İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

# COLOR IMAGE SEGMENTATION:
# MULTITHRESHOLDING AND CONSTRAINT SATISFACTON METHODS

PhD Thesis by
Fatih KURUGÖLLÜ, Msc.

504930011012



Date of submission : 30 September 1999

Date of defence examination: 21 February 2000

Supervisor (Chairman): Prof. Dr. Emre HARMANCI (İ.T.Ü)

Members of the Examining Committee Prof.Dr. Bülent SANKUR (B. Ü.)

Prof.Dr. Muhittin GÖKMEN (İ.T.Ü.)

Assoc. Prof.Dr. Lale AKARUN (B.Ü.)

Assoc. Prof.Dr. Işıl CELASUN (İ.T.Ü.)

FEBRUARY 2000

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# RENKLİ İMGE BÖLÜTLEME:
# ÇOKLUEŞİKLEME ve KISIT SAĞLAMA YÖNTEMLERİ

**DOKTORA TEZİ**
**Y. Müh. Fatih KURUGÖLLÜ**
**504930011012**

**Tezin Enstitüye Verildiği Tarih : 30 Eylül 1999**
**Tezin Savunulduğu Tarih : 21 Şubat 2000**

| | |
|---|---|
| **Tez Danışmanı :** | **Prof.Dr. Emre HARMANCI** |
| **Diğer Jüri Üyeleri** | **Prof.Dr. Bülent SANKUR (B.Ü.)** |
| | **Prof.Dr. Muhittin GÖKMEN (İ.T.Ü.)** |
| | **Doç.Dr. Lale AKARUN (B.Ü.)** |
| | **Doç.Dr. Işıl CELASUN (İ.T.Ü.)** |

**ŞUBAT 2000**

## ACKNOWLEDGMENTS

This thesis would not be finished without the help of a number of people who offered me different kinds of support throughout my study. First of all, I have to thank for my supervisor Prof. Dr. Emre HARMANCI for introducing me to the field of computer engineering. His guidance in my research from undergraduate to Ph.D. has always been very valuable. I also appreciate his constructive ideas and help for letting me to select an appropriate research topic.

I have to express my gratitude to Prof. Dr. Bülent SANKUR, my co-supervisor. His immense patience in discussing with me on both academic and personal matters throughout my six years of study is incredible. I learned from him how a scientist must study. He always encouraged me to finish my thesis and to publish my research results. Without him, this thesis would not be as complete as it is.

I also thank for Assoc. Prof. Dr. Muhittin GÖKMEN who introduced me image processing and computer vision topics delightfully during my course. The homework assignments and final projects were very instructive for me. My interest on image processing rose from his lectures.

I must give my special thanks my colleagues in Marmara Research Center. I am grateful to a number of friends, especially Serdar BİRECİK, Selim ÇETİN, Derya DEMİR and Mehmet SEZGİN, who have shared with me their problems, joy and happiness in their research as well as personal life. I will never forget pleasant times with my friends at MRC. I would like to thank Cengiz GÜNGÖR, Namık ÜNLÜ Celalettin KARAMAN and Aydın YAVUZ for their help for color printing of the thesis.

I thank for the continuous support and concern from my parents, Celal and Emine KURUGÖLLÜ. They always want the best wishes for me. They also want me become a useful person for our country. I hope I will achieve that goal. Last but not least, I must thank my wife, Yasemin. She has waited patiently for me to finish this thesis without any complaining. When I have been studying on my thesis in endless nights, she always waited me patiently and consoled herself with our children, Zeynep and Mustafa. I hope that completion of this thesis allows me to have more time with my family.

## TABLE OF CONTENTS

## ABBREVIATIONS

| | |
|---|---|
| **1D** | : 1 Dimensional |
| **2D** | : 2 Dimensional |
| **3D** | : 3 Dimensional |
| **AIC** | : Akaike Information Criteria |
| **ANN** | : Artificial Neural Network |
| **AUMA** | : Absolute Ultimate Measurement Accuracy |
| **B-CSNN** | : Boundary Constraint Satisfaction Neural Network |
| **BT** | : Backtracking |
| **CE** | : Classification Error |
| **CNN** | : Cellular Neural Network |
| **CSNN** | : Constraint Satisfaction Neural Network |
| **CSP** | : Constraint Satisfaction Problem |
| **DRNN** | : Deterministic Relaxation Neural Network |
| **FGBFNN** | : Fuzzy Gaussian Basis Function Neural Network |
| **GRMF** | : Gauss Markov Random Field |
| **G-T** | : Generate-and-Test |
| **ICM** | : Iterated Conditional Modes |
| **LTM** | : Long-term Memory |
| **MAP** | : Maximum A Posterior Probability |
| **MR** | : Magnetic Resonans |
| **MRF** | : Markov Random Field |
| **MRF-CSNN** | : Constraint Satisfaction Neural Network with Markov Random Field Extension |
| **MS-CSNN** | : Multi-scan Constraint Satisfaction Neural Network |
| **P-CSNN** | : Pyramidal Constraint Satisfaction Neural Network |
| **RUMA** | : Relative Ultimate Measurement Accuracy |
| **SAR** | : Synthetic Aperture Radar |
| **SOM** | : Self Organizing Map |
| **STM** | : Short-term Memory |
| **TAF** | : Turning Angle Function |
| **UMA** | : Ultimate Measurement Accuracy |

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF SYMBOLS

| | |
|---|---|
| $\lambda$ | : Difficulty level of a synthetic test image |
| **CE** | : Average of $CE_I$ and $CE_{II}$ |
| **$CE_I$** | : Classification Error I |
| **$CE_{II}$** | : Classification Error II |
| **$C_{ij}$** | : $(i,j)^{th}$ entry of confusion matrix |
| **$H_{ijk}$** | : Total contribution of neighbors of $U_{ijk}$ at $k^{th}$ layers |
| **$M_p$** | : Minkowski summation of CE and TAF |
| **O()** | : Complexity of an algorithm |
| **$O_{ijk}$** | : The output of the $U_{ijk}$ |
| **TAF** | : Turning angle function |
| **$T_{lk}$** | : Label transfom from segmented map i to segmented map j |
| **$U_{ijk}$** | : The neuron in the $k^{th}$ layer at the (i,j) location in CSNN |
| **$W_{ij,qr,k,l}$** | : The connection weight (constrain) between $U_{ijk}$ and $U_{qrl}$ |

## COLOR IMAGE SEGMENTATION: MULTITHRESHOLDING AND CONSTRAINT SATISFACTION METHODS

## SUMMARY

Image segmentation is the process that divides the given image into regions in terms of specific uniformity criteria. Segmentation is used in many image processing applications. The desired performance of this processing is tightly coupled to the performance of the segmentation stage since it directly affects the consecutive steps of analysis. Thus the segmentation can be viewed as the most critical stage in the image analysis.

The segmentation methods could be broadly divided into two classes: Problem-oriented and general methods. Problem-oriented methods are developed according to the nature of the given problem while general methods try to carry out the segmentation without any a-priori knowledge about the scene so that they are bound to be ad hoc and intuitive approaches. General methods can be subdivided into four categories: Region-based, boundary-based, measurement space-based and hybrid approaches. Region-based methods take into account pixel similarity in the region while boundary-based methods use assumption of discontinuity between the regions. Measurement-space-based methods carry out the segmentation task in the measurement domain usually without spatial constraints. In the hybrid methods, a combination of the above approaches is employed.

An important issue is the evaluation and assessment of the goodness of an image segmentation algorithm. Evaluation and assessment criterion could firstly provide guidance for the selection among several algorithms. Secondly, most of the image segmentation methods require some initial setting or subsequent adjustment of parameters, and the segmentation outcome often depends upon the choice of these parameters. Choosing the correct parameters for the given application can also be instrumented via a performance criterion. Finally, the third importance of performance indicators comes up in the feedbacked image analysis operations. These types of systems try to improve their performance by modifying their strategy according to the feedback from the current result. Then the quantitative measure of

the performance plays the key role in providing a feedback path from the current outcome to the algorithm. Consequently, the performance evaluation and comparison of image segmentation algorithms is an indispensable tool in the segmentation problem. In this thesis a novel evaluation criterion that combines classification error and shape deformation has been proposed. Segmentation evaluation criteria must be able to quantify two important aspects: Good detection of classification and good detection of shape formation. Thus the evaluation criterion must give high score the algorithms that generate segments more closely to the reference one in the spatial domain and generate segments whose shapes are more similar to the reference one. In this thesis classification error measure proposed by **Yasnoff et.al (1977)** is adopted as detection of classification error. Shape deformation is detected by a measure based on comparison of the turning angle function of objects. These two measures have sometimes-complementary properties. In fact, classification error measure has good detection of classification but poor shape distortion monitoring ability, while turning angle function has good shape distortion monitoring but poor detection of classification capability. It has been conjectured that their linear combination of these two criteria will satisfy both the localization and shape requirements. In addition, an automatic test image generation procedure has also been proposed in this thesis.

Among the two novel segmentation algorithms, proposed the first one relies on multithresholding. Multithresholding is an attractive and widely used tool that is based on image histogram. While multithresholding for gray level images is relatively straightforward, it becomes a complex problem for color images since their histograms are three-dimensional. In this thesis, a novel multithresholding technique based on 2-Dimensional (2D) histograms and decision fusion has been proposed. In this technique, pairs of three color bands are selected and their three 2D histograms are constituted. These histograms are partitioned according to their dominant peaks. Each band pair image is segmented using its corresponding partitioned 2D histogram. The three segmented maps are fused to obtain a final segmentation by taking into account label concordances in each segmented map. This result is filtered morphologically so that small segments are eliminated. The proposed method has been compared with its 1-Dimensional (1D) histogram and 3-Dimensional (3D) histogram based counterparts. It is observed that the proposed method can catch segments that can be missed in the 1D-histogram based method Thus its performance is superior with respect to the 1D case while its computational complexity is only about two times that of 1D case. Although the 3D-histogram based method gives the best results since the threshold blobs are inherently delineated very well, the computational complexity of 3D case is about four hundred times that of the 2D case. Consequently, the proposed method is an attractive alternative to its 1D and 3D counterparts in the manner both computational complexity and segmentation performance.

The second innovation proposed consists of various adaptations of the Constraint Satisfaction Neural Network (CSNN) based segmentation, which is proposed by **Lin et. al. (1992)**. Although it is proposed as a neural network implementation of the constraint satisfaction problem, it can be interpreted as a probabilistic relaxation scheme. Various drawbacks of the algorithm can be summarized as follows: i) The number of segments are given the algorithm by means of a-priori information. ii) Only gray level image is used in CSNN. iii) Initialization of CSNN is carried out by an ad hoc method using SOM clustering. iv) Convergence time of CSNN is very long. Depending on this long convergence time, CSNN generates generally oversmoothed segments. Some modifications to overcome the shortcomings of the original algorithm will be presented. First, the number of segments is determined automatically from a given image by means of a cluster validity criterion. For this purpose, a modified AIC cluster validity measure is used. Second, the initial architecture is generalized to color images. Third, the initialization stage is modified by using the fuzzy c–means algorithm rather than the SOM algorithm to avoid the ad hoc fuzzification in the initialization. To overcome oversmoothing and long convergence time problems, some structural modifications have been imposed to CSNN. For example, boundary constraints obtained from a coarse scale Marr-Hildreth edge detector have been added to CSNN interconnections. The proposed multi-scan CSNN (MS-CSNN) or multi-grid sampling CSNN algorithm introduced a multiresolution nature to CSNN. A second way to provide multiresolution behaviors was the pyramid architectural applied to CSNN. This yields third method called pyramidal CSNN (P-CSNN). Finally, CSNN has been altered to solve the Markov Random Field (MRF) segmentation problem. Since it was observed that CSNN actually carries out overall energy minimization but using only neighborhood contributions, it was sufficient to add a data dependency term to the CSNN formulation to implement the MRF-based minimization.

The performance of all proposed methods are compared by using the proposed evaluation criteria and test images that are generated by proposed test image generation procedure. It has been observed that the segmentation performances of the proposed methods are better than those of the competitor methods in the literature. The computational complexities of the proposed methods have been also reduced in terms of their competitors.

## RENKLİ İMGE BÖLÜTLEME: ÇOKLUEŞİKLEME ve KISIT SAĞLAMA YÖNTEMLERİ


### ÖZET

İmge bölütleme verilen bir imgeyi belli bir tekdüzelik kriterine göre homojen bölgelere ayırma sürecidir. Bölütleme bir çok imge çözümleme uygulamasında kullanılır. Bölütlemenin başarımı çözümlemede kendisini izleyen adımların başarımını doğrudan etkilediğinden uygulamanın istenen başarımı, bölütleme evresinin başarımına sıkı bir şekilde bağlıdır. Dolayısıyla bölütlemenin imge çözümlemenin en kritik evresi olduğunu söylemek yanlış olmaz.

Bölütleme yöntemleri kabaca iki ana gruba ayrılabilir: Probleme yönelik yöntemler ve genel-geçer yöntemler. Probleme yönelik yöntemler verilen problemin doğasına ve bazı önsel bilgilere göre geliştirilirken genel-geçer yöntemler bölütlemeyi sahne hakkında herhangi bir önsel bilgi olmadan yerine getirmeyi amaçlar. Bu nedenle genel-geçer yöntemler daha çok buluşsal ve sezgisel yaklaşımlara dayanır. Genel-geçer yöntemler, bölge tabanlı, sınır tabanlı, ölçüm uzayı tabanlı ve melez yöntemler olmak üzere dört alt gruba ayrılabilir. Bölge tabanlı yöntemler bölge içindeki piksellerin benzerliğini temel alırken sınır tabanlı yöntemler bölgeler arasındaki süreksizliği sezmeyi temel alırlar. Ölçüm uzayı tabanlı yöntemler ise bölütlemeyi uzamsal kısıtları göz ardı ederek ölçüm uzayında gerçekleştirirler. Melez yöntemler yukarıdaki üç yaklaşımın herhangi bir kombinasyonunu kullanır.

Bölütlemede önemli bir problem bölütlemenin değerlendirilmesi ve bölütleme başarımının sıralanmasıdır. Değerlendirme ve sıralama kriterleri çeşitli yöntemlerden belli bir uygulama için en iyi başarım gösteren yöntemi seçmede kullanılabilir. Öte yandan birçok bölütleme yöntemi bazı önsel parametre ayarı gerektirmektedir. Bölütleme sonucu bu parametre ayarına bağlı olmaktadır. Verilen bir uygulama için en doğru parametre seçimi, değerlendirme kriteri ile yapılabilir. Değerlendirme kriterinin diğer önemli bir kullanım alanı geribeslemeli image analizidir. Bu tip uygulamalarda sürecin her adımında imge bölütleme sonucu değerlendirilir ve gerekli parametre ve strateji değişiklikleri bu değerlendirmeye göre yapılır. Burada sonucun nicel değerlendirilmesi anahtar rol oynamaktadır. Bu tezde

sınıflama hatasını ve şekil bozukluklarını birleştiren yeni bir değerlendirme yöntemi önerilmiştir. Herhangi bir deneysel değerlendirme yöntemi sınıflama hatasını ve şekil bozulmalarını nicel olarak ölçmelidir. Daha açıkça, değerlendirme yöntemi uzamsal olarak dayanak haritasıyla daha iyi örtüşen ve şekil olarak referansa daha yakın bölütler üreten bir algoritmaya daha fazla puan vermelidir. Bu tezde Yasnoff et.al. (1977) da önerilen sınıflama hatası ölçütü sınıflama hatasını değerlendirmede kullanılırken, şekil bozulmaları için nesnelerin dönme açısına dayalı yeni bir yöntem önerilmiştir. Dönme açısı işlevi nesnelerin şekil karakteristiğini yansıtmaktadır. Dönme açısı ve sınıflama hatası kimi durumlarda birbirini tamamlayıcı özelliğe sahiptir. Sınıflama hatası ölçütü doğru sınıflamayı sezmede başarılı olurken şekil bozulmalarını ortaya çıkaramamaktadır. Dönme açısı ölçütü ise şekil bozulmalarını başarılı bir şekilde sezerken sınıflama başarımını göz ardı etmektedir. Dolayısıyla bu iki ölçütün birleştirilmesiyle elde edilecek yeni bir ölçüt, hem sınıflama hatasını hem de şekil bozulmalarını göz önüne almaktadır. Bu tezde her iki ölçüt Minkowski toplamı ile birleştirilerek bölütlemenin değerlendirilmesinde kullanılabilecek yeni bir kriter önerilmiştir. Önerilen kriteri değişik algoritmaların başarımını sıralamada kullanmak için bir dizi test imgesine de gerek vardır. Bu amaçla bazı koşulları otomatik olarak sağlayan bir test imge üretme yordamı da geliştirilmiştir.

Bölütlemedeki diğer ilginç bir problem de renkli imgelerin çoklu-eşiklenmesidir. Çoklu-eşikleme imge histogramına dayalı oldukça basit ve sıkca başvurulan bir bölütleme yöntemidir. Gri tonlu imgeler üzerinde çoklu-eşiklemeyi gerçekleştiren bir çok yöntem mevcuttur. Ancak renkli imgeler için çoklu-eşikleme renk histogramının üç boyutlu olması nedeniyle aynı kolaylıkla kullanılamamaktadır. Bu tezde renkli imgeler için iki boyutlu renk histogramlarına ve tümleştirmeye dayalı yeni bir çoklu-eşikleme yöntemi önerilmiştir. Bu yöntemde renk bileşenleri ikişer ikişer eşlenerek her bir renk bileşen çifti için iki boyutlu histogramlar oluşturulur. Bu iki boyutlu histogramlar, eşik yüzeyleri oluşturmak amacıyla doruklarına göre bölüntülenir. Her bir bileşen çifti, ilgili bölümlenmiş iki boyutlu histogramdaki eşik yüzeylerine göre bölütlenir. Elde edilen üç bölüt haritası etiket uyumlaştırma ve bir tümleştirme yöntemi ile birleştirilerek sonuç bölütleme haritası elde edilir. Ortaya çıkabilecek küçük bölütleri yok etmek için sonuç bölütleme haritası morfolojik kapama süzgeçi ile süzgeçlenir. Önerilen yöntem bir boyutlu ve üç boyutlu histograma dayalı eşdeğer yöntemler ile karşılaştırılmıştır. Bir boyutlu yöntemde ortaya çıkan tek renk geçişi ile oluşan bölütlerin kaybolması problemi önerilen yöntemle giderilmiştir. Önerilen yöntemin yürütüm süresinin, bir boyutlu yöntemin yürütüm süresinin sadece iki katı olduğu gözlenmiştir. Öte yandan üç boyutlu histograma dayalı yöntem doğası gereği önerilen yönteme göre biraz daha iyi sonuçlar üretmesine rağmen yürütüm süresi iki boyutlu yönteme göre yaklaşık dörtyüz kat daha fazla olmaktadır. Sonuçta iki boyutlu histograma dayalı yöntem bir boyutlu eşdeğerine göre daha iyi başarıma sahip, üç boyutlu eşdeğerine göre ise işlemsel olarak daha etkin bir bölütleme yöntemi sunmaktadır.

Bu tezde ölçüm uzayında topaklama ve imge uzayında gevşeme işlemlerini bir yapay sinir ağı mimarisinde birleştirerek bir arada kullanan Kısıt Sağlayan Yapay Sinir Ağı (KSYSA) da irdelenmiştir. KSYSA istenen kısıtları gerçekleştiren basit mimarisi nedeniyle bölütleme için etkin bir araç sağlamaktadır. KSYSA aslında kısıt sağlama problemine yapay sinir ağı çözümü getirmek için tasarlanmış olmasına rağmen olasılıksal gevşeme yöntemine dayalı bir bölütleme yöntemi olarak da yorumlanabilmektedir. Yöntemin bir dizi olumsuzlukları vardır. Bunlar i) Bölüt sayısının algoritmaya önceden verilmesi zorunluluğu ii) Sadece gri tonlu imgelerde kullanılmak üzere tasarlanmış olması iii) Önerilen yapay sinir ağının ilklendirilmesinin SOM a dayalı sezgisel bir yöntemle gerçekleştirilmesi iv) Yakınsama süresinin çok uzun olması ve buna bağlı olarak genellikle çok fazla yumuşatılmış sonuç üretmesi olarak sıralanabilir. Bu tezde yöntemin bu olumsuzluklarını ortadan kaldırmak üzere bir dizi iyileştirmeler gerçekleştirilmiştir. Öncelikle bölüt sayısının verilen imgeden otomatik olarak belirlenmesine yönelik bir topak geçerlilik ölçütü kullanılmıştır. İkinci olarak yapay sinir ağının ön koşullamasında kullanılan ön sınıflama evresi renkli imgeleri de kullanacak şekilde genişletilmiştir. Son olarak ön sınıflamada SOM yöntemi yerine bulanık c-ortalamaları yöntemi kullanılmıştır. Böylece SOM da önce etiket kararı verip sonra sezgisel olarak bulanıklaştırma yerine doğrudan bulanık çıkış üreten bir yöntem kullanılmıştır. Yumuşatılmış çıkış ve uzun yakınsama süresi problemlerini aşmak için KSYSA'nın mimarisini değiştiren dört yeni yöntem önerilmiştir. Birinci yöntem Sınır kısıtı sağlayan yapay sinir ağı olarak adlandırılmaktadır. Bu yöntemde KSYSA da mevcut kısıtların yanı sıra kaba ölçekte çalışan bir Marr-Hildreth ayrıt sezicisinden elde edilen sınır kısıtları da kullanılmıştır. İkinci yöntem Çoklu-taramalı KSYSA yöntemidir. Bu yöntemde KSYSA nın performansı çoklu-çözünürlük yaklaşımı ile iyileştirilmiştir. Çoklu-çözünürlüğü gerçekleştirmek için çoklu-tarama yaklaşımı kullanılmıştır. Çoklu-çözünürlük kullanılarak gerçekleştirilen diğer bir yöntem de piramit temelli KSYSA dır. Burada piramitin her katmanında ayrı bir KSYSA kullanılmıştır. Son olarak Markov rasgele alanları (MRA) yaklaşımı ile KSYSA birleştirilerek yeni bir yöntem önerilmiştir. KSYSA sadece komşululuklara göre enerji azaltma işlevi görmektedir. Önerilen yöntemde KSYSA nın mimarisine, MRA temelli enerji azaltmayı gerçekleştirmek üzere veriye bağımlılık terimi de eklenmiştir.

Bu tezde önerilen bütün yeni yöntemlerin başarımları önerilen değerlendirme kriteri ve test imge yordamı kullanılarak değerlendirilmiş ve karşılaştırılmıştır. Önerilen yöntemlerin literatürdeki rakiplerine göre daha üstün bölütleme başarımına sahip olduğu, ayrıca yürütüm sürelerin de daha kısa olduğu gözlenmiştir.

# 1. INTRODUCTION: IMAGE UNDERSTANDING AND IMAGE SEGMENTATION

## 1.1 Introduction

Digital image processing, which appeared in 1920s due to improve the appearances of digitized the newspaper images, is a process that aims to enhance the image quality or to obtain the scene description from the given image **(Gonzalez and Wintz, 1987)**. This process is carried out naturally by human eye and brain. Although perception and interpretation of the scene by human beings seems easy and is a natural process, this task is very difficult for computers. On the other hand, the need for automatic analysis and interpretation of scene information from the images using computers has grown up significantly. This has let to an increasing number of applications based on image processing. Some of these image processing applications can be summarized as follows:

- Analysis of medical images for diagnosis
- Automatic character recognition for document analysis
- Quality control and product pursuit in industrial applications
- Automatic finger-print and face recognition for law enforcement applications
- Target detection and identification for military purposes
- Analysis of aerial and satellite images for meteorological and military purposes
- Image compression for effective storage and transmission of images
- Restoration of images that are corrupted by noise, motion, etc.

Image processing applications can be divided into four major categories:

- Improvement of image quality for human interpretation.
- Restoration of images that are degraded in some corruption effects such as noise, blurring, motion, defocusing, etc.
- Coding of images for effective storage or transmission tasks.
- Image understanding that aims to symbolically represent the scene from given image.

Image understanding has a basic different property. In image understanding, the input is an image and the output is the description of the scene while both the input and the output are images in image processing **(Lim, 1990)**. The description of scene includes quantitative data such as number of objects in the scene, morphological features of the objects, position relationship between objects. Image understanding consists of following principle steps:

- Image acquisition
- Image enhancement
- Image segmentation
- Feature extraction
- Object recognition
- Scene description

Image acquisition is the first step in image understanding. Acquiring image from which type of sensors (visual band, infrared band, multispectral band, etc.) determines algorithms that is to be used in consecutive steps. Image enhancement aims to improve the visibility of the image and to make easy jobs of other steps on the image understandings. Noise removal, deblurring, and histogram equalization can be given as examples. Image segmentation is the process dividing the given image into regions, which does not occlude each other. It is hopefully considered that these regions correspond to the objects in the scene. Feature extraction includes extracting the quantitative data, which are used to distinguish the objects, form the segments. For instance, moments, Fourier descriptors, segment signatures, topological descriptors can be used as features. The extracted features are used to recognize the objects. For this aim, feature classifiers are used. Thus the objects are recognized and labelled after object recognition. Scene description step brings into light the relationship between objects and interprets the scene.

Image acquisition, image enhancement, image segmentation and feature extraction constitute the low-level vision whereas object recognition and scene description are considered high-level vision. Signal processing tools are used in low-level vision while high-level vision often uses artificial intelligence tools as well as signal processing tools **(Rosenfeld, 1984)**.

An example of image understanding application is given in Figure 1.1. In this example, image processing steps used in the analysis of a medical tissue image acquired from a microscope are depicted. First, the image is filtered by means of a median filter to remove the noise. After that, the image is subjected to a histogram equalization process to increase the visibility of the image. These steps are followed by the segmentation step. Boundaries of the cells are found by means of a contour tracing algorithm. Then the segments are labelled in terms of these boundaries. Thus the segmented image is obtained. The features are extracted from the segmented objects. These features are subjected into a classifier so that finding which segment corresponds to which object. Thus the analysis procedure is finished. The obtained information can be used high-level procedures such as determination of pathological cells.

The segmentation step plays crucial role in image understanding since its performance directly effects the performance of the subsequent steps. In this thesis, the general segmentation problem is addressed. In this context, definition of segmentation problem, taxonomy of segmentation methods, evaluation of segmentation methods, multithresholding based color image segmentation and constraint satisfaction neural network based segmentation are examined.

The organization of the thesis is as follows: Image segmentation problem is addressed the following sections of Chapter 1. The taxonomy of segmentation methods are briefly examined in Chapter 2. Chapter 3 addresses the evaluation of segmentation methods. Multithresholding based color image segmentation is presented in Chapter 4. Constraint Satisfaction Neural Network based image segmentation methods are introduced in Chapter 5. Chapter 6 presents the conclusion.

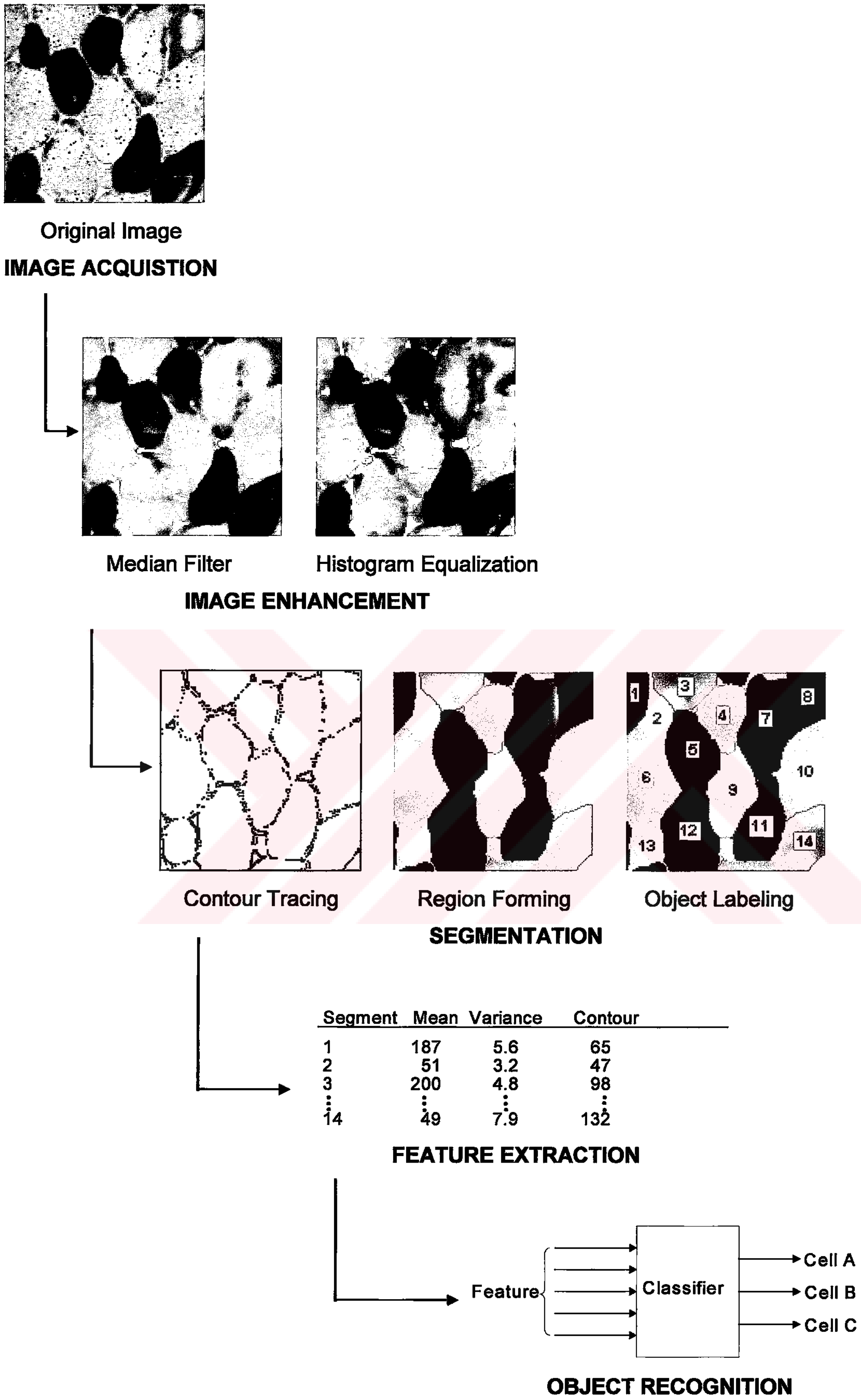



Figure 1.1 Steps of a medical image analysis application.

## 1.2 Image Segmentation

Image segmentation is the process that divides the given image into regions in terms of some specific uniformity criteria. These regions should not occlude each other. They may be homogenous and uniform. A new region that is constituted by combining adjacent two regions violets the uniformity criteria. The union of all regions yields the entire image.

Segmentation process receives an image. This can be original image as well as a modified image by the enhancement step in the analysis. After segmentation, a segmentation map is obtained. It is assumed that the segments correspond to the objects in the scene. Figure 1.2 depicts a segmentation map of the “House” image. Segmented image separates and labels the objects without any interpretation about them. Interpretation about objects is carried out by higher level of the entire analysis.

Original image　　　　Manually segmented image

Figure 1.2 (a) Original “House” image. (b) Manually segmented image. Sky, roof, house, windows and gutters are separated and labelled with different labels. These labels are depicted by using different gray levels.

Segmentation is used in many applications such as listed in Section 1.1. The desired performance of the application is tightly coupled to the performance of the segmentation stage since it directly effects the consecutive steps of analysis. Thus the segmentation is the most critical stage in the image understanding **(Rosenfeld 1984)**.

A formal definition of segmentation has been given by **Gonzalez and Witz (1987)** as follows:

Let F denotes the set of all pixels that resides in the image. P() represents uniformity criteria that is defined on the connected pixels of the segments. Segmentation can be defined as a process which divides F into sub-sets that consist of connected pixels ($S_1$, $S_2$, $S_3$, ... $S_N$). This partition must obey the following conditions:

1. $\bigcup_{i=1}^{N} S_i = F$ (Union of all regions must yield entire image)
2. $S_i \bigcap S_j = \varnothing \quad i \neq j$ (Intersection of two regions must be null set)
3. $\forall i \quad P(S_i) = \text{true}$ (Each region must be satisfy the uniformity criteria)
4. If $S_i$ and $S_j$ is adjacent, $P(S_i \bigcup S_j) = \text{false}$ (Union of two adjecent region must no satisfy the uniformity criteria)

It is desired that the segments resulting from segmentation must be "meaningful" in terms of the application. However, to describe to a computer program what means "meaningful segmentation" is very difficult task. Instead of that, it is desired that the results from the segmentation should obey the following rules **(Haralick and Shapiro, 1993)**:

- Segments must be homogenous and uniform with respect to specific characteristic such as gray tone, texture, color, etc.
- Segment interiors must be simple and does not contain many small holes.
- Adjacent segments should have significantly different values with respect to the specific characteristic.
- The boundary of segments should be simple as well as located correct spatial position. It should not be ragged.

Development of such algorithm that satisfies all above rules is very difficult task. Even the segments satisfies the rules, their semantic contents may not be reflect the actual scene reality. On the other hand, the segments that are produced by practically used segmentation methods usually contain lots of small holes and irregular boundaries. To eliminate this problem, the value of specific characteristic can be increased. In this case, some meaningful segments could be merged their neighbours. Thus the segmentation methods try to come to an agreement between two cases: Over-segmentation and under-segmentation **(Haralick and Shapiro, 1993)**.

On the other hand, a characteristic that is indicator of uniformity for one image may not be used as uniformity measure for the others. **Gurari and Wechsler (1982)** pointed out that an algorithm, which finds out that a specific characteristic can be used as a uniformity measure for a given image, cannot be developed. Instead of such an algorithm, it can be desired finding out the best possible characteristic for a given image. Nevertheless, the concept of "the best" varies from one image to other **(Gurari and Wechsler, 1982)**.

Because of aforementioned problems, there exists no general segmentation method that is suitable for all image in the literature even though thousands of them has been proposed. Although many segmentation methods have been successful on specific problems, it is not proposed a general segmentation method that is successful on all problems. Performance of any method varies according to given image or problem as well as various methods that performs on a specific image yield different results.

The main reason of these problems is that there is no general mathematically theory of the segmentation. Proposed methods so far usually are problem-oriented, ad-hoc based and intuitional. These approaches take usually into accounts nature of image type. Thus if they were used on a different image type they would not produce successful results since they do not run on different nature of images with harmony. For example, an industrial inspection problem that aims to inspect products on a line is considered. If a convenient lighting is applied the image taken form the line consists of a uniform and homogeneous background (line itself) and the objects (products on the line) whose contrast is very high according to the background. For this kind of images, a simple multi-level thresholding method would be adequate for segmentation **(Demir et.al, 1994)**. On the other hand, if the problem were to detect the deformation in a textile production line simple multilevel thresholding would not be successful due to textural nature of the textile product. It is required to employ more intelligent multilevel thresholding **(Sezgin et.al, 1997)**.

Some methods try to use mathematical model of the given image approximately. However, it is not expected to fit the model to the actual scene so the performance of the method highly depends on the accuracy of the model **(Rosenfeld and Davis, 1979)**. Therefore, more parametric conforming to a flexible model, more successful result. However to constitute a general model that includes all image classes is very difficult task. In addition some adequate models are so complicated that the algorithms become beyond a practical tool.

Another problem of the segmentation is the evaluation and assessment of segmentation results. As mentioned above there is no general segmentation method, which is appropriate for all applications. Identifying the segmentation method for a given application is a crucial problem. In this respect, quantitative comparison criteria could first provide guidance for the selection among several algorithms. Secondly, most of image segmentation methods require some initial setting or subsequent adjustment of parameters, and the segmentation outcome often depends upon the choice of these parameters. Choosing the correct parameters for a given application can also be instrumented via performance criteria. **(Zhang, 1996)**. Consequently, the performance evaluation and comparison of image segmentation algorithms is an indispensable tool in the segmentation problem. Although hundreds of image segmentation techniques have been proposed so far, there has been little progress in the performance evaluation of these techniques.

### 1.3 Scope of the thesis

In this thesis, the color image segmentation problem is addressed. First of all various segmentation algorithms have been examined. Due to plethora of segmentation algorithms, it has been realized that a taxonomy of segmentation methods is required to examine them efficiently. Thus, a taxonomy of existing segmentation methods is presented in Chapter 2.

One of the essential problems of segmentation is the evaluation of segmentation methods. Since various novel segmentation algorithms have been developed in this thesis, a quantitative evaluation tool is indispensable to prove the benefit of the novel methods. Thus, Chapter 3 addresses the evaluation of segmentation results. A novel evaluation criterion combining classification error measure and shape error measure based on turning angle function is proposed in Chapter 3.

Multithresholding provides a simple segmentation tool for some applications that have real-time constraint and have controlled environment. For instance, it is widely used in visual quality control of products since the environment is generally under user control. There exist a lot of multithresholding method that employees only gray level information. However, information of color images has greater than that of gray level ones. Thus, methods of multithresholding for color images should be confliderest. The difficulty of color multithresholding rises from the three-color band nature of the color images. Three-dimensional histogram based multithresholding suffers from computational burden. Instead of using three-dimensional histogram, using three separate one-dimensional color band histograms can be an intuitive approach. In this approach, each color band is subjected to multithresholding operation and the results are fused. Although computational complexity of the method is very low this approach suffers from its inability to detect the regions that have a single color band transition. Thus in Chapter 4, a method using two-dimensional histograms for multithresholding of color images has been investigated and a novel segmentation algorithm based on two-dimensional histograms and fusion has been proposed.

Another interesting tool for image segmentation is the Constraint Satisfaction Neural Network (CSNN) based image segmentation. Its simple network architecture and its constraint scheme used in segmentation process make it an attractive tool. This method has been exhaustively examined in Chapter 5. Its advantages and disadvantages have been brought into light. To overcome shortcomings of the method, two categories of modification have been carried out. First category of modification aims to enhance the initial conditions of the algorithm while the modifications mapped to second category change the structure of the algorithm. Four different structural modifications of CSNN have been proposed in Chapter 5.

All proposed methods have been evaluated and compared with predecessor method using the proposed evaluation criterion in Chapter 3. It has been concluded that all results are promising.

Finally, conclusions and themes for further research are presented in Chapter 6.

# 2. TAXONOMY OF IMAGE SEGMENTATION METHODS

## 2.1 Introduction

In Chapter1, it was stated that there is plethora of segmentation algorithms in the technical literature. Thus, it becomes convenient to examine these methods by means of a taxonomy tree. There have been several attempts to classify the segmentation methods such as in **Pal and Pal (1993)**, **Haralick and Shapiro (1985)**, **Kanade(1980)** and **Fu and Mui (1981)**. **Sahoo et.al. (1988)** have also attempted to give a taxonomy on thresholding methods.

The segmentation methods could be broadly divided into two classes: Problem-oriented methods and general methods. Problem-oriented methods are developed according to the nature of the given problem. They can be subdivided into two main categories: Knowledge-based (or expert system based) and model-based approaches. General methods try to carry out the segmentation without any a-priori knowledge on the scene so that they are bound to be ad hoc and intuitive approaches. They can be subdivided into four categories: Region-based, boundary-based, measurement space-based and hybrid approaches. This general categorization of image segmentation methods is illustrated in Figure 2.1 as a taxonomic tree.

