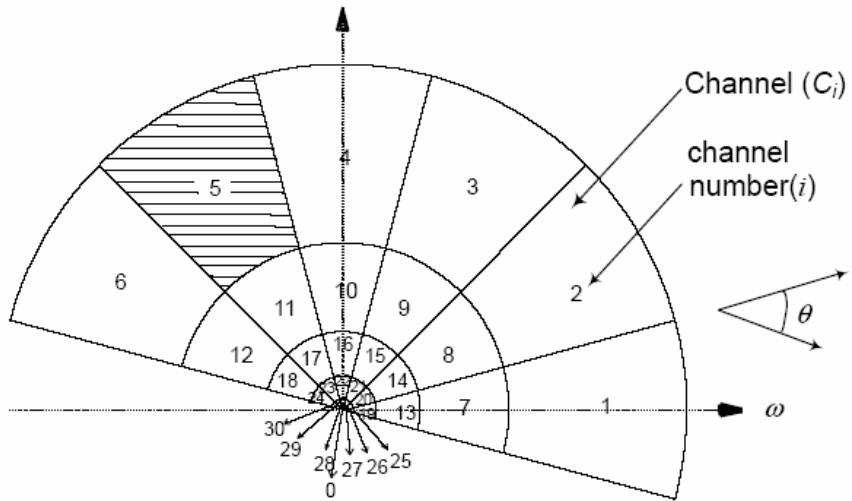


Figure 4.31: Feature layout for texture feature extraction.

4.4.2 Simulation Results

The test images of Figs. 3.4, 3.5, 3.7, 4.1 and 4.2 are utilized to assess the performance of proposed algorithm. Segmented images are displayed in Figs. 4.32 to 4.36 and the results are summarized in Table 4.4. Segmentation performance of this method is similar to segmentation performance of first proposed method, namely region intensity modeling by histogram. The proposed algorithm is more successful especially in segmenting textured images, compared to classical RSST

algorithm, and slightly better than region intensity modeling by histogram. The ground truth errors of this method are generally better than those of classical RSST except for test image 2. However, elapsed times of the algorithm are very high.

Table 4.4: Numerical results of RSST algorithm with Region Intensity Modeling by texture for test images.

IMAGE	K	Ground Truth Error	Elapsed Time (sec)
Test Image 2	4	0,3394	356,157
Test Image 3	4	0,0575	435,594
Test Image 5	3	0,0851	411,766
Test Image 6	7	0,1018	349,937
Test Image 7	8	0,4086	401,234



Figure 4.32: Segmentation of test image 2 with RSST with Region Intensity Modeling by texture.

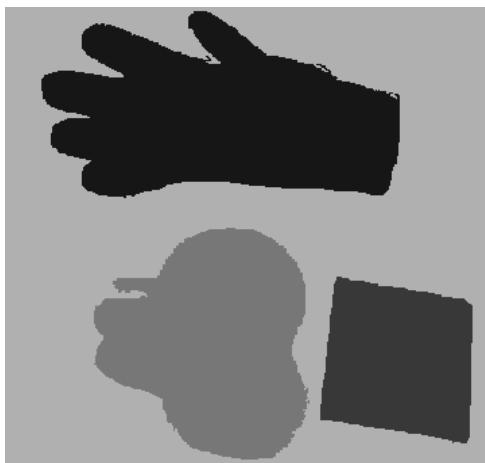


Figure 4.33: Segmentation of test image 3 with RSST with Region Intensity Modeling by texture.

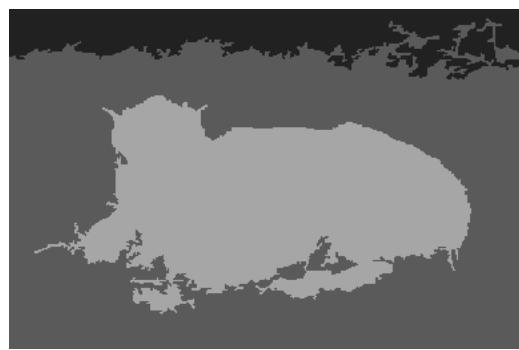


Figure 4.34: Segmentation of test image 5 with RSST with Region Intensity Modeling by texture.



Figure 4.35: Segmentation of test image 6 with RSST with Region Intensity Modeling by texture.



Figure 4.36: Segmentation of test image 7 with RSST with Region Intensity Modeling by texture.

4.5 Comparative Analysis

The proposed methods are applied on five test images. Ground truth error is utilized as the performance metric. Ground truth error is bounded between 0 and 1 with smaller values indicating better segmentation. The comparison of ground truth error and elapsed time values of proposed methods are shown in Fig. 4.37 and 4.38.

The experimental results show that segmentation performances of RSST with region intensity modeling by histogram and by texture are better than segmentation performance of classical RSST with region modeling by intensity means, especially in textured images. However, region intensity modeling based on joint histogram utilized in second proposed method fails to improve the segmentation performance.

