3-1-2009

Fast unsupervised multiresolution color image segmentation using adaptive gradient thresholding and progressive region growing

Sreenath Rao Vantaram

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact rit scholar works@rit.edu.
FAST UNSUPERVISED MULTIRESOLUTION COLOR IMAGE SEGMENTATION USING ADAPTIVE GRADEINT THRESHOLDING AND PROGRESSIVE REGION GROWING

by

Sreenath Rao Vantaram

A Thesis submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

Approved by:

Prof. _______________________________
Thesis Advisor – Dr.Eli Saber

Prof. _______________________________
Thesis Committee Member – Dr.Sohail A. Dianat

Prof. _______________________________
Thesis Committee Member and Head of the Department of Electrical Engineering – Dr.Vincent Amuso

Department of Electrical Engineering,
Kate Gleason College of Engineering,
ROCHESTER INSTITUTE OF TECHNOLOGY,
Rochester, New York

March 2009
Thesis Author Permission Statement

Name of Author : Sreenath Rao Vantaram
Degree : Master of Science
Program : Electrical Engineering
College : Kate Gleason College of Engineering

I understand that I must submit a print copy of my thesis or dissertation to the RIT Archives, per current RIT guidelines for the completion of my degree. I hereby grant to the Rochester Institute of Technology and its agents the non-exclusive license to archive and make accessible my thesis or dissertation in whole or in part in all forms of media in perpetuity. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Print Reproduction Permission Granted:
I, ___Sreenath Rao Vantaram___, hereby grant permission to the Rochester Institute Technology to reproduce my print thesis or dissertation in whole or in part. Any reproduction will not be for commercial use or profit.

Signature of Author: _____________________________________ Date: ____________

Inclusion in the RIT Digital Media Library Electronic Thesis & Dissertation (ETD) Archive
I, ___Sreenath Rao Vantaram___, additionally grant to the Rochester Institute of Technology Digital Media Library (RIT DML) the non-exclusive license to archive and provide electronic access to my thesis or dissertation in whole or in part in all forms of media in perpetuity.
I understand that my work, in addition to its bibliographic record and abstract, will be available to the world-wide community of scholars and researchers through the RIT DML. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation. I am aware that the Rochester Institute of Technology does not require registration of copyright for ETDs.
I hereby certify that, if appropriate, I have obtained and attached written permission statements from the owners of each third party copyrighted matter to be included in my thesis or dissertation. I certify that the version I submitted is the same as that approved by my committee.

Signature of Author: _____________________________________ Date: ____________
DEDICATION

This thesis dedicated to my family

To my father, Mr. Viswanath Rao Vantaram, for his unparalleled and unique ways of motivating me from time to time

To my mother, Mrs. Dharmavani Rao Vantaram, for her never-ending love

To my sister, Preethi Rao Vantaram, for always being there for me
ACKNOWLEDGMENTS

I feel it to be a unique privilege combined with immense happiness to acknowledge the contributions and support of all the wonderful people who have been responsible for the completion of my master’s degree. The two years of graduate study at RIT has taught me that creative instinct, excellent fellowship and perceptiveness are the very essence of engineering. It not only imparts knowledge but also lays emphasis on the overall development of an individual. I am extremely appreciative of RIT, especially Department of Electrical Engineering in this regard.

I would like to thank Dr. Sohail Dianat, Dr. Vincent Amuso and Dr. Eric Peskin for their valuable assistance that has made this work possible. Also, I extend my sincerest gratitude to all my friends/colleagues in and outside the ‘signal processing community’ at RIT for their encouragement and being a part of a grueling yet enjoyable educational experience. I am grateful to Mr. Mark Shaw, Mr. Ranjit Bhaskar and the Hewlett Packard Company for their generosity in supporting and sponsoring this research and showing tremendous confidence and satisfaction in my work and the results achieved.

Finally, I am extremely indebted to the man whose unique ways of teaching and relentless guidance is solely responsible for transmuting an ordinary student like me to a confident graduate researcher. His inspired notion that “if intellect can find the right question, learning and research help find the answer, seek and thou shall find”, has been instrumental in what I am today. Thank you Dr Eli Saber for everything that you have done for me, it’s been a pleasure being your student.