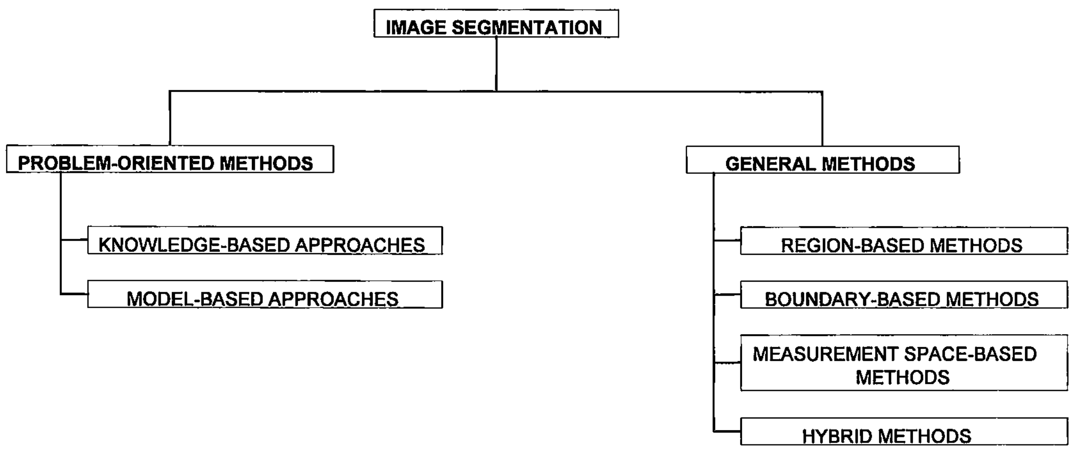



Figure 2.1 Taxonomy of image segmentation methods.

## 2.2 Problem-Oriented Methods

Problem-oriented segmentation methods carry out the segmentation task according to the nature of the given problem and by constraining the application domain. For example, brain MR images cannot vary too much from one patient to another, so that knowledge about anatomical structure of the brain and the other expertise could be used in the segmentation process. Problem-oriented methods can be divided into two main categories: Knowledge-based and model-based methods.

### 2.2.1 Knowledge-Based Methods

These methods carry out segmentation task by using a priori knowledge about the problem. This knowledge usually comes from the human beings expertise on the given application. Consequently, an expert system could be constructed for the segmentation task.

A prototype paper in the use of knowledge-base is the one by **Nazif and Levine (1984)** who proposed a rule-based segmentation system. The method contains six main processes: The initializer, the line analyzer, the region analyzer, the area analyzer, the focus of attention and the scheduler. Two memories are also utilized in the system: The short-term and the long-term memories. The short-term memory (STM) stores the input images while segmentation rules and control strategies are stored in the long-term memory (LTM). A process matches the rules in the LTM against data in the STM. If any matching occurs, the rule is fired and the action associated to the rule is performed. The model stored in LTM has three levels of rules: Knowledge rules, control rules and strategy rules. Knowledge rules deal with properties of region, line and areas. They have actions such as region merging, region splitting, merging two lines, and extending a line. These knowledge rules come from expertise of segmentation procedure. For instance, rules in the region splitting-based segmentation methods the region under consideration is divided into subregions if the region size is sufficiently high, region average gradient is high, and the histogram of the region is not unimodal. This specific knowledge is encoded in **Nazif and Levine (1984)** as follows:

IF: 1. The REGION SIZE is NOT LOW
2. The REGION AVERAGE GRADIENT is HIGH
3. The REGION HISTOGRAM is BIMODAL
THEN: SPLIT the REGION according to HISTOGRAM

The Nazif-Levine algorithm is replete with tens of such knowledge rules. Control rules control the process by determining the focus of attention. They include metarules, called inference rules, which are responsible for changing order of actions. Strategy rules, which reside in the highest level, select the set of control rules. They try to choose the most appropriate control strategy for application. Nazif and Levine method has general rules that could be changed according to the application without changing the main code. For instance, **Raya (1990)** used this method to segment MR brain images, just changing rules in terms of MR images. Similarly **Darwish and Jain (1988)** proposed a rule-based approach for inspecting printed circuit board. They mainly used morphological techniques in the system, while the expert knowledge was encoded as rules in the Nazif and Levine method.

**Lundervold and Storvik (1995)** used a-priori anatomical knowledge to segment MR brain images. Their method uses multispectral nature of MR, prior knowledge of the anatomy and an algorithm which approximates the maximum a posteriori probability (MAP) of the model for the given image.

Some expert system algorithms use directly the knowledge of humans who are expert on image processing. They encode knowledge of expert about selection of image processing subroutines in terms of application. For instance, **Grimm and Bunke (1993)** developed such a system, where the expert system uses an image processing package, called SPIDER, that includes image processing subroutines. The system uses a knowledge base to choose an appropriate action for the given problem. Its knowledge base was capable of determining the general strategy of problem solving by means of a software package, classifying the problem, decomposing it into subproblems, and selecting the appropriate algorithms, subroutines and parameters.

### 2.2.2 Model-Based Methods

Another category of problem-oriented methods is the model-based methods. In these methods, models of objects are obtained from a training set of previous images. Image modelling aims to understand the nature of scene from where the image is generated so as to represent the character of the image with a few parameters **(Dubes and Jain, 1989)**. Models used in the segmentation can be broadly divided into two main categories: Physics-based and stochastic models.

Physics-based models take into account various physical effects such as shading, highlight, interreflection, and texture. The physical models are extracted using the image formation process that describes pixel measurements in terms of light sources, surfaces, and image sensors. These effects are modelled by using some training set and some a priori knowledge about the scene **(Healey, 1996)**. For example, **Healey (1992)** used the dichromatic reflection model derived for metal surfaces. The proposed segmentation method takes into account changes in surface orientation and highlights in color images of metal and dielectric materials using derived reflection models.

Stochastic modelling assumes that there exists a probability distribution that can capture to some extent the variability and the interactions of different sets of relevant image attributes **(Perez, 1998)**. Gibbs and Markov random fields are used in this respect since they permit the introduction of spatial context into the segmentation process. Furthermore, stochastic models provide careful formulations for ill-posed problems in a Bayesian framework. Bayesian decision theory can also be used on random field models as they represent prior information about the image. In these schemes one starts by selecting a model, thereby specifying the probability distribution based on the underlying the application and image properties. Secondly, samples are drawn from the chosen distribution, which can be viewed as generating a realization of the random field, given the model. Model parameters are specified or can be estimated from a training set. Finally, one configures the segment labels that is the best fit to the model. This is usually carried out maximizing the likelihood between the modeled distribution and the label configurations. The considerable computational complexity inherent in this scheme can be reduced by using the equality between the Markov random fields and Gibbs distribution. This enables simpler calculations on smaller neighborhoods instead of the entire image. Besides Markov random fields other models such as autoregressive models and random mosaic models can be used in segmentation. Details of these methods can be found in **Haralick and Shapiro (1992)**.

## 2.3 General Methods

General methods do not take into account specifically either the task or the nature of the given image. The general methods try to segment an image in terms of some heuristic or ad hoc assumptions, like region uniformity, edge continuity, etc. Although they may not be as successful as specific problem oriented methods, they provide a methodological basis of tools that problem oriented methods can exploit. The myriad of general segmentation methods could be divided into four main categories: Region-based, boundary-based, measurement-space-based and hybrid methods.

Region-based methods carry out the segmentation task by taking into account certain region properties. They are based on the assumption that the objects are represented in an image by uniform and homogenous regions, therefore, pixel similarity-based predicates are used. Boundary-based methods, on the contrary, use the assumption of discontinuity between regions. Discontinuity assumption addresses the abrupt changes in an image that possibly correspond to object boundaries. Measurement space-based methods carry out the segmentation task in the measurement domain usually without spatial constraints. Certain pixel features are clustered in the measurement space. Finally, in the hybrid methods a combination of the above approaches is employed. For example, rough contours can be extracted first, then they are elaborated by a region growing algorithm.

### 2.3.1 Region-Based Methods

Region-based methods are executed in the spatial domain by taking into account similarity predicates between pixels. Choosing an adequate criterion for similarity predicate is the main problem. This criterion is often defined a priori and it does not depend directly on the given image.

A well-known region-based method is the region growing approach. Region growing tries to group pixels that are similar to each other in terms of chosen features. Most of region growing methods start the process at seed points and the other neighbor pixels are appended to seed points if they satisfy the similarity criterion. In spite of its simplicity, region growing suffers from some problems. The first problem is choosing the correct threshold value for similarity predicate. If one chooses a high threshold, this causes oversegmentation. On the contrary, if a low threshold is chosen, this yields undesired segment merging. Another problem is the weakness of linkage procedure between regions. If a pixel on the boundary of two distinct regions satisfies the similarity predicate for both regions, this single pixel can cause merging of these two regions.

Besides choosing an adequate similarity criterion, determining the location of seed points is yet another problem since the result of segmentation depends highly on the location of seed points **(Gonzalez and Wintz, 1987)**. Although seed points maybe determined manually, they can also be determined automatically by a priori knowledge. For example, **Adams and Bischof (1994)** proposed to use convergence squares algorithm to find the seed points in the X-ray images of human chest. The seed points must be located in the left and right lung fields and the mediastium (field between two lungs). Two high-level knowledge rules are used to set the seed points. The three seeds chosen consist, for the left and right lung fields, of the minima of the two minimum density regions, and a third suitable seed for mediastium as the maximum of the region of maximum density.

A variation of region-based methods is the split-and-merge methods. It starts from the entire image, and the image is divided sequentially into smaller regions unless they satisfy the uniformity predicate. Then adjacent small regions are merged if the newly merged region satisfies the uniformity criterion as well. Quadtree structure is often used in split-and-merge because of its simplicity and computational efficiency **(Gonzalez and Wintz, 1987)**. It, however, has an inherent drawback which is its insensitivity to image semantics. The rigid rectilinear image tessellation causes edges of segments that consist of only horizontal and vertical orientations. This rigid tessellation procedure increases also the number of split and merge operations. Other tessellation procedures have been proposed to overcome this drawback. **Gevers and Kajcovski (1994)**, for instance, proposed the incremental Delaunay triangulation as a directed region partitioning technique. The initial triangulation consists of two triangles from which the vertices correspond to three arbitrary image corners. This initial coarse triangulation is splitted successively if it does not obey the homogeneity criterion. Vertices are chosen according to a maximum difference criterion defined over region intensity and low-order polynomials calculated using edges found by means of Canny edge detector and line fitting. Thus the new triangulation is adjusted according to the underlying structure of image data.

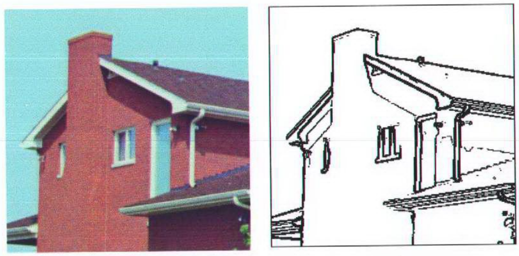
A different region-based method is the watershed transformation, which is a mathematical morphological tool. The watershed algorithm, as implied by its name, segments the regions into catchment basins. In this algorithm, the image is considered as a topographic surface. This surface is gradually flooded by water from its minima, conceptually. The merging of the water that comes from different minima is prevented and the boundary between two catchment basins is the watershed line. Thus the given image is divided into two different sets: Catchment basins and watershed lines. The catchment basins correspond to the segments while the watershed lines denote the boundaries between the segments **(Vincent and Soille, 1991)**. The minima can be located by gradient of the given image. The main drawback of this method is that it causes oversegmentation. Marker-controlled watershed can be used to overcome this problem. In this approach flooding the surface is started from a set of markers that are previously defined manually or automatically. An approach that avoids oversegmentation is the hierarchical watershed transformation that defines a hierarchy among the catchment basins. The gradient image is subjected to an initial watershed transformation resulting in a mosaic image that is associated to the gradient. Another watershed transformation with a different criterion function is applied to the mosaic image. This new criterion is based on relative heights of the walls separating the initial catchment basins **(Beucker, 1999)**.

Segmentation using minimum spanning tree is also a region-based method. The method proposed by **Xu and Uberbacher (1997)** constructs a minimum spanning tree representation of the given image. Thus the region partitioning problem turns to a minimum spanning tree partitioning problem. This causes reduction of the computational complexity of the region partitioning. The tree-partitioning algorithm partitions the minimum spaning tree into subtrees. These subtrees represent different homogenous regions. Partitioning is carried out by minimizing the sum of variations of gray levels over all subtrees under two constraints: First, each subtree should have at least a specified number of nodes, second two adjacent subtrees should have significantly different average gray-levels. A dynamic programming technique is used in tree partitioning process.

Region-based methods can take into consideration the multiresolution data. by using pyramidal structures **(Burt et.al, 1981)**. In this scheme, the image is smoothed and subsampled by a factor of, typically, 2, to constitute a pyramid structure. Each level is linked to upper and lower levels by a linkage rule, known as parent-child relationships. For example, each child node may have four parents in the upper level while each parent may have sixteen children in the lower level. The transition between levels has also some specific rules. In **Burt et.al (1981)**, for instance, each parent takes the value of the mean of its children, and each child takes the value of the parent whose value is closest to itself. These rules are applied from bottom-to-top and top-to-bottom iteratively. The process is continued until no changes occur from one iteration to the next. After convergence, the bottom layer which represents the entire image is adopted as the segmentation map.

### 2.3.2 Boundary-Based Methods

In contrast to the region-based methods, boundary-based methods carry out segmentation task by taking into account discontinuity. Discontinuities or abrupt changes in the image occur due to the surface reflectance, illumination, occlusion or even noise **(Lim, 1990)**. The ensemble of discontinuity points represents the boundaries of the segments called edge. In Figure 2.2, the edge map of the "House" image is represented.



(a) (b)

Figure 2.2 Edge map of "House" image. (a) Original image (b) Edge map

Edges can be detected using simple gradient or Laplacian based edge detectors. These detectors use the local derivative operators which produces high value in the image pixels whose gray levels change rapidly **(Gonzalez and Wintz, 1987)**. Sobel, Prewitt, Roberts, Frei-Chen, Canny and Deriche are well known edge detection operators. Details of these methods can be found in **(Pratt, 1991)**. **Marr and Hildreth (1980)** proposed the Laplacian of Gaussian (LoG) operator. This operator, represented by $\nabla^2 G$, consists of the second partials of the Gaussian function, that is, $\frac{\partial^2}{\partial x \partial y}\left(e^{x^2+y^2}/2\pi\sigma^2\right)$. While the Gaussian function of LoG smooths the image, the Laplacian portion in the LoG is used to detect the edges along zero transition locations. The results at different scales can be obtained by varying the standard deviation parameter, $\sigma$. One important problem is that it is not possible to know a priori which gradients correspond to a meaningful edge and which do not.

Finally boundaries can be detected from these edges by energy minimization based methods. These methods try to find the boundary of segments by minimizing some energy functional. Among several approaches, active contour and regularization based methods are the most popular.

Active contour models, or snakes, were popularized by the seminal paper of **Kass et.al (1987)**. Snakes can be defined as an energy minimizing spline whose energy depends on its shape and proximity to the features in the image. The minimization process should reward the boundaries, which are locally smooth and located in high-gradient image regions. The initialization of a snake critically affects its performance. **(Lisin, 1999)**.

The energy functional consists of internal and external forces. The internal forces come from the shape of the snakes, while the external ones come from the image itself. The snake is defined in a parametric form along the contour as a function of the arclenght s: v(s) = ( x(s), y(s) ) where x(s) and y(s) are x and y coordinates along the contour with arclength $s \in [0,1]$. The energy functional to be minimized is as follows:

$$E_{snake} = \int_0^1 \left(E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))\right) ds \tag{2.1}$$

The internal energy can be written as Equation 2.2:

$$E_{int} = \alpha(s)\left|\frac{dv}{ds}\right|^2 + \beta(s)\left|\frac{d^2v}{ds^2}\right|^2 \tag{2.2}$$

where $\alpha(s)$ and $\beta(s)$ specify the elasticity and stiffness of the snake. Thus the first term makes the snake act like a membrane whereas the second one makes it act like a thin plate **(Kass et.al., 1987)**.

The second term of the energy is computed over the image area that the snake lies. This term can be constructed as a weighted combination of three different functionals as seen in Equation 2.3. These functionals attracts the snakes to lines, edges and terminations. The formulation of these terms is given in Equations 2.4, 2.5 and 2.6. The weights are used to control the behaviour of the snake.

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term} \tag{2.3}$$

$$E_{line} = f(x,y) \tag{2.4}$$

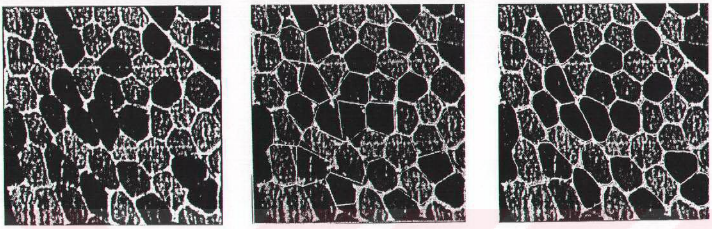
$$E_{edge} = -\left|\nabla f(x,y)\right|^2 \tag{2.5}$$

$$E_{term} = \frac{\left(\frac{\partial^2 g}{\partial y^2}\right)\left(\frac{\partial g}{\partial x}\right)^2 - 2\left(\frac{\partial^2 g}{\partial x \partial y}\right)\left(\frac{\partial g}{\partial x}\right)\left(\frac{\partial g}{\partial y}\right) + \left(\frac{\partial^2 g}{\partial x^2}\right)\left(\frac{\partial g}{\partial y}\right)^2}{\sqrt{\left(\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2\right)^3}} \tag{2.6}$$

In these formulations, f(x,y) and g(x,y) denote the original image and the smoothed image, respectively, while $\nabla$ denotes the gradient operator.

The third term denotes the constraints imposed by user. These constraints are used to force the snake toward or away from some features.

If a priori information about the target objects is available, snakes can be effectively used to segment them. For example, muscle fibre cell images were successfully segmented by means of Voronoi diagrams and active contours in **Klemencic et.al. (1996)**. In this study, firstly a Voronoi diagram is constituted according to the center of cells. These diagrams are used to initiate the snakes. Then the active contour models are executed to fit the snake the exact boundaries of the cells. An example of this process can be seen in Figure 2.3.



(a) (b) (c)

Figure 2.3 Example of snake based segmentation. (a) Original muscle fibre image. (b) Initial snakes by obtained Voronoi diagrams. (c) After active contour process. Voronoi diagrams close to cell boundaries. (Reprinted from **Klemencic et.al. (1996)** )

Another popular energy based approach is regularization-based edge detection. In general, early vision problems are characteristically of ill-posed nature. If either the solution of a problem does not exist, is not unique, or does not vary continuously with the data, the problem is called as ill-posed. Ill-posed problems can be converted to well-posed ones by imposing some constraints on the solution. These constraints can restrict the number of solutions or to insure that solutions vary smoothly in terms of data. Regularization theory gives the following functional, which imposes smoothness constraint, to be minimized:

$$E_m(f) = \iint_{\Omega} (f-d)^2 \, dx\, dy + \lambda \iint_{\Omega} (f_x^2 + f_y^2)\, dx\, dy \tag{2.7}$$

where d denotes the data, f represents the solution. $f_x$ and $f_y$ denotes derivation of solution along x and y direction. The first integral term insures that the solution depends on the data while the second one imposes the smoothness constraint on the solution. $\lambda$ controls the degree of smoothing **(Gokmen and Li, 1992)**. Several approaches can be used to determine solution space, f. For example, Canny suggested the first derivative of a Gaussian as the function that minimizes this functional approximately **(Canny, 1986)**. The second term which depends on first derivatives is called membrane model. The second derivatives can also be used. In this case the term is named as plate model. These models can be combined to obtain a new energy functional as in Equation 2.8. The functional can be scaled by varying $\lambda$ and $\tau$ parameters. In this case, multiscale edge detection is possible **(Gokmen and Jain, 1997)**.

$$
\begin{aligned}
E_m(f) = & \iint_{\Omega} (f-d)^2 \, dx\,dy + \lambda \iint_{\Omega} (f_x^2 + f_y^2)\, dx\,dy + \\
& \tau \iint_{\Omega} (f_{xx}^2 + 2 f_{xy} f_{yx} + f_{yy}^2)\, dx\,dy
\end{aligned}
\tag{2.8}
$$

The main drawback of these methods is that there usually exist gaps in the detected edges so that the contours are not complete. On the contrary, some false edges could appear due to noise, which cause small holes in the segments. Thus one of the requirements of the segmentation stated in Chapter 1 is violated. Therefore some postprocessing operations, called edge linking, are required to enhance the boundary integrity.

Various edge linking techniques can be used to enhance the edges and to fill the gaps. A simple edge linking technique is the curve fitting. In this method, the broken part can be connected to the contour by using polynomial, spline or piecewise linear curve fitting. This method is effective for simply structured objects. If the image contains many overlapping objects or its edge map contains branch structure the fitting process may be cumbersome **(Pratt, 1991)**.

A widely used method in edge linking is the Hough transform. The edge map is subjected to the Hough transformation tuned to straight lines. The sufficiently large cells on the Hough space correspond to the straight lines in the image plane. The line is overlayed with the edge map. If the line coincides with any edge it can be used as a mask to fill gaps in the edge **(Pratt, 1991)**.

There exist some other heuristic edge linking methods. For example, one of the simplest ways of linking is analysing the characteristics of edge points in a small neighborhood such as 3x3 or 5x5. If all edge pixels are similar, they are linked. Two properties are used for similarity: First is the strength of the response of the gradient operator and second is the direction of the gradient. If the absolute difference of the gradients of two neighbour edge points is under a specific threshold, the first similarity predicate is satisfied. Similarly, if the absolute difference of the gradient angle of two neighbour edge points is under another specific threshold, the second predicate is also satisfied **(Gonzalez and Wintz, 1987)**.

Roberts proposed another heuristic edge linking technique. In this method, the edge gradients are examined in 4x4 pixels blocks. If the magnitude of the pixel that has the largest magnitude in the block is greater than a threshold, it is declared as a tentative edge point. Then the four main oriented lines of length five are fitted to the gradient data about the tentative point. A ratio of the best fit to the worse fit that is measured according to fit correlation is calculated. If this value is greater than a specific threshold the tentative point is declared valid and it is assigned to the direction of the best fit. Next, straight lines are fitted between pairs of edge points if they are in adjacent 4x4 blocks. This method is capable of eliminating small triangles by deleting their longest side. Small rectangles are replaced by their longest diagonal. Short spur lines are also deleted **(Gonzalez and Wintz, 1987)**.

### 2.3.3 Measurement Space-Based Methods

Measurement space-based methods take into account only measurements obtained from the pixels such as their gray level, color, texture parameters, local means, and local variances, etc. These algorithms do not consider in general the positions of the pixels. The segmentation decision is carried out by grouping of measurement values. There exist two main categories of the measurement space-based methods: Multithresholding and clustering.

Multithresholding is a tool for segmentation exploiting the histogram information of the image. Given a histogram and (n-1) threshold values $t_1$, $t_2$, ..., $t_{n-1}$, all pixels with gray values between $t_i$ and $t_{i+1}$ ( i = 0, 1,....n; $t_0$=0, $t_{n+1}$=255) are said to belong to the same segment. For examples for n = 2, there exists only one threshold value and the pixels whose gray value are under this threshold are mapped to the background while the others are said to belong to the objects. There are lots of thresholding methods in the literature **(Sahoo et.al., 1988) (Pal and Pal, 1993)**.

The main issue in thresholding is to determine the threshold value automatically. If the gray level distributions belonging to object and background are very different from each other, their histogram has two well-separeted modes as seen in Figure 2.4.a. In this case, the threshold value can be chosen in the sharp valley between the two modes. However if the distributions of object and background overlap each other, choosing an appropriate threshold becomes a more challenging problem, since the valley may disappear as seen in Figure 2.4.b. In this case, the threshold can be found by using various other criteria such as entropy, error probability, Kullback information distance, clustering measure, etc.

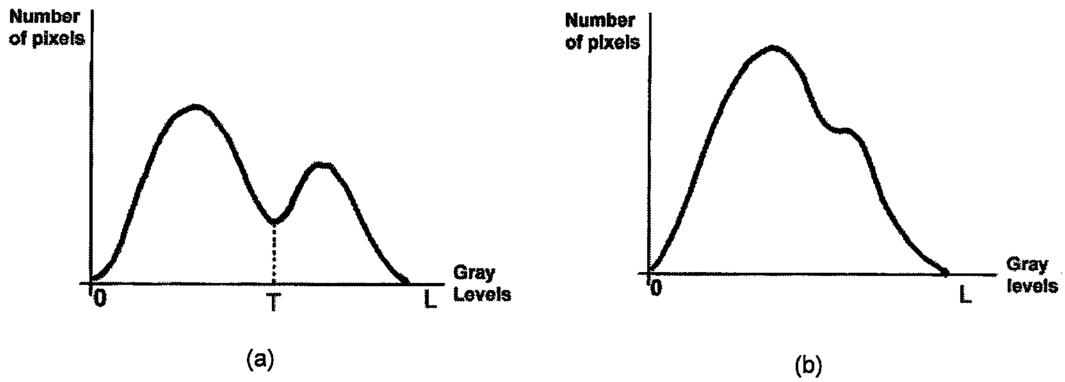
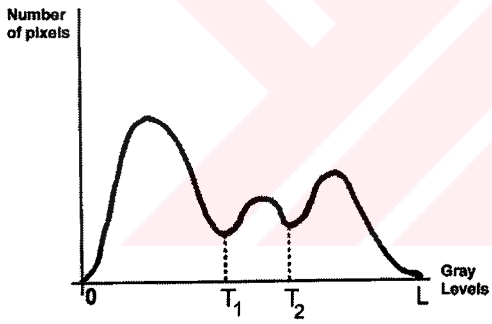



Figure 2.4 Various image histograms. (a) A bimodal histogram. Thresholding is easily carried out by choosing threshold value as T. (b) A histogram with overlapping modes. Choosing threshold value is difficult.

In general, the image may have several objects in the background, resulting in a multimodal histogram as shown in Figure 2.5. In this histogram, the objects can be separated from each other as well as from background by choosing two different thresholds ($T_1$ and $T_2$).




$$f_T(i,j) = \begin{cases} f(i,j) \le T_1 & \text{Background} \\ T_1 < f(i,j) \le T_2 & \text{Object 1} \\ f(i,j) > T_2 & \text{Object 2} \end{cases}$$

Figure 2.5 Histogram of an image that has two objects on the background.

The segmentation problem can be cast as a clustering problem if the spatial relationship between pixels is not taken into account. The objective of clustering is to group the pixel feature vectors and to assign the same label to all the pixels that are classified into the same cluster. Clustering is an important tool in pattern recognition. Some clustering algorithms are summarized in following paragraphs.

**Haralick and Kelly (1969)** proposed a clustering tool to extract similar regions in the multispectral aerial images of agricultural land. For this aim, first the feature vectors are extracted from pixels. Such features as multispectral or color components, local mean and local standard variation are considered. If the chosen features are effective for segmentation, feature vectors obtained from the same segment are similar each other. In other words, if the feature vector has N features (N-tuple), these vectors tend to come together in the N-dimensional space. Figure 2.6 shows clustering of data into three segments in the two dimensional space **(Pratt, 1991)**.

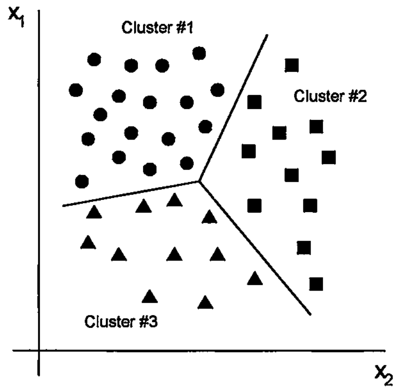



Figure 2.6 For two dimensional features, classification boundaries in the feature space. Each class corresponds to a different segment.

**Coleman and Andrews (1979)** proposed another clustering based segmentation method. The flowchart of the algorithm is given in Figure 2.7. The first step of the algorithm is the feature extraction step where twelve features computed over windows of sizes 1, 3, 7 and 15 pixel are used. In the second step, the optimal number of classes is determined by means of centers of each cluster in the measurement space. A class quality factor computed over the trace of between-class and within-class matrices is defined and used for this aim. The number of classes that maximizes this criteron is chosen as optimal number of classes. In the last step, a classifier labels the pixels by assigning its feature vector to the closest center of a class **(Pratt, 1991)**.

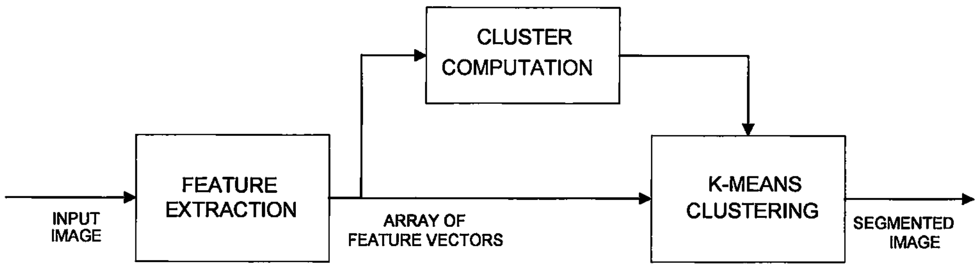



Figure 2.7 Flowchart of Coleman and Andrews method.

There are a lot of clustering based segmentation methods in literature such as parametric classifiers, ANN based classifiers, fuzzy classifiers **(Pal and Pal, 1993)**. Diversity among these methods arises in the design of clustering. In principle, all methods classify features obtained from images. Disadvantages of clustering-based segmentation is that they do not consider the spatial relationship between pixels.

The pixels that do not reside in the same segment but have similar feature values in measurement space may be included into the same segment, thus causing noisy segments.

## 2.4 Hybrid Methods

Some general methods can combine two or more different paradigms to enhance the performance of segmentation. **Pavlidis and Liow (1990)**, for example, used both split-and-merge and edge detection techniques for segmentation. First the image is segmented by using a split-and-merge technique. Then the artificial edges coming from the split-and-merge process is removed by using a priori horizontal and vertical edge detection. After this elimination process, a contour smoothing algorithm on the edges is performed to approximate the artificial edges to actual edges. **Chu and Aggarwal (1993)** and **Xiaohan and Yla-Jaaski (1992)** proposed a method that combines region growing and edge detection. **Zugaj and Lattuati (1998)** extended this approach to color images. **Zhu and Yuille (1996)** proposed a method that unifies snakes, region growing and Bayes/MDL approaches.

Relaxation based segmentation methods also combine measurement space and region-based methods. The given image is initially segmented by means of any clustering method. Then a relaxation method makes the initial labels of pixels concordant with its neighbours. In this method measurement-based and region-based approaches are used together.

Measurement-based and boundary-based methods are combined in **Qian and Zhao (1997)**. The method has two stages. In the first stage, two initial segmentation processes are carried out by means of a Fuzzy Gaussian Basis Function Neural Network (FGBFNN) and Canny edge operator. The number of classes and fuzzy memberships of pixels are found by FGBFNN. An edge map of image is obtained by using Canny edge operator. In the second stage, a modified fuzzy c-means algorithm (MFCM) is used to obtain the final segmentation.

## 2.5 Conclusion

A taxonomy of segmentation methods has been given in this chapter. It was emphasised that problem-oriented methods give better results than general methods since they inherently take into account knowledge about specific scene or given application. But problem-oriented methods usually carry out the segmentation using some tools derived from general methods in terms of given application. Thus, thousands of general methods provide a methodological basis for problem-oriented ones.

Measurement space-methods provide simply tools for especially real-time applications. In this thesis, extension of multithresholding to color images has been considered. A novel 2-Dimensional histogram and fusion based method has been proposed in Chapter 4.

In Chapter 5, Constraint satisfaction neural network (CSNN) based segmentation method has been examined. The method consists of two main stage: Initialization and relaxation. The former corresponds to measurement space- based while the latter can be considered as relaxation. Some enhancements have been carried out in the initialization stage to improve the performance of the CSNN. Four different methods which is obtained by modifying the network structure of the method have also been proposed to enhance the performance.

# 3. EVALUATION AND ASSESSMENT OF IMAGE SEGMENTATION

## 3.1 Introduction

As mentioned Chapter 1, there is no general segmentation method, which is appropriate for all applications. Thus identifying the suitable segmentation method for the given application is a crucial problem. In this respect, quantitative comparison criteria could provide guidance for the selection among several algorithms. These criteria, under certain conditions, may provide an objective assessment of segmentation performance in lieu of visual subjective comparisons. Secondly, most of image segmentation methods require some initial setting or subsequent adjustment of parameters, and the segmentation outcome often depends upon the choice of these parameters. Choosing the correct parameters for the given application can also be instrumented via performance criteria. **(Zhang, 1996)**. Finally, the third importance of performance indicators comes up in the feedbacked image analysis operations. This type of system tries to improve its performance by modifying its strategy according to the feedback from the current result. Here the quantitative measure of the performance plays the key role in providing a feedback path from the current outcome to the algorithm **(Levine and Nazif, 1985)**. Consequently, the performance evaluation and comparison of image segmentation algorithms is an indispensable tool in the segmentation problem.

Although hundreds of image segmentation techniques have been proposed so far, there has been little progress in the performance evaluation of these techniques. In this chapter, first a critical assessment of existing performance evaluation methods will be presented. Second, the novel proposed evaluation strategy will be explained. Finally merits of the proposed strategy will be shown in the test images. In addition, an image generation procedure, which creates test images with ground-truth information, is presented in this chapter. The real world images, which will be used as test cases in this thesis, are also introduced.

## 3.2 Evaluation Methods for Image Segmentation

A thorough review of evaluation methods for image segmentation can be found in **Zhang (1996)**'s seminal paper. According to Zhang's taxonomy, the evaluation methods are broadly divided into two categories as shown in Figure 3.1: Analytical and empirical methods.

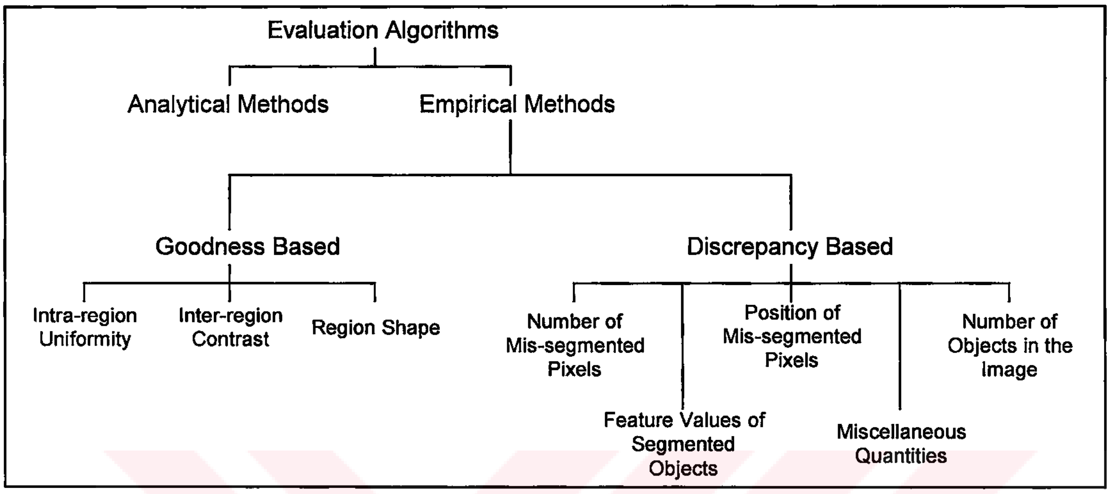



Figure 3.1 Taxonomy of Segmentation Evaluation Algorithms

In the analytical methods, the segmentation algorithms are directly evaluated by taking into account their principles, properties, requirements, utilities, strategies, complexities, etc. The analysis of a segmentation method is only carried out by considering the implementation of the algorithm, while the segmented output is not taken into consideration **(Zhang, 1996)**. This class of evaluation methods gives only information about algorithmic structure of the method under consideration. Although certain properties such as the processing strategy and resolution of segmentation can be obtained easily by analysis, some properties such as the leverage of a priori knowledge of the image cannot be assessed precisely. Thus, the application of this kind of evaluation methods would not be satisfactory for quantitative comparison of the algorithms.

The empirical methods deal with the results of segmentation algorithms rather than their structure, thus offering more quantitative measures than the analytical ones. The two kinds of empirical methods are the "goodness" based ones and the "discrepancy" based ones. The goodness-based methods evaluate the segmented image to indirectly assess the performance of algorithms, whereas the discrepancy-based ones require a reference test image, which is also called ground-truth image. In the discrepancy-based case, the test image, which is often synthesized and rendered starting from a ground-truth image, is segmented by using the algorithm under consideration and the segmented output is compared with the reference ground-truth map. The difference between the ground-truth and the test image is used to assess the performance.

Goodness-based methods can be categorized into three classes: They are based on intra-region uniformity, inter-region contrast and region shape. Discrepancy-based methods are divided into five groups: They are based on the number of mis-segmented pixels, the position of mis-segmented pixels, the number of objects in the image, the feature values of segmented objects and miscellaneous quantities. A brief description of these methods is as follows. The details can be found in **Zhang (1996)** and related references.

- Goodness-based methods focusing on region uniformity assume that an adequate segmentation yields higher intra-region uniformity. For this aim, **Weszka and Rosenfeld (1978)** used a busyness measure based on co-occurence matrix that reflects the uniformity of thresholded images. **Levine and Nazif (1985)** proposed a gray-level region uniformity criterion based on the variance of the gray-levels in the detected regions. **Sahoo et.al (1988)** also used a normalized version of **Levine and Nazif (1985)** approach.

- Goodness-based methods focused on region contrast use the fact that the contrast between the adjacent regions must be high. This measure has been proposed by **Levine and Nazif (1985)**.

- **Sahoo et.al. (1988)** proposed another measure which takes into account the shape quality of the thresholded objects such as smoothness of the boundaries. This method can be considered as goodness-based on region shape method.

- Discrepancy-based methods focused on the number of mis-segmented pixels use the confusion matrix to derive a criterion. The confusion matrix is constructed by matching the segmentation results of the ground-truth reference and test images. Each entry, $C_{ij}$, represents the number of class j pixels of the reference image classified as class i in the test image. More details of these methods will be given in Section 3.3.

- Discrepancy-based methods focused on the position of mis-segmented pixels take into account not only the number of mis-classified pixels but also their positions to involve spatial information. **Yasnoff et.al. (1977)**

proposed to use the distance between a mis-classified pixel and the nearest pixel that actually belongs to the mis-classified class. **Abdou and Pratt (1979)** also proposed a similar figure-of-merit to evaluate the edge detectors. **Strasters and Gebrands (1991)** used this figure-of-merit approach for evaluating segmentation results.

- The number of objects in both the reference and test image must be equal. This is the key idea behind the discrepancy-based methods focusing on the number of objects.

- The shape of the segmented objects must be similar to the ones in the reference image as much as possible. Then the feature values of the segmented objects such as area, perimeter, form factor and eccentricity provide well-defined discrepancy-based measures. Details of this type of methods are discussed in Chapter 3.4.

- The other discrepancy measures that do not fit into any of above categories are collected under discrepancy-based methods concerned with other miscellaneous quantities. For example, figure of certainty, which takes into account the gray-level information accompanying the spatial information to define a discrepancy measure, as proposed by **Straster and Gebrands (1991)**, can be considered to be in this category. Symmetric divergence, which uses the a posteriori probabilities of a pixel to be in the reference and test images proposed by **Pal and Bhandari (1993)**, can also be given as an example.

In this thesis, the performance of segmentation algorithms is quantified by using discrepancy-based methods whenever synthetic images are used. Goodness-based methods are utilized for natural test image for which it is not possible to know exactly the exact segmentation map.

Any empirical evaluation method must be able to quantify two important aspects:

- Good detection of classification difference: It must give higher score to the algorithms that generate segments more similar to the reference one in the spatial domain. This can be accomplished by taking into account classification error resulting from the segmentation process.

- Good detection of shape formation: It should yield higher scores for the algorithms that generate segments whose shapes are more similar to the reference one.

In the following sections, two discrepancy features that are the number of mis-segmented pixels and the turning-angle function of segmented objects are comparatively examined. Their shortcomings and strengths are illustrated. Finally, a novel evaluation technique, which combines these two features, is presented.