The ground truth errors of histogram based and texture based RSST are generally better than those of classical RSST. On the other hand, the ground truth errors of joint histogram based RSST are worse than classical RSST.

The elapsed times of classical RSST and histogram based RSST are close to each other and are at acceptable levels. The elapsed times of joint histogram based and texture based RSST are higher than these two methods; texture based RSST having the highest elapsed time values among all.

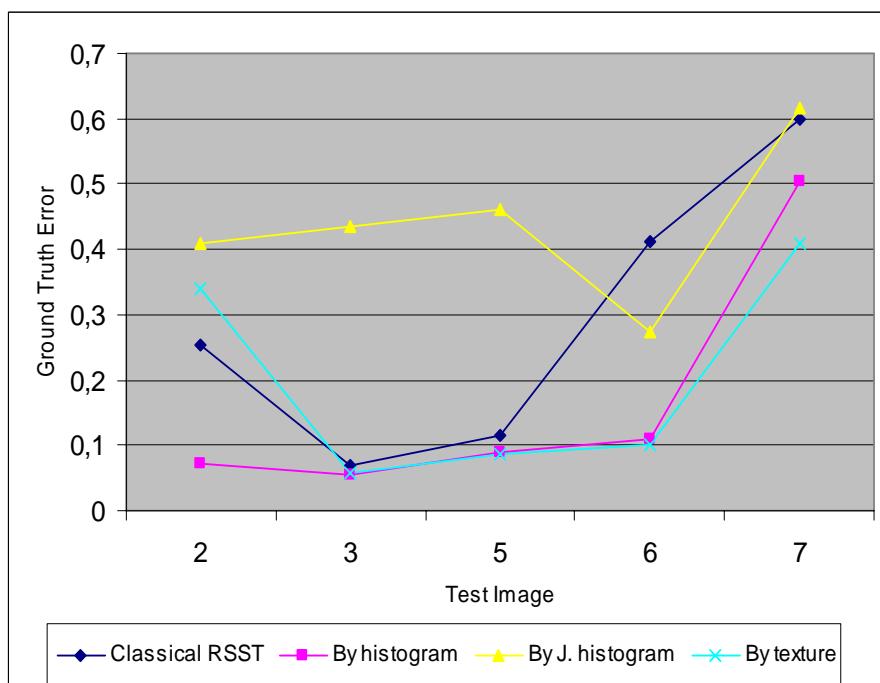


Figure 4.37: Chart showing Ground Truth Errors of proposed methods during simulations.

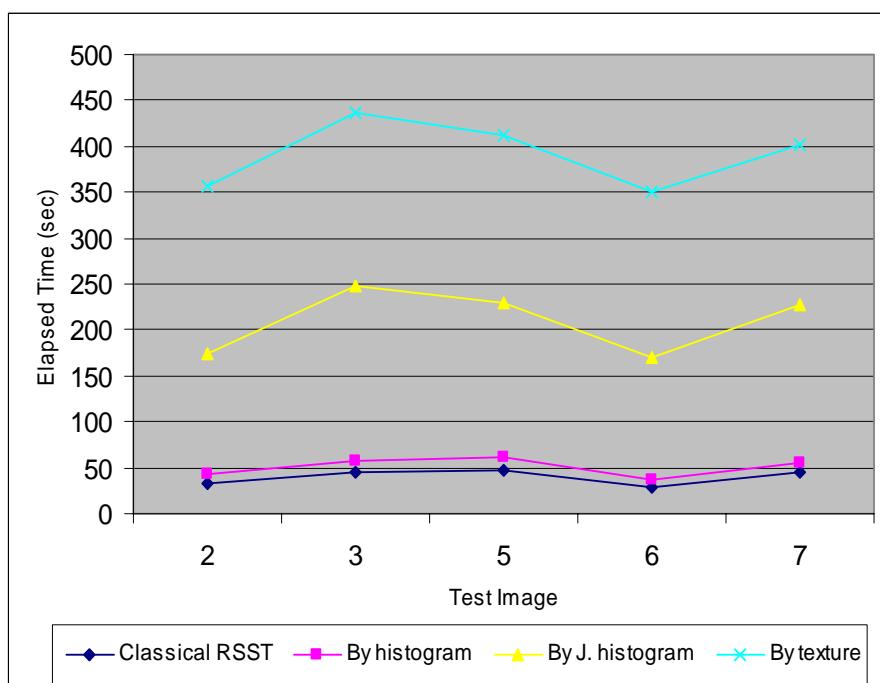


Figure 4.38: Chart showing Elapsed Times of proposed methods during simulations.

CHAPTER 5

CONCLUSIONS

This thesis is focused on comparison of fundamental image segmentation algorithms. Furthermore, possible improvements to one of the algorithms, the RSST algorithm, are also investigated. The conclusions, which are based on simulation results, can be summarized as follows:

The experimental results show that each algorithm has advantageous properties as well as some specific drawbacks. None of them gives a complete solution to this challenging problem.