Sreenath Rao Vantaram
In this thesis, we propose a fast unsupervised multiresolution color image segmentation algorithm which takes advantage of gradient information in an adaptive and progressive framework. This gradient-based segmentation method is initialized by a vector gradient calculation on the full resolution input image in the CIE L*a*b* color space. The resultant edge map is used to adaptively generate thresholds for classifying regions of varying gradient densities at different levels of the input image pyramid, obtained through a dyadic wavelet decomposition scheme. At each level, the classification obtained by a progressively thresholded growth procedure is combined with an entropy-based texture model in a statistical merging procedure to obtain an interim segmentation. Utilizing an association of a gradient quantized confidence map and non-linear spatial filtering techniques, regions of high confidence are passed from one level to another until the full resolution segmentation is achieved. Evaluation of our results on several hundred images using the Normalized Probabilistic Rand (NPR) Index shows that our algorithm outperforms state-of-the-art segmentation techniques and is much more computationally efficient than its single scale counterpart, with comparable segmentation quality.
# TABLE OF CONTENTS

Thesis Author Permission Statement ................................................................. II

Dedication ........................................................................................................ III

Acknowledgments ............................................................................................. IV

Abstract .............................................................................................................. V

Table of Contents .............................................................................................. VI

List of Figures ................................................................................................... VIII

List of Tables .................................................................................................... XVII

Chapter 1: INTRODUCTION .............................................................................1

1.1 Objectives and Motivations .......................................................................1

1.2 Literature Review .......................................................................................1

1.3 Contributions .............................................................................................6

1.4 Potential Applications ................................................................................8

1.5 Thesis Outline ...........................................................................................11

Chapter 2: BACKGROUND ..............................................................................12

2.1 Wavelet Transform ....................................................................................12

2.2 Multiresolution Image Decomposition/Representation .........................14

2.3 CIE L*a*b* Color Space ............................................................................19

Chapter 3: Proposed Algorithm ......................................................................21