### 3.3 Discrepancy-Based on the Number of Mis-segmented Pixels

The most intuitive discrepancy-based method is the percentage of the mis-segmented pixels. A confusion matrix can be used to obtain this basic discrepancy measure. The confusion matrix is constructed by using the ground-truth reference and the test images. If the reference image has N segments, the matrix consists of N rows and N columns, which correspond to the segments of the test and the ground-truth reference images, respectively. Each entry of the confusion matrix, $C_{ij}$, represents the number of pixels which belong segment j in the reference image and that are segmented as class i by the segmentation algorithm. The trace of the matrix represents the number of correctly segmented pixels.

**Yasnoff et.al. (1977)** proposed two types of error measure for each segment class. The first one called Type I error (Classification Error I, $CE_I$) takes into account the ratio of the number of pixels of segment k not classified as k to the total number of pixels of segment k. It, therefore, reflects the percentage of pixels that are attributed to incorrect segments:

$$0 \leq CE_I^{(k)} = \frac{\sum_{i=1}^{N} C_{ik} - C_{kk}}{\sum_{i=1}^{N} C_{ik}} \leq 1 \tag{3.1}$$

where $C_{ik}$ denotes the count of pixels of class i classified as in class k. The other one, called Type II error (Classification Error II, $CE_{II}$), gives the ratio of the number of pixels belonging to other classes but labelled as class k to the total number of pixels of other classes. Then this measure represents the error for wrongly segmented pixels of class k.

$$0 \leq CE_{II}^{(k)} = \frac{\sum_{j=1}^{N} C_{kj} - C_{kk}}{\sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij} - \sum_{i=1}^{N} C_{ik}} \leq 1 \tag{3.2}$$

The total classification error for a class, $CE^{(k)}$, is given as the average value of these errors:

$$CE^{(k)} = \frac{1}{2}(CE_{I}^{(k)} + CE_{II}^{(k)}) \tag{3.3}$$

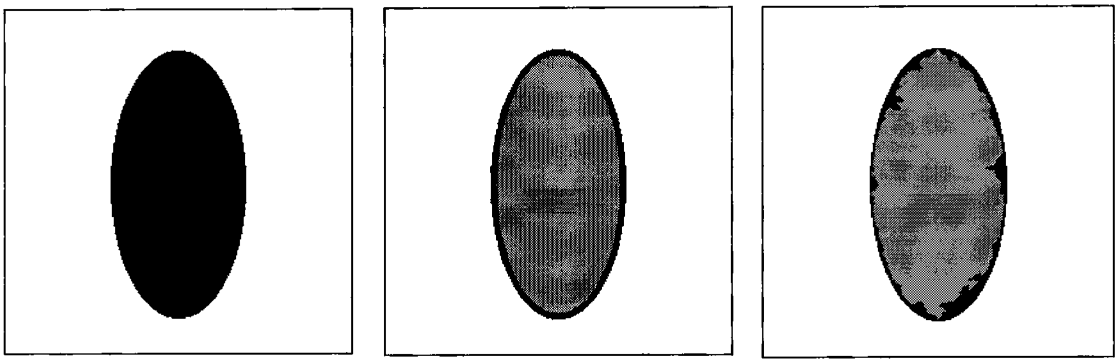
The overall error measure, CE, can be obtained by taking the average over the per class classification errors:

$$0 \leq CE = \frac{1}{N}\sum_{k=1}^{N} CE^{(k)} \leq 1 \tag{3.4}$$

This error as a performance indicator provides good detection of classification difference, and it is sensitive to the position of objects. But it fails to detect any deformation of the shape of objects. This effect is depicted in Figure 3.2. Figure 3.2.a shows the ground-truth of a reference image consisting of a single ellipse. Two different symbolically segmented images obtained with different algorithms are also depicted in Figure 3.2.b and Figure 3.2.c. While the two segmentation results have exactly the same CE figures, they differ from each other in shape properties. It is seen that the first segmented image has a shape much closer to the reference one. This is in fact, as discussed later in Section 3.4, shown quantitatively in Table 3.1 by comparing CE score with the shape deformation score measured by the turning-angle function (TAF). Thus, while the CE measure gives the same result for both segmented images, the shape deformation indicators differ between Image I and Image II by a factor of more than 6. As a consequence, although the CE measure provides good detection of classification difference it does not take into account the shape factor of the segmented objects.

Table 3.1 Classification error (CE) versus Turning-angle function (TAF) for the example given in Figure 3.2

| | CE measure | TAF measure |
|---|---|---|
| Segmented Image I | 0.063 | 0.029 |
| Segmented Image II | 0.063 | 0.186 |



(a) Ground-truth image (b) Segmented image I (c) Segmented image II

Figure 3.2 Illustration of the CE measures. (a) The ground-truth image. (b) and (c) Segmented images with different algorithms. Gray region represents the segmented shape while the black regions represent the missing pixels that were assigned to the background. Notice that the two segmented regions have the same area but different shapes.

### 3.4 Discrepancy-Based on the Feature Values of the Objects

In the higher-level image analysis, certain object features such as area, perimeter, form factor obtained from the segmented objects may play a dominant role. Thus these features can be employed in the segmentation evaluation metric. In this context, a segmented image has the highest quality if the object features extracted from it precisely match the features of those in the reference image **(Zhang, 1996)**.

**Zhang and Gerbrands (1994)** proposed such a criterion for performance assessment called ultimate measurement accuracy (UMA). They considered that different object features can be important in different applications, and proposed a goal-oriented criterion. Two sorts of measures are proposed depending on how the comparison is to be made: Absolute UMA (AUMA) and relative UMA (RUMA).

$$\mathrm{AUMA} = |R_f - S_f| \tag{3.5}$$

$$\mathrm{RUMA} = \frac{|R_f - S_f|}{R_f} \tag{3.6}$$

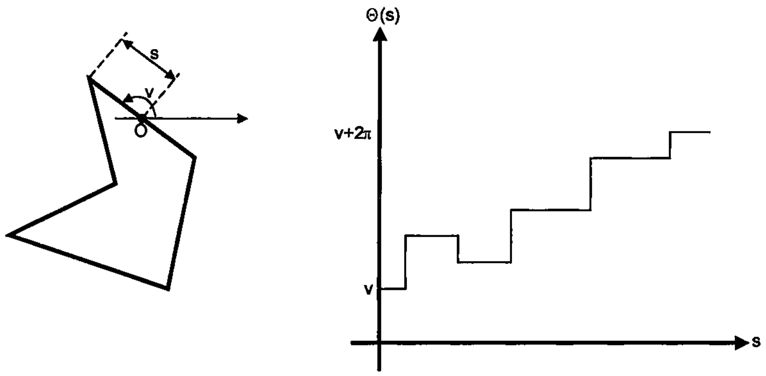
where $R_f$ and $S_f$ denote the feature values extracted from the reference and segmented images, respectively. The AUMA and RUMA are inversely proportional to the segmentation quality. The smaller the values, the better the segmentation performance.

An object feature can be chosen to fit the application or different features can be combined by means of a weighted summation according to the importance of each feature. In the image applications of cell biology, robotics and industrial inspection, geometric features of objects such as area, perimeter, form factor, eccentricity, circularity have been used. For example, the form factor given in 3.7 provides a good feature for circular or elliptical objects:

$$F = \frac{P^2}{4\pi A} \qquad (3.7)$$

where P and A denotes the perimeter and the area of the object. Nevertheless, it fails for complex shaped objects.

A geometric feature that is also a metric of shape characteristics of an object is the turning-angle function (TAF) **(Arkin et.al., 1991)**. Turning-angle, $\Theta(s)$, is a function that represents the boundary characteristic of an object. It measures the angle of the counterclockwise tangent with respect to the x-axis as a function of the arc-length s. In **(Arkin et.al., 1991)** the turning-angle has been used to compare and sort polygonal shapes. Measurement begins from a reference point on the object boundary and it takes into account vertices on the boundary. The angle between breakpoints in the boundary, v, is calculated according to x-axis. Measurement keeps track of the turnings at the vertices until reaching the reference point. If the turn takes place on the left-hand side, the function is increased, otherwise, it is decreased. An example of turning-angle function of a polygonal shape is given in Figure 3.3. In this example, O represents the reference point and the perimeter of the object is normalized to 1. Since the polygon has a closed shape, $\Theta(s)$ begins from some value v and increases to v + $2\pi$.




(a) (b)

Figure 3.3 Definition of the turning-angle function. (a) A polygonal object (b) Turning-angle function of the object. O represents the starting point. v indicates the turning-angle according to x-axis. s denotes the arc length.

Since **Arkin et.al. (1991)** has used the turning-angle function to compare only polygonal shapes. Therefore, the definition of the turning-angle must be modified for arbitrary shapes occurring in segmentation evaluation process.

The difficulty with the arbitrary non-polygonal shapes is that the turning-angle function is very noisy if it is estimated on a short support region. The intuitive choice of arc length is one pixel interval. But in digital spaces this yields turning-angles which are only 0, 90, 180, 270 or 360 degrees. It is experimentally determined that three-pixel support for the computation of the turning angle gives adequate results. Calculation of turning angle for a boundary point is depicted in Figure 3.4. The differential angles between the line, which passes through successive and preceding pixels, and the x-axis is adopted as the turning-angle of the pixel under consideration. The calculated turning-angle function is smoothed by using a simple arithmetic smoothing filter so that digital quantization effect is eliminated.

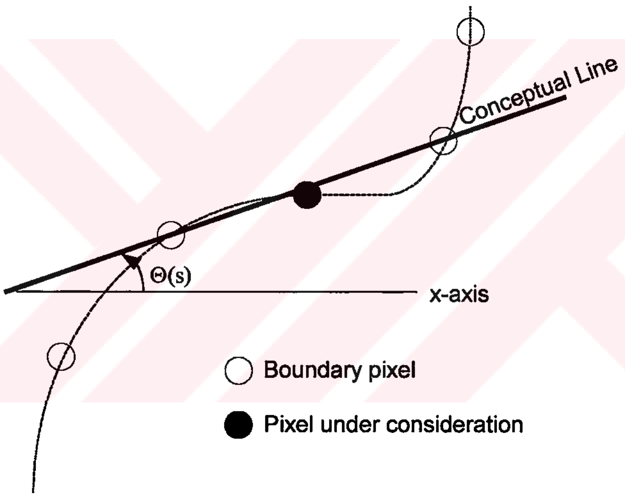


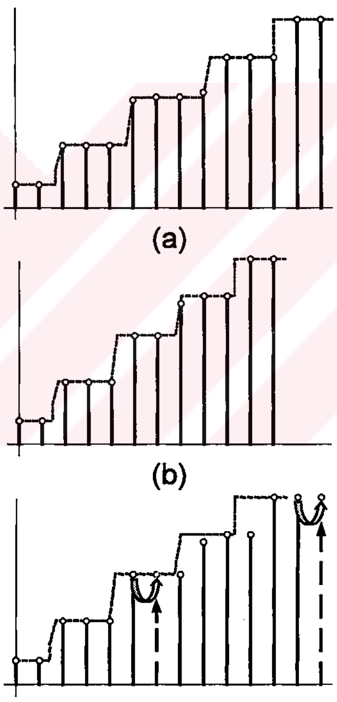

Figure 3.4 Calculation of turning angle. The angle of the pixel under consideration is computed by using the pixels which are two forward and backward from the current pixel.

The turning-angle function (TAF) of any object provides a shape metric. For this aim, the turning-angle functions of the objects in both the reference and segmented images, $\Theta_R(s)$ and $\Theta_S(s)$, are calculated. By using the $L_1$ norm, the TAF measure can be given as follows:

$$0 \leq \mathrm{TAF} = \frac{\sum_{s=1}^{P} \left| \Theta_R(s) - \Theta_S(s) \right|}{P * 2\pi} \leq 1 \tag{3.8}$$

where P is the perimeter of the object. Since the maximum difference between any two pixel orientations is $2\pi$, this measure takes values between 0 and 1. Thus, it is commensurate to be paired with the classification error, CE.

The perimeters of the two objects to be compared may in general differ. In this case, the turning-angle functions do not have the same length. To enable the calculation of TAF as in Equation 3.8 one can use upsampling. The turning-angle function whose length is smaller is upsampled so that its length should match the one that has bigger length. Zero-order interpolation is used for upsampling operation as shown in Figure 3.5. The number of points sizes of difference length between bigger and smaller functions should be inserted to the smaller function. The insertion locations are distributed as equal step size. At the insertion locations, the preceding value of turning-angle function is repeated.



(a)

(b)

(c)

Figure 3.5 Upsampling Operation. (a) Turning-angle function belonging to shape whose perimeter is bigger. (b) Turning-angle function belonging to shape whose perimeter is smaller. (c) Upsampled version of (b). Its size matches the (b). At the insertion points, the preceding value of function is repeated.

The turning-angles of two objects to be compared may differ depending on the starting points and the rotation of the objects. However in our case the reference and test objects can be initialized at the same point provided they have a common boundary point and there is no relative rotation between the two objects.

In Figure 3.6 the turning-angle functions of some regular and irregular shapes are illustrated.

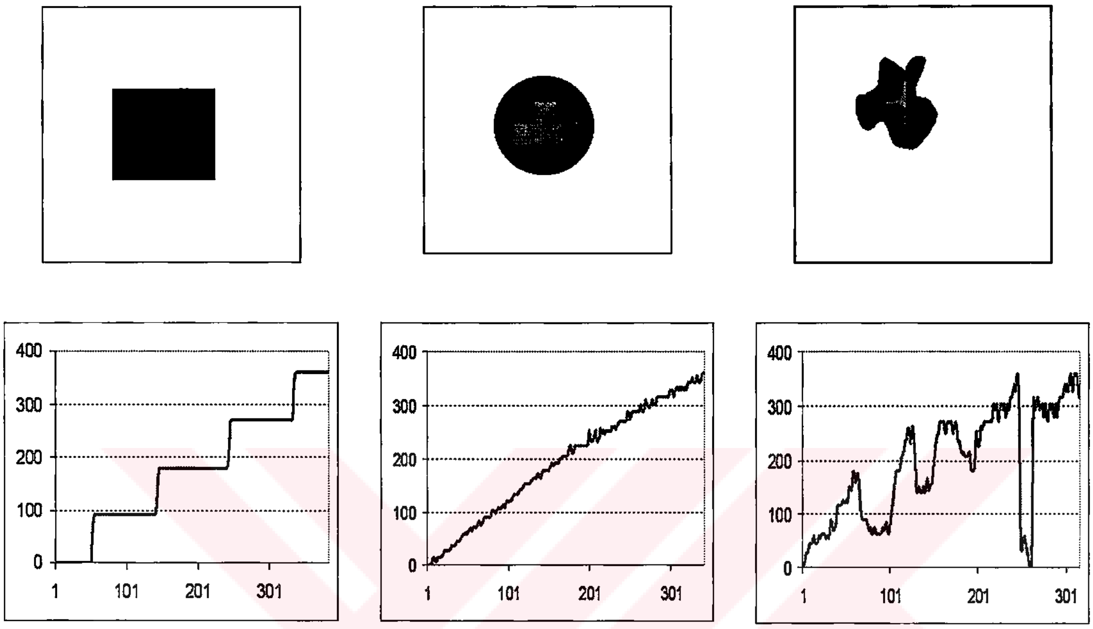
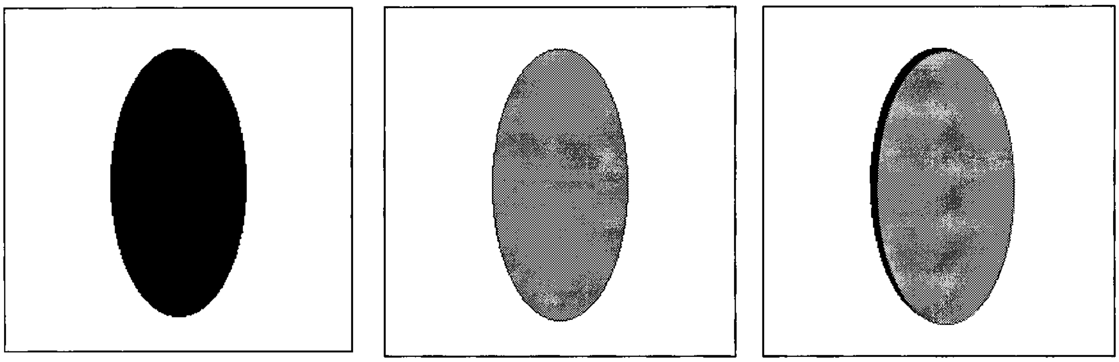



Figure 3.6 Turning-angle functions of different objects.

As a consequence, in any performance evaluation task, the turning-angle functions of each object can be extracted from both the reference and test objects and the measure given in 3.8 can be calculated. The overall performance criterion is obtained by the averaging over TAF values of all objects.

It can be said that TAF shows the performance of the shape extraction ability of the segmentation algorithm quite well. However, it fails to indicate some classification errors. If a segmentation algorithm is able to extract the objects without corrupting their shape but in the wrong position, TAF gives a good score. This effect is illustrated in Figure 3.7. The reference image is depicted in Figure 3.7.a. The two symbolical segmentation results are also given in Figure 3.7.b and Figure 3.7.c. In the first segmented image the object is perfectly formed and located correctly. In the second one, it is also well formed but is extracted in the wrong position. Turning-angle functions give the same result for these segmentation results as depicted in Table 3.2. But it is evident that the first segmentation result is superior to the second one. This is brought into light by means of the CE measure, where indeed, CE gives the perfect score for the first segmentation while a lower score for the second one. Similar cases occur when the segmented object is almost uniformly contracted or thickened vis-à-vis the ground-truth object.



(a) Ground-truth image (b) Segmented image III (c) Segmented image IV

Figure 3.7 Example for TAF measures. (a) The ground-truth image. (b) and (c) Two different segmentation results. The grey region represents the segmented shape while the black region represents the pixels wrongly assigned to the background. Notice that the two segmented regions have the same shape but they are located in different positions.

Table 3.2 Turning-angle function (TAF) versus classification error (CE) for the example given in Figure 3.7

| | TAF measure | CE measure |
|---|---|---|
| Segmented Image III | 0.000 | 0.000 |
| Segmented Image IV | 0.000 | 0.120 |

## 3.5 Combination of CE and RUMA Criteria: A Novel Evaluation Criterion

As shown in Chapters 3.3 and 3.4, the two evaluation criteria, i.e, CE and TAF have sometimes complemental properties. CE has good detection of classification but poor shape distortion monitoring ability, while TAF has good shape distortion monitoring but poor detection of classification capability. It is conjectured that the combination of these two criteria will satisfy both the localization and shape formation requirements. Such a combination can be achieved by using their Minkowski sum:

$$M_p = \left(\alpha CE^p + \beta TAF^p\right)^{1/p} \tag{3.9}$$

where p is order of the metric, $\alpha$ and $\beta$ are the weights for the measures. This metric allows not only weighted contributions from the two criteria but also can adjust the penalty via the p power. For example, if p is equal to 1 with $\alpha=\beta=½$ this metric becomes the arithmetic average. It becomes the root mean square average for p = 2. Finally, if p goes to infinity, then the $M_p$ becomes a max operation, that is of takes the maximum one between the two criteria.

The weights and p can vary according to the application. For instance, if one desires to give more importance to the detection of classification property of the algorithm, $\alpha$ must be greater than $\beta$. On the contrary, if the correctly formed shape is more important than classification then $\beta$ must be greater than $\alpha$.

To assess the relevance of p, $\alpha$ and $\beta$ tests are conducted on the shapes in Figure 3.8. Table 3.3 shows the various results for the examples given in Figures 3.8 in terms of different assignments in $M_p$. The metrics used are given in the following equations:


Figure 3.8 Test images using in assessment of p, $\alpha$ and $\beta$ parameters. Gray region reflects the result of segmentation while black region represents wrongly classified pixels.

$$M_I = \frac{1}{2}(CE + TAF) \qquad p = 1, \alpha = \beta = \frac{1}{2} \tag{3.10}$$

$$M_{II} = 0.9\,CE + 0.1\,TAF \qquad p = 1, \alpha = 0.9, \beta = 0.1 \tag{3.11}$$

$$M_{III} = 0.1\,CE + 0.9\,TAF \qquad p = 1, \alpha = 0.1, \beta = 0.9 \tag{3.12}$$

$$M_{IV} = \sqrt{CE^2 + TAF^2} \qquad p = 2, \alpha = \beta = 1 \tag{3.13}$$

The first metric gives equal importance to classification error and shape formation. The second metric brings forth the localization while the third one penalizes more the shape-forming property. Notice that the third segmentation result (Image III) is the perfect one, and then it is ranked the highest result in all metrics. The fourth segmentation result has correct shape but it is badly localized. Then it is ranked the worst result in the $M_{II}$ but the second good result in the $M_{III}$. On the contrary, the second segmentation result is well localized but badly shaped. Then it is ranked the worst case in $M_{III}$ but has a moderate score in $M_{II}$.

Since it has been not deal with a specific segmentation problem in this thesis, in Chapter 4 and 5 the algorithmic performance of the two segmentation schemes will be measured using $M_{II}$ metric that gives the average value of two criterions.

Table 3.3 Results obtained with different $M_p$ for the examples.

| | CE | TAF | $M_I$ | $M_{II}$ | $M_{III}$ | $M_{IV}$ |
|---|---|---|---|---|---|---|
| Image I | 0.063 | 0.029 | 0.046 | 0.059 | 0.032 | 0.069 |
| Image II | 0.063 | 0.186 | 0.124 | 0.075 | 0.173 | 0.196 |
| Image III | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Image IV | 0.120 | 0.000 | 0.060 | 0.108 | 0.012 | 0.120 |

## 3.6 Generation of the Reference Test Images

For discrepancy-based segmentation evaluation techniques ground-truthed reference images are required. Therefore, artificially generated test images become imperative. **Zhang and Gerbrands (1994)** proposed the following test image design considerations:

1. Test images should be suitable for quantitative evaluation. Thus, only synthetic test images can exactly fulfill this requirement.

2. Synthetic test images should reflect common characteristics of real images. For example, some image corruption factors such as noise and blurring must be taken into account. Variation in the parameters of objects such as size and shape must be simulated as well as transitions between objects.

3. Sets of synthetic test images should be made with the grading change of every parameter.

In contrast to the simplistic test images used in **(Zhang and Gebrands, 1994)**, which consisted of a single ellipsoid against a uniform background, a different test image generation procedure based on occluding circular objects used by **Stuller**

**and Shah (1998)** is proposed in this thesis. The following considerations are taken into account for this goal:

1. The test image should contain variable number of objects.
2. The positions of the objects must be distributed randomly in the image.
3. Random occlusions must be allowed.
4. One must be able to control the distribution of object sizes.
5. There should be a stochastic process to "paint" the objects, that is texture or color rendering.

The proposed test image generation scheme, which is constituted of circular objects with random centers and radii, is illustrated in Figure 3.9. The steps of test image generation process are as follows:

**Shape Generation**: Circles are generated to fill all of the image area, so that the image plane is completely covered. The center and radius of each circle are determined by a random process that has a uniform distribution. The center takes value from 3 to 253 so those centers of circles reside in the image area whose size is 256-by-256. The value of radius also lies between 10 and 71 so circles do not exceed the image area and circles that have very small are is restricted. In this random covering process arbitrary occlusions occur resulting in a rich variety of shapes. Finally the small objects that have area under 200 pixels are eliminated by forcing them into one of the neighbouring objects.

**Object Coloring**: Objects are attributed a mean color called base color which is chosen randomly from a base color library. The base colors are obtained from some real world images so that they reflect true colors occurring in nature. For this aim, pixels of the real world images are clustered using the fuzzy c-means clustering algorithm. The center of each cluster is then adopted as base colors.

**Color Variety**: After the selection of the base color, object pixels are painted with Gaussian distributed colors so that the appearances of the objects become more natural. Thus while the base colors (mean color) is chosen from a palette realized from natural objects, color variety is obtained by selecting standard deviations of the Gaussian distribution randomly. In other word, the standard deviation of each color, components for each object is chosen from a uniform distribution between two limits $L_1$ and $L_2$ as given in Table 3.4. Standard deviation for main colors, called color spread, is forced into four categories. These categories are listed in Table 3.4. Each variance spread category indicates also the level of difficulty with which it is possible to classify and segment the image. In fact, the $\lambda$ index given below is indicative of the classification difficulty.

$$\lambda = \frac{\text{Sum of within-segment variance}}{\text{Sum of between-segment variance}} = \frac{\sum_{S} \sigma_S^2}{\sum_{S}\sum_{S'} (\mu_S - \mu_{S'})^2} \tag{3.14}$$

where $\mu$ and $\sigma$ are the centers of the base colors and standard deviations of color spreads, respectively. It is also noted that $\mu$ and $\sigma$ are vector form representing R, G and B component of the color.

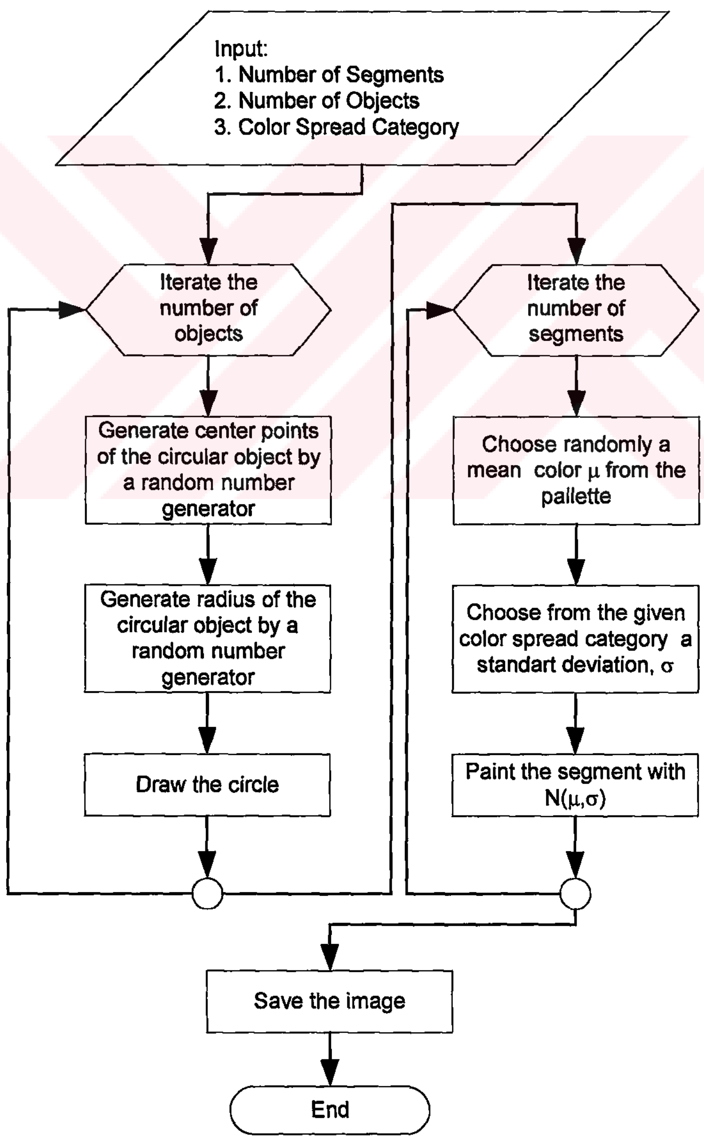



Figure 3.9 Procedure of the Test Image Generation

Figure 3.10 gives an example of a test image. Figure 3.10.a depicts a base test image with five segments painting with randomly chosen base colors while Figure 3.10.b, 3.10.c, 3.10.d and 3.10.e shows Gaussian variations on the mean-colored image. Each standard deviation in each category is randomly chosen and the difficulty level of the test image using Equation 3.14 is also given.

Table 3.4 Color spread categories used in the test images.

| Category | Standard deviation limits ($L_1$-$L_2$) |
|---|---|
| 1 | 5 – 10 |
| 2 | 10 – 20 |
| 3 | 20 – 30 |
| 4 | 30 – 40 |

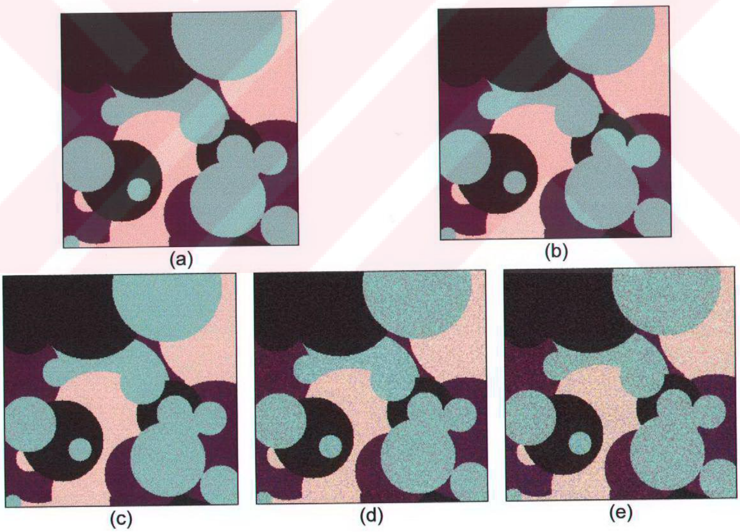
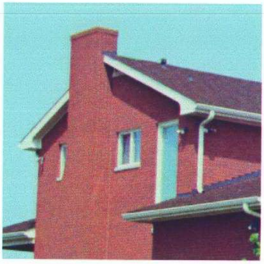


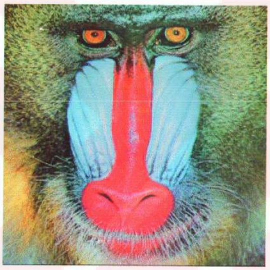
Figure 3.10 Generation of Test Image. (a) Image painting with four base colors. RGB values of the segments are (50-10-10), (95-25-65), (145-160-165), (230-195-175). (b) Image with standard deviations (7-6-7) and with difficulty level $\lambda$ = 0.0009. (c) Image with standard deviations (19-12-14) and with difficulty level $\lambda$ = 0.0050. (d) Image with standard deviations (27-26-26) and with difficulty level $\lambda$ = 0.0149. (e) Image with standard deviations (32-32-35) and with difficulty level $\lambda$ = 0.0239.

Besides artificial test images, the specific real-world images shown in Figure 3.11 utilized in the tests for visual comparison of the segmentation results.



House

Peppers



Mandrill

Mars

Figure 3.11 Real-world images.

## 3.7 Conclusion

In this chapter, the issue of performance evaluation of segmentation algorithms was addressed. A commonly used evaluation measure, namely CE is adopted while A new shape metric, namely turning-angle function, is also proposed. After considering their relative merits and weaknesses of both measures, a novel measure is proposed based on Minkowski averaging of them. A test image generation scheme is also introduced that provides sufficient variety and realism to test comparatively the segmentation algorithms.

# 4. MULTITHRESHOLDING OF COLOR IMAGES

## 4.1 Introduction

Multithresholding is an important and widely used tool in image segmentation because of its simplicity. Basic assumption of multithresholding is that the various objects give rise to peaks in the histogram corresponding to color clusters in the image. Then the segmentation task is reduced to finding hyperplanes dissecting the image histogram. Although multithresholding has been widely used to segment gray level images **(Sahoo et.al., 1988)**, there has been little amount of work on multithresholding of color images **(Çelenk and Haag, 1998)**. It is also expected that extending the gray-level multithresholding scheme to color images will provide in general better results due to richer information in color.

In this chapter, a scheme for color image multithresholding is proposed. A novel multithresholding technique based on two-dimensional (2D) histograms and fusion of their results is proposed . In this technique, pairs from the three color bands are selected and their 2D histograms are constituted. These histograms are partitioned according to their dominant peaks to constitute the threshold surfaces. Each pair is segmented in terms of its corresponding partitioned 2D histogram. The resulting three segmented maps are fused to obtain a final segmentation by taking into account the label concordance in each segmented map. This segmentation result is further filtered by a morphological filter so that small segments are eliminated **(Kurugöllü and Sankur, 1997) (Ünal, Kurugöllü and Sankur, 1999)**.

Effects of the histogram dimension on color multithresholding will be examined in Section 4.2. The proposed novel algorithm based on two-dimensional (2D) histogram will be detailed in Section 4.3. Results of the algorithm will be compared with its 1D and 3D counterparts in a quantitative manner in Section 4.4. Conclusions will be presented in Section 4.5.

## 4.2 Multithresholding of Color Images

In gray level images, multithresholding is relatively straightforward by choosing threshold values in the valleys between peaks in the image histogram, if these peaks are well delineated. Otherwise one must use image performance measure like boundary stable state to implement it. However, for color images, multithresholding is not a trivial task since the information comes from three separate bands, eg., red (R), green (G) and blue (B). A simple but logical extension of gray-level multithresholding to color multithresholding consists of multithresholding each color band separately, and then fuses together these three segmentation maps to obtain the final segmentation result. This approach is equivalent to partitioning the three-dimensional (3D) histogram into rectangular prisms as shown in Figure 4.1. It is evident that this type of partitioning can be improved by designing discriminating surfaces that adopted to the actual clusters in 3D-measurement space.

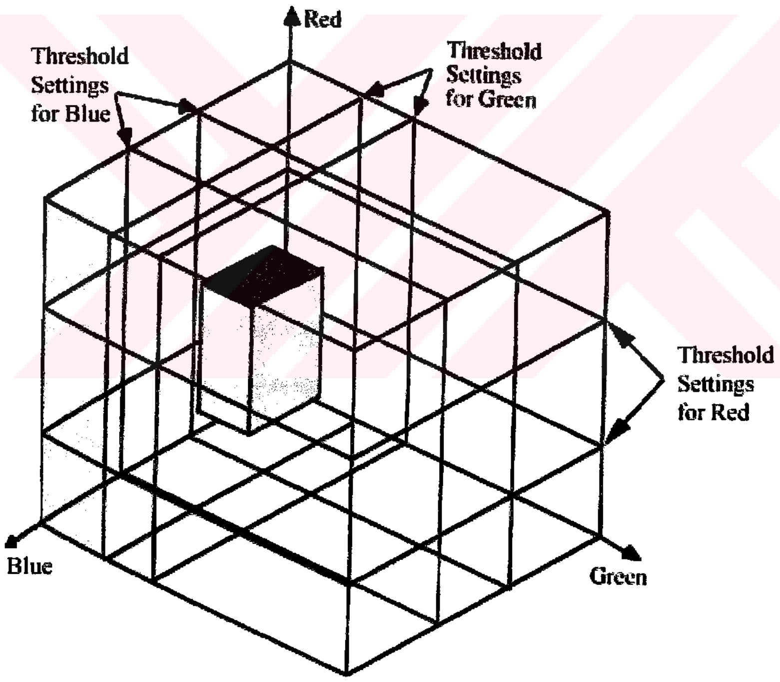
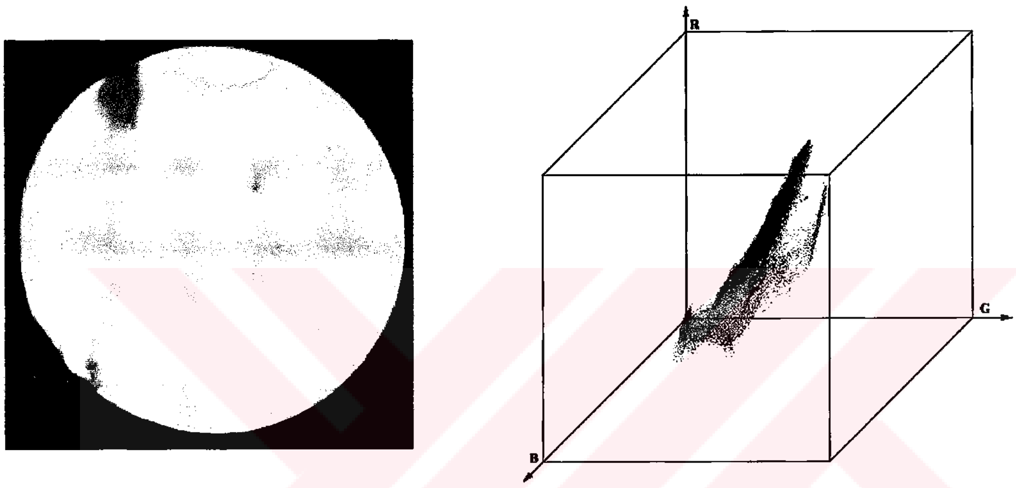



Figure 4.1 3D-histogram interpretation of multithresholding using 1D-histogram thresholds calculated on each color band. Each rectangular prism corresponds to a segment. (Reprinted from **Russ (1995)** ).

In the 3D-histogram, the "peaks" correspond to the bins with maximal accumulation of mass. The valleys become basins which lie between the "peaks". The threshold surfaces (e.g., 2D planes) should follow as much as possible the sparsely populated bins.

One problem is the sparseness of data. For example, for a color image with size of 256 by 256, and with 256 color tones in each band, there are only 65.536(256x256) three-dimensional color vectors while there exist 16.777.216 (256x256x256) three-dimensional (3D) histogram bins. Thus it is not feasible to attain a reliable estimate of the histogram. Thus "peak" picking algorithms on this 3D-histogram should deal with this sparsity problem. For example, the 3D-histogram of color "Mars" image is given in Figure 4.2. One can notice the sparseness of the histogram. Third problem is that the complexity of the problem grows rapidly up from O(n) to O($n^3$). Thus especially real-time applications are suffered from this high complexity.




(a) (b)

Figure 4.2 (a) "Mars" image (b) 3D-histogram of color "Mars" image.

On the other hand, the 2D-histogram has been sometimes used in thresholding of images with two-dimensional features. For instance, **Russ (1995)** introduced a 2D-histogram based method to segment the images that have regions separated by different spatial textures and/or by different mean brightness levels. By using the 2D-histogram of texture and brightness, the regions that cannot be discriminated using 1D-histograms of brightness and of texture are brought into light. Another approach of 2D thresholding, proposed by **Cheng and Chen (1999)**, uses fuzzy entropy approach for threshold selection in a 2D-histogram.

2D-histogram based approach presents a trade-off between three 1D-band histograms and a single 3D-color histogram. It delineates the histogram clusters, which correspond to the segments, better than each 1D-band histogram. On the other hand, they mitigate the data paucity problem in the 3D-histogram case. 2D-histograms can be considered as projection of 3D-histogram onto color pair planes as seen in Figure 4.3. The shape of clusters in 2D case are approximated to the original 3D better than 1D case since only rectangular prism shape can be delineated in 1D case. The 2D-histograms of the color "Mars" image are illustrated in Figure 4.4. Approximation of cluster shapes can be seen more clearly if they are compared with those Figure 4.2. A detailed examination of approximation of 3D histogram with 2D histograms can be found in Appendix A.

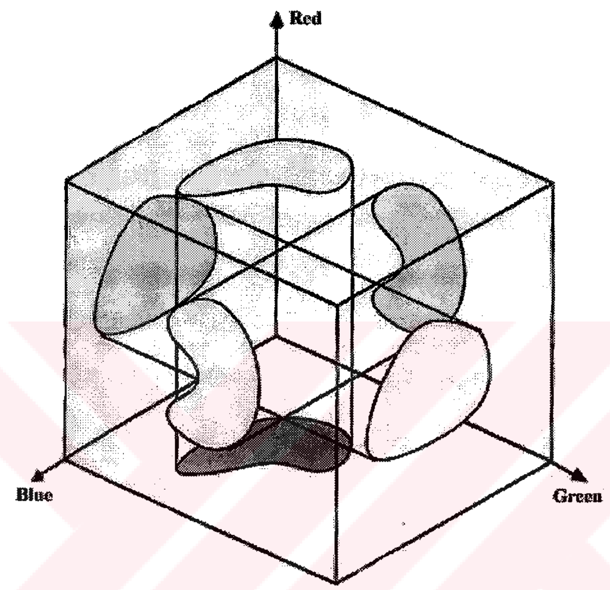
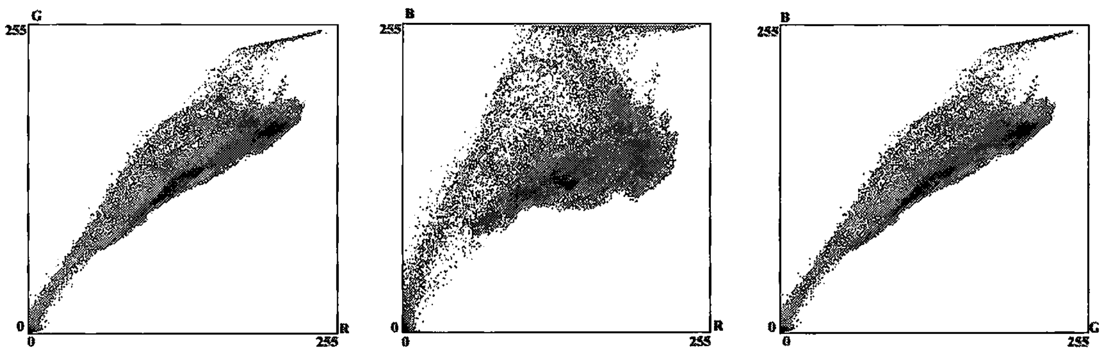



Figure 4.3 Projection of cluster shapes onto color pair planes ( Reprinted from **Russ (1995)** ).