Intensity clustering methods provide low computation times. However, since they neglect spatial information of the pixels, the resulting regions are not connected and are mostly widely scattered, especially in textured images. Between the two clustering algorithms tested, Fuzzy C-Means algorithm [] does not provide significantly better results than the classical K-Means algorithm []. However, due to the fuzziness concept included in the Fuzzy C-Means algorithm, its computational load is extremely high compared to K-Means.

The region-based methods, Seeded Region-Growing [] and RSST [] algorithms provide satisfactory results with relatively acceptable computation times.

KMC algorithm [] combines clustering and region-based approaches. The connected-component labeling algorithm within KMC enables getting connected regions and also provides a dynamic determination of number of regions regardless of the initial value of K . However, the resulting computational burden it brings causes the algorithm to significantly slow down, especially in textured regions.

For improving the region modeling of RSST, three methods are proposed, namely region intensity modeling by histogram and joint histogram as well as by texture. Among the proposed methods, better segmentation performances are obtained by histogram-based and texture-based methods, compared to classical RSST algorithm. However, joint histogram utilized in the second proposed method does not appear to be a good choice for region modeling of RSST. Among the three proposed methods, region intensity modeling by use of texture feature provides best segmentation performances on textured test images. The homogeneous texture descriptor, utilized in this method, enables to segment textured areas in the images more successfully. However, elapsed times of region intensity modeling by histogram are far better compared to elapsed times of region intensity modeling by texture.

REFERENCES

- [1] R. O. Duda and P. E. Hart, “*Pattern Classification and Scene Analysis*”, John Wiley&Sons, New-York, 1973.
- [2] V. Ramos and F. Muge, “*Image Colour Segmentation by Genetic Algorithms*”, CVRM, Lisboa, Portugal, 2000.
- [3] L. Lucchese and S. K. Mitra, “*Color Image Segmentation: A State-of-the Art Survey*”, “*Image Processing, Vision, and Pattern Recognition*,” Proc. of the Indian National Science Academy (INSA-A), New Delhi, India, Vol. 67 A, No. 2, pp. 207-221, Mar. 2001.
- [4] M. S. Ergin, “*Scene Segmentation Comparative Analysis*”, MS Thesis, METU, 2001.
- [5] A. H. Bhalerao, “*Multiresolution Image Segmentation*”, PhD Thesis, University of Warwick, 1991.
- [6] X. M. Pujol, “*Image Segmentation Integrating Colour, Texture and Boundary Information*”, PhD Thesis, University of Girona, 2002.
- [7] D. L. Pham, C. Xu and J. L. Prince, “*A Survey of Current Methods in Medical Image Segmentation*”, Annual Review of Biomedical Engineering, 1998.
- [8] R. C. Gonzales and R. E. Woods, Digital Image Processing, Addison-Wesley Pub. Co., 1992.
- [9] E. Tuncel, “*Utilization of Improved Recursive-Shortest Spanning Tree Method for Video Object Segmentation*”, MS Thesis, Bilkent University, 1997.
- [10] A. Mehnert and P. Jackway, “*An Improved Seeded Region Growing Algorithm*”, Pattern Recognition Letters 18 pp.1065-1071, 1997.
- [11] S. H. Kwok, A. G. Constantinides and W. Siu, “*An Efficient Recursive Shortest Spanning Tree Algorithm Using Linking Properties*”, IEEE Transactions, Circuit and Systems for Video Technology, Vol. 14, 2004.

- [12] I. Kompatsiaris and M. G. Strintzis, “*Spatiotemporal Segmentation and Tracking of Objects for Visualization of Videoconference Image Sequences*”, IEEE Transactions, Circuit and Systems for Video Technology, Vol. 10, 2000.
- [13] P. Guha, “*Automated Visual Inspection of Steel Surface, Texture Segmentation and Development of a Perceptual Similarity Measure*”, Indian Institute of Technology, Kanpur, 2001.
- [14] Y. M. Ro, M. Kim and H. K. Kang, “*MPEG-7 Homogeneous Texture Descriptor*”, ETRI Journal, Vol. 23, No. 2, 2001.
- [15] E. B. Ranguelova, “*Segmentation of Textured Images on Three-Dimensional Lattices*”, PhD Thesis, University of Dublin, 2002.
- [16] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, “*Automatic segmentation, classification and clustering of broadcast news audio*”, in Proc. of the DARPA speech recog. workshop, Chantilly, VA, pp. 97-99, Feb. 1997.
- [17] Z. Kato, “*Modelisations markoviennes multiresolutions en vision par ordinateur. Application a la segmentation d'images SPOT*”, PhD Thesis, University of Nice, 1994.
- [18] D. L. Pham, J. L. Prince, “*An Adaptive Fuzzy C-Means Algorithm for Image Segmentation in the Presence of Intensity Inhomogeneities*”, SPIE Medical Imaging: Image Processing, San Diego, CA, Feb 21-27, Proceedings of SPIE vol. 3338, pp. 555-563, 1998.