(a) (b) (c)

Figure 4.4 2D-histograms of "Mars" image. (a) RG (b) RB (c) GB histograms.

## 4.3 Multithresholding of Color Images Using 2D-Histogram and Fusion

The flowchart of the proposed algorithm is depicted in Figure 4.5. The algorithm consists of two stages:

1. Multithresholding using 2D-histograms.
2. Fusion of segmentation maps.

These stages are detailed in the following sections.

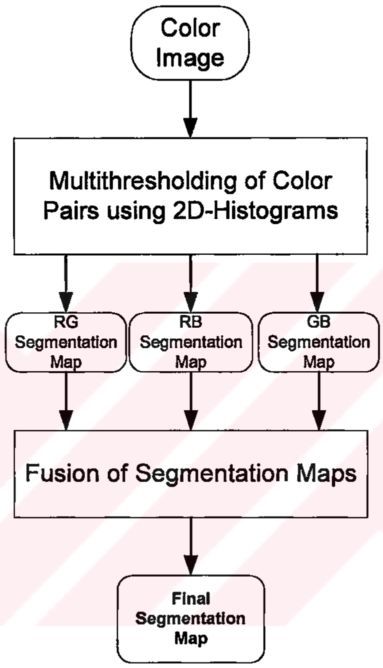



Figure 4.5 The flowchart of the color multithresholding algorithm.

### 4.3.1 Multithresholding of 2D-histograms

In this stage, "peaks" of the 2D-histograms of color band pairs are searched since the segments in the given image tend to cluster around a "peak" point in the 2D-histogram as seen in Figure 4.4. The "peak" points are considered as centers of clusters and can be used as representative of the segments. The other bins in the 2D-histogram are assigned to these "peaks". Thus a 2D-histograms is partitioned using its "peak" bins, hopefully each partition corresponding to a segment.

The steps of multithresholding stage are illustrated in Figure 4.6. These steps are detailed in the following subsections.

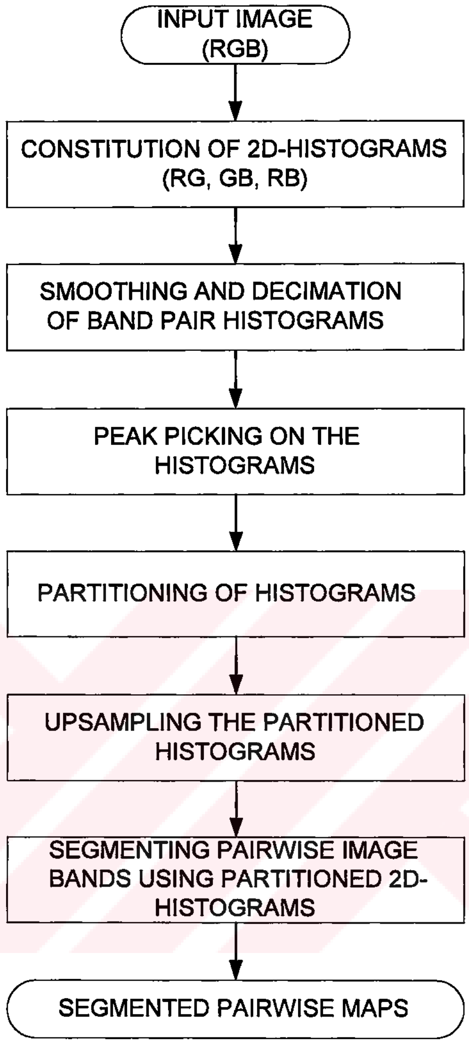



Figure 4.6 Steps of the first stage of the algorithm.

#### 4.3.1.1 Constitution of 2D-histograms

The first step is the constituting of the three 2D-histograms: RG, RB and GB. The histogram bins store the number of pixels having color pair values encountered in the corresponding color bands. Figure 4.7 shows the matching color bands scheme and Figure 4.8 depicts the RG histogram of the “Mars” image.

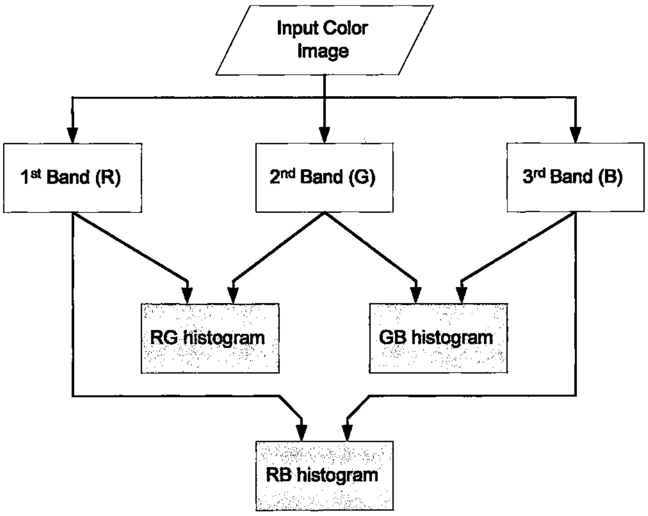



Figure 4.7 Constitution of 2D-histograms. The color bands are paired two-by-two and their 2D-histogram is calculated.

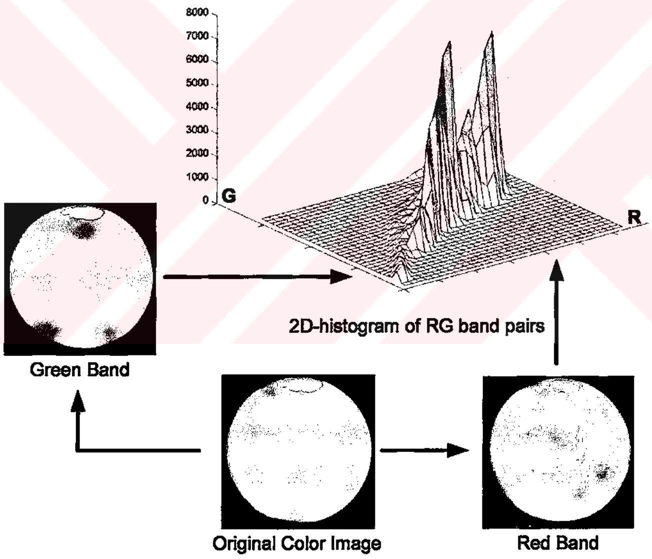



Figure 4.8 RG histogram of the "Mars" image. Each bin contains the number of color pairs encountered in the image pairs.

#### 4.3.1.2 Smoothing and Decimation of 2D-histograms

Since the 2D-histogram has 65536 (256x256) bins the surface of the histogram may be noisy due to sparseness problem stemming from insufficient statistics. This problem can be eliminated by changing the resolution of the histogram. Downsampling is used to reduce the resolution. Thus every reduced histogram bin contains proportionally more data. Decimation procedure is given in Figure 4.9. The histogram is subjected to a smoothing process and a downsampling iteratively to reduce its dimensions from 256x256 to 32x32. A 5-by-5 Gauss filter whose standard deviation is 0.625 is applied to histogram bins at each step in the smoothing process. The filter coefficients, that are determined experimentally, are given in Table 4.1. After filtering the histogram, it is downsampled by summing up all the bins in each 2-by-2 neighbourhood to get one bin value for each block as seen in Figure 4.10. Since this procedure is applied three times successively, the size of the histogram is reduced from 256x256 to 32x32.

Table 4.1 Coefficient of Gauss filter used in smoothing process.

| | | | | |
|---|---|---|---|---|
| 0.0000 | 0.0007 | 0.0024 | 0.0007 | 0.0000 |
| 0.0007 | 0.0314 | 0.1131 | 0.0314 | 0.0007 |
| 0.0024 | 0.1131 | 0.4067 | 0.1131 | 0.0024 |
| 0.0007 | 0.0314 | 0.1131 | 0.0314 | 0.0007 |
| 0.0000 | 0.0007 | 0.0024 | 0.0007 | 0.0000 |

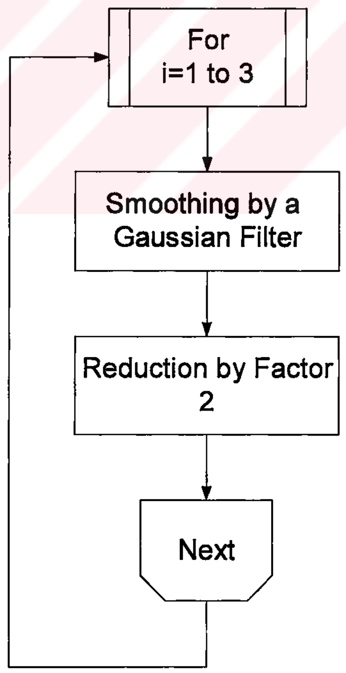



Figure 4.9 Downsampling of the 2D-histogram. The histogram is subjected to Gaussian filtering and downsampling iteratively three times to reduce its dimensions from 256x256 to 32x32.

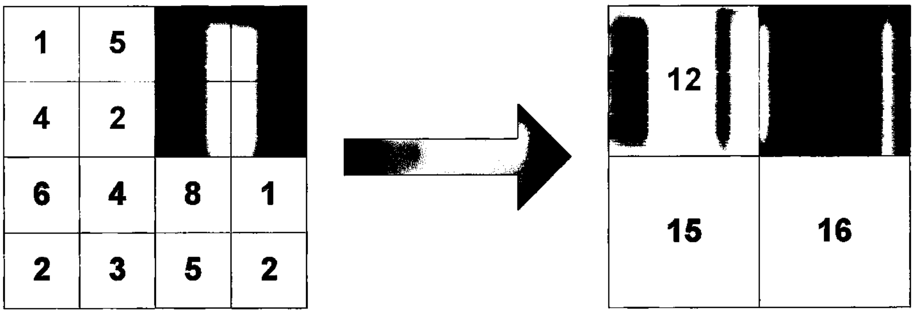



Figure 4.10 Reduction by factor 2. Each bin is formed by summing up 2x2 block of bins in the upper resolution.

#### 4.3.1.3 Finding Peaks on the 2D-histogram

The main assumption of the algorithm is that objects cause clusters in the feature space which in turn correspond to peaks of the histogram. Thus the determination of peak location in 2D-histogram is essential, analogous to peak finding in the 1D case. Thus the M distinct peaks must be found in the 2D-histogram, if M denotes the number of segments in the given image. This number can be estimated directly form the given image or should be given as a priori knowledge. Alternatively one can obtain a range of different results with different number of segments and choose the best result in terms of an evaluation criterion **(Kurugöllü and Sankur, 1997)**.

Two methods have been proposed for peak picking. The first method is an artificial neural network (ANN) based one proposed by **Vinod et.al (1992)**. Actually, Vinod's ANN has been designed to detect the peak locations in the Hough space. This ANN can be used in the same manner in a 2D-histogram. A neighborhood parameter, which is adjusted by user, influences the number of picked peaks. Smaller value causes more peaks, while a larger parameter resulting less peaks. Obviously the large neighborhood search the more the smoothing which reduces peak candidates by merging the ones that are close to each other. As a consequence, Vinod's ANN determines the peak location by adjusting its neighborhood parameter **(Kurugollu and Sankur, 1997)**. Details of the algorithm can be found in **Vinod et.al.(1992)**.

The second peak picking method is based on the algorithm proposed by **Koonty, Narendra and Fukunga (1976)**. This method has been adopted in this work since it is more reliable than Vinod's method. The flowchart of this algorithm is illustrated in Figure 4.11. This algorithm successfully used by **Khotanzad and Bourfa (1990)** in a clustering method. In this algorithm, each histogram bin is connected to a parent bin that has the biggest value in its neighborhood. Then the all histogram bins are connected in terms of a child-parent relationship. The bins that are not connected to a parent but have children are declared as the local peaks. An example of this chaining process is shown in Figure 4.12. If the number of determined peaks is equal or less than the desired number of segments, M, the algorithm is terminated. Otherwise the local peaks are eliminated iteratively until the number of peaks reaches to M.

Each peak is considered the center of a region formed by all bins that are its children. Then each peak is represented by a value which is the sum of all its children bins, as illustrated in Figure 4.12. The children bins are set to zero. Thus the 2D-histogram is converted to one that has only peaks whose value is proportional to its underlying mass as illustrated in Figure 4.13. Thus peaks that

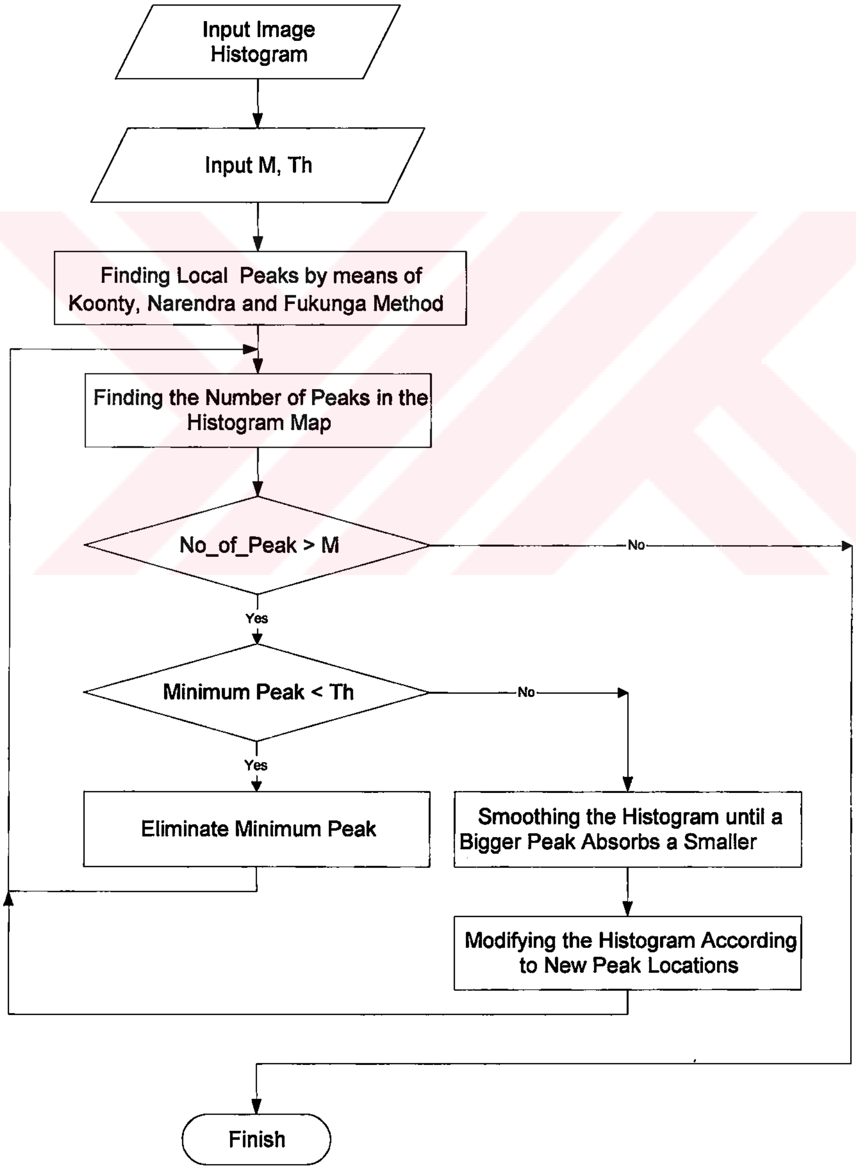
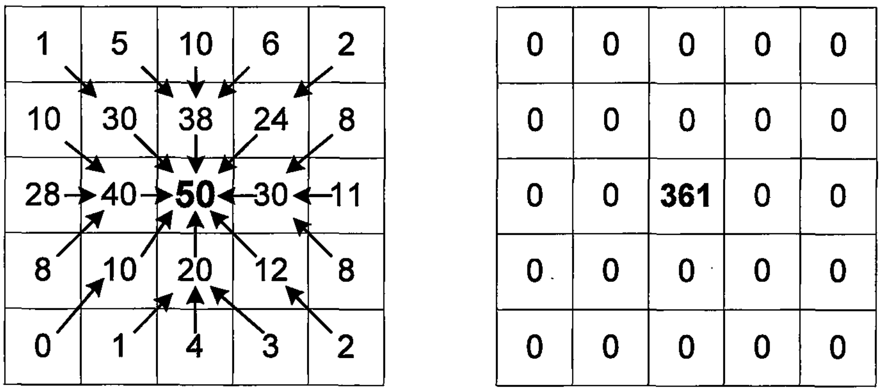



Figure 4.11 Flowchart of the peak picking based on the modification of the . **Koonty, Narendra and Fukunga (1976)** method.

have the same mass, but different supports and shapes will be represented by the same representative value.




(a) (b)

Figure 4.12 Finding peaks by **Koonty, Narendra and Fukunga (1976)** method. (a) Each bin is connected to a neighbor bin which has maximum value in its neighborhood. (b) The bins that are not connected to any neighbor are declared as peaks. Each peak is represented by a peak value, which is the sum of all its child bins. All child bins are then set to zero.

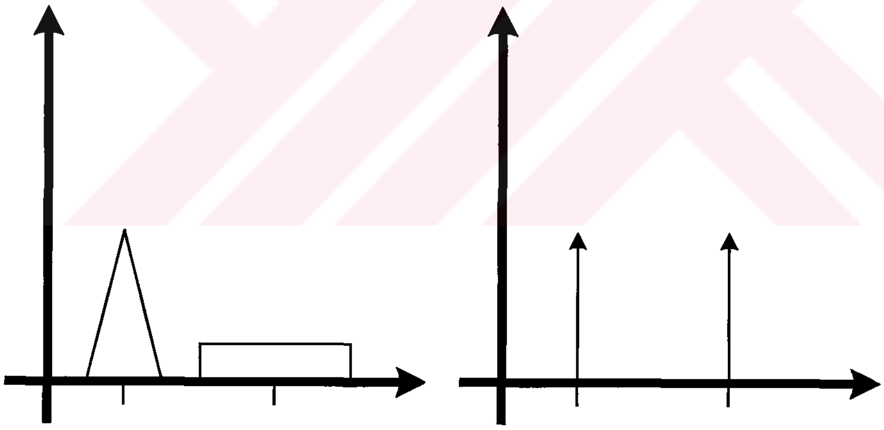


Figure 4.13 Illustration of peak representation in 1D case. Two region that have same area are represented by same value.

In the elimination process, the peaks are examined according to their representative values. The peak that has minimum value is considered: If this value is under a threshold, typically 1% of total image size, this peak is eliminated. All minimum valued peaks are eliminated until the number of peaks reaches to the given number of segments. If there are no peaks whose representative values is under the threshold and the number of peaks did not yet reach the given value, the 2D-histogram is smoothed by using a Gaussian filter that is given in Table 4.1 until a peak obliterates one or more. The obliterated peak is then eliminated. An essential point here is that every time the peak number decreases, i.e., a peak obliterates another peak, the initial 2D-histogram is restored excluding the eliminated peak. This removes the effects of the previous smoothing operations while continuing the peak elimination with the remaining peaks. Another reason for restoration is that it eliminates the small shifts in the peak locations. Consequently, the most dominant peaks are found in their exact locations and the number of peaks is equal to the given value.

#### 4.3.1.4 Partitioning the 2D-histograms

After having found that peaks, the 2D-histograms are partitioned into regions. Each peak is considered as the center of a region. Each 2D-histogram bin is attributed to a specific peak by taking into account the location and strength of the peaks. The attraction of a bin to a peak can be modelled as the "gravity force". In other words, each non-zero histogram bin is attributed to the peak that exerts the maximum gravitational pull on that bin. The gravity exerted on a bin by the $i^{th}$ peak is given as:

$$F_i = \frac{M_i}{d_i^2} \qquad i = 1,2,....,N \tag{4.1}$$

where $F_i$ : Attraction force of the $i^{th}$ peak

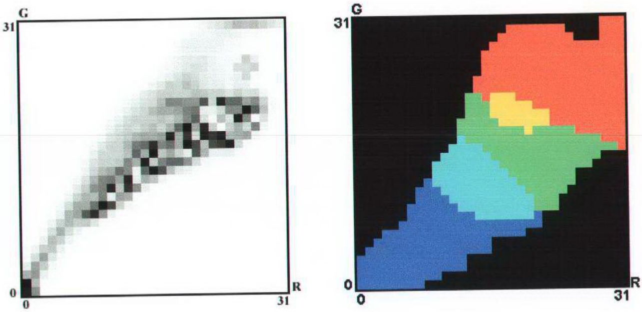
$M_i$ : Strength (Representative value) of the $i^{th}$ peak

$d_i$ : Euclidean distance from $i^{th}$ peak to the histogram bin

N : The number of peaks

Thus, a histogram bin belongs to the $j^{th}$ peak if $j = \text{Argmax}\{F_1, F_2, ...., F_N\}$.

The downsampled and partitioned 2D-histogram of RG bands of "Mars" image is given as an example of 4.14. Each gray level represents a histogram region while the black regions correspond to zero-valued histogram bins.




(a) (b)

Figure 4.14 An example of partitioning process. (a) Decimated 2D-histogram of RG bands of Mars image. (b) Correponding partitioned histogram. Each gray level except for black represents a cluster. Black regions correspond to zero-valued bins.

### 4.3.1.5 Interpolation of the Partitioned 2D-histograms

The partitioned 2D-histograms are upsampled back to their original sizes, 256x256, by replication each bin value in the 8x8 block. This process is depicted in Figure 4.15.

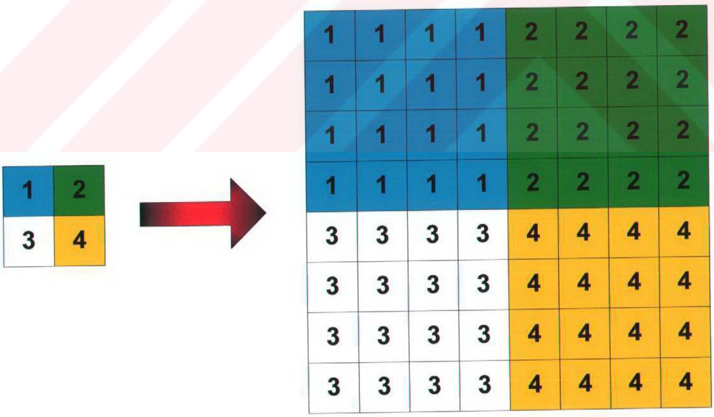



Figure 4.15 Upsampling process. Each histogram bin is simply repeated to 8x8 block.

#### 4.3.1.6 Segmentation of Pairwise Image Bands

The three pairwise image bands are segmented by using their partitioned 2D-histograms. Each region in the partition of the histogram corresponds to a segment in the corresponding in two-band image. Thus the segmentation maps of pairwise image bands are obtained by using the regions in the partitioned 2D-histograms. This process is illustrated in Figure 4.16. Consequently, three segmentation maps are obtained. Segmentation results of "Mars" image are given in Figure 4.17.

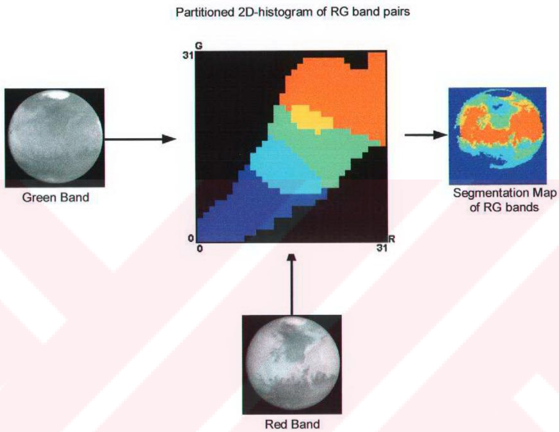



Figure 4.16 Segmentation process of pairwise image bands.

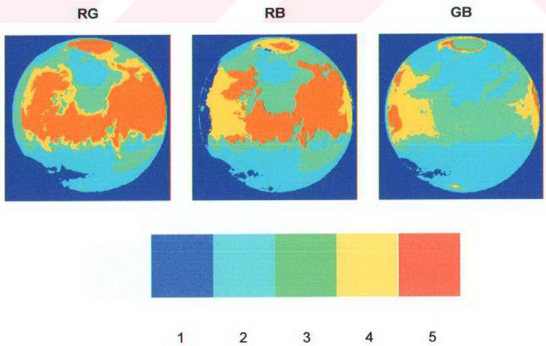



Figure 4.17 Segmentation results of RG, RB and GB pairwise bands of Mars image. Each color represents a label.

### 4.3.2 Fusion of Segmentation Maps

The second stage of the algorithm realizes the fusion of three segmented maps obtained from the two-band multithresholding stage. This stage consists of three steps: 1) Label concordance, 2) spatial-chromatic majority filtering and 3) postprocessing as morfological filtering. These steps of this stage is illustrated in Figure 4.18 and they are detailed in the following sections.

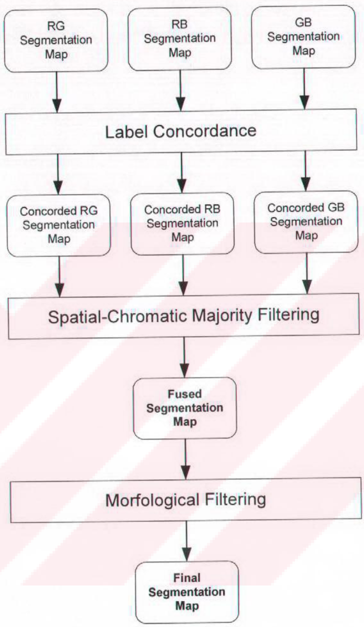



Figure 4.18 Steps of Fusion Stage.

#### 4.3.2.1 Label Concordance

In the multithresholding stage, every image band pair was separately segmented by using the corresponding partitioned 2D-histogram. However there is no guarantee that the same cluster labels will be assigned to pixels with the same addresses in different maps. This problem can be clearly seen in Figure 4.16 where pixels belonging to the same segments are labelled with different labels. Without a concordance of cluster labels from the pairwise bands, no fusion algorithm can be implemented. Thus, a label concordance operation is required before fusion of segmentation maps.

Before explaining the concordance algorithm, let's that examine the clearest cases of color transitions giving rise to different segments and how these changes are reflected in the histograms. Three different situations can occur: All three color bands change, two color bands change and only one color band changes. An image that has all three situations is depicted in Figure 4.19. The first region has the color-band values of [R G B] = [0 0 0] that corresponds black color. The second region whose color is blue ( [R G B] = [0 0 255] ) is the example of one color-band change situation since transition from black to blue occurs only in one color band. The third region is the example of two color-bands change situation. Its color is light-blue and color-band values is [R G B] = [0 255 255]. This region is also an example for the formation of a new region with changes in two color-band values. The last situation is three color-band change which is given by the fourth white color region ( [R G B] = [255 255 255] ) .

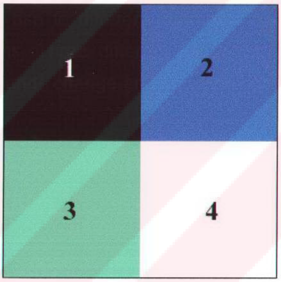



Figure 4.19 An example image that illustrates the three color transition types.

The reflecting of these three transition types to histograms is as follows:

1. *Three color-bands change*: Such a change will reflect in all three 2D-histograms, because all of the color bands change. This case does not cause any ambiguity.

2. *Two color-bands change*: Such a change will also reflect in all three 2D-histograms, because at least one of these two changed bands will be a variable in a 2D-histogram. This case also does not cause any ambiguity.

3. One color-band change: Such a change will reflect in only two of the 2D-histograms, because this changing band will be a variable in only two of

the three 2D-histograms. Since the region will not be detected in one of the 2D-histograms, this situation causes ambiguity.

From these observations, one can conclude that if a region is formed with three and two color-band changes, some corresponding region will exist in all three segmentation maps. It remains to obtain concordance of their labels. On the other hand, if a region is formed with only one color-band change, this region will have a counterpart in only two maps. However, this region is also recovered by the majority filtering since the region take two labels, out of three, from two bands. But this conclusion is valid under the assumption that the region takes same label value. This assumption does not always true.

To overcome this problem, the label concordance algorithm developed in this thesis is shown in the following Figure 4.20.

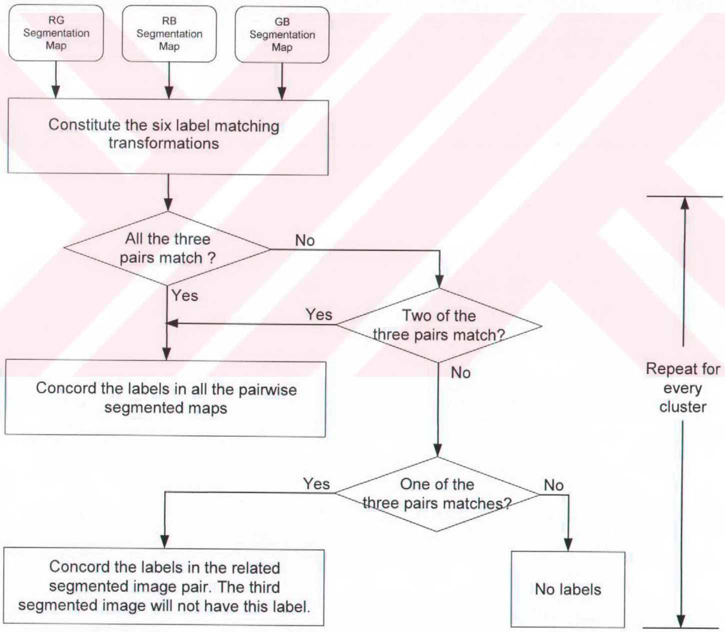



Figure 4.20 Flowchart of Label Concordance Algorithm.

The first step of the algorithm is to constitute the six label matching transformations. Each of these transformations maps the labels of one segmentation map to related labels of another map. The transformation of a label, say x, from a segmented map to another equals to the label of the target map if the overlap proportion is the largest with that of the source map, say y. That label transformation is defined as follows:

$$T_{ik}(x) = y \tag{4.2}$$

where x: The source label in the segmented map i.

$T_{ik}$: Label transformations from segmented map i to segmented map k.

y: The target label in the segmented map k, which overlaps most with x.

Equation 4.2 means that the corresponding region of x in segmented map i is the region y in the segmented map k. Thus, a total of six transformations are formed: $T_{12}$, $T_{21}$, $T_{13}$, $T_{31}$, $T_{23}$ and $T_{32}$ where the subscripts 1, 2 and 3 denote RG, RB and GB segmented maps, respectively.

Using these definitions of transformations, the bilateral matching cases are checked to find out regions to be identically labelled. A match is defined as in Equation 4.3.

$$T_{ki}(T_{ik}(x)) = x \tag{4.3}$$

where x: A label in segmented image i.

$T_{ik}$: Transformation from segmentation map i to segmentation map k.

$T_{ki}$: Transformation from segmentation map k to segmentation map i.

Equation 4.3 means that the transformation of label x in segmentation map i is mapped onto segmentation map k. Then the label $y = T_{ik}(x)$ in segmentation map k is mapped back onto segmentation map i. If the result is label x again in the segmentation map i, this is a "match" between labels x in segmentation map i and y in segmentation map k.

The following matching scenarios can be encountered:

1. RG-RB and RG-RB and RB-GB all match

2. RG-RB and RG-GB match, RB-GB does not match
   RG-RB and RB-GB match, RG-GB does not match
   RG-GB and RB-GB match, RG-RB does not match

3. RG-RB matches only (change in only R-band forms a region)
   RG-GB matches only (change in only G-band forms a region)
   RB-GB matches only (change in only B-band forms a region)

In the first scenario, the region exists in all of the three segmented maps. There is no conflict in this scenario. However, in the second scenario, which is rarely occurring, one of the three matches is missing. Its occurrence must be due to partial deformations and small variations, which in turn result in errors between the matching regions. In this case, although one match case is missing, one can deduce that the segment pair for which the match does not hold must be matching in reality and can align all with the labels of the other two pairs. The third scenario arises when only one color-band variation is responsible for a segment. In this case, only the related segmented maps' labels are concorded and the third segment is not considered. Forcing such a change would cause a pseudo-region. Such pseudo-regions in the segmented maps are not labelled and will not be considered while performing the fusion process.

Figure 4.21 depicts the effects of label concordance algorithm on the segmented maps of "Mars" image and the color label scale. By noticing Figure 4.16, Regions 4 in RB and GB maps are the same regions, but there is no corresponding region in RG map since the varying color-component is the B component. This is an example of the third scenario. One can notice that the region 4 in RG segmented map and the region 5 in the GB segmented map have no corresponding regions in the other segmented maps. Therefore, they are not real regions, they are pseudo-regions. These regions do not take any label.

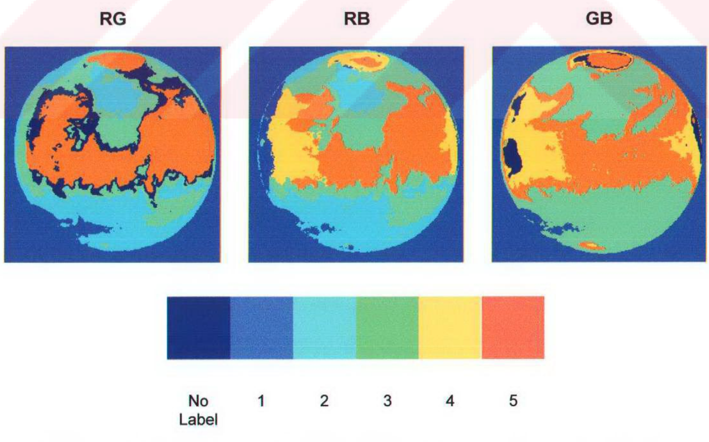



Figure 4.21 Segmentation Maps After Label Concordance Operation.

#### 4.3.2.2 Spatial-Chromatic Majority Filtering

After label concordance step, the final segmentation is obtained by fusing the concorded segmentation maps of all three band pairs. For this aim, a majority filtering operation is applied on both spatial and chromatic dimensions. The filtering operation is illustrated in Figure 4.22. A 5x5 neighborhood of a pixel in all three band pairs is considered. Therefore the label of the pixel becomes the most frequently encountered label value in the 3x5x5 neighborhood. In the case of close votes, the spatial size of the neighborhood is increased to achieve a clear majority voting. This fusion scheme incorporates spatial information into the segmentation process since the label of any pixel is determined by taking into account the labels of total 75 neighbor pixels. The result of majority filtering operation for "Mars" image is given in Figure 4.23.

The filter size has been chosen experimentally by using test images and evaluation criteria explained in Chapter 3. The filter size has been changed from 3x3 to 5x5 and has been applied to test images. The average $M_p$ values of the results are depicted in Figure 4.24. As seen in this figure, the best scores for all three methods are obtained in 5x5 filter size.

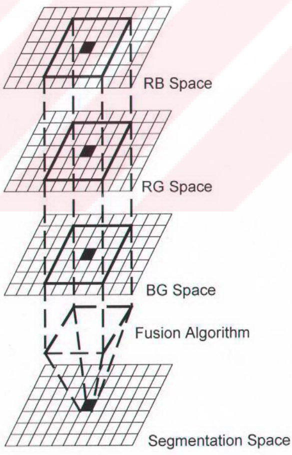



Figure 4.22 Fusion Scheme: The results of the all three bands are combined by using a 3x5x5 neighborhood.

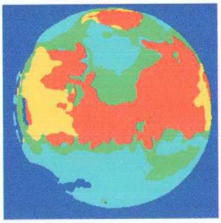


Figure 4.23 The fusion operation result of "Mars" image.

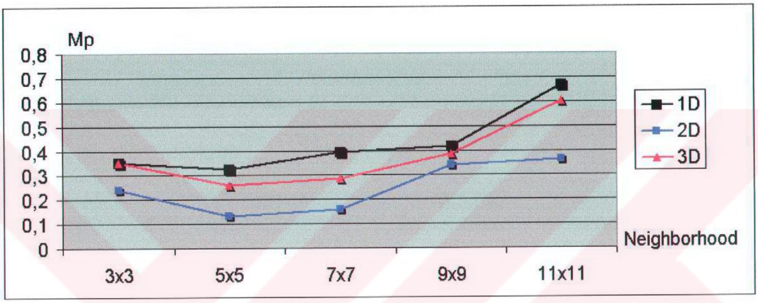
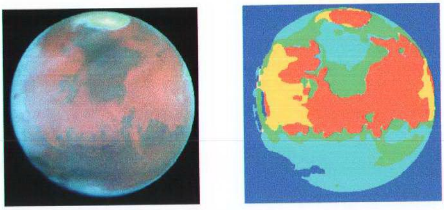



Figure 4.24 Average Mp results with respect to different filter sizes.

#### 4.3.2.3 Morphological Filtering

Despite the fact that a spatial-chromatic majority filtering is used in the fusion step, the result may still be noisy. Furthermore, there may be some meaningless regions with small areas. These problems are removed by applying a morphological closing operation to the final segmentation result. Since the final segmentation result has several regions labelled with different label, closing operation is carried out on each label field. This operation also smooths the result. Thus the final segmentation result is obtained. Figure 4.25 shows the final result of "Mars" image.



(a) (b)

Figure 4.25 The final segmentation result of "Mars" image after morphological closing operation.

## 4.4 Results

In this section, the results of the proposed algorithm will be given by comparing it vis-à-vis the 1D-histogram and 3D-histogram counterparts of the algorithm. In the 1D-histogram case, each color band can be segmented by using its "gray" value histogram. Then the three segmentation maps can be fused using the fusion algorithm detailed in Section 4.3.2. This scheme is depicted in Figure 4.26. In 3D-histogram case, highest density histogram bins, which are counterparts of peaks in 2D case, are found by a 3D version of Koonty, Narendra and Fukunga method. Then threshold blobs are constituted by using "gravity force" of the highest density bins as same as its 2D counterparts. The segments are directly constructed by using these multithreshold blobs without any label concordance and fusion operation.

### 4.4.1 Computational Complexity

Computational complexity of 1D-histogram, 2D-histogram and 3D-histogram cases are O(n), O($n^2$) and O($n^3$), respectively. All three cases are programmed by using Microsoft Visual C++ 4.2 compiler under Windows NT 4.0 operating system. The number of CPU clocks for all three cases, with an image sizes of 256x256, have been determined by means of clock() routine. The number of clocks for 1D, 2D and 3D cases by using a Intel Pentium II – 266 Mhz computer are found 430, 730 and 130000 respectively. Figure 4.27 illustrates the number of clocks in the logarithmic scale. While complexity of 3D-histogram case is about one hundred and eighty times of that of 2D case, complexity of 2D-histogram based method is about two times of that of 1D case. The detailed analysis of computational complexity of the methods can be found in Appendix B.

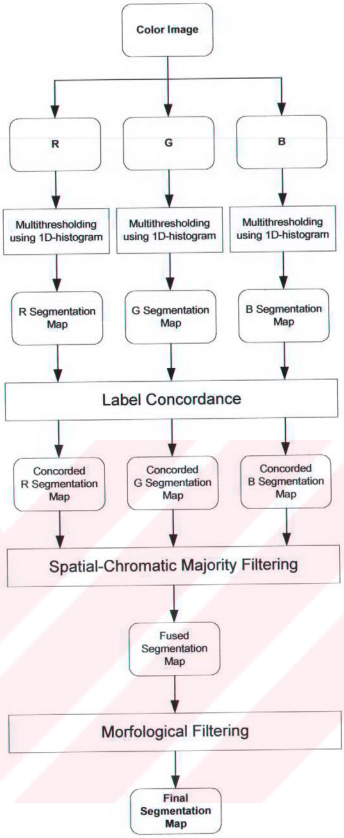



Figure 4.26 Flowchart of the multithresholding of color images using 1D-histograms. This scheme is the 1D counterpart of the proposed 2D-histogram based color image multithresholding.

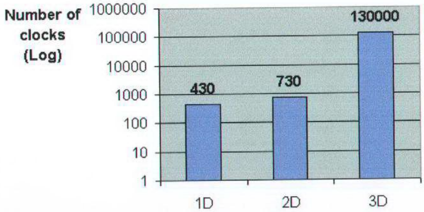



Figure 4.27 Comparison of computational complexities of the three methods.

### 4.4.2 Results of Real World Images

The results of real world images given in Chapter 3 are depicted in Figure 4.28. The numbers of segments are determined by a cluster validity process that will be detailed in Chapter 5.

It is observed that 1D-histogram based method gives the worst results for all images. For instance, sky and house segments in "House" images have been merged since there exist one color band transition between these regions. This problem has been eliminated in the 2D-histogram based method. The results of the 3D-histogram based method is more detailed than that of the 2D case. For example, roof detail on the right hand side of "House" image has been merged to sky segment in the 2D case while it is preserved in the 3D case since their clusters (sky and right roof) are better discriminated in the 3D case.

In general the following conclusions can be given:

1. In the 1D case, some segments can be merged due to single color band transition.

2. In the 2D case, the problem of merged segments is eliminated. Some segments that do not well defined peaks in the 2D histograms cannot be detected.

3. The 3D case gives the best result since the multithresholding is carried out inherently. But it requires much larger computation power.

### 4.4.3 Quantitative Results

The performance of the proposed 2D-histogram based method has been compared with those of the 1D and 3D-histogram based methods. The evaluation criterion, $M_p$, proposed in Chapter 3 has been used for this aim. The algorithms have been run on 200 test images and the corresponding $M_p$ values have been calculated.

The performance of proposed 2D-histogram based multithresholding method has been found to be better than that of the 1D case for 187 test images out of 200 images. Thus the performance ratio of 2D case is 93%. The detailed test scores are given in Table 4.2 and Table 4.3 Table 4.2 shows the performance comparison as a function of the number of segments while Table 4.3 gives the scores as a function of the color spread. In each table, the second and third columns indicate the number of cases when the $M_p$ score was found superior with respect to the other method. The fourth column shows the percentage of cases when 2D case scored higher that 1D case. One can notice that the performance of 2D case increases when the number of segments increases while it slightly declines when the color spread increases.

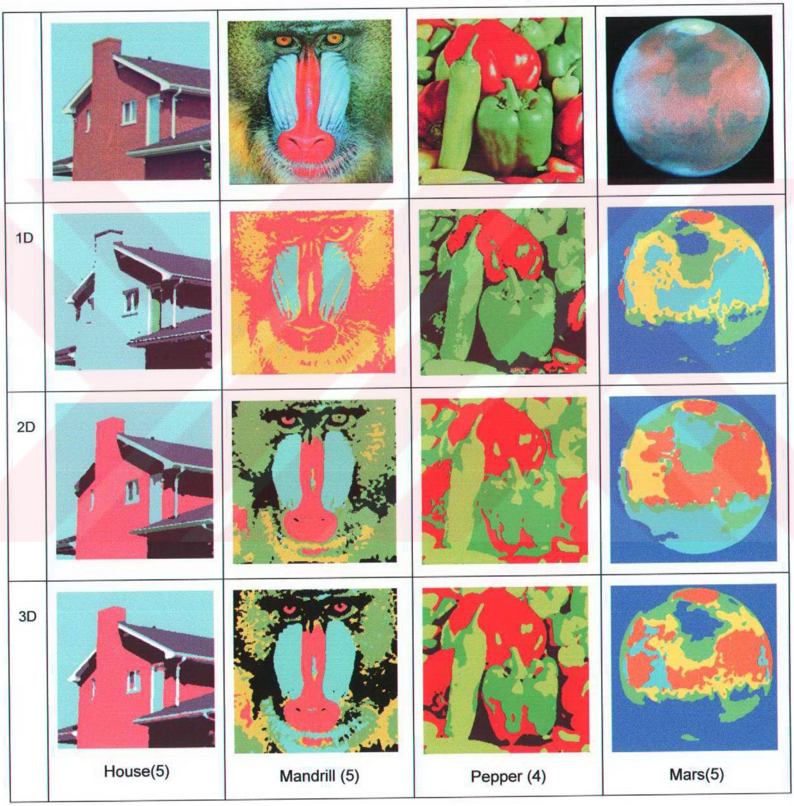



Figure 4.28 Results of real world images.

Table 4.2. Comparison of performance of 1D and 2D cases according to number of segments.

| Number of Segments | 1D case that better than 2D case | 2D case that better than 1D case | Percentage 2D case is better than 1D case |
|---|---|---|---|
| 2 | 4 | 36 | 90% |
| 3 | 3 | 37 | 92% |
| 4 | 3 | 37 | 92% |
| 5 | 2 | 38 | 95% |
| 6 | 1 | 39 | 97% |
| **Total** | **13** | **187** | **93%** |

Table4.3. Comparison of performance of 1D and 2D cases according the color spread.

| Category | 1D case that better than 2D case | 2D case that better than 1D case | Percentage 2D case is better than 1D case |
|---|---|---|---|
| 1 | 1 | 49 | 98% |
| 2 | 2 | 48 | 96% |
| 3 | 3 | 47 | 94% |
| 4 | 7 | 43 | 86% |
| **Total** | **13** | **187** | **93%** |

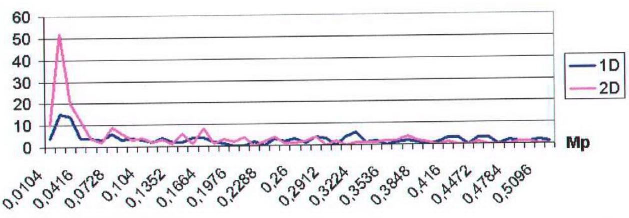



Figure 4.29 Histograms of $M_p$ for 1D and 2D case. Mean value of 2D case is lower than that of 1D case.

Figure 4.29 depicts the histogram of $M_p$ for both methods. It is noticed that the mean value of $M_p$ for 2D case is lower than that of 1D case.

When a similar comparison is done between the 2D-histogram and the 3D-histogram based multithresholding methods, it has been found out that the performance of the 3D case is only slightly better than the 2D case for 111 images out of 200 images. Thus the performance ratio of the 3D case is 55%. This result is to be expected since using 3D histogram for multithresholding is more inherently than the other methods. Table 4.4 and Table 4.5 are counterparts of Table 4.2 and Table 4.3.

Table 4.4. Comparison of performance of 2D and 3D cases according to number of segments.

| Number of Segments | 2D case that better than 3D case | 3D case that better than 2D case | Percentage 3D case is better than 2D case |
|---|---|---|---|
| 2 | 10 | 30 | 75% |
| 3 | 18 | 22 | 55% |
| 4 | 19 | 21 | 52% |
| 5 | 18 | 22 | 55% |
| 6 | 24 | 16 | 40% |
| **Total** | **89** | **111** | **55%** |

Table4.5. Comparison of performance of 2D and 3D cases according the color spread.

| Category | 2D case that better than 3D case | 3D case that better than 2D case | Percentage 3D case is better than 2D case |
|---|---|---|---|
| 1 | 1 | 49 | 98% |
| 2 | 15 | 35 | 70% |
| 3 | 30 | 20 | 40% |
| 4 | 43 | 7 | 24% |
| **Total** | **89** | **111** | **55%** |

It is interesting that the performance of the 3D case dramatically declines with increasing color spread. Its performance is definitely worse than 2D case in the fourth color spread category while its performance is superior in the first color spread category. This phenomenon can be seen in Figure 4.30, which depicts the histogram of $M_p$ values of both 3D and 2D cases. The $M_p$ values of the 3D-histogram based method are gathered on the two extremes of the histogram.

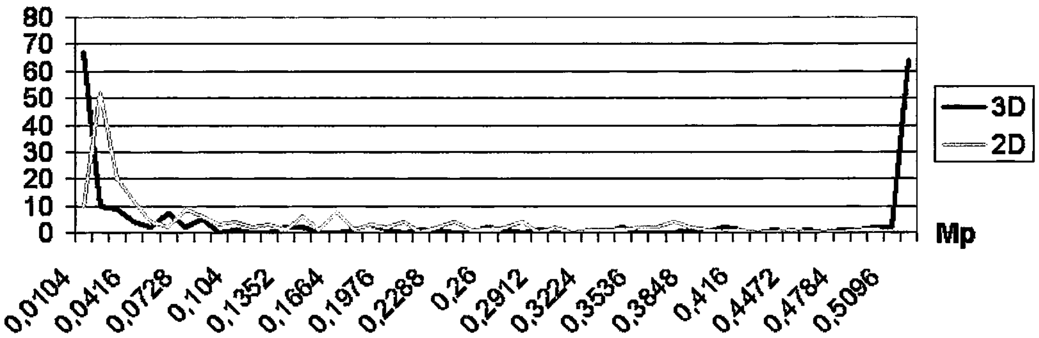



Figure 4.30 Histograms of $M_p$ for 3D and 2D case. Values of 3D case are gathered to two sides of the histogram.

## 4.5 Conclusion

In this chapter, multithresholding of color images has been investigated and a method that uses 2D-histograms and a fusion algorithm has been proposed. The proposed method has been compared with its 1D-histogram and 3D-histogram based counterparts. It is observed that the proposed method results more often in the correct segmentation map than the 1D-histogram based method. The computational complexity of the proposed method is only approximately two times of that of the 1D case.

It was expected that the 3D-histogram based method would give the best results since the histogram peaks or alternatively the corresponding clusters in the measurement space are inherently better delineated. The quantitative results have supported this expectation. On the other hand, the computational complexity of 3D case is about one hundred and eighty times of that of 2D case.

Consequently, the proposed method is computationally much more efficient than the 3D case but its performance is only slightly inferior.

The proposed method can be applied to any multispectral images such as LANDSAT images. In this case, the bands of multispectral images can be combined two-by-two and the segmentation can be carried out by using 2D-histograms of these combined images. Then the segmentation results can be fused to obtain the final segmentation map. For instance, LANDSAT images have seven bands. Thus combination of these bands gives 21 band pairs ( $C(7,2) = 21$ ). These 21 band pairs can be segmented by using proposed 2D-histogram based multithresholding method, and the resulting 21 segmentation maps can be fused to obtain final segmentation result.

# 5. CONSTRAINT SATISFACTION NEURAL NETWORK BASED SEGMENTATION

## 5.1 Introduction

This chapter presents various extensions of the Constraint Satisfaction Neural Network (CSNN) based image segmentation method. CSNN, which is proposed by **Lin et. al. (1992)**, is an attractive tool for image segmentation because of its simple network architecture and its constraint scheme used in segmentation process. Although it is proposed as a neural network implementation of the constraint satisfaction problem, it can be interpreted a probabilistic relaxation scheme as well. In this chapter, firstly the mathematical preliminaries of Constraint Satisfaction Problem (CSP) and the CSNN structure will be introduced. Various drawbacks of the algorithm as in **Lin et. al. (1992)** will be pointed out. Some modifications to overcome the shortcomings of the original algorithm will be presented. The first extension of CSNN is called the Boundary CSNN (B-CSNN), where boundary constraints obtained from a coarse scale Marr-Hildreth edge detector have been added. The second proposed method, called multi-scan CSNN (MS-CSNN), uses a multiresolution approach to enhance the performance of the CSNN. Multiresolution property in MS-CSNN is introduced by a multi-scan scheme. Pyramid type multiresolution technique has been also applied to CSNN. This yields the third method called pyramidal CSNN (P-CSNN). The last modification has been carried out by coalescing the Markov Random Field (MRF) model and the CSNN tool. CSNN actually carries out energy minimisation using only neighbourhood contributions. In this method, a data dependency term is added to the CSNN formulation to implement the MRF-based minimization.

## 5.2 Constraint Satisfaction Problem and Constraint Satisfaction Neural Network for Image Segmentation (CSNN): Preliminaries

### 5.2.1 Constraint Satisfaction Problem

Recently there has been considerable interest on Constraint Satisfaction Problem (CSP) in artificial intelligence and computer science applications. Many problems encountered in these disciplines such as scheduling, temporal reasoning, graph problems, planning can be modelled as special cases of CSP. Especially in image processing, many problems can be viewed as a CSP problem **(Kumar, 1992)**. Line labelling, for example, matching in two-dimensional shape recognition and stereo matching can be mapped to a CSP **(Haralick and Shapiro, 1993)**. A CSP is a problem comprised of a finite set of variables, each of which is associated with a finite domain, a finite set of labels, and a set of constraints which confine the labels the variables can simultaneously take. The goal is to find a solution which assigns a label to each variable satisfying all constraints. More formally, the CSP can be defined as follows:

A CSP is a 4-tuple problem: (V, D, C, F) where V denotes a finite set of variables $\{v_1, v_2, \ldots\ldots, v_n\}$ (or given data) and D is set of domains for variables $\{D_1, D_2, \ldots. D_n\}$ which consists of labels coming from a label set, L, $\{l_1, l_2, \ldots. l_m\}$. The variable $v_i$ can only take a label from its domain, $D_i$. F is a function which maps V into D by assigning a label, $l_k$, that belongs to the domain of a variable, $D_i$, to the corresponding variable, $v_i$. C is a finite set of constraints on an arbitrary subset of variables in V that restricts the values of those constrained variables can take simultaneously. The goal of CSP is to find a mapping between V and D that satisfies all constraints, C **(Kwan, 1997)**.

A typical example of a CSP is the N-Queen problem. A chessboard with size NxN and N queens are given in this problem. The desired solution is that the queens must be placed on the chessboard in such a way that no queen can capture any other one. The problem can be cast as CSP in such a manner that the queens are the variables, the locations in the chessboard are labels and the chess rules about movement of queen, that a queen can move on the same row, the same column or the same diagonal, are the constraints. A different interpretation of the problem can be given such that the rows of the chessboard are considered the variables while the columns of that are chosen as labels. If the movements rule of the queen were taken into account, there would be only and only one queen per row and per column. Thus one constraint is that pair of on same rows and on same column will restrict each other. This is expressed as follows: $C_1 = \{ (v_i,l_m), (v_j,l_n) \mid v_i,v_j \in V , l_m, l_n \in L, \text{and } i \neq j \Rightarrow m \neq n \}$. The other constraint comes from the diagonal condition and can be represented as: $C_2 = \{ (v_i,l_m), (v_j,l_n) \mid v_i,v_j \in V , l_m, l_n \in L , \text{and } m \neq n \Rightarrow | i - j | \neq | m - n | \}$ **(Haralick and Shapiro, 1993)**.

A CSP can be solved using the generate-and-test (G-T) method. In this method, all possible label assignments are constituted and tested systematically. The first configuration that satisfies all the constraints is adopted as the solution. The number of all possible configurations is the size of the Cartesian product of all variable domains. Thus the search space can become huge even though the size of domains is moderate. The number of possible solutions grows exponentially with the number of variables. This is, therefore, the disadvantage of the G-T method.

A more acceptable method used in a CSP is the backtracking (BT) method. BT is actually a tree search algorithm **(Haralick and Shapiro, 1993)**. In this method, the assignment of labels to variables is carried out sequentially by taking into account the constraints. If an assignment violates any of the constraints, the backtracking is performed. This is accomplished by re-labeling the most recently assigned variable if there exists still alternative labels in the variable domain available. Although performance of BT is better than that of G-T, its complexity is still exponential **(Kumar, 1992)**. To overcome this weakness of BT and G-T, several CSP algorithms such as BT with Forward Checking, Partial Lookahead, Really Full Lookahead, iterative repair were also proposed. Details of these algorithms can be found in **Kumar (1992)** and **Eugene and Wallance (1992)**.

On the other hand, there has been recent interest in using Artificial Neuron Network (ANN) for the solution of the CSP domain because of its massively parallel processing capability. For instance, energy minimization based ANNs such as Hopfield networks have been used to solve problems such as traveling salesman, N-queens or Tangram puzzles. The proposed ANN models to solve CSP are not only problem oriented as studied by **Privault and Hérault (1998)** but also generic such as GENET proposed by **Lee and Tam (1995)**.

Image segmentation can be also considered as a CSP by assigning segment labels into the image pixels under certain spatial constraints. In the next chapter, the theoretical foundations of the CSNN will be introduced.

### 5.2.2 Constraint Satisfaction Neural Network for Image Segmentation

**Lin et. al. (1992)** viewed the image segmentation problem as a CSP, then proposed a neural network architecture to solve the CSP. They used the region homogeneity property of the image segmentation that emphasizes that the pixels belonging to the same homogenous region should be mapped into the same segment. They constructed a neural network architecture, called CSNN, to perform image segmentation. CSNN performs its task by using the constraints between a pixel and the pixels that reside in its neighborhood as well as the network topology. Topology of the CSNN and the connection weights between the neighboring pixels reflect the nature of CSP. They used the relationship between the CSP and the relaxation process implicitly to accomplish the solution of CSP. The overall processing in CSNN as in **Lin et. al. (1992)** can be summarized as follows:

First of all, features extracted from the given image are clustered. In **Lin et. al. (1992)** the feature set consists only of the gray values of the pixels. A Kohonen's self-organizing map is used for this task. Then the results of the clustering are fuzzified to initialize the CSNN neurons that hold the probability of belonging to a specific segment of a pixel site. After initialization, CSNN performs a parallel and iterative process to accomplish the segmentation by taking into account the spatial context through constraints. In the following sections the mapping CSP to CSNN, dynamics of CSNN and constraint specifications will be briefly presented.

#### 5.2.2.1 Mapping CSP to CSNN

CSP is reinterpreted to construct the CSNN in **Lin et.al. (1992)**. This interpretation will be briefly presented in this chapter while more details can be found in **Lin et.al. (1992)** :

1. Variables are denoted by $O=\{O_i, i=1.2.,,n\}$ where n is the number of the variables (ie., feature value at a pixel location i).

2. Labels are represented by $\Lambda=\{\lambda_j, j=1,2,....,m\}$ where m is the number of the labels (ie., classes or segments).

3. A neighborhood constraint set of a variable, $O_i$, is denoted by $\Omega_i$. $N_i = \{O_{i1}, O_{i2},...,O_{ip}\}$ consists of the neighbor variables of $O_i$. There are some spatial constraints between a variable and its neighbors.

4. Binary constraints between two neighbor variables, $O_i$ and $O_{i1}$, over a set of labels $\{\lambda_j,\lambda_{j1}\}$ can be defined as:

$$\Lambda_{ii_1} : \Lambda_i \times \Lambda_{i_1} \rightarrow [\min, \max] \tag{5.1}$$

where min and max denote certain negative and positive real numbers, respectively. The constraint $\Lambda_{ii_1}(\lambda_j, \lambda_{j_1})$ can be interpreted as a compatibility measure of assigning label $\lambda_j$ to variable $O_i$ and label $\lambda_{j_1}$ to variable $O_{i_1}$. It can be formulated as the following expression:

$$\Lambda_{ii_1}(\lambda_j, \lambda_{j_1}) = \begin{cases} \max & \text{If label } \lambda_{j_1} \text{ on variable } O_{i_1} \\ & \text{supports label } \lambda_j \text{ on } O_i \\ \min & \text{If label } \lambda_{j_1} \text{ on variable } O_{i_1} \text{ and} \\ & \text{label } \lambda_j \text{ on } O_i \text{ are inconsistent} \\ 0 & \text{If } O_i \text{ and } O_{i_1} \text{ are independent} \end{cases} \tag{5.2}$$

where high and low denote a positive and negative real number, respectively.

CSNN topology is constructed to reflect this CSP interpretation (i.e., pixel locations). It consists of nxm neurons, where each neuron belongs to one variable and its specific label. Constraints are represented by connection weights between neurons. They are chosen in such a manner that their value becomes positive and high whenever neurons support each other, whereas they take negative values if the labels of the neurons are inconsistent. If two neurons are independent, the connection weight between them becomes 0. Thus these connections do not exist in the neural net.

#### 5.2.2.2 Image Segmentation with CSNN

In the image segmentation problem, the variables are the image pixels while the labels correspond to segment labels. The topology of the CSNN is depicted in Figure 5.1. CSNN consists of layers where each of them corresponds to a different segmentation map. The number of layers, M, is given as a priori information. Each layer has size NxN, which is the size of the given image. Neuron outputs in the layers hold the probability that the pixel belongs to the segment represented by corresponding layer. As shown in Figure 5.2, each neuron has connections to eight neighborhood neurons in the same layer as well as in all the other layers. Connection weights represent evidences or counter-evidences for segment decisions. They are determined in such a manner that a neuron excites those neurons that has the same label, and inhibits the ones that have different labels. In CSNN, first of all, the number of segments is determined by some a-priori knowledge and initial label assignments are given by fuzzificaiton. Thus the "probability" of each pixel to belong to different M segments is determined and given to the constraint satisfaction neural network as the initial conditions. These segment probabilities are updated iteratively by taking into account their neighbours and the spatial constraints. The network converges to a solution which also satisfies the given constraints. After converging, the label which holds the largest probability value is assigned to the corresponding pixel.

The notation used to explain the dynamics of the CSNN is as follows:

| | |
|---|---|
| NxN | :Image size |
| M | :Number of layers (number of segments) |
| $U_{ijk}$ | :The neuron in the $k^{th}$ layer at the (i,j) location. |
| $O_{ijk}$ | :The output of the $U_{ijk}$ |
| $N_{ij}$ | :Set of neighbors of (i,j) location |
| p | :Size of $N_{ij}$ |
| $H_{ijk}$ | :Total contribution of neighbors of $U_{ijk}$ at $k^{th}$ layers |
| $W_{ij,qr,k,l}$ | :The connection weigth (constraint) between $U_{ijk}$ and $U_{qrl}$ |

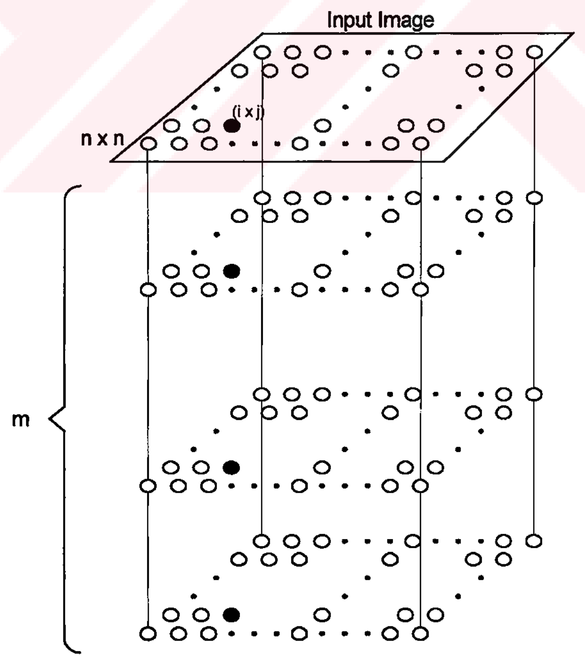



Figure 5.1 Constraint satisfaction neural network topology. Each layer represents the segments. $(i,j)^{th}$ neuron in each layer holds the probability that $(i,j)^{th}$ pixel belongs to the segment represented by the layer.

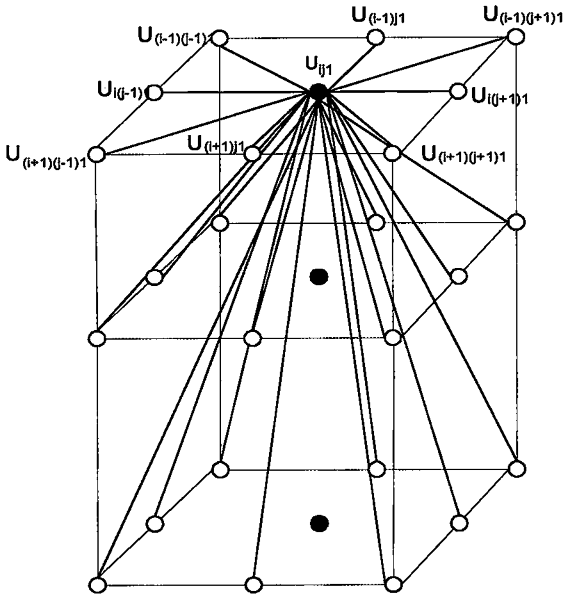



Figure 5.2 Connection between a neuron and its neighbors. The weights of these connections are interpreted as constraints.

Each neuron in the CSNN receives support from its neighborhood as well as the feedback from its output. The update scheme is depicted in Figure 5.3. The output of any neuron depends on its previous state (feedback signal) and the neighborhood contributions from each of the segment plane. Neighborhood contributions come from the outputs of the neighbor neurons through connection weights that represent the spatial constraints. The contribution of the neighbors has the following properties:

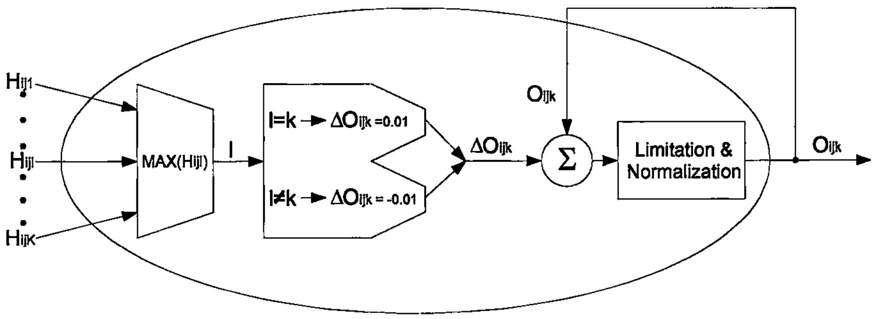



Figure 5.3 Structure of a neuron. Input signals come from the states of neighbours as well as feedback from neuron's past state. The current output of the neuron is determined by means of a nonlinear function of the neighbourhood contribution, H, and the feedback signal.

1. If both $O_{qrl}$ and $W_{ij,qr,k,l}$ ($qr \in N_{ij}$ $k,l = 1,..,m$) are positive, then $O_{qrl}W_{ij,qr,k,l}$ should contribute a positive amount to $U_{ijk}$

2. If $O_{qrl}$ is positive but $W_{ij,qr,k,l}$ are negative, then $O_{qrl}W_{ij,qr,k,l}$ should contribute a negative amount to $U_{ijk}$

3. If $W_{ij,qr,k,l}$ is zero then $O_{qrl}W_{ij,qr,k,l}$ should not contribute to $U_{ijk}$

These properties are in agreement with CSP formulation given in Equation 5.2. It is noted that the outputs are always positive since the mapping function limits the output between 0 and 1. The contribution of the neighborhood at $l^{th}$ layer at the iteration number t is given as follows:

$$H_{ijl}^{t} = \sum_{(q,r)\in N_{ij}} W_{ij,qr,k,l} O_{qrl}^{t} \tag{5.3}$$

The nonlinear updating rule is as follows:

$$O_{ijk}^{t+1} = \frac{Pos(O_{ijk}^{t} + \Delta O_{ijk}^{t})}{\sum_{l=1}^{m} Pos(O_{ijl}^{t} + \Delta O_{ijl}^{t})} \tag{5.4}$$

where $$\Delta O_{ijk}^{t} = \begin{cases} \delta & \text{if} \quad H_{ijk}^{t} = \max_{l}\{H_{ijl}^{t}\} \\ -\delta & \text{otherwise} \end{cases} \tag{5.5}$$

$$Pos(x) = \begin{cases} x & \text{if} \quad x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

and where $\delta$ is a small positive number, typically chosen as 0.1, and it determines the step size between two consecutive iterations. The denominator term in Equation 5.4 is the normalization factor which keeps the total sum of the outputs of the neurons residing in a specific location less than 1. The updating rule, given from 5.3 to 5.6, is based on two principles:

1. If $H_{ijk}$ is the maximum one among all contributions at location (i,j), $H_{ij.}$, a positive contribution corroborates the $U_{ijk}$. Thus the output of $U_{ijk}$ is increased.

2. If $H_{ijk}$ is not the maximum, a negative contribution decrements the output of $U_{ijk}$.

After initialization, the CSNN updates all the neurons according to the updating rule. Among all the neurons, which are located at a specific pixel site, only one neuron output can converge to 1 while the others are forced to 0. After convergence, the layer with output 1 becomes the segment label at that specific location.

The initial conditions for CSNN are quite crucial for convergence to a consistent solution since the CSNN actually performs deterministic relaxation operation on the fuzzy measure of the image features such as gray level, color, etc. as will be explained in Section 5.3. The result therefore depends highly on the initial conditions in the deterministic relaxation because the deterministic relaxation cannot crank back, ie., it goes forward from the initial conditions to a local minimum. There is no mechanism to backtrack if the process is stuck in a local minimum. Therefore a good initial condition is essential for convergence to the global minimum.

One way to initialize the neurons is carried out by using clustering and fuzzification in **Lin et. al. (1992)**. Kohonen's self-organising map (SOM) has been adopted for clustering purpose. The gray levels in the given image are classified into M classes by means of SOM.

After this clustering, the initial probabilities of the neurons are determined by a fuzzification rule: If the pixel site under consideration has been clustered to the segment with label L, then a larger probability value is assigned to the neuron which resides in the $L^{th}$ layer and smaller probability values are assigned to the neurons which are adjacent to the $L^{th}$ layer. Probability values of neurons which are far from $L^{th}$ layer are set to 0. The sum of the probability values for a pixel site is 1.

The connection weights play a crucial role in CSNN since they are interpreted as spatial constraints. The weights are determined in a heuristic manner in CSNN. The weight between $U_{ijk}$ and $U_{qrl}$, $W_{ij,qr,k,l}$, is given by the rule in 5.7.

$$W_{ijj,qr,k,l} = \frac{1}{p}\left(1 - \frac{2|k-l|}{M}\right) \tag{5.7}$$

where p is the size of neighborhood. The motivation behind this choice of weights is that a neuron must excite other neurons representing the labels of similar image features and must inhibit other neurons representing labels of quite different features. In this context, the maximum value of W, that is $\frac{1}{p}$, is obtained for k=l, that is whenever two neurons reside in the same segment layer. The minimum value, that is $\frac{1}{p}\left(1-\frac{2(M-1)}{M}\right)$, is also obtained when k=1, l=M or k=M, l=1, which is the neuron is far from the other one. Table 5.1 shows the weights when p=M=8.

Table 5.1 Connection weights between layers for p=m=8

| k↓ l→ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.125 | 0.093 | 0.062 | 0.031 | 0 | -0.031 | -0.062 | -0.093 |
| 2 | 0.093 | 0.125 | 0.093 | 0.062 | 0.031 | 0 | -0.031 | -0.062 |
| 3 | 0.062 | 0.093 | 0.125 | 0.093 | 0.062 | 0.031 | 0 | -0.031 |
| 4 | 0.031 | 0.062 | 0.093 | 0.125 | 0.093 | 0.062 | 0.031 | 0 |
| 5 | 0 | 0.031 | 0.062 | 0.093 | 0.125 | 0.093 | 0.062 | 0.031 |
| 6 | -0.031 | 0 | 0.031 | 0.062 | 0.093 | 0.125 | 0.093 | 0.062 |
| 7 | -0.062 | -0.031 | 0 | 0.031 | 0.062 | 0.093 | 0.125 | 0.093 |
| 8 | -0.093 | -0.062 | -0.031 | 0 | 0.031 | 0.062 | 0.093 | 0.125 |

The CSNN process proceeds in parallel and iteratively until the network converges. The total square difference of neuron outputs CE in consecutive iteration steps is adopted as a convergence criterion as given in Equation 5.8.

$$CE=\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m}\left(O_{ijk}^{t+1}-O_{ijk}^{t}\right)^2 \tag{5.8}$$

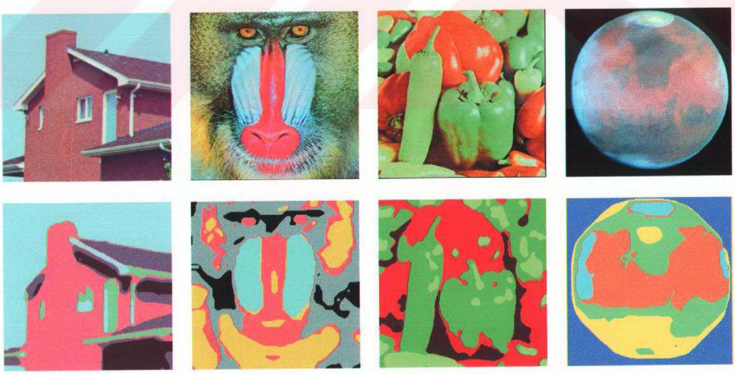
The convergence proof of the CSNN can be found in **Lin et.al. (1992)**.

The algorithm of CSNN is depicted in Figure 5.4. The segmentation results of some color images are shown in Figure 5.5. The number of segments belonging to each image was determined by using a cluster validity method, which is to be explained in Section 5.5. Numbers of segments are determined as 5, 5, 4, and 5 for the "House", "Mandrill", "Peppers" and "Mars" images, respectively.

```
CSNN Algorithm:

Cluster the given image using SOM.
Initiate the neurons by means of fuzzification using the pre-clustering.
t = 0
Repeat until (the convergence criterion satisfied, (CE < 0.01) )
        For (each pixel)
                For (each label, k )
                        Calculate H_ijk
                Endfor
                Select minumum U_ijl with the argument l.
                For (each label, k)
                        If ( k = l) then
                                ΔO_ijk=δ
                        Else
                                ΔO_ijk=-δ
                        Endif
                        Update the output of the neuron according to 5.4
                Endfor
        Endfor
Endrepeat
```

Figure 5.4 The pseudo-code of CSNN algorithm.



House (M = 5) Mandrill (M = 5) Pepper (M = 4) Mars(M = 5)

Figure 5.5 CSNN segmentation results of real world color images.

### 5.2.3 Interpretation of CSNN as a Relaxation

Relaxation labeling is widely used in image processing. Its attractiveness arises from its capability to incorporate contextual information about the scene. Relaxation has been applied to various image processing problems such as edge enhancement **(Schachter et.al., 1977)**, image enhancement **(Davis and Rosenfeld, 1978)**, scene labeling **(Rosenfeld et.al., 1976)**, pixel classification **(Eklundth et.al., 1980) (Therrien, 1989)** and thresholding **(Rosenfeld and Smith, 1981)**.

Broadly speaking, the relaxation is carried out in two main steps:

1. Initial labeling of objects independently from the spatial context.
2. Relax the initial labeling by taking into account the spatial context via some compatibility coefficients.

The aim of the relaxation is to find a consistent label assignment as a function of spatial context. This task is carried out by achieving a global consistency by using some local constraints. From this point of view, CSP can be viewed as a relaxation operation **(Haralick and Shapiro, 1993) (Tsao and Lin, 1991) (Lin et.al., 1992) (Hummel and Zucker, 1983)**. In this section, firstly the relaxation process will be briefly presented. Then the equivalence between the CSNN and the relaxation will be demostrated.

Relaxation can be formally defined as follows **(Borotschnig et.al., 1998)**: Consider a set of objects $i \in O$, to be labeled with a set of labels $\lambda \in \Lambda_i$, for each object i. $p_{i\lambda}$ is the assignment weight of the specific labeling $(i, \lambda)$ ($p_{i\lambda} \in [0,1]$). The consistency of label assignments between two objects is denoted by $R_{ij\lambda\mu}$, which is called the support value. $R_{ij\lambda\mu}$ measures the support for label $\lambda$ at object i arising from label $\mu$ at object j where $i \neq j$. $R_{ij\lambda\mu}$ encodes how strongly two labelings $(i,\lambda)$ and $(j,\mu)$ support each other. A positive high value of $R_{ij\lambda\mu}$ means that the labeling $(i,\lambda)$ supports the labeling $(j,\mu)$. On the contrary, a negative value of that denotes that the labeling $(i,\lambda)$ does not support the labeling $(j,\mu)$. If $R_{ij\lambda\mu}$ equals 0, then the two objects are considered as independent of each other. The support values reflect the spatial contextual information. After an initialization stage, the assigned weights are updated iteratively according to the chosen relaxation scheme by using support values.

Support values, $R_{ij\lambda\mu}$, can be computed from the given data. An intuitively way to determine these coefficients is that of taking the coefficients to be zero for non-neighbouring data and of giving for neighbouring pairs the weight of the mutual information given in Equation 5.9 **(Rosenfeld and Smith, 1981)**.

$$\log\left(\frac{\Pr ob(\text{object i in label } \lambda, \text{object j in label } \mu)}{\Pr ob(\text{object i in label } \lambda) \Pr ob(\text{object j in label } \mu)}\right) \tag{5.9}$$

But the determination of the joint probability function in Equation 5.9 is impractical for many images. Thus the coefficients are determined by using some ad-hoc approaches according to application **(Therrien, 1989)**.

Relaxation algorithms vary according to interpretation and usage of assignment weights **(Borotschnig et.al., 1998)**: They can be diversified broadly into two classes in terms of the normalization constraint on the assignment weights: Normalized models in which the assignment weights are restricted to the interval [0,1] and non-normalized models in which this restriction is not imposed. Another classification is based on the values taken by assignment weights: Discrete relaxation and continuous relaxation. In the discrete relaxation, the assignment weights can only take 0 or 1 values, whereas they can take the values in the interval [0,1] in the continuous case. A further classification is given in terms of the methodology used: Probabilistic relaxation and Fuzzy relaxations. There is major difference between the probabilistic and fuzzy relaxation. In the probabilistic relaxation, the assignment weights are interpreted as probability values, while in the fuzzy method the assignment weights denote the degrees of membership to a particular class of labels. Therefore, the fuzzy relaxation models use the fuzzy logic rules to update the assignment weights while the probabilistic ones use the classical probability techniques. It is evident that the probabilistic models must belong to the normalized relaxation. All non-normalized models can be considered as fuzzy ones, though some fuzzy relaxation methods use normalized models.

All models use the following general update scheme in the second stage of the relaxation operation:

$$p^{t+1} = F(p^{t}, R) \tag{5.10}$$

where F varies according to the model used. Whichever model is used, however, F must obey to some common rules. F must be designed so that $p_{i\lambda}$ should be increased if another object, j, has high assignment weight for label $\mu$, $p_{j\mu}$ , and the $R_{ij\lambda\mu}$ is high too. If $R_{ij\lambda\mu}$ is low then $p_{i\lambda}$ should be decreased. On the other hand, If $p_{j\mu}$ is low there must be no contribution to $p_{i\lambda}$. Table 5.2 summarizes this desired characteristic of the F function where $\Delta p_{i\lambda}$ denotes the change in $p_{i\lambda}$ between two consecutive relaxation steps, "+" and "-" express the increase and decrease in $p_{i\lambda}$, respectively **(Borotschnig et.al., 1998)**:

Table 5.2 Characteristic properties that the F function must possess.

| | | $R_{ij\lambda\mu}$ | |
|---|---|---|---|
| | $\Delta p_{i\lambda}$ | High | Low |
| $p_{i\mu}$ | High | + | - |
| | Low | 0 | 0 |

The most commonly used normalized model consists of two major steps **(Borotschnig et.al., 1998) (Rosenfeld et.al., 1976) (Therrien, 1989) (Hummel and Zucker, 1983)**. The first step calculates the total support for $(i,\lambda)$ from all other objects j, which is denoted by $q_{i\lambda}$ :

$$q_{i\lambda} = \sum_{\substack{j=1 \\ j\neq i}}^{n} \sum_{\mu=1}^{m} R_{ij\lambda\mu} p_{j\mu} \tag{5.11}$$

The compatibility coefficient remains in the interval [-1,1]. 1 denotes full compatibility between the assignments while –1 corresponds to total incompatibility. The second step is the calculation of the current value of the assignment weight:

$$p_{i\lambda}^{t+1} = \frac{p_{i\lambda}^{t}(1+q_{i\lambda}^{t})}{\sum_{\mu=1}^{m_i} p_{i\mu}^{t}(1+q_{i\mu}^{t})} \tag{5.12}$$

It can be realized that CSNN actually performs a probabilistic relaxation scheme if the formulation of CSNN compared with that of relaxation. In Equation 5.13, the update equations for the CSNN and the probabilistic relaxation are given together.

$$O_{ijk}^{t+1} = \frac{Pos(O_{ijk}^{t} + \Delta O_{ijk}^{t})}{\sum_{l=1}^{m} Pos(O_{ijl}^{t} + \Delta O_{ijl}^{t})} \quad \equiv \quad p_{i\lambda}^{t+1} = \frac{p_{i\lambda}^{t}(1+q_{i\lambda}^{t})}{\sum_{\mu=1}^{m_i} p_{i\mu}^{t}(1+q_{i\mu}^{t})} \tag{5.13}$$

Recall that $\Delta O_{ijk}^{t}$ is determined from the neighbourhood contribution ($H_{ijk}$) through constraints while $q_{i\lambda}^{t}$ denotes the support from neighbourhood through compatibility coefficients. Thus, CSNN performs a variation of the probabilistic relaxation, at least formally.

The relaxation results of test images are given in Figure 5.6. The initial map and compatibility coefficients are chosen the same as in the CSNN scheme. Thus both are initialized with fuzzified Kohonen's SOM and the constraint scheme given at Equation 5.7 is adopted as compatibility coefficients. As seen in the examples the results of relaxation are much more detailed than those of CSNN since architecture of the CSNN yields smoother segments. The two methods are compared using ground truth test-bed images and evaluation criteria, which are presented in Chapter 3. The performance of CSNN has been found to be better than relaxation by 68%.

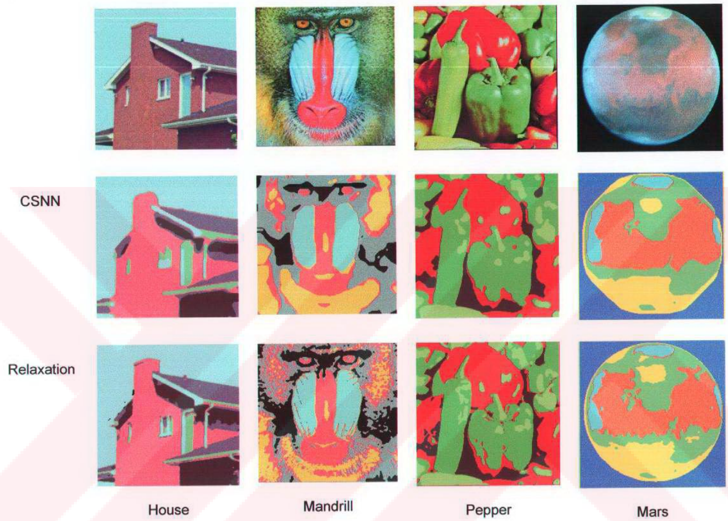



Figure 5.6. Results of relaxation. Segments are more detailed than CSNN.

### 5.2.4 Criticism of CSNN: Advantages and Disadvantages

CSNN is a tool for image segmentation that combines the clustering operation in measurement space with the relaxation method in spatial domain. Thus the aforementioned insufficiency of the clustering for spatial data in Chapter 2 is eliminated by relaxation in CSNN.

The first advantage of CSNN based segmentation is that the neural network structure provides an efficient and simple way to solve the CSP. The relaxation rate can be easily controlled by means of varying the connection weights that correspond to constraints or compatibility coefficients and neighbourhood size around each pixel site. The behaviour of the network, therefore, can be adjusted according to applications without any changes in the network structure. For example, the smoother segments could be obtained by increasing the neighbourhood size.

Secondly, because relaxation is carried out in the spatial domain, it is a region-oriented method, provides hence the spatial continuity. On the other hand, the most region based segmentation methods use some energy functional to accomplish their segmentation task (Some of them will briefly reviewed in Chapter 5.8). The proper formulation of the energy functional is crucial for these methods. In contrast, CSNN does not require any explicit energy formulation. Its structure implies an MRF based energy minimization, as to be shown in Section 5.8.

Thirdly, the real-time demand plays dominant role in many applications. The neural network structure of CSNN can satisfy real-time constraints for these applications. Its neurons can be implemented by using VLSI chips. Furthermore, its massively parallel structure can be used in distributed computation **(Lückenhaus and Eckstein, 1997) (Zikos et.al., 1997)**.

Fourthly, CSNN as proposed in **Lin et.al. (1992)** uses only gray level values of the given image to initiate the its structure. Since multiband images such as color images and Landsat images generally provide more information than gray tone, however, it is desired to enhance to CSNN to accommodate them. It is straightforward to augment the CSNN to use multiband images by using a multiband-clustering algorithm in the initialization step of the algorithm.

On the other hand, CSNN has some disadvantages too. First, determining the number of segments from a given image is a general problem in image segmentation algorithms as well as in the CSNN method. In general, this information is given to the algorithm as some a priori information. However, the number of segments is not known in many applications. Thus, the determination of the number of segments automatically from the given image is a task to be done in any unsupervised segmentation. As addressed in Section 5.3, this problem can be solved to a certain extent by means of a cluster validity algorithm.

Second, CSNN has a long convergence time. Furthermore, the convergence error, given in Equation 5.8, does not decrease beyond a certain value. This problem arises from the segment boundaries, as the algorithm hesitates in the assignment of pixels around the segment boundaries, causing a lot of futile iterations. The resulting segment borders are also noisy as illustrated in Figure 5.7. In this example, segmentation results of a test image , with four segments, are shown. As seen in this example, most of the pixels within the segments are labelled correctly in the first few steps. On the other hand, most of the boundary pixels are not correctly labelled even after the $50^{th}$ step. This can be seen more clearly in Figure 5.8. Two convergence errors are computed, one from the pixels which reside within the regions and the other from border pixels. One can notice that while the within segment error converges rapidly to 0, the boundary error remains at a standstill.

A third shortcoming of the CSNN algorithm is that it generates under-segmented results as small segments can be captured by big ones. This effect can be seen in Figure 5.5. For example, The details of the window segment on the "house" image are absorbed by the neighborhood segments.

To overcome these problems several modifications that augment and enhance the CSNN are proposed in this thesis. For example, in the initialization step, the given image is firstly clustered by using a SOM, then the crisp result of SOM is fuzzified by an ad-hoc technique. This method of initialization is not a natural one. Alternative or better initializations have been investigated and fuzzy c-means clustering has been adopted. Using a fuzzy clustering for initialization of CSNN is a more convenient approach since the fuzzy measure values of the pixels can be directly obtained without any artificial fuzzification process. These modifications will be introduced in the following sections.

## 5.3 Modification I: Unsupervised Segmentation Using Cluster Validity Analysis in CSNN

The number of segments is presented to the CSNN segmentation as an a priori information. Most other algorithms necessitate the knowledge of the number of classes. A similar problem arises in the clustering as well. Since most of the clustering algorithms are unsupervised, the outcome of the algorithm must be verified and this verification is known as cluster validity.

Cluster validity can be used to determine the number of clusters in the given data as well as to specify the performance of the used clustering technique. There exist several cluster validity methods and models in the literature. **Dubes and Jain (1977)** presented a comprehensive survey about the cluster validity methods.

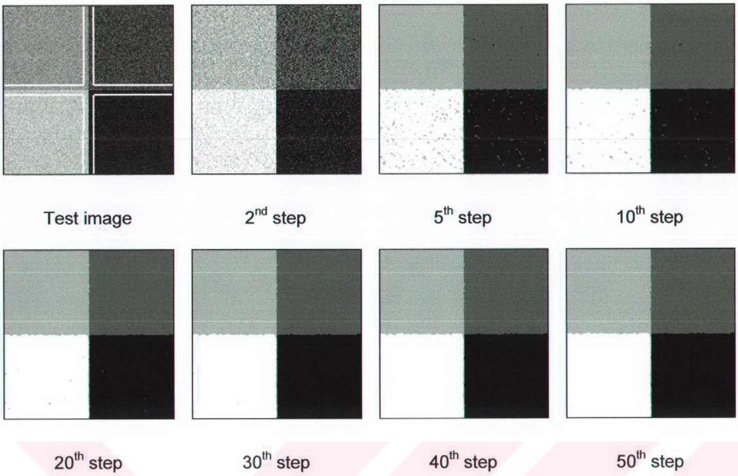



Figure 5.7. The results of the CSNN at some specific iteration step.

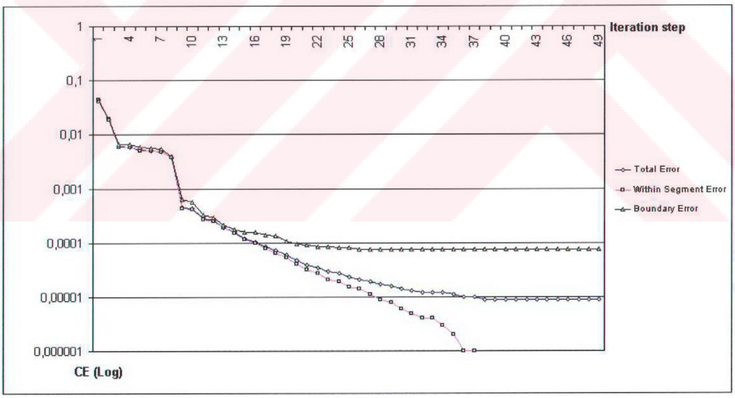



Figure 5.8. Convergence Error versus iteration steps for test image. Within segment error is rapidly converges to 0 whereas the boundary error remains constant after $20^{th}$ step.

In the cluster validity application, several clustering algorithms or same algorithm with different parameter settings can be applied to the data to determine which result fits the validity index best. Thus cluster validity schemes can be used to determine the number of segments automatically. This is achieved by performing the clustering operation by varying the number of clusters, say from 2 to some maximum number and the one that yields the best validity index is chosen as the number of classes in the given image. The process is depicted in the Figure 5.9. Some of the cluster validity algorithms will be summarized the following paragraphs.

**Davis and Bouldin (1979)** proposed an optimization index which varies in terms of the number of clusters. The lowest value of this index is chosen as the best number of clusters. The proposed cluster validity index takes into account that the clusters must have minimum scatter within the cluster and maximum distance between clusters. It determines the number of clusters which yields compact and separate clusters as possible.

**Rezaee et.al. (1998)** proposed a validity scheme for fuzzy c-means algorithms. They took into consideration compactness of any fuzzy c-partition as well as its separation. Their scheme, which is very similar to the Davies and Bouldin's one, combines within and between scattering.

**Zhang and Modestino (1990)** proposed a cluster validation method based on model fitting approach. They cast the cluster validity problem as a multiple decision

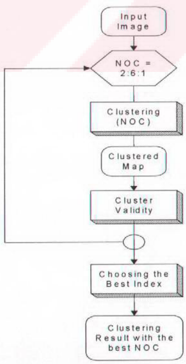



Figure 5.9 Cluster Validity Process. The number of clusters is varied from 2 to a maximum value (6, in this thesis) in the clustering phase. Cluster validity index is computed for each clustered map. Then the best one is chosen as the clustering result and its index is adopted as the number segments for the CSNN.

problem. The method is based on a model parameterized by the number of clusters (K). The aim is to find the best set of K parameter which best fits to the data. Akaike's information criterion (AIC) which was used in determining the order and the parameters of an AR model is adopted. The parameter that minimizes the AIC was chosen as the number of clusters that best fit to the given data.

Let $\mathbf{Y} = \{ \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N \}$ denote the data which consists of N random vectors. It is assumed that $\mathbf{Y}$ contains T clusters and can be modeled by a probability density function (pdf), $p_K(\mathbf{Y}|a^{(T)})$. $a^{(T)}$ represents the model parameter vector. Under these assumptions, the problem of cluster validity can be considered as the model order determination which best fits to the data. AIC which is chosen as the goodness of fit is given in equation 5.14.

$$\mathrm{AIC}(K) = -2\log\{p_k(Y \mid \hat{a}_{ML}^{(K)}) + 2K' \tag{5.14}$$

where $\hat{a}_{ML}{}^{(K)}$ is the maximum likelihood (ML) estimate of the model parameter vector a(K), and K' is the number of independently adjustable parameters of the K-class data model. Thus the estimated number of clusters is determined as follows:

$$K_0 = \underset{1 \le K \le K_{max}}{\arg\min} \quad \mathrm{AIC}(K) \tag{5.15}$$

As seen as in Equation 5.14, AIC consists of two terms: The first one is the model fitting penalty and the second one is the excess parameter penalty. The model fitting term monotonically decreases in terms of the number of clusters whereas the penalty term increases. Thus the sum of the two should hopefully yield convex curve, and the minimum point of this curve should indicate the best index.

We have witnessed that the two terms are not always commensurate such that the model fitting term dominates the AIC. In this thesis, to alleviate this handicap, we have tested a heuristic method that is a modification of the AIC criterion. The heuristic role consists in normalizing the two terms to their ranges. Thus this yields a desired index which has a well defined minimum **(Kurugöllü and Sankur, 1998a)**.

Three of these methods are applied to 10 test-bed images. The results are summarized in Table 5.3, Table 5.4 and Table 5.5, that are for Davis-Bouldin index, Rezaee criterion and Modified AIC criterion respectively. In all tables, the rows correspond to the number of segments while the columns indicate color spread categories listed in Table 3.4. The cells of the table indicate the number of cases for which the number of classes was correctly determined by the algorithm. One can notice that the Modified AIC criterion has the best score on the average, that is 68%.

Thus it is used as the cluster validity index in the CSNN algorithm. It is worth noting that the performance of all indexes decreases when the number of classes increases. The performance of Modified AIC index declines with the growing of the color spread, faster than that of the other two methods.

Table 5.3 Performance scores of the Davies-Bouldin index according to color spread

| | Category 1 | Category 2 | Category 3 | Category 4 | Total |
|---|---|---|---|---|---|
| 2 | 10/10 | 10/10 | 10/10 | 10/10 | **40/40 – 100%** |
| 3 | 7/10 | 7/10 | 5/10 | 6/10 | **25/40 – 62%** |
| 4 | 3/10 | 4/10 | 2/10 | 1/10 | **10/40 – 25%** |
| 5 | 3/10 | 1/10 | 2/10 | 1/10 | **7/40 – 24%** |
| 6 | 0/10 | 0/10 | 1/10 | 0/10 | **1/40 – 2%** |
| Total | **23/50 – 46%** | **22/50 – 44%** | **19/50 – 38%** | **18/50 – 36%** | **82/200 – 41%** |

Table 5.4 Performance scores of the Rezaee criterion according to color spread

| | Category 1 | Category 2 | Category 3 | Category 4 | Total |
|---|---|---|---|---|---|
| 2 | 10/10 | 4/10 | 8/10 | 6/10 | **28/40 – 85%** |
| 3 | 3/10 | 8/10 | 7/10 | 8/10 | **26/40 – 65%** |
| 4 | 7/10 | 7/10 | 2/10 | 4/10 | **20/40 – 50%** |
| 5 | 5/10 | 5/10 | 2/10 | 3/10 | **15/40 – 37%** |
| 6 | 1/10 | 0/10 | 0/10 | 0/10 | **1/40 – 2%** |
| Total | **26/50 – 52%** | **24/50 – 48%** | **19/50 – 38%** | **21/50 – 42%** | **90/200 – 45%** |

Table 5.5 Performance scores of the Normalized AIC criterion according to color spread

| | Category 1 | Category 2 | Category 3 | Category 4 | Total |
|---|---|---|---|---|---|
| 2 | 10/10 | 10/10 | 10/10 | 8/10 | **38/40 – 95%** |
| 3 | 10/10 | 10/10 | 10/10 | 6/10 | **36/40 – 90%** |
| 4 | 8/10 | 8/10 | 6/10 | 4/10 | **26/40 – 65%** |
| 5 | 6/10 | 7/10 | 5/10 | 2/10 | **20/40 – 50%** |
| 6 | 7/10 | 5/10 | 3/10 | 2/10 | **17/40 – 42%** |
| Total | **41/50 – 82%** | **40/50 – 80%** | **34/50 – 68%** | **22/50 – 44%** | **137/200– 68%** |

When the modified AIC validity index is applied to the natural test images in Chapter 3, it is also observed that the modified AIC criterion gives very reasonable number of clusters. Table 5.6 shows the number of clusters of the test images which are determined by using modified AIC.

Table 5.6 Number of Clusters of test images determined using modified AIC index

| Image | Number of Clusters |
|---|---|
| House | 5 |
| Mandrill | 5 |
| Peppers | 4 |
| Mars | 5 |

## 5.4 Modification II: Improvement in the Initialization Stage

It is evident that the initial labeling in the relaxation algorithm plays a crucial role in the overall process. More consistent labeling in the initialization step results in a better segmentation result. Thus any enhancement in the initialization step of the CSNN should improve the performance of the algorithm.

It is well known that color grants more useful information than gray level about the scene. For example, if there exit several objects with different surface color characteristic but identical reflectance property it is not possible to distinguish the objects from each other using only gray level information. It is assumed that the pixels coming from the same surface will cluster in the color space **(Valavanis et.al., 1996)**. Thus color information is especially used in the clustering based image segmentation algorithms **(Coleman and Andrews, 1979) (Healey, 1996) (Uchiyama and Arbib, 1994)**. As regarding using only gray level images in CSNN, it is expected that using color images improve the performance of the algorithm significantly. This is achieved by using color vectors in the pre-clustering step instead of gray levels. The two cases are compared using ground truth test-bed images and the evaluation criterion, which are presented in Chapter 3. The performance of CSNN that uses color information is better than gray level one with the 85% ratio.

As mentioned in Section 5.2.2.2, Kohonen's SOM algorithm is used as the clustering tool in the pre-clustering step of CSNN in the original algorithm. After clustering, the initial label assignments are determined by a technique that applies an ad-hoc fuzzification process to the clustering results. It gives a big score to the winner label while gives smaller scores to the neighbour labels. The labels that are far from the winner one take zero score. Thus, their winning chance in the relaxation step is very low. Thus instead of using SOM, other fuzzy clustering such as fuzzy c-means is used. This eliminates the above problem. Any fuzzification process is not required in this case since the clustering result is already fuzzy. All labels take any score that is different from zero. Thus, the winning chance of the labels that are far from the winner one (that has biggest fuzzy score) is increased. In this thesis, fuzzy c-means algorithm is adopted instead of SOM method for clustering rule. The test-bed is used to compare how two clustering tools affect the CSNN performance. The performance of the fuzzy c-means has been found to be better than that of SOM by a 72% ratio.

## 5.5 Modification III: Including Boundary Constraints to CSNN: Boundary Constraint Satisfaction Neural Network (B-CSNN)

To solve the convergence problems around the boundaries, a modification called **B**oundary-**C**onstraint **S**atisfaction **N**eural **N**etwork (B-CSNN) is proposed in this thesis. To this purpose, a coarse boundary map of the input image is obtained by means of Marr-Hildreth edge detector and the neuron connections across the boundaries are severed by setting them to 0. So the contributions of the boundary pixels to their neighbors are precluded as depicted in Figure 5.10 **(Kurugöllü et.al.,**

**1996)**. This can be viewed as additional constraints imposed to segmentation process. As a consequence, both the boundary constraints are used to process the image pixels in the B-CSNN algorithm as well as the spatial constraints. This modification prevents futile iterations at the segment boundaries. Consequently, both the numbers of iterations and the convergence error of the B-CSNN are reduced significantly as seen in Figure 5.14. The flow diagram of the algorithm is shown in Figure 5.11. Various segmentation results are depicted in Figure 5.12. It can be noticed that segmentation boundary noise and absorption of the small segments by their larger neighbors are avoided in the B-CSNN algorithm. Details can be seen in Figure 5.13.

B-CSNN method gives better results than CSNN method in a quantitative manner. Performance ratio of B-CSNN has been found to be better than that of CSNN by 66%. More detailed quantitative results are given in Section 5.9.

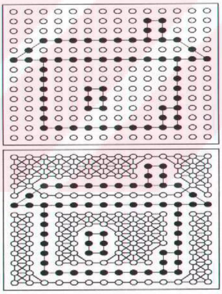


Figure 5.10 Adjustment on connection weights by using edge map of the input image. The black dots indicate the edge pixels. The contribution of the boundary pixel to their neighbor are precluded by assigning 0 to its connection weights.

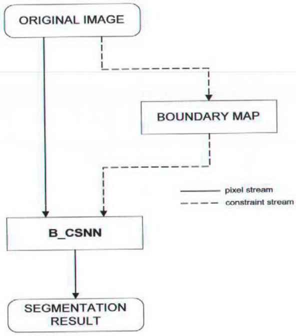



Figure 5.11. Flow diagram of the B_CSNN algorithm. Besides the original image, a coarse boundary information is used to determine segment boundary.

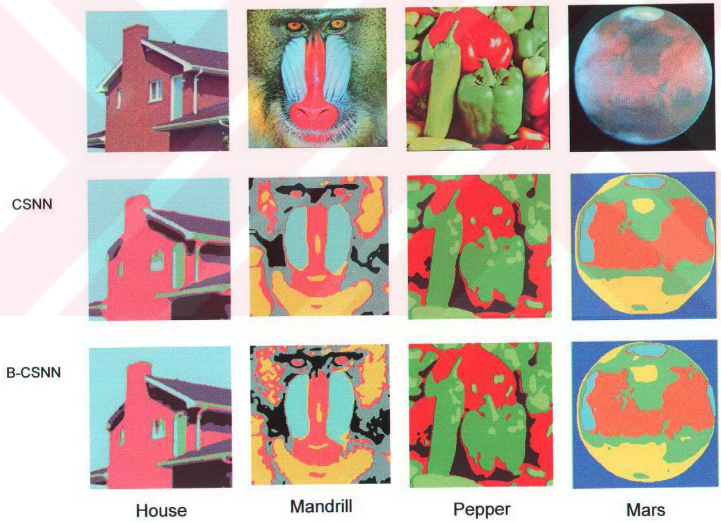
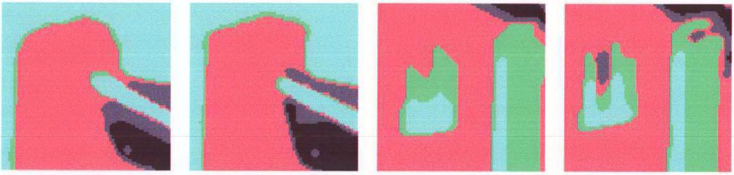



Figure 5.12. Segmentation results obtained from CSNN and B-CSNN algorithms.



CSNN B-CSNN CSNN B_CSNN

Improvement in segment boundaries Improvement in absorption problem

Figure 5.13. Detail comparisons of the algorithms.

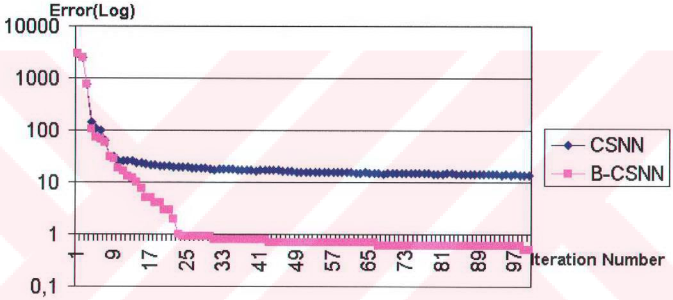
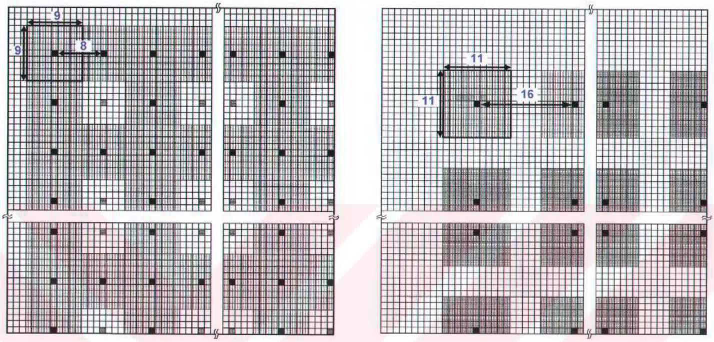



Figure 5.14. Convergence error with respect to the number of iterations of both algorithm for a 256 by 256 image. At 23rd iteration step, convergence error of B_CSNN is under 1 while that of CSNN is still 10

## 5.6 Modification IV: Multiresolution approach to CSNN: Multi-Scan Constraint Satisfaction Neural Network (MS-CSNN)

A structural modification of the novel method proposed in this thesis, called Multi-Scan CSNN (MS-CSNN), aims to enhance the performance of CSNN on both the rate of convergence as well as to reduce the misclassified pixels. This is carried out by using multiresolution scheme in the CSNN method **(Kurugöllü and Sankur, 1998a) (Kurugöllü and Sankur, 1999b)**.

The topology of the MS-CSNN algorithm is basically similar to that in Figure 5.1 and Figure 5.2. However it differs in the scanning schedule of the pixels and their neighborhood sizes, which effectively emulate a multiresolution approach. Notice that in this technique "scanning" means updating the label of specifically visited pixels via a CSNN using their neighborhood label information. Thus a separate CSNN is run till convergence for each "resolution" stage. Each such CSNN has the same number of layers (classes) as optimally determined with cluster validation. The CSNN structures differ however in the pixels they are handling at each stage and the corresponding neighborhood sizes (Figure 5. 15).




(a) Scanning scheme in 4$^{th}$ Level.

(b) Scanning scheme in 3$^{rd}$ Level.

■ CURRENT VISITING PIXEL
▦ NEIGHBORHOOD PIXEL
▧ VISITED PIXEL IN UPPER LEVELS

Figure 5.15 Scanning scheme in the first two levels. The neighborhood of the scanned pixel is decreased in the lower levels as well as the step size.

Another interpretation of the multi-scan approach is a "pyramidal" scheme, where the segmentation that takes place at the top level is propagated toward lower levels till the original lattice at the bottom. The higher levels are characterized by more sparse pixels and by larger neighborhoods. The lower levels have denser pixels to be visited but smaller neighborhoods as illustrated in Figure 5.15. In the example of an image of 256x256, and five resolution stages, one updates at the top level, only every 16$^{th}$ pixel along any row or column, each with a 11x11 neighborhood. The third level will handle pixels spaced by 8x8, skipping the ones handled already in the 4$^{th}$ level. At the bottom level fully 128x128 pixel labels are updated, but each with 3x3 neighborhoods. The pixel labels that have been updated, let's say the 16x16 labels in the 4$^{th}$ level (Figure 5.15.a) are not changed any more in any of the lower levels, but they serve as neighborhood information for other pixels in the lower levels. This process consists of converging the labels of subsets pixels in each resolution level using a CSNN, and inheriting the fixated pixel labels from one level to the lower levels. In other words, the MS-CSNN begins the segmentation process from the top ($4^{th}$) level using the same dynamics as the CSNN algorithm. Whenever it converges to a solution, it begins to another process in the subsequent lower level. This process continues until one reaches the bottom ($0^{th}$) level.

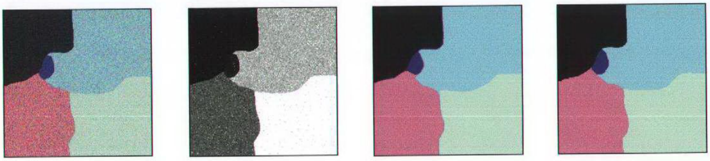
We have found that five levels overall, including the original image (at level 0), were adequate. Levels from 4 to 0 are treated with a visiting schedule that gets sparser toward higher levels. The neighborhood sizes and the number of visiting pixels are shown in Table 5.7. The visiting schedule is run by subsampling the image at a rate $2^k$ at the $k^{th}$ level. More specifically, at the $k^{th}$ level, one advances by steps of length $2^k$ in the horizontal or vertical directions. Furthermore in the lower levels one does not visit the pixels already visited in one of the upper levels. Thus in the $k^{th}$ level one ends up visiting only $\left(\frac{N}{2^k} x \frac{N}{2^k}\right) - \sum_{i=k+1}^{K}\left(\frac{N}{2^i} x \frac{N}{2^i}\right)$ pixels. The number of visited pixels and the neighborhood sizes with respect to levels are given in Table 5.7. Two stages of this scanning scheme are also illustrated in Figure 5.15.

In summary large neighborhoods used in the upper levels give coarse but reliable segmentation, while at the lower levels one recourses to a finer representation. In this manner, the busyness due of swapping pixel labels, especially at the boundaries, is avoided. This increased stability of segmentation does not augment the computational cost, as the number of visited pixels, amounts to, in the final analysis to, NxN. Thus, the complexity of the proposed MS-CSNN method is the same as that of CSNN algorithm. But the performance and the convergence rate of the novel method are better than the CSNN algorithm.

Table 5.7 Number of visited pixels and the neighborhoods sizes used in the levels.

| **Level** | **Number of Visited Pixels** | **Neighborhood sizes** |
|---|---|---|
| 4 | 16x16 | 11x11 |
| 3 | 16x16 | 9x9 |
| 2 | 32x32 | 7x7 |
| 1 | 64x64 | 5x5 |
| 0 | 128x128 | 3x3 |
| **Total** | 256x256 | |

Figure 5.16 shows the results of both CSNN and MS-CSNN algorithm. Detail of these results are also depicted in Figure 5.17. The small middle region has been segmented by the MS-CSNN more correctly.



Colored test image    Clustering map    CSNN result    MS_CSNN result

Figure 5.16. The segmentation result of CSNN and MS_CSNN methods in the colored test image.

Colored test image    CSNN result    MS_CSNN result

Figure 5.17. Detail of segmentation in Figure 5.15.

The second improvement realized from the MS-CSNN algorithm was a higher convergence rate. In Figure 5.18, the logarithm of convergence error against recursion steps for the "Landsat" image is illustrated. The error in the case of CSNN segmentation falls off gradually but does not diminish beyond a level of "10" even after 100 iterations. The MS-CSNN segmentation has a very different nature: At every resolution stage it converges more rapidly, but it has saccadic upswings at instants when not yet updated pixels are considered from level to level. Note that the asymptotic error always becomes lower than that of the CSNN segmentation.

The results of real world images are depicted in Figure 5.19. It is noticed that segments are preserved by proposed algorithm as well as segment boundaries are more realistic.

The proposed method has been applied to ground truth test-bed data. MS-CSNN method has given better result according to CSNN method for 72% of test-bed images.

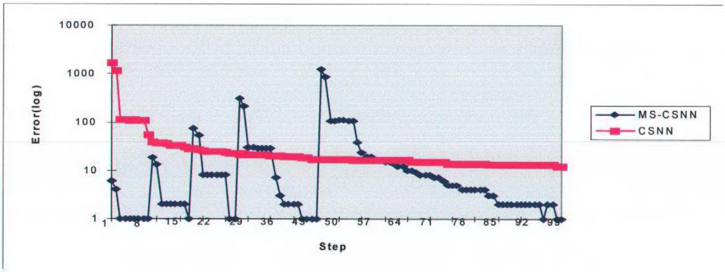



Figure 5.18. Comparison of the convergence rates of CSNN and MS_CSNN. Notice how the error jumps at steps when new pixels whose labels are not yet updated via CSNN are first handled.

As a consequence, improvements of the proposed segmentation method based on the CSNN can be summarized as follows:

- This method uses the basic CSNN structure for segmentation, but the scanning scheme of the pixels is modified to achieve a multiresolution character, where pixels are scanned with sparse intervals but with large neighborhoods in the first level of the algorithm. The intervals between pixels are reduced as well as the neighborhood size toward lower layers.

- It appears that the main bodies of the segments are constructed in the high levels of the pyramid, while only fine details are dealt with in the lower levels. Thus the time spent to work on details is reduced and the convergence rate of the algorithm is improved.

- The other innovation of the proposed method is using the cluster validity to determine automatically the number of segments in the given image. The number of clusters in the given image is found by using AIC cluster validity index.

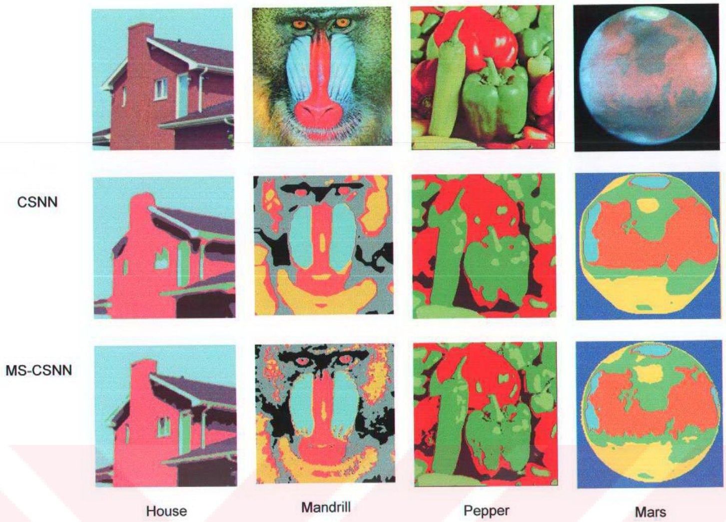



Figure 5.19. Results of MS-CSNN. Segments are preserved in MS-CSNN against oversegmentation of CSNN.

## 5.7 Modification V: Cooperation of Different CSNNs in Different Resolutions: Pyramidal Constraint Satisfaction Neural Network (P-CSNN)

In Section 5.6 multiresolution approach is introduced to the CSNN algorithm by using a different pixel scanning scheme. Another way to implement a multiresolution scheme is to assemble a pyramidal scheme to the CSNN. In this way each resolution level of the pyramid consists of different number of pixels that decreases from bottom to top level. Each level has its own CSNN structure. **(Kurugöllü and Sankur, 1998b)**.

In this modification, CSNN method has been enhanced with multiresolution capability. This is accomplished by modifying the new neural network structure based on pyramid of data as shown in Figure 5.20. Notice that the pyramid is not constructed by successive filtering of the image data but from the full-resolution label space using father-child relationship. Each level of the pyramid consists of a separate CSNN structure but with interconnections across resolution levels. This pyramid structure is established in the initialization step. The clustered map is used for this purpose. The clustered map is initialized as in the conventional CSNN,which has the original image dimensions, and constitutes the base of the pyramid structure

($0^{th}$ level). Then this segmentation label map is transformed to $1^{st}$ level label map using an approach suggested by **Burt et.al (1981)**. In this approach, a parent pixel in the upper level is related with its 16 child pixels in the lower level. A child pixel in the lower level has also 4 parent pixels in the upper level as seen as Figure 5.21. A voting among the child pixels is performed and the maximum encountered segment label of the child pixels is assigned to the parent pixel. Thus the established label map at the $1^{st}$ level is used to generate the $2^{nd}$ level label map, and so on.

After initialization of the pyramidal CSNN as described above, the neural network runs to a segmentation result which must satisfy all the constraints in the whole pyramid. The constraints are determined by the synaptic weights between neurons and each neuron synapses to

a) 16 child sites in the lower resolution level,

b) 4 parent sites in the higher resolution level,

c) 8 neighboring sites in the same resolution level, for all segments.

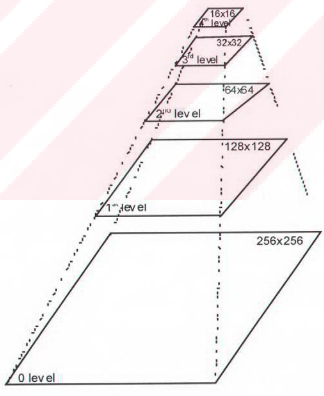
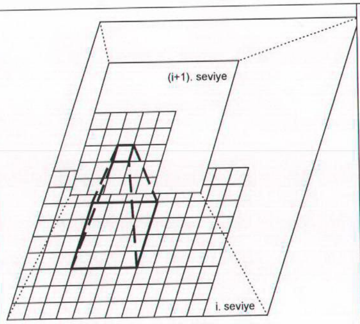
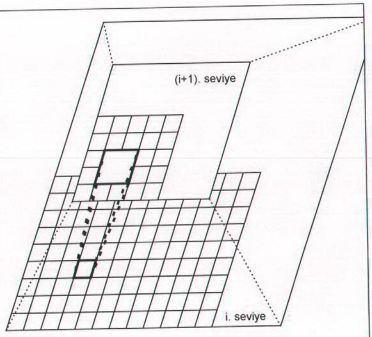



Figure 5.20. Structure of Pyramidal CSNN. Each level is operated on by a separate CSNN, with the same number of segments.




(a) Relationship between top-bottom levels. One parent at the $(i+1)^{th}$ level has 16 children at the $i^{th}$ level




(b) Relationship between bottom-top levels. A child at the $i^{th}$ level has 4 parents at the $(i+1)^{th}$ level.

Figure 5.21 Connections between levels. These connection weights can be adjusted to obtain more smoothed or more detailed segmentation result.

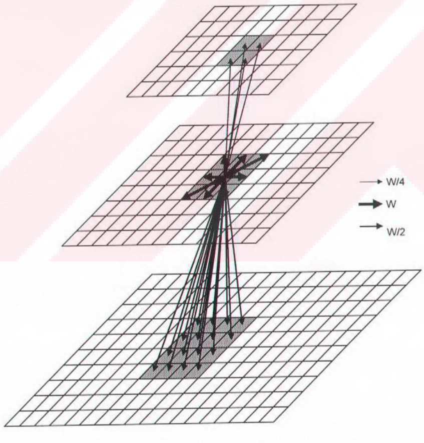
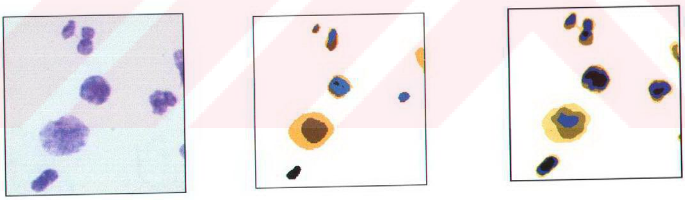



Figure 5.22. Relationship between levels of the pyramid.

Each site contains the stated number of segments neurons in the segmentation levels as depicted Figure 5.22. The weights of these connections determine the constraints and the dynamic structure of the neural network. For example, if high connection weights are attributed to lower-to-higher resolution synapses more smoothed segments will be obtained because smoothed pixels have dominant role in this scheme. On the contrary, if high connection weights are chosen for higher-to-lower resolution synapses more detailed segments will be obtained. Consequently, the proposed method gives the user flexibility to tune the algorithm for user's demands.

Both coarser and finer analysis, for instance, may be required in some medical image analysis applications. In Figure 5.23.a, a cell image is depicted. The cells are painted purple by chemical mean. Desired information from this cell image is a) The morphology of the cell (Perimeter, area, form factor, etc.), b) The irregular area in the nucleoid. These two tasks are carried out successfully by the pyramidal CSNN by its tuning the connection weights. In Figure 5.23.b, a more smooth segmentation is obtained for calculation of the morphology of the cells. The more detailed segmentation which exposes irregular areas is used for examination of the nucleoid as depicted in Figure 5.23.c. Thus the user can obtain two desired solution with a single P-CSNN algorithm by just tuning synaptic weights between resolution levels.

The convergence error is also improved in the P-CSNN scheme as seen in Figure 5.24. The control of smoothness and details is illustrated in Figure 5.25. Notice how by changing the emphasis for higher resolution and lower resolution images, more detailed and more smoothed images are obtained, repectively.



(a) Original Cell Image (b) Smoothed Segmentation Result (c) Detailed Segmentation Result

Figure 5.23. Result of P-CSNN with different details

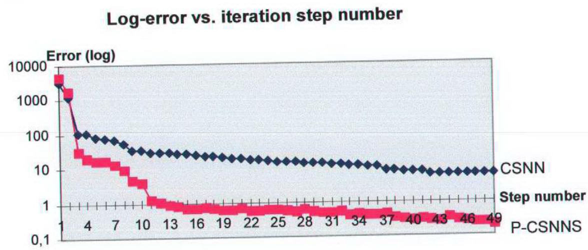



Figure 5.24. Convergence rate of P-CSNN: The error goes more rapidly to a lower level.

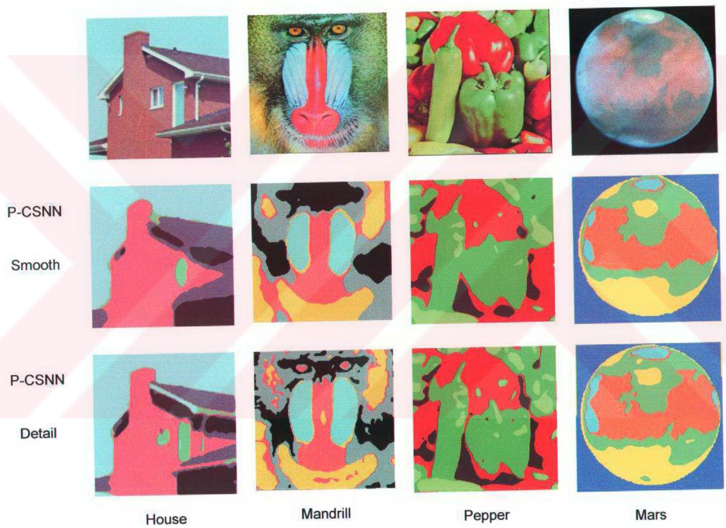



Figure 5.25. Smoother and more detailed segmentation maps of real world images.

## 5.8 Modification VI: Applying Markov Random Field Approach to CSNN: MRF-CSNN

As stated in Chapter 5.3, CSNN uses a relaxation scheme that exploits dependence among objects in a heuristic manner. Statistical models of image formation such as Markov Random Field (MRF) models are constructive approaches to such dependence **(Therrien, 1989)**. This approach causes to get the advantage of using data, world and sensor models **(Szeliski, 1989)**. Thus it is conjectured that CSNN can be applied as an alternative solution to the MRF based segmentation. In the following sections, firstly Bayesian approach of image segmentation will be briefly presented. Then the augmentation of CSNN to MRF approach will be introduced.

### 5.8.1 A Brief Statement of Bayesian Image Segmentation using MRF:

The segmentation problem can be formulated as a Bayesian estimation of a maximum a posterior probability of the hidden pixel labels. The spatial dependence of the labels are modelled as an MRF while the image features such as gray level, color, texture parameters, etc. are modelled via their respective probability density functions **Geman and Geman's (1984) (Dubes and Jain, 1989) (Perez, 1998) (Szeliski, 1989)**. Among several possible solutions to the MRF segmentation problem ways such as Maximizer of Posterior Marginals and Expectation Maximization to express Bayesian approach, only MAP estimation is dealt with in this study.

The goal of segmentation, from the point of Bayesian interpretation, is to estimate the original scene, **S**, from the given image, **I**. This estimation process consists of finding the maximum a posterior probability (MAP) of **S** given **I**, P(**S**|**I**) **(Szeliski, 1989)**. To compute this posterior model, the Bayes' rule can be used. Thus, the estimation of the actual scene, say $\hat{S}$, can be defined as follows:

$$\hat{S} = \arg\max_{S}\{p(S \mid I)\} = \arg\max_{S}\left\{\frac{p(I \mid S)p(S)}{p(I)}\right\} \tag{5.16}$$

Since the divisor in the right hand side of the equation 5.16 is independent from argument, **S**, it can be dropped. Then the MAP estimation becomes:

$$\hat{S} = \arg\max_{S}\{p(S|I)\} \propto \arg\max_{S}\{p(I|S)p(S)\} \tag{5.17}$$

In this MAP formulation there are two components: The first one, P(**S**), is the joint probability of the labels over the whole image lattice. The second component is the image model, P(**I**|**S**), that is pixel features is given the corresponding segmentation label. In the light of Clifford-Hammersley theorem, the MRF problem can be expressed as a Gibbsian distribution. Thus the right hand side of 5.16 can be combined as a Gibbsian energy, which is like that in 5.18, under MRF assumptions.

$$P(w) = \frac{1}{Z}\exp\left\{-\frac{U(w)}{T}\right\} \qquad Z = \sum_{w\in\Omega}\exp\left\{-\frac{U(w)}{T}\right\} \tag{5.18}$$

where w, T and Z denotes the state of the system, i.e., current segmentation, the temperature parameter that controls the peak of the distribution and the normalization constant, respectively. Thus, maximum a priori segmentation as in 5.16 reduces to the minimization of the corresponding Gibbsian energy, U(w). In other words, the minimum energy configuration is the MAP segmentation segmentation of the image.

Gibbsian energy is derived using the Markovianity condition as well as the relationship between MRF and Gibbs distribution. Markovian condition of a MRF, stated in Equation 5.19, expresses how to make a global inference using only local information. Without this condition, the computation of the conditional probability of a pixel becomes an insurmountable task. It emphasizes that conditional probability of a pixel, $x_{ij}$, over entire image, L, depends only on its neighbour pixels.

$$P(x_{ij} \mid x_{L-ij}) = P(x_{ij} \mid x_{kl},\ (k,l) \in N_{ij}) \tag{5.19}$$

where L-ij denotes entire image except for pixel $x_{ij}$.

Evaluation of conditional probability in systems such an image which have large configuration space becomes very hard problem. Fortunately, an important theorem is stated by Hammersley-Cilfford about equivalence between MRF and Gibbs distributions. This theorem furnishes a mathematically manageable mechanism to specify the conditional probability of an MRF. The theorem emphasizes that **X** is an MRF with respect to N if the probability that all of the realizations of the random field, (P(x)) is a Gibbs distribution with respect to the neighbourhood system N, and vice versa. This means that the joint probability of a MRF obeys a Gibbs distribution. An elegant proof of the theorem can be found in **Li ( 1995)**. A Gibbs distribution is stated as in Equation 5.18. Z given in Equation 5.17 is a normalizing constant, called partition function. U is an energy term. T named temperature controls the peak of the distribution. The neighbourhood system is enacted via cliques so that the entire Gibbsian energy can be calculated by summing the contributions of clique potentials, $V_c(w)$, at each site.

Let c denote a clique. Then U can be calculated by summing over all clique potentials:

$$U(w) = \sum_{c \in C} V_c(w) \tag{5.20}$$

C represents the set of all cliques contained in the image lattice. $V_c(w)$ depends only on the pixel values in the clique c. The form of clique potentials, $V_c(w)$, determines the model which depends on the application **(Dubes and Jain, 1989)**.

Clique potentials can be divided into two categories: Clique associated pixel under consideration and cliques associated neighborhood of pixel under consideration. The former one corresponds to P(**I**|**S**), which is used for modelling the image formation process, while the latter one corresponds to P(**S**), that models the scene, in Equation 5.17. Thus the energy can be written with two terms as in Equation 5.21. The first term controls the degree of closeness of the state to the data, called data closeness term, while the second one permits to relax the energy over neighbour states, which coincides with the smoothness term.

$$U(w) = U(w)_D + U(w)_S \tag{5.21}$$

By taking into account 5.18 the MAP estimation problem, which is a formulation of the image segmentation problem, turns then into an optimization one. This optimization can be carried out by different approaches. Actually, the image segmentation algorithms based on Bayesian approach particularly vary according to two main aspects: The model types for model for the priors and their conditional probabilities, and the algorithm used in the optimization process. Several methods have been used for optimization such as dynamic programming **(Hansen and Elliot, 1982) (Derin and Elliot, 1987)**, regularization **(Li, 1995) (Daily, 1992)**, stochastic relaxation (simulated annealing) **(Geman and Geman, 1984)**, deterministic relaxation **(Dubes and Jain, 1982) (Memin et. al., 1994) (Perez, 1998)**.

The major handicap of these optimization algorithms is their requirement for big computation power. Since energy functions used in the segmentation optimization could be decomposed into local interactions using MRF models as described above. The parallelism that is inherent in artificial neural networks can be exploited to the purpose for implement the deterministic relaxation. For instance, energy functions used in deterministic relaxation can be easily mapped to a neural network architecture. Interconnections between the neurons can be arranged to use the Gibbs-Markov equivalence property. Based on the local characteristic of the MRF model, a neuron is only connected to its neighbors. This simplifies extremely the overall structure of the network. The other advantage of using ANN in computer vision area such as image restoration, computing optical flow and stereo matching is that their inherent parallel structure enables them to implement real-time application without any performance degradation.

For example, **Manjunath et.al. (1990)** proposed a Hopfield type ANN to implement deterministic relaxation-based texture segmentation. They used image intensity and texture label process based on Gauss Markov Random Field (GRMF) to express the MAP formulation. Then they mapped the energy function that should be minimized into a Hopfield type energy function.

Another Hopfield type approache to implement MAP minimization is the Deterministic Relaxation Neural Network (DRNN) proposed by **Raghu et.al. (1997)**. They use DRNN as classification of texture. To reach this goal, they proposed three processes called feature quantization, partition and competition processes. The posterior probabilities are derived from these processes by using Gibbs distributions. The corresponding energy is then mapped to a Hopfield type energy model. Finally, a deterministic relaxation scheme is applied to obtain the MAP solution.

**Chen and Chen (1998)** proposed a modified Cellular Neural Network (CNN) architecture for MAP estimation task. They combined the adaptive K-means clustering algorithm, which is developed by **Pappas (1992)**, and the GRF approach. The MAP estimation problem can be carried out by using a combined method based on adaptive K-Means algorithm and Iterated Conditional Modes (ICM). They use a modified CNN architecture for this MAP estimation task.

There are also plenty of studies on the Bayesian image segmentation topic; such as, multigrid MRF for MAP estimation **(Memin et.el., 1994) (Zhang and Wang, 1998)**, various methods to select a model for texture (**Kervrann and Heitz, 1993) (Zhu et.al., 1998)**, MAP relaxation of region adjacency graph **(Kim and Yang, 1994)**, Bayes smoothing based segmentation **(Derin et.al., 1984)**, various MAP minimization methods **(Meier et.al., 1997)**, various criterion for segmentation such as Maximization of Posterior Marginals and Expectation-Maximization **(Comer and Delp, 1996)**. Examination of these topics, however, is beyond this thesis. In the following section, we will point out the analogy between the CSNN and the MRF approaches and we will show how the performance of the CSNN based segmentation can be improved by using an MRF model.

### 5.8.2 Generalization of CSNN to MRF Models: MRF-CSNN

It was stated in Section 5.2.3 that CSNN performs formal probabilistic relaxation based segmentation. On the other hand, a MAP estimation of the Bayesian segmentation can be solved by using stochastic or deterministic relaxation will be depicted in this section. The close relationship between CSNN architecture and

MRF based segmentation will be brought into light and the CSNN architecture will be adapted to perform MRF based segmentation. The benefit of this approach over the CSNN itself is the re-introduction of data fitting term which improves the performance of the original CSNN in **Lin et.al (1992)**.

The energy term that is minimized in the MAP estimation depicted in 5.20 consists of the clique potential over the entire image:

$$U(l)=\sum_{c\in C}V_c(l) \tag{5.22}$$

The right hand side of 5.22 can be decomposed into potentials of different sized cliques:

$$U(l)=\sum_{(i,j)\in C_1}V_{C_1}(l_{ij})+\sum_{\substack{(i,j),\\(k,l)\in C_2}}V_{C_2}(l_{ij},l_{kl})+\sum_{\substack{(i,j),(k,l),\\(m,n)\in C_3}}V_{C_3}(l_{ij},l_{kl},l_{mn})+\ldots\ldots. \tag{5.23}$$

Each term in Equation 5.23 refers to a clique of a certain size. The first term denotes the cliques that have only one pixel, the second one corresponds to the pair-site cliques, the third one represents the cliques containing three sites, and so on. In most cases only up to second-order is taken into account because of its adequacy and simplicity:

$$U^{(2)}(x)=\sum_{(i,j)\in C_1}V_{C_1}(x_{ij})+\sum_{\substack{(i,j),\\(k,l)\in C_2}}V_{C_2}(x_{ij},x_{kl}) \tag{5.24}$$

Then the energy of the system, U, is divided into two different terms. The first term can be ascribed as data dependency term since it involves only the current pixel. The second one can also be interpreted as neighborhood term since it reflects the pair-site interaction in the neighborhood. It can be stemmed from two different processes: Data dependency process and neighborhood process. These two processes correspond to the data closeness and to the label smoothness terms defined in Equation 5.21.

With the data fitting term, one minimizes the expected difference between the observed data and the mean value of the segment to which belongs. The stationary probability distribution of the realization, which is denoted by w, can be assumed as a Gibbs distribution given in Equation 5.25 **(Raghu et.al., 1997)**:

$$P_D(w)=\frac{1}{Z}\exp\{-U_D(w)\} \tag{5.25}$$

where $U_D(w) = \sum_{m=0}^{M-1}\sum_{y \in R_l} d(y, \mu_l)$ is the average distance between data and the centroids of the segments and Z is the normalizing constant so that $\sum_{w \in \Omega} P_D(w) = 1$. m, $R_m$ and $\mu_m$ represent the segment labels of the $m^{th}$ segment and the mean value of the data belonging to $R_m$, respectively. The smaller this average distance, the higher probability of the realization, hence the better the segmentation.

The neighborhood term that models the interaction between neighbor pixels is generally used to impose the label smoothness. This energy term is proportional to the probability of a label assigned to a pixel site, $l_{ij}$, in relation to the labels of the neighbor sites. It can also be described as a Gibbs distribution as follows:

$$\begin{aligned} P_S(w) &= \frac{1}{Z}\exp\{-U_s(w)\} \\ &= \frac{1}{Z}\exp\{-U_s(x_{ij} \mid x_{kl}, (k,l) \in N_{ij})\} \end{aligned} \qquad 5.26$$

$U_s$ , which is called neighborhood contribution, is the energy function such that when it is small, then that pixel site has high probability whenever the label of this site is similar to those of the neighborhood pixels. This can be achieved with different energy functions like those **(Raghu et.al., 1997) (Li, 1995)**. The label interaction term in 5.26 can be expressed one of the following ways:

$$U_S = -\sum_{(k,l) \in N_{ij}} \beta\,\delta(x_{ij} - x_{kl}) \qquad \delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise,} \end{cases} \qquad (5.27)$$

$$U_S = -\sum_{(k,l) \in N_{ij}} \beta\, x_{ij} x_{kl} \qquad (5.28)$$

$$U_S = -\sum_{(k,l) \in N_{ij}} \beta(x_{ij} - \mu_{x_{ij}})(x_{kl} - \mu_{x_{kl}}) \qquad (5.29)$$

$$U_S = \begin{cases} \beta & \text{if all pixels on the neighborhood are similar} \\ -\beta & \text{otherwise} \end{cases} \qquad (5.30)$$

where $\beta$ is a positive constant, $\delta$ is the Kronecker delta function, $\mu_x$ denotes the mean value of the pixels having label x.

On the other hand, the system energy, U, depicted in 5.24 can be decomposed into the label levels **(Raghu et. al., 1997)**. In this case, the MRF structure is represented by a neural network structure such as CSNN or DRNN. This type of ANN structures have size of NxNxM. NxN denotes the image lattice while M denotes the label fields. For example, the energy function of DRNN at a pixel site is given at equation 5.31.

$$U_{ijl}^{t} = \sum_{(i',j'),l' \in N_{ij}} W_{ij,i'j',l,l'} O_{i'j'l'} - d(y_{ij}, \mu_l) \tag{5.31}$$

The first term of the right hand side of 5.31 corresponds the neighborhood contribution whereas the second one coincides the data dependency term. This energy term is the dual of 5.24. This duality stems from the definition of DRNN **(Raghu et. al., 1997)**. In DRNN scheme, the energy functions belonging to each layer at the each pixel site, $U_{(i,j),.}$ are calculated and the maximum one is chosen as the winner neuron in the pixel site under consideration. The output of the winner neuron is set to 1 while those of the others are set to 0. Thus, DRNN minimizes the energy by means of discrete relaxation. 5.31 can be easily changed like that 5.3. Then the neuron that has the minimum energy value is chosen as the winner one. $Q_{(i',j'),l'}$ and $W_{(i,j),l,(i',j'),l'}$ denotes the outputs of the neuron in the neighborhood and connection weights between the pixel under consideration and the pixel located in the neighborhood, respectively. The neighborhood contribution term is derived from 5.26 by means of Hopfield type ANN by choosing the symmetric connection weights, which satisfies the Hopfield convergence condition.

The structure of DRNN is very similar to that of CSNN. There exist, however, some differences between the two approaches. First of all there is no data dependency term in the CSNN. Second, the outputs of the label layer neurons at each pixel are updated in the CSNN according to equations from 5.3 to 5.6. Thus, a label value is changed step by step in CSNN whereas it is changed in quantum jumps in DRNN. In DRNN the label of a pixel is changed radically from one label to the other at each iteration. But in CSNN the label changes to the other one gradually. This is achieved by using probabilistic relaxation scheme in CSNN. Therefore the changes in label in CSNN is more reliable than DRNN **(Kurugöllü and Sankur, 1999a)**.

The MRF approach can be adapted to CSNN with two easy steps. First, $H_{ijk}$ term is chosen as the neighborhood contribution term. Although $H_{ijk}$ term is very similar to the neighborhood contribution term, the energy minimization scheme of this term must be proven because the energy minimization property of DRNN comes directly from the Hopfield convergence condition. There is no direct interpretation about minimization energy property of CSNN in **Lin et. al. (1992)**. Convergence proof of CSNN has been proved by changing interpretation of CSNN formulation given from 5.32 to 5.35. This proof can be found in Appendix C.

The second step to MRF representation is adding a data dependency term. An auto-normal model with unity variance is adopted for this term like DRNN **(Li, 1995) (Raghu et. al., 1997)**. For this aim the CSNN formulation can be modified as follows:

$$W'_{ij,qr,k,l} = -W_{ij,qr,k,l} \tag{5.32}$$

$$H^t_{ijk} = \sum_{(q,r),l\in N_{(i,j)}} W'_{ij,qr,k,l} O^t_{qrl} = \sum_{l=1}^{m}\sum_{r=1}^{p} W'_{kl} O^t_r \tag{5.33}$$

$$\Delta O^t_{ijk} = \begin{cases} \delta & \text{if} H^t_{ijk} = \underset{l}{\text{minimum}}\left(H^t_{ijl}\right) \\ -\delta & \text{otherwise} \end{cases} \tag{5.34}$$

$$O^{t+1}_{ijk} = \frac{Pos(O^t_{ijk} + \Delta O^t_{ijk})}{\sum_{i=1}^{m} Pos(O^t_{ijk} + \Delta O^t_{ijk})} \tag{5.35}$$

In this new formulation, signals of the connection weights are changed. But under this change the neuron that has the minimum neighborhood contribution is chosen as the winner rather than the one having the maximum contribution. The new update scheme is depicted in Figure 5.26.

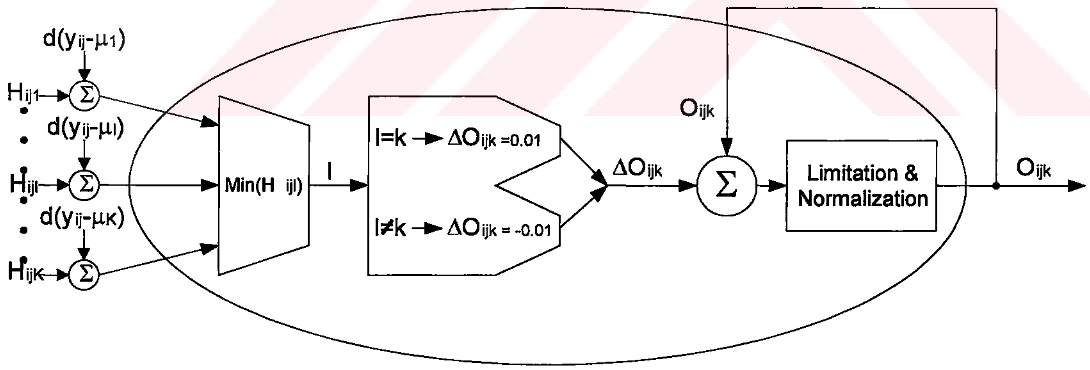



Figure 5.26. Update scheme of MRF-CSNN. Each neuron receives a signal from data dependency process as well as neighborhood contribution process.

The energy of a neuron vis-à-vis the k[th] segment decision can also be written as follows:

$$U^t_{(i,j),k} = d(y_{(i,j)},\mu_k) + H^t_{(i,j),k} = d(y_{(i,j)},\mu_k) - \sum_{l=1}^{m}\sum_{r=1}^{p} W_{k,l} O^t_r \tag{5.36}$$

The total energy function of a pixel site can be chosen as summation of all neurons that represents this site:

$$U_{ij} = \sum_{k=1}^{M} d(y_{ij}, \mu_k) + \sum_{k=1}^{M} H_{ijk} \tag{5.37}$$

If any regular configuration of pixel labels yields decreasing in this energy while the irregular one increases that, this energy could be used as Gibbsian energy in the segmentation process. Since the structure of the ANN reflects the Markovianity condition overall process would become a MRF based segmentation.

The energy given in Equation 5.37 consists of aforementioned data dependency and neighborhood terms. It is straightforward that any regular configuration decreases the data dependency term. In this case, distance between the data and its center would become minimum. This causes decreasing in the energy. On the other hand, any irregular configuration causes increasing the energy. Thus this would be a penalty to the configuration.

For the neighborhood contribution or smoothness term in Equation 5.37, any regular configuration causes a neuron to receive the signals of the neurons that reside close segment layers. In this case, output of these neighbor neurons has a large positive value while the others have a small positive value or no value (e.g., 0). Since the connection weights, $W'_{kl}$, have large negative values for close neighbors and have

```
MRF-CSNN Algorithm:

Cluster the given image by using a fuzzy c-means clustering according to a determined
number of clusters.
Initiate the neurons denoting the pixel labels according to the result of the clustering.
Repeat until (the convergence criterion satisfied)
        Determine the mean value of the label fields, μk .
        For (each pixel)
                For (each label, k )
                        Calculate energy of the neuron, Uijk
                Endfor
                Select minumum Uijl with the argument l.
                For (each label, k)
                        If ( k = l) then
                                ΔOijk=δ
                        Else
                                ΔOijk=-δ
                        Endif
                        Update the output of the neuron according to 5.36
                Endfor
        Endfor
Endrepeat
```

Figure 5.27. The flowchart of the MRF-CSNN algorithm

large positive value for far neighbors, neighborhood contribution, $H_{ijk}$, becomes negative. Then the regular configuration causes decreasing in the energy. On the contrary, if the pixel site is not compatible its neighbors far neighbors have positive contribution with positive connection weights. Thus the energy increases in this situation. This penalties the configuration.

The proposed MRF-CSNN algorithm is summarized in Figure 5.27.

The result of MRF-CSNN on real world images can be seen in Figure 5.28. The segments in the MRF-CSNN results are more plausible than those that CSNN yields, presumably due to the added data dependency term. In Figure 5.29 the convergence error versus the iteration number are compared for both the CSNN and the MRF-CSNN algorithms. The performance of the MRF-CSNN is slightly better than the CSNN. Further quantitative figures are given in the following section.

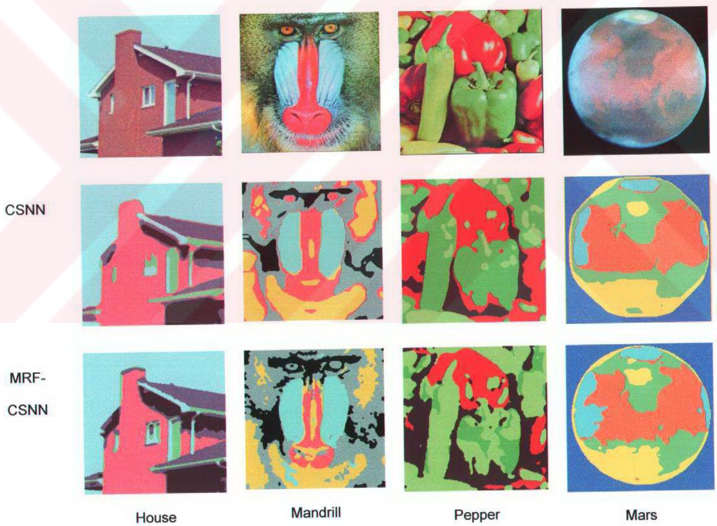



Figure 5.28. Results of real world images obtained from CSNN and MRF-CSNN methods

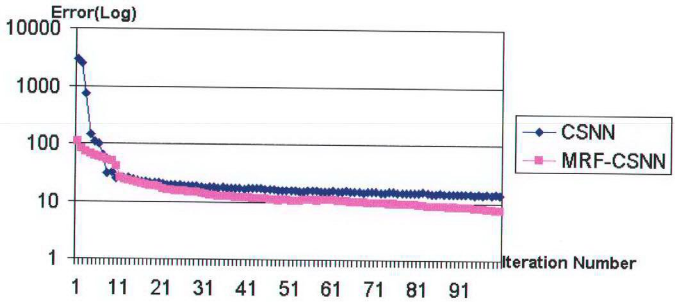



Figure 5.29. Convergence error of MRF-CSNN. The convergence performance is MRF-CSNN is slightly better than that of CSNN.

### 5.9 Quantitative Results

The performance of the proposed CSNN methods except for pyramidal CSNN have been compared with that of the original CSNN algorithm by using the evaluation criterion, $M_p$ in Chapter 3. For this aim, the algorithms are run on 200 synthetic test images and the corresponding $M_p$ scores have been calculated. Since the results of the pyramidal CSNN depends highly on connection weights between pyramid levels and these weights are determined in terms of the given application, the quantitative evaluation of pyramidal CSNN without any specific application is not plausible. Thus it is not considered in the quantitative assessment.

#### 5.9.1 B-CSNN versus CSNN

The performance of B-CSNN algorithm proposed in Section 5.5 has been found to be better than that of CSNN for 137 test images out of 200 images. Thus the performance advantage of B-CSNN is 68%. The detailed test scores are given in Table 5.8 and Table 5.9. Table 5.8 shows the performance comparison as a function of the number of segments while Table 5.9 gives the scores as a function of the color spread categories. In each table the second and third columns indicate the number of cases when the $M_p$ score was found superior with respect to the other method. The fourth column shows the percentage of cases when B-CSNN scored higher that CSNN. Figure 5.30 depicts the histogram of $M_p$ for the both methods. One can notice that the mean value of $M_p$ for B-CSNN is lower than that of CSNN.

Table 5.8. Performance of B-CSNN according to number of segments

| Number of Segments | B-CSNN score | CSNN score | Percentage B-CSNN is better than CSNN |
|---|---|---|---|
| 2 | 29 | 11 | 72% |
| 3 | 36 | 4 | 90% |
| 4 | 28 | 12 | 70% |
| 5 | 24 | 16 | 60% |
| 6 | 20 | 20 | 50% |
| **Total** | **137** | **63** | **68%** |

Table 5.9.Performance of B-CSNN according to color spread

| Category | B-CSNN score | CSNN score | Percentage B-CSNN is better than CSNN |
|---|---|---|---|
| 1 | 39 | 11 | 78% |
| 2 | 34 | 16 | 68% |
| 3 | 34 | 16 | 68% |
| 4 | 30 | 20 | 60% |
| **Total** | **137** | **63** | **68%** |

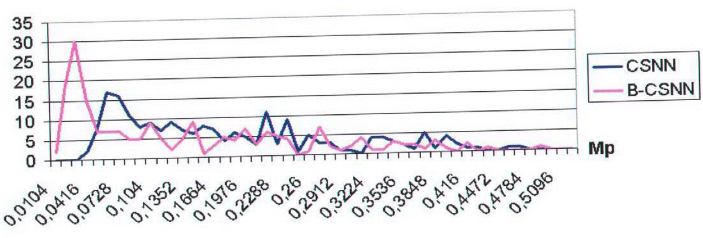



Figure 5.30. Histograms of $M_p$ for B-CSNN and CSNN. Mean value of B-CSNN is lower than that of CSNN.

### 5.9.2 MS-CSNN versus CSNN

The performance of MS-CSNN has been found as 72%. MS-CSNN is better than CSNN for 144 images out of 200 test images. Details are given in Table 5.10 and 5.11 that have same interpretation as Table 5.8 and Table 5.9 respectively. Histograms of $M_p$ for both methods are also illustrated in Figure 5.31. Mean value of MS-CSNN is definitely lower than that of CSNN. If one compare B-CSNN and MS-CSNN it is realized that performance of MS-CSNN is better than that of B-CSNN.

Table 5.10. Performance of MS-CSNN according to number of segments

| Number of Segments | MS-CSNN score | CSNN score | Percentage MS-CSNN is better than CSNN |
|---|---|---|---|
| 2 | 37 | 3 | 92% |
| 3 | 29 | 11 | 72% |
| 4 | 23 | 17 | 57% |
| 5 | 27 | 13 | 67% |
| 6 | 28 | 12 | 70% |
| **Total** | **144** | **56** | **72%** |

Table 5.11. Performance of MS-CSNN according the color spread

| Category | MS-CSNN score | CSNN score | Percentage MS-CSNN is better than CSNN |
|---|---|---|---|
| 1 | 38 | 12 | 76% |
| 2 | 36 | 14 | 72% |
| 3 | 37 | 13 | 74% |
| 4 | 33 | 17 | 66% |
| **Total** | **144** | **56** | **72%** |

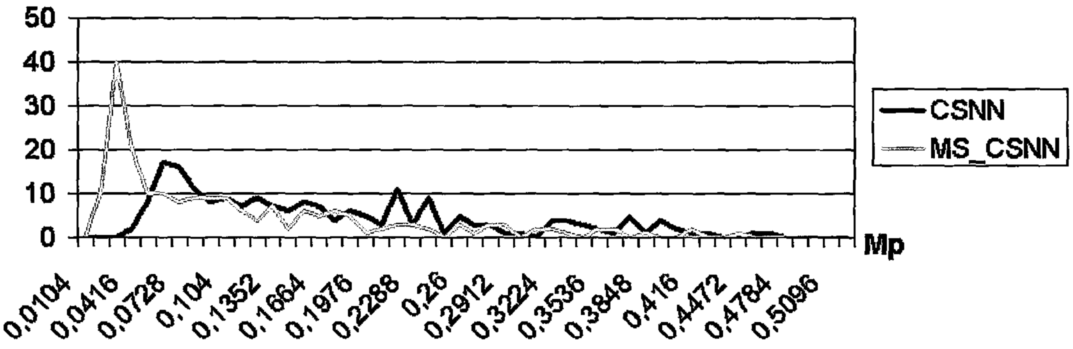



Figure 5.31. Histograms of $M_p$ for MS-CSNN and CSNN. Mean value of MS-CSNN is lower than that of CSNN.

### 5.9.3 MRF-CSNN versus CSNN

The performance of MRF-CSNN is only slightly better than that of CSNN and the advantage is not as impressive as those for B-CSNN and MS-CSNN. The performance of MRF-CSNN was worse than that of CSNN when the difficulty or color spread of the test images is high. On the other hand, the algorithm gives much better results than CSNN when the color spread is low. This fact must be due to the presence of that data dependency term. While the data dependency term prevents an oversmoothed segmentation result, it also reduces noise. If the given image is very noisy or color spread is high, the noise may not be eliminated by the relaxation algorithm. Thus, for example, small holes in an actual segment can be formed which in turn the performance to drop. On the contrary, in the low noise case, data dependency term helps to corroborate the correct segment decision. Quantitative results are given in Table 5.11 and 5.12 in the same manner as Table 5.8 and Table 5.9, respectively. The histograms of the $M_p$ values are also given in Figure 5.32. The histogram for MRF-CSNN has two peaks, one is near the 0 and the other is near the 0.35 because of aforementioned noise sensivity of the algorithm.

Table 5.12. Performance of MRF-CSNN according to number of segments

| Number of Segments | MRF-CSNN score | CSNN score | Percentage MRF-CSNN is better than CSNN |
|---|---|---|---|
| 2 | 28 | 12 | 70% |
| 3 | 29 | 11 | 72% |
| 4 | 24 | 16 | 60% |
| 5 | 23 | 17 | 57% |
| 6 | 17 | 23 | 42% |
| **Total** | **121** | **79** | **60%** |

Table 5.13. Performance of MRF-CSNN according the color spread

| Category | MRF-CSNN score | CSNN score | Percentage MRF-CSNN is better than CSNN |
|---|---|---|---|
| 1 | 44 | 6 | 88% |
| 2 | 41 | 9 | 82% |
| 3 | 24 | 26 | 48% |
| 4 | 12 | 38 | 24% |
| **Total** | **121** | **79** | **60%** |

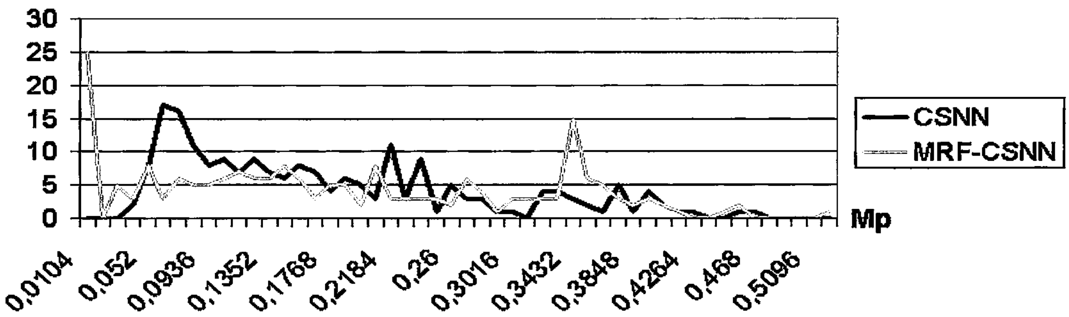



Figure 5.32. Histograms of $M_p$ for MRF-CSNN and CSNN. It must be noticed that histogram of MRF-CSNN has two peaks.

### 5.10 Conclusion

In this chapter CSNN-based image segmentation has been explored. Its benefits and weaknesses have been presented. They can be summarized as follows:

CSNN provides an efficient and simple ANN architecture for segmentation combining the clustering operation in measurement space and the relaxation method in image spatial domain. Behavior of the algorithm can be adjusted via connection weight and neighborhood size according to applications without changing the network structure. The local interconnection nature of CSNN makes its adaptation to MRF problems easy.

We had observed that CSNN as proposed in **Lin et.al (1992)** either had some shortcomings or that it could be improved or made more flexible. The observed problems and proposed solutions are summarized as follows:

1. The algorithm necessitated the number of segments as an input information. This number must either be known as part of the domain knowledge or must be determined automatically from the given image. A solution has been provided in Section 5.5

2. CSNN uses only gray level information of the given image. The algorithm has been generalized to accommodate multiband images.

3. In the initialization step, the given image was clustered by using the SOM algorithm, then the crisp result of the SOM was fuzzified by an ad-hoc technique. We have adapted to fuzzy c–means clustering algorithm instead of Kohonen's SOM. Thus, the ad hoc fuzzification step was avoided by using directly the fuzzy c-means algorithm for initialization of the pixel label.

4. After a rapid stage of adjustment, CSNN is tended to have a very slow convergence. This convergence has been accelerated in the B-CSNN, MS-CSNN and P-CSNN algorithms significantly.

5. CSNN could generate under-segmented results since some small segments were captured by big ones. Thus segments were not always constituted properly.

6. Its convergence around the boundaries was not very satisfactory. In the B-CSNN method, boundary constraints obtained from the edge map

imposed as spatial constraints. These additional constraints eliminated the noisy segment boundaries and the convergence rate became much better. This problem was also solved by multiresolution schemes. For example, both the MS-CSNN scheme and P-CSNN scheme improved the segmentation fidelity, boundary smoothness and convergence rate. In addition P-CSNN gave an additional tool for controlling the support of each label and its interaction with lower and higher resolution maps.

Finally, CSNN has been adapted to the solution of MRF-based segmentation.

# 6. CONCLUSIONS

## 6.1 General Conclusions

This thesis has advanced novel methods for general purpose segmentation and has addressed the problem of measurement of segmentation. Special emphasis is given to multispectral images, e.g., color images.

A taxonomy of image segmentation methods was introduced in Chapter 2. Such a taxonomy is an aid in situating and associating novel proposed algorithms with respect to old ones on one hand; on the other hand it may reveal where and how there is an opportunity to advance a new method. We have observed that hybrid techniques that combines measurement space-based, region-base and boundary-based paradigms are promising.

In Chapter 3, a novel segmentation evaluation method resulting from Minkowski average of measures of classification errors and shape deformation has been proposed. Classification error measure is a performance indicator of correct assignment of pixels to respective segments. It, however, is not sufficiently sensitive to deformations of the shape of objects. On the other hand, turning angle function is sensitive to shape deformation of objects while there are cases it does not reflect adequately classification errors. Thus it is conjectured that a combination of these measures could give a superior evaluation criterion for image segmentation. This capability of the proposed measure has been shown over a number of test cases. Furthermore, a test image generation scheme has been described that provides a rich variety of shape occlusions and a controlled amount of contrast and color spread. Both the segmentation performance measure and test image generation scheme can be used in the experiments for comparative assessment of general segmentation methods. The contributions of this chapter are as follows:

1. A new index based on turning angle function that indicates shape deformation of the objects has been developed.

2. A novel criteria that combines classification error measure and shape deformation index has been proposed.

3. A new test image generation system that can be used for quantitative evaluation of segmentation methods has been defined.

The multispectral image segmentation method proposed in Chapter 4 has explored segmentation via multithresholding of lower dimensional (marginal) histograms. The proposed technique is based on 2D-histogram multithresholding and fusion of segmentation maps. The proposed technique has been compared with its 1D-histogram and 3D-histogram counterparts. As expected, segmentations (multithresholding and fusion) with N dimensions are better than the results with (N-1) dimensions; though at a higher cost. More specifically the results of 3D case are slightly better than those of 2D case as expected. However, the 3D algorithmic calculation is much higher (by a factor of 180) than that of the 2D-histogram based method. On the other hand, the performance of 2D-histogram based method is considerably better than that of 1D case with only about twice its computational complexity. Thus it is concluded that the proposed 2D-histogram based method provides a tool that has better performance than 1D case and is more efficient than 3D case. The contributions of this chapter can be summarized as follows:

1. A 2D-histogram based multithresholding method for color images has been developed.

2. A fusion algorithm based on label concordance and spatial-chromatic majority filtering has been proposed to combine segmentation results of color band pairs. This method is applicable and generalizable to any N-band multispectral image. It provides a practical segmentation tool that is particularly effective when the multispectral feature vectors are well clustered in the N-dimensional space.

In Chapter 5, constraint satisfaction neural network based segmentation method has been examined. In essence this is a hybrid method where a segmentation maps is initialized via measurement based clustering followed by relaxation. The role of CSNN is to instrument the relaxation with controlled support and interconnection weights. Although CSNN provided an interesting tool for image segmentation, it had some disadvantages. First, the number of segments was to be given as a priori information and the pixel labels are initialized based on a rather ad hoc fuzzification method. Secondly, CSNN had a long convergence time due to futile iterations on object boundaries, yielding often oversegmented results. To overcome these problems some modifications have been carried out. First a cluster validity scheme has been added to the initialization step to estimate the number of segments from a given image. Label initialization process modified by using fuzzy c-means clustering instead of Kohonen's SOM algorithm. Since fuzzy c-means algorithm already gives fuzzy decisions about clustered data there is no need for any ad hoc fuzzification. To improve the convergence time and avoid oversmoothed results, the architecture of CSNN has been modified in four different ways. Our main contributions to CSNN can be listed as follows:

1. CSNN architecture has been generalized to accommodate multispectral data, e.g., color images.
2. The performance of CSNN has been improved by using fuzzy c-means clustering instead of Kohonen's self organizing maps in the initialization stage.
3. Boundary based CSNN algorithm that incorporates boundary constraints has been shown to yield smoother and more precise boundaries.
4. Multi-scan based CSNN and Pyramidal CSNN methods are presented as two multiresolution realizations of CSNN that resulted in both improved convergence and controllable segmentation resolution.
5. MRF-CSNN method has been proposed as an alternative method to solve the MRF estimation problem.

In summary, the CSNN based segmentation method with its proposed improvements provides an attractive tool for segmenting multispectral images. The advantages lie in the ability to introduce easily various constraints under the user control such as degree of smoothness, the location of known edges, label relations over a neighborhood etc.

## 6.2 Perspectives for Future Works

- Evaluation of segmentation performance

Other intrinsic criteria such as region smoothness and contrast can be incorporated in the Minkowski average in Equation 3.9, along with extrinsic measures such as classification error and boundary deformation. A better definition of TAF based on higher order polynomial fitting remains to be worked out.

- Multithresholding Segmentation

Performance of 2D-histogram based multithresholding method maybe improved by using various other decision fusion methods. The fusion methods to be considered could be parametric, e.g., Bayesian, or non-parametric such as logistic regression and nearest neighbor. Fuzzy logic is another fusion scheme.

Another avenue of research would be to determine the "complementarity of band pairs". In other words given N bands, a method to select K-tuple of bands in such a way that the $\binom{N}{K}$ combinations are the most complementary to each other. Investigation of suitable color coordinates beyond RGB is one such case in this point.

For multispectral images, combinations of higher dimensional histograms should be examined. In this case, label concordance algorithm should be refined to handle these new circumstances.

- CSNN Segmentation

The constraints of CSNN method that are represented by connection weights remain unchanged during the process. This constraints maybe updated according to the current segmentation. Some local information can be used for this purpose. Constraints between the neighbor neurons belonging to a uniform segment should be relaxed. On the contrary, if the neighbor neurons belong to different segments the constraint between these neurones should be strengthened.

The adjustment of neighborhood support and multiresolution interconnections should be done semi-automatically as a function of the noise level. For example SAR images, which are corrupted by heavy multiplicative noise remains a testbed to prove the merits of the CSNN algorithms.

# REFERENCES

**Abdou, I.E., Pratt, W.K.**, 1979. Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors. *Proceedings of IEEE*, **67**, 753-763.

**Adams, R., Bischof, L.**, 1994. Seeded Region Growing. *IEEE Trans. on PAMI*, **16**, 641-647.

**Arkin, E.M., Chew, L.P., Huttenlocker, D.P., Kedem, K., Mitchell, J.S.B.,** 1991. An Efficient Computable Metric for Comparing Polygonal Shapes. IEEE Tran. on PAMI, **13**, 209-215.

**Beuker, S.**, 1999. http://cmm.ensmp.fr/~beuker/wtshed.html

**Borotschnig, H., Pinz, A., Bloch, I.**, 1998. Fuzzy Relaxation Labeling Reconsidered, *Proceedings of IEEE World Congress on Computational Intelligence (FUZZ-IEEE 1998)*, Anchorage, Alaska, USA, May 1998, 1417-1423.

**Burt, P.J., Hong, T.H., Rosenfeld, A.**, 1981. Segmentation and Estimation of Image Region Properties through Cooperative Hierarchical Computation, *IEEE Trans. On SMC*, **11**, 802-809.

**Canny, J.F.,** 1986. A Computational Approach to Edge Detection. *IEEE Trans. On PAMI*, **8**, 679-698.

**Chen, C.W., Chen, L.**, 1998. Cellular Neural Network Architecture for Gibbs Random-Field-Based Image Segmentation, *Journal of Electronic Imaging*, **7**, 45-51.

**Cheng, H.D., Chen, Y.H.**, 1999. Fuzzy Partition of Two-dimensional Histogram and Its Application to Thresholding, *Pattern Recognition*, **32**, 825-843.

**Chu, C.C., Aggarwal, J.K.**, 1993. The Integration of Image Segmentation Maps Using Region and Edge Information. *IEEE Trans. on PAMI*, **15**, 1241-1252.

**Coleman, G.B., Andrews, H.C.**, 1979. Image Segmentation by Clustering. *Proceedings of IEEE*, **67**, 773-785.

**Comer, M.L, Delp, E.**, 1996. The EM/MPM Algorithm for Segmentation of Textures Images: Analysis and Further Experimental Results, *Proceedings of the International Conference on Image Processing (ICIP'96)*, vol. 3, Laussane, Switzerland, October 1997, 947-950.

**Çelenk, M., Haag, M.U.**, 1998. Optimal Thresholding for Color Images, *SPIE Conference on Nonlinear Image Processing IX*, San Jose, ABD, 1998.

**Daily, M.J.**, 1992 Color Image Segmentation, in *Advances in Image Analysis*, pp.131-161, Eds. Mandavieh, Y. and Gonzales, R.C., Society of Photo Optical, USA.

**Darwish, A.M., Jain, A.K.**,1988. A Rule Based Approach for Visual Pattern Inspection, *IEEE PAMI*, **10**, 56-68.

**Davies, D.L., Bouldin, D.W.**, 1979. A Cluster Separation Measure. *IEEE Trans. on PAMI*, **1**, 224-227.

**Davis, L.S., Rosenfeld, A.**, 1978. Noise Cleaning by Iterative Local Averaging, *IEEE Trans. on SMC*, **8**, 705-710.

**Demir, D., Birecik, S., Kurugöllü, F., Sezgin, M., Bucak, İ.Ö, Sankur, B., Anarım, E.**, 1994. Quality Inspection in PSBs and SMDs Using Computer Vision Techniques, *International Conference on Industrial Electronics Control and Instrumentation (IECON'94)*, vol:2, Bologna, ITALY, 5-9 September 1994, 857-861.

**Derin, H., Elliot, H., Cristi, R., Geman, D.**, 1984. Bayes Smoothing Algorithms for Segmentation of Binary Images Modeled by Markov Random Fields, *IEEE Trans. on PAMI*, **6**, 707-720.

**Derin, H., Elliot, H.**, 1987. Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields, *IEEE Trans. on PAMI*, **9**, 39-55.

**Dubes, R.C., Jain, A.K.**, 1977. Models and Methods in Cluster Validity, Technical Report, **No. TR-77-05**, Michigan State University, College of Engineering, Department of Computer Science, Michigan, USA.

**Dubes, R.C., Jain, A.K.**, 1989. Random Field Models in Image Analysis, *Journal of Applied Statistics*, **16**, 131-162.

**Eklundh, J.O., Yamamoto, H., Rosenfeld, A.**, 1980. A Relaxation Method in Multispectral Pixel Classification, *IEEE Trans. on PAMI*, **2**, 72-75.

**Eugene, C.F., Wallance, R.J.**, 1992. Partial Constraint Satisfaction, *Artificial Intelligence*, **58**, 21-70.

**Fu, K.S., Mui, J.K.**, 1981. A Survey on Image Segmentation, *Pattern Recognition*, **13**, 3-16.

**Geman, S., Geman, D.**, 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, *IEEE Trans. on PAMI*, **6**, 721-741.

**Gevers, T., Kajcovski V. K.**, 1994. Image Segmentation by Directed Region Subdivision, Proceedings of the 12$^{th}$ IAPR International Conference on Pattern Recognition, Jerusalem, ISRAEL, October 1994, 342-346.

**Gonzalez, R.C., Wintz, P.**, 1987. Digital Image Processing. Addision-Wesley Pub., Reading, Massachusetts, USA.

**Gökmen, M., Li, C.C.**, 1992. Multiscale Edge Detection Using First Order R-filter. Proceedings of IEEE Conference on Pattern Recognition, 307-310, The Hague, The Netherlands.

**Gökmen, M., Jain, A.K.**, 1997. $\lambda$-$\tau$ Space Representation of Images and Generalized Edge Detection, *IEEE Trans. on PAMI*, **19**, 1997, 545-563.

**Grimm, F., Bunke, H.**, 1993. An Expert System for the Selection and Application of Image Processing Subroutines, *Expert Systems*, **10**, 61-74.

**Gurari, E. M., Wechsler, H.**, 1982. On the Difficulties Involved in the Segmentation of Pictures, *IEEE PAMI*, **4**, 304-306.

**Hansen, F.R., Elliot, H.**, 1982. Image Segmentation Using Simple Markov Field Models, *Computer Graphics and Image Processing*, **20**, 101-132.

**Haralick, R.M, Kelly, G.L.**, 1969. Pattern Recognition with Measurement space and Spatial Clustering for Multiple Images, Proc. IEEE, **57**, 654-665.

**Haralick, R.M, Shapiro, L.G.**, 1985. Survey, Image Segmentation Techniques, *Computer Vision, Graphics and Image Processing*, **29**, 100-132.

**Haralick, R.M., Shapiro, L.G.**, 1993. Computer and Robot Vision. Addison-Wesley Pub., Reading, Massachusetts, USA:

**Healey, G.**,1992. Segmenting Images Using Normalized Color. *IEEE Trans. on SMC,* **22**, 64-73.

**Healey, G.**, 1996. Modeling Color Images for Machine Vision. Advances in Image Processing and Machine Vision, Springer-Verlag, Ed. J. Sanz 129-146.

**Hummel, R.A., Zucker, S.W.**, 1983. On the Foundations of Relaxation Labeling Process, *IEEE Trans. on PAMI*, **5**, 267-287.

**Kanade, T.**, 1980. Region Segmentation: Signal vs. Semantics, *Computer Graphics and Image Processing*, **13**, 279-297.

**Kass, M., Witkin, A., Terzopoulos, D.**, 1987. Snakes: Active Contour Models, *International Journal of Computer Vision*, **1**, 312-331.

**Kervrann, C., Heitz, F.**, 1993. A Markov Random Field Model-Based Approach to Unsupervised Texture Segmentation using Local and Global Spatial Statistics, Publication Interne, **No. 752**, Institut de Receherche en Informatique et Systemes Aleatories, Cedex, France.

**Khotanzad, A., Bourfa, A.**, 1990. Image Segmentation by a Parallel, Non-parametric Histogram Based Clustering Algorithm, *Pattern Recognition*, **23**, 961-973, 1990.

**Kim, I.Y., Yang, H.S.**, 1994. A Systematic Way for Region-Based Image Segmentation Based on Markov Random Field Model, *Pattern Recognition Letters*, **15**, 969-976.

**Klemencic, A., Pernus, F., Kovacic, S.**, 1996. Segmentation of Muscle Fibre Images Using Voronoi Diagrams and Active Contour Models, Proceedings of ICPR'96, 538-542.

**Koonty, W., Narenda, P.M., Fukunya, F.**, 1976. A Graph Theoretic Approach to Non-parametric Cluster Analysis, *IEEE Trans. on Computer*, **25**, 936-944.

**Kumar, V.**, 1992. Algorithms for Constraint Satisfaction Problems. A Survey, *Artificial Intelligence Magazine*, **13**, 32-44.

**Kurugöllü, F., Birecik, S., Sezgin, M., Sankur, B,**, 1996. Image Segmentation based on Boundary Constraint Neural Network, *Proceedings of Third*

*International Workshop on Image and Signal Processing on the Theme of Advances in Computational Intelligence, Manchester, United Kingdom, 4-7*, November, 1996, 353-356.

**Kurugöllü, F., Sankur, B.**, 1997. Color Image Segmentation Based on Multithresholding and Fusion, *1997 IEEE International Conference on Image Processing - ICIP'97*, 26-29 October,1997 Santa Barbara, California, USA.

**Kurugöllü, F., Sankur, B,**, 1998a. Image Segmentation using Multi-Scan Constraint Satisfaction Neural Network, *Proceedings of 13th International Symposium on Computer and Information Sciences,* Antalya, Turkey, 26-28 October, 1998, 190-197.

**Kurugöllü, F., Sankur, B,**, 1998b. Color Cell Image Segmentation using Pyramidal Constraint Satisfaction Neural Network, *Proceedings of IAPR Workshop on Machine Vision Applications*, Makuhari-Chiba, Japan, 17-19, November, 1998, 85-88.

**Kurugöllü, F., Sankur, B,**, 1999a. MAP Segmentation of Color Images using Constraint Satisfaction Neural Network, IEEE *International Conference on Image Processing*, Kobe, Japan, 24-28, October, 1999.

**Kurugöllü, F., Sankur, B,**, 1999b. Image Segmentation Based on Multi-scan Constraint Satisfaction Neural Network, *Pattern Recognition Letter*, Accepted for publication.

**Kwan, A.C.M.**, 1997. A Framework for Mapping Constraint Satisfaction Problems to Solution Methods. Ph.D. Thesis, Department of Computer Science, University of Essex, Essex, UK.

**Lee, J.H.M., Tam, V.W.L.**, 1995. A Framework for Integration of Artificial Neural Networks and Constraint Logic Programming, *International Journal on Artificial Intelligence Tools*, **4**, 3-32.

**Levine, M.D., Nazif, A.M.**, 1985. Dynamic Measurement of Computer Generated Image Segmentations. *IEEE Trans. on PAMI*, **7**, 155-164.

**Li, S.Z.**, 1995. Markov Random Field Modeling in Computer Vision. Springer-Verlag, USA.

**Lim, J.S.**, 1990. Two-Dimensional Signal and Image Processing. Prentice Hall. New Jersey, USA.

**Lin, W.C., Tsao E.C. and Chen, C.T.**, 1992. Constraint Satisfaction Neural Networks for Image Segmentation, *Pattern Recognition*, **25**, 679-693.

**Lisin, D.**, 1999. http://www.wpi.edu/~dima/ummed/proposal/index.html.

**Lundervold, A., Strovik, G.**, 1995. Segmentation of Brain Parenchyma and Cerebrospinal Fluid in Multispectral Magnetic Resonance Images, *IEEE Trans. on Medical Imaging*, **14**, 339-349.

**Lückenhaus, M., Eckstein, W.**, 1997. A Multi-agent Based System for Parallel Image Processing. *Proceedings of SPIE*, **3166**, 31-40.

**Manjunath, B.S., Simchony, T., Chellappa, R.**, 1990. Stochastic and Deterministic Networks for Texture Segmentation, *IEEE Trans. on ASSP*, **38**, 1039-1049.

**Marr, D., Hildreth, E.**, 1980. Theory of Edge Detection. *Proceedings R. Soc. London Ser.*, **207**, 187-217.

**Memin, E., Heitz, F., Charot, F.**, 1994. Efficient Parallel Non-Linear Multigrid Relaxation Algorithms for Low-Level Vision Applications, Publication Interne, **No. 790**, Institut de Receherche en Informatique et Systemes Aleatories, Cedex, France.

**Nazif, A.M., Levine, M.D.**, 1984. Low-Level Image Segmentation: An Expert System, *IEEE Trans. on PAMI*, **6**, 555-577.

**Pal, N.R., Bhandari, D.**, 1993. Image Thresholding: Some New Techniques. *Signal Processing*, **33**, 139-158.

**Pal, N.R., Pal, S.K.**, 1993. A review on Image Segmentation Techniques, *Pattern Recognition*, **26**, 1277-1294.

**Pappas, T.**, 1992. An Adaptive Clustering Algorithm for Image Segmentation, *IEEE Trans. on Signal Processing*, **40**, 901-914.

**Pavlidis, T., Liow, Y.**, 1990. Integration Region Growing and Edge Detection, *IEEE Trans on PAMI*, **12**, 225-233.

**Perez, P.**, 1998. Markov Random Field and Images, Publication Intern, **No. 1196**, Institut de Receherche en Informatique et Systemes Aleatories, Cedex, France.

**Pratt, W.K.**, 1991. Digital Image Processing. John Wiley & Sons, Inc. New York, USA.

**Privault, C., Hérault, L.**, 1998. Constraint Satisfaction Through Recursive Neural Networks with Mixed Penalties: A Case Study, *Neural Procesing Letters*, **8**, 15-26.

**Qian, Y., Zhao, R.**, 1997. Image Segmentation Based on Combination of the Global and Local Information, *Proceedings of IEEE International Conference on Image Processing (ICIP'97)*, Santa Barbara, California, USA, October 1997, vol. 1, 204-207.

**Raghu, P.P., Poongodi, R., Yegnanarayana, B.**, 1997. Unsupervised Texture Classfification Using Vector Quantization and Deterministic Neural Network, *IEEE Trans. on Image Processing*, **6**, 1376-1387.

**Raya, S.P.**, 1990. Low-Level Segmentation 3-D Magnetic Resonance Brain Images – A Rule-Based System, *IEEE Trans. on Medical Imaging*, **9**, 327-337.

**Rezaee, M.R., Leieveldt, B.P.F., Reiber, J.H.C.**, 1998. A New Cluster Validity Index for the Fuzzy c-mean, *Pattern Recognition Letters*, **19**, 237-246.

**Rosenfeld, A., Hummel, R.A., Zucker, S.W.**, 1976. Scene Labeling by Relaxation Operations, *IEEE Trans. on SMC*, **6**, 420-433.

**Rosenfeld, A., Davis, L.S.**, 1979. Image Segmentation and Image Models, *Proc. of the IEEE*, **67**, 764-772.

**Rosenfeld, A., Smith, R.C.**, 1981. Thresholding Using Relaxation. *IEEE Trans. on PAMI*, **3**, 598-606.

**Rosenfeld, A.**, 1984. Computer Vision: Signals, Segments, and Structures *IEEE ASSP Magazine*, **2**,11-18.

**Russ, J.C.**, 1995. The Image Processing Handbook. CRC Press, Boca Raton, Florida, USA.

**Sahoo, P.K., Soltani, S., Wong, A.K.C., Chen, Y.C.**, 1988. A Survey of Thresholding Techniques. *Computer Vision, Graphics and Image Processing*, **41**, 233-260.

**Schachter, B.J., Lev, A., Zucker, S.W., Rosenfeld, A.**, 1977. An Application of Relaxation Methods to Edge Reinforcement, *IEEE Trans. on SMC*, **7**, 813-816.

**Sezgin, M., Kurugöllü, F., Birecik, S.**, 1997. An Application of Multilevel Thresholding Methods to Visual Cloth Quality Inspection, *International Conference on Quality Control by Artificial Vision - QCAV'97*, Le Creusot, France 28-30, May,1997, 281-286.

**Strasters, K., Gebrands, J.J.**, 1991. Three-dimensional Image Segmentation Using a Split, Merge and Group Approach, *Pattern Recognition Letters*, **12**, 307-325.

**Stuller, J., Shah, R.**, 1998. An Image Model Based on Occluding Object Images and maximum Entropy, *IEEE Trans. on Image Processing*, **7**, 1300-1310.

**Szeliski, R.**, 1989. Bayesian Modeling of Uncertainty in Low-level Vision. Kluwer Academic Pub. Massachusetts, USA.

**Therrien, C.W.**, 1989. Decision Estimation and Classification. John Wiley & Sons Inc. USA.

**Tsao, E.C.K., Lin, W.C.**, 1991. Constraint Propagation Neural Networks for Huffman-Clowes Scene Labeling, *IEEE Trans. on SMC*, **21**, 1536-1548.

**Tsao, E.C.K., Bezdek, J.C., Pal, .R.**, 1994. Fuzzy Kohonen Clustering Networks. *Pattern Recognition*, **27**, 757-764.

**Uchiyama, T, Arbib, M.A.**, 1994. Color Image Segmentation Using Competitive Learning. *IEEE Trans. on PAMI*, **16**, 1197-1206.

**Ünal, Ü., Kurugöllü, K., Sankur, B.**, 1999. Color Image Segmentation based on Multithresholding and Fusion (in Turkish), *Proceedings of Conference on Signal Processing and Its Applications (SIU'99)*, Ankara, TURKEY, 16-19 June, 1999, 489-492.

**Valavavis, K.P, Zheng, J., Paschos, G.**, 1996. A Total Color Difference Measure for Segmentation in Color Images, *Journal of Intelligent and Robotic Systems*, **16**, 269-313.

**Vincent, L., Soille, P.**, 1991. Watershed in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations, *IEEE Trans. On PAMI*, **13**, 583-598.

**Vinod et.al**, 1992. A Connectionist Approach for Peak Detection in Hough Space, *Pattern Recognition*, **25**, 1253-1264.

**Weszka, J.S:, Rosenfeld, A.**, 1978. Threshold Evaluation Techniques. *IEEE Trans. on SMC*, **8**, 622-629.

**Xiaohan, Y., Yla-Jaaski, J.**, 1992. Image Segmentation Combining Region Growing and Edge Detection, IEEE Conference on Computer Vision and Pattern Recognition, 481-484.

**Xu, Y., Uberbacker, E.C.**, 1997. 2D Image Segmentation using Minimum Spaning Trees, *Image and Vision Computing*, **15**, 47-57.

**Yasnoff, W.A., Mui, J.K., Bacus, J.W.**, 1977. Error Measures for Scene Segmentation, *Pattern Recognition*, **9**, 217-231.

**Zhang, J., Modestino, J.W.**, 1990. Model-Fitting Approach to Cluster Validation with Application to Stochastic Model-Based Image Segmentation. *IEEE Trans. on PAMI*, **12**, 1009-1017.

**Zhang, J., Wang, D.**, 1998. Image Segmentation by Multigrid Markov Random Field Optimization and Perceptual Consideration, *Journal of Electronic Imaging*, **7**, 52-60.

**Zhang, Y.J, Gerbrands, J.J.**, 1994. Objective and Quantitative Segmentation Evaluation and Comparison. *Signal Processing*, **39**, 43-54.

**Zhang, Y.J.**, 1996. A Survey on Evaluation Methods for Image Segmentation. *Pattern Recognition*, **29**, 1335-1346.3.

**Zhu, S.C., Yuille, A.**, 1996. Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation. *IEEE Trans. on PAMI*, **18**, 884-900.

**Zhu, S.C., Wu, Y., Mumford, D.**, 1998. Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling, *International Journal of Computer Vision*, **27**, 107-126.

**Zikos, M., Kaldoudi, E., Orphanoudakis, S.C.**, 1997. DIPE: A Distributed Environment for Medical Image Processing. *Medical Informatics Europe, MIE'97,* Porta Carras, Greece.

**Zugaj, D., Lattuati, V.**, 1998. A New Approach of Color Images Segmentation Based on Fusion Region and Edge Segmentation Outputs. *Pattern Recognition*, **31**, 105-113.

## APPENDIX A: APPROXIMATION OF 3D HISTOGRAM WITH 2D HISTOGRAMS

To show approximation of 3D histogram with 2D histograms, the following test image will be used.

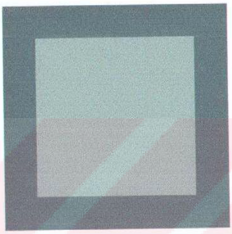


Figure A.1. Test image: It contains two regions whose sizes are equal.

First, the effect of color spread is examined. For this aim, the regions are colored with [120,120,120] and [180,180,180] color RGB vectors. Then the regions are contaminated with additive Gaussian noise whose standard deviations are 5, 10, 20, 30, 40 in all three color vectors as seen in Figure A.2. Thus, two spheres are formed in the 3D histogram with different radii. Corresponding 3D and 2D histograms are also given in Figure A.2. The peaks are separated each other very well in the lower spread cases. But the spread becomes wider the separation between peaks disappears. In this case, this property is observed in the 2D histograms.

In the second case, the color spread is changed only one color band. Standard deviation of spread is varied from 10 to 40 in the B band while that of other color bands remains unchanged at 5. The effect of this case is illustrated in Figure A.3. Form of RG 2D histograms are not affected while form of other two 2D histograms change with spread. Thus sketch of 3D histogram are capture by these two 2D histogram.

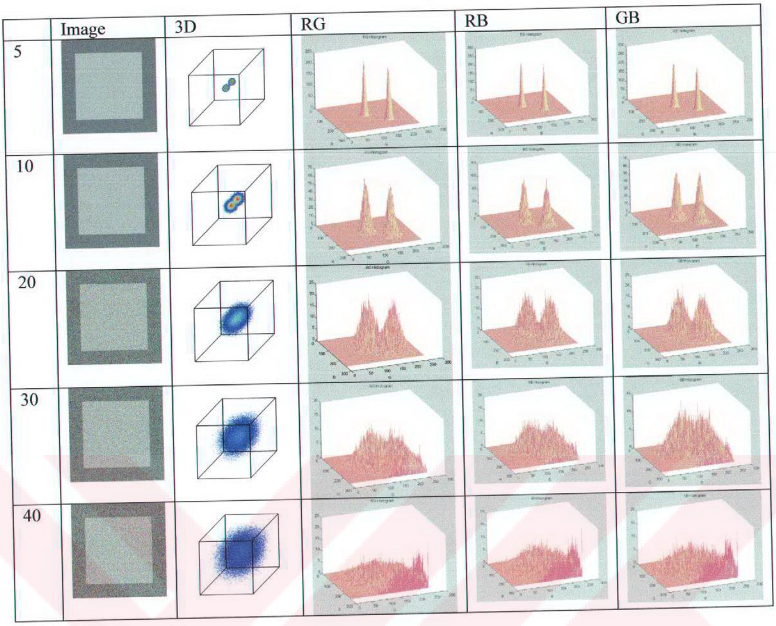



Figure A.2. Effect of color spread in all color bands.

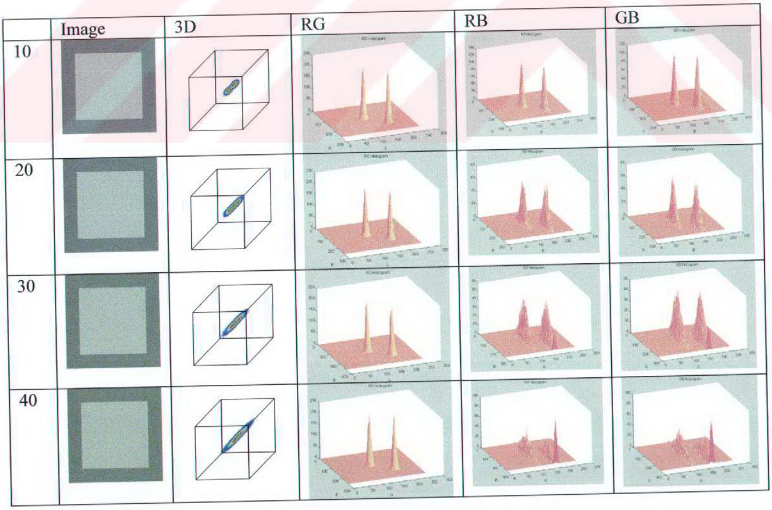



Figure A.3. Effect of color spread in only blue band.

The spread of color can be changed differently in three color bands. This case is shown in Figure A.4 by means of two examples. The standard deviation of color spread is [5, 10, 20] for the first one, while that is [10, 30, 5] for the second one. The form of 3D histogram are delineated by means of three 2D histogram in this cases.

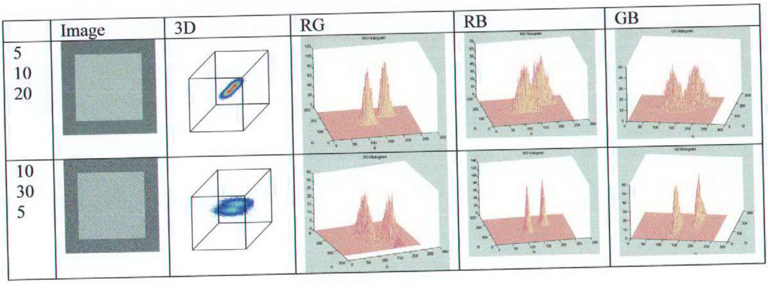



Figure A.4. Effect of color spread which is different in all three bands.

The other important aspect is color transitions between regions. Especially if there is one color transition between two regions these regions cannot be separated by means of 1D histograms. But as seen in Figure A.5, these regions can be delineated by 2D histograms. In this example, the RGB color vectors of two region are chosen as [120, 120, 120] and [180,120,120], i.e., only red band value is changed between two regions. Colour spread is 30 in all bands. These two regions can be separated by using RG and RB 2D histograms although they are not apparent in GB 2D histogram. This is accomplished by fusion step of the algorithm.

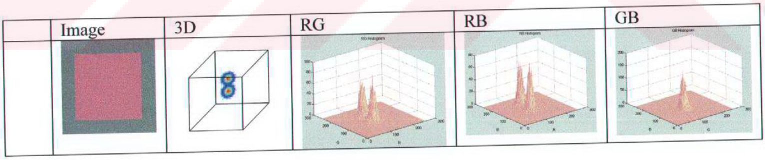



Figure A.5. Effect of one color transition between two regions

Two color transition case is more easier. The regions are formed in all 2D histograms and can be separated easily. Figure A.6 shows this case. The RGB color vectors of two region are chosen as [120, 120, 120] and [180,180,120], i.e., red and green band values are changed between two regions. Colour spread is also 30 in all bands.

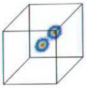
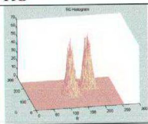
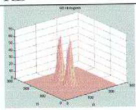
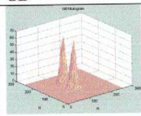
| | Image | 3D | RG | RB | GB |
|---|---|---|---|---|---|
| |  |  |  |  |  |

Figure A.6. Effect of two color transition between two regions.

## APPENDIX B : COMPLEXITY ANALYSIS OF MULTITHRESHOLDING METHODS

The general steps of 1D, 2D and 3D-histogram based multithresholding methods are as follows:

1. Histogram calculation
2. Histogram condensing
3. Connection histogram bins to local parents
4. Peak finding and elimination
5. Upsampling the partitioned histogram and segmentation color band
6. Matching the color labels
7. Fusion color bands

The steps form 1 to 6 are repeated three times for each color band or color band pairs for 1D and 2D based methods, respectively, while they are passed only one time in 3D based method. Furthermore, $6^{th}$ step is not considered in 3D case because of there exits only one segmentation map. $7^{th}$ step also is replaced a 5x5 voting filter to contribute the smoothing effect taking in fusion step for 1D and 2D cases.

The methods are programmed using Microsoft Visual C++ 4.2 compiler under Windows NT 4.0 operating system and Intel Pentium II-266 Mhz computer. The execution times are determined by using clock() routine for an image sizes of 256x256. The execution times of each step for all methods are given in Table B.1.

Table B.1. Execution times of each step. All measurement are given in millisecond.

| Step | 1D | 2D | 3D |
|---|---|---|---|
| 1 | 0.2 | 15 | 19000 |
| 2 | 0.03 | 8 | 93000 |
| 3 | 0.03 | 5 | 380 |
| 4 | 0.02 | 2 | 290 |
| 5 | 4.2 | 56 | 18000 |
| Sub-total for one band | **4.48** | **86** | - |
| Sub-total for three bands | **13.44** | **258** | **130670** |
| 6 | 130 | 180 | - |
| 7 | 287 | 290 | 110 |
| **Total (approximate)** | **430** | **730** | **130000** |

As seen Table B.1, the main complexity comes from 6$^{th}$ and 7$^{th}$ steps for 1D and 2D methods, while 1$^{st}$, 2$^{nd}$ and 5$^{th}$ steps for 3D method. 6$^{th}$ and 7$^{th}$ steps have exactly same body for 1D and 2D methods. Then their execution times are almost same. For 3D case, 6$^{th}$ step is ignored and the 7$^{th}$ step are replaced a 5x5 voting filter instead of 3x5x5 voting filter that are used in 1D and 2D cases. Then the execution time of 7$^{th}$ step in 3D method is faster than the other cases approximately three times.

In 1D case, only 1D histograms are used in the steps from 1 to 5. Only one image band is also used. However 2D histograms and two image bands are used in 2D based method. The histograms used in 2D method are represented by 2D arrays while 1D arrays are used in 1D case. This increases the complexity from O(n) to O($n^2$). Then the complexity of 2D method is higher than 1D one.

There exits an interesting difference between 3D case and the others for 1$^{st}$, 2$^{nd}$ and 5$^{th}$ steps. This difference comes from the constraints of computer system resources rather than algorithmic complexity. It is expected that the complexity of the 3D method $O(n^3)$ because of representing the histograms by 3D arrays. But the size of histograms becomes 16777216 byte (16 MB) in this case. All three bands of input image are also used together in all steps. This increases requirement of data memory. Then the operating system becomes to use paging technique to meet this requirement. Then the paging time are added to computation time in every step of process.

On the other hand, the condensed histograms are used in 3rd and 4th steps. Then there is no paging operation in these steps thus the execution times are more plausible with respect to expected complexity.

## APPENDIX C : ENERGY MINIMIZATION IN CSNN

In CSNN approach, the energy term of each pixel site can be chosen as total neighborhood contributions of a pixel as follows:

$$U_{ij}^{t} = \sum_{k=1}^{m} H_{ijk}^{t} = \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{r=1}^{p} W'_{kl} O_{r}^{t} \tag{C.1}$$

where r denotes neighbor pixels around the pixel (i,j), k and l scan the label layers, m is for the number of layers (segments), p represents the number of neighborhood and $W'_{kl}$ denotes the modified connection weights as depicted in C.2.

$$W'_{kl} = -\frac{1}{p}\left(1 - \frac{2|k-l|}{m}\right) \tag{C.2}$$

$W'_{kl}$ takes the minimum value, which is $\left(-\frac{1}{p}\right)$, when k equals to l and takes the maximum value, which is $\left(-\frac{1}{p}\left(1-\frac{2(m-1)}{m}\right)\right)$, when k and l equal 1 and m, respectively or visa versa. This chosen energy term must be decreased at successive iteration steps. Thus difference between two sequential iteration epochs must be less than 0.

$$\Delta U_{ij} = U_{ij}^{t+1} - U_{ij}^{t} < 0 \tag{C.3}$$

$$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{r=1}^{p} W'_{kl}(O_{r}^{t+1} - O_{r}^{t}) < 0 \tag{C.4}$$

We can write the CSNN update rule instead of $O_{r}^{t+1}$ in equation C.4.

$$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{r=1}^{p} W'_{kl}\left(\frac{\mathrm{Pos}(O_{r}^{t} + \Delta O_{r}^{t})}{\sum_{i=1}^{m}\mathrm{Pos}(O_{r}^{t} + \Delta O_{r}^{t})} - O_{r}^{t}\right) < 0 \tag{C.5}$$

The denominator term, $\sum_{i=1}^{m} Pos(O_i^t + \Delta O_i^t)$, is used for normalization. If t, the number of iterations, goes to infinity, only one neurons in the label layers goes to 1 while the others goes to 0. Since Pos(.) is an upper limiter function that is limited the result between 0 and 1, the denominator term goes to 1. We can also ignore the Pos(.) function in the numerator. Thus, C.5 can be simplified as in C.5 without loss of generality:

$$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{r=1}^{p} W'_{kl}(O_r^t + \Delta O_r^t - O_r^t) < 0 \tag{C.6}$$

$$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{r=1}^{p} W'_{kl}\Delta O_r^t < 0 \tag{C.7}$$

As a consequence, if the expression given in C.7 is proven then the energy minimization in CSNN would be shown.

$\Delta O_r^t$, which is given in 5.5, shows the small positive or negative contributions of the neurons in the neighborhood. At a specific iteration epoch, only one neuron whose output has positive contribution is excited whereas the others whose outputs have negative contribution are inhibited in a pixel site. Thus, the total number of excited neurons in a neighborhood is only p while those of inhibited ones is (m-1)p. Then there exits two marginal cases. First, the all positive contributions come through the maximum connection weights and the all negative contributions appear with the minimum connection weights. Second, positive contributions are with the negative connection weights while the others are positive ones. If these two marginal cases obey the condition given in C.7, the moderate cases between marginal ones also obey the minimization condition. Thus, the energy minimization in CSNN is proven. In the following two cases will be examined:

Case 1. Positive contributions with maximum weights, negative ones with minimum ones:

There exit p neurons that has positive contribution and (m-1)p neurons that negative one. Thus C.7 can be expressed as follows:

$$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{r=1}^{p} W'_{kl}\Delta O_r^t = p\left(-\frac{1}{p}\left(1-\frac{2(m-1)}{m}\right)\right)\delta + (m-1)p\left(-\frac{1}{p}\right)(-\delta) < 0 \tag{C.8}$$

$$\delta\left\{\frac{2(m-1)}{m} - 1 - (m-1)\right\} < 0 \tag{C.9}$$

Since $\delta$ and m are greater than 0,

$$-m^2 + 2m - 2 < 0 \tag{C.10}$$

$$(m-1)^2 + 1 > 0 \tag{C.11}$$

Since $m \geq 2$ in real applications, proposition given in C.11 is true. Then the first case obeys the minimization condition given in C.7.

Case 2. Positive contributions with minimum weights, negative ones with maximum ones:

Like case 1, the expression C.7 can be written according to case 2 as follows:

$$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{r=1}^{p} W'_{kl}\Delta O_r^t = p\left(-\frac{1}{p}\right)\delta + \sum_{k=1}^{m}\sum_{\substack{l=1 \\ k\neq l}}^{m}\sum_{r=1}^{p-1}\left(-\frac{1}{p}\left(1-\frac{2|k-l|}{m}\right)\right)(-\delta) < 0 \tag{C.12}$$

$$-\delta + \delta\sum_{k=1}^{m}\sum_{\substack{l=1 \\ k\neq l}}^{m}\left(\frac{p-1}{p}\right)\left(1-\frac{2|k-l|}{m}\right) \quad < \quad 0 \tag{C.13}$$

$$\delta\left\{-1+\frac{p-1}{p}\left[\sum_{k=1}^{m}\sum_{\substack{l=1 \\ k\neq l}}^{m}1 - \frac{2}{m}\sum_{k=1}^{m}\sum_{\substack{l=1 \\ k\neq l}}^{m}|k-l|\right]\right\} \quad < \quad 0 \tag{C.14}$$

Since if k equals l then $|k-l|=0$ and $\delta > 0$,

$$-1+\frac{p-1}{p}\left[m(m-1) - \frac{2}{m}\sum_{k=1}^{m}\sum_{l=1}^{m}|k-l|\right] \quad < \quad 0 \tag{C.15}$$

$$-\frac{2}{m}\sum_{k=1}^{m}\sum_{l=1}^{m}|k-l| < \frac{p}{p-1} - m(m-1) \tag{C.16}$$

$$\sum_{k=1}^{m}\sum_{l=1}^{m}|k-l| > \frac{m}{2}\left(\frac{p}{p-1} - m(m-1)\right) \tag{C.17}$$

Since $|k-l| \geq |k| - |l|$,

$$\sum_{k=1}^{m}\sum_{l=1}^{m}|k-l| \geq \sum_{k=1}^{m}\sum_{l=1}^{m}|k|-|l| > \frac{m}{2}\left(\frac{p}{p-1} - m(m-1)\right) \tag{C.18}$$

$$\sum_{k=1}^{m}\left(mk - \frac{m(m-1)}{2}\right) \quad > \quad \frac{m}{2}\left(\frac{p}{p-1} - m(m-1)\right) \tag{C.19}$$

$$\frac{m^2(m+1)}{2} - \frac{m(m+1)}{2} \quad > \quad \frac{m}{2}\left(\frac{p}{p-1} - m(m-1)\right) \tag{C.20}$$

$$m(m+1) - (m+1) \quad > \quad \frac{p}{p-1} - m(m-1) \tag{C.21}$$

$$(m-1)(m+1) + m(m+1) \quad > \quad \frac{p}{p-1} \tag{C.22}$$

$$\frac{(m-1)(2m+1)(p-1)}{p} \quad > \quad 0 \tag{C.23}$$

In real applications m and p are greater than 2, than proposition given in C.23 is satisfied. Thus the second case also obeys the minimization condition given in C.7. As a consequence, the energy update scheme in CSNN always minimizes the energy. Then this type of energy and ANN can be used in MRF representation.

## RESUME

Fatih KURUGÖLLÜ was born on August 1, 1968 in Bursa, Turkey. He received Bachelor of Science and Master of Science degrees in control and computer engineering department from Istanbul Technical University, Istanbul in 1989 and 1994, respectively. He has been working as a senior researcher in the Marmara Research Center since 1991. His research interests include image processing, computer vision, neural networks and signal processing applications using DSP chips.