3.1 Adaptive Gradient Thresholding ...............................................................23

3.2 Dyadic Wavelet Decomposition ...............................................................32

3.3 Multiresolution Region Growing .............................................................34

3.4 Inter-resolution Information Transfer .....................................................53
LIST OF FIGURES

Figure 1: Overview of proposed approach .................................................................6
Figure 2: Image Rendering utilizing MAPGSEG .........................................................9
Figure 3: CBIR utilizing MAPGSEG ......................................................................10
Figure 4 a): Multiresolution image representation ..................................................18
Figure 4 b): Analysis filter bank .............................................................................18
Figure 5: Block diagram of MAPGSEG .................................................................22
Figure 6 a): Gradient histogram of ‘Parachute’ image ..........................................23
Figure 6 b): Gradient histogram of ‘Cheetah’ image .............................................23
Figure 6 c): Gradient histogram of ‘Cars’ image ..................................................23
Figure 7 a): Gradient histogram RGB Vs. CIEL*a*b* of ‘Parachute’ image .........26
Figure 7 b): Gradient histogram RGB Vs. CIEL*a*b* of ‘Cheetah’ image ..........26
Figure 7 c): Gradient histogram RGB Vs. CIEL*a*b* of ‘Cars’ image ...............26
Figure 8: Histogram based adaptive gradient thresholding .....................................28
Figure 9: Flowchart of adaptive gradient thresholding ............................................30
Figure 10 a): Static Vs Adaptive thresholds .........................................................31
Figure 10 b): Multiresolution gradient histograms ...............................................31
Figure 11 a): Determination of number of decomposition levels ............................33
Figure 11 b): Two level decomposition with corresponding designations ..........33
Figure 12: Progressive region growth involving distributed dynamic seed addition.35
Figure 13 a): Cars-L*a*b* (81 X121) ....................................................................36
Figure 13 b): Corresponding color gradient .........................................................36
Figure 13 c): Initial clusters at \( \lambda \) (5), 50*MSS ...............................................36
Figure 13 d): Logical seed map .......................................................................................36
Figure 13 e): Logical seed map after dilation ..................................................................36
Figure 13 f): Padded seeds in the gradient map ............................................................36
Figure 13 g): Initial clusters at $\lambda + 5 (10), 25^\circ \text{MSS}$ .....................................36
Figure 13 h): Parent seeds ...............................................................................................36
Figure 14 a): Logical PS map ..........................................................................................37
Figure 14 b): Unassigned pixels ......................................................................................37
Figure 14 c): Large unassigned regions .........................................................................37
Figure 14 d): Small and isolated unassigned pixel locations .........................................37
Figure 14 e): Small and isolated unassigned pixel map after dilation ...........................37
Figure 14 f): Small and isolated unassigned pixel borders ...........................................37
Figure 14 g): Small and isolated unassigned pixel neighborhood labels .....................37
Figure 14 h): Small and isolated unassigned pixel label assignment ............................37
Figure 15 a): Parent seeds map after seed saturation ..................................................40
Figure 15 b): New seeds after threshold increment ......................................................40
Figure 15 c): Parent seed borders ...................................................................................40
Figure 15 d): Adjacent child seeds map ........................................................................40
Figure 15 e): Seed map after one interval of the region growth procedure ...................40
Figure 15 f): Seeds obtained during the first stage of dynamic seed addition procedure ..........................................................................................................................40
Figure 15 g): Parent seeds for the next region growth interval ...................................40
Figure 16 a): Cars-L*$a*b*$ (161X241) .............................................................................43
Figure 16 b): Corresponding color gradient .....................................................................43
Figure 16 c): Interim segmentation (81X121) ...............................................................43
Figure 16 d): High confidence seeds (161X241) ................................................................43
Figure 16 e): Padded high confidence seeds in the gradient map.................................43
Figure 17 a): Gradient histogram comparison of ‘Cars’ image Vs unsegmented pixels, at the CDS (161X241): Entire histogram ...........................................................45
Figure 17 b): Gradient histogram comparison of ‘Cars’ image Vs unsegmented pixels, at the CDS (161X241): Zoomed view 1 ..............................................................45
Figure 17 c): Gradient histogram comparison of ‘Cars’ image Vs unsegmented pixels, at the CDS (161X241): Zoomed view 2 ..............................................................45
Figure 18 a): Classifying threshold intervals for DDSA...............................................46
Figure 18 b): Zero crossing curve between red and green curves in (a).........................46
Figure 19 a): At decision boundary 1 (gradient value 22): Agglomeration of seeds obtained .............................................................................................................................49
Figure 19 b): At decision boundary 1 (gradient value 22): Overall seed map prior to region growth ...................................................................................................................49
Figure 19 c): At exact point of intersection (gradient value 108): (c) Agglomeration of seeds obtained ..............................................................................................................49
Figure 19 d): At exact point of intersection (gradient value 108): (c) Overall seed map prior to region growth .............................................................................................49
Figure 20 a): Cars-L*a*b* (321X481) ............................................................................50
Figure 20 b): Corresponding color gradient .....................................................................50
Figure 20 c): Interim segmentation (161X241) .............................................................50
Figure 20 d): High confidence seeds (321X481) ............................................................50
Figure 20 e): Padded high confidence seeds in the gradient map..........................50
Figure 20 f): Agglomeration of seeds obtained at various thresholds lower than the decision gradient value........................................................................................................50
Figure 21: Gradient histogram comparison of ‘Cars’ image Vs unsegmented pixels, at the CDS (321X481) ........................................................................................................51
Figure 22 a): MAPGSEG region growth map: Level1..........................................52
Figure 22 b): MAPGSEG region growth map: Level0..........................................52
Figure 22 c): Neighborhood label assignment: Level1 .........................................52
Figure 22 d): Neighborhood label assignment: Level0.........................................52
Figure 22 e): Iterative morphological label assignment: Level1 .............................52
Figure 22 f): Iterative morphological label assignment: Level0 .............................52
Figure 22 g): V2.2 region growth map: before residual pixel label assignment ....52
Figure 22 h): V2.2 region growth map: after residual pixel label assignment .......52
Figure 23: Seed transfer module ..........................................................................54
Figure 24 a): Interim output: Level2 ......................................................................54
Figure 24 b): Interim output: Level1 ......................................................................54
Figure 24 c): Zero insertion yielding: Level1 .........................................................54
Figure 24 d): Zero insertion yielding: Level0............................................................54
Figure 24 e): Neighborhood label assignment: Level1 .........................................54
Figure 24 f): Neighborhood label assignment: Level0 .........................................54
Figure 25: Quantized gradient map at level1 (161X241) ........................................57
Figure 26 a): High confidence pixel locations corresponding to quantization levels 5 and 12 at a level1: Logical map...................................................................................57
Figure 26 b): High confidence pixel locations corresponding to quantization levels 5 and 12 at a level1: Color map .................................................................57
Figure 26 c): High confidence pixel locations corresponding to quantization levels 5 and 12 at a level1: Zoomed in version of circular area in (b) ....................57
Figure 27 a): High confidence pixel locations color map ......................................60
Figure 27 b): MSBR .........................................................................................60
Figure 27 c): High confidence pixel locations color map after MSBR removal ....60
Figure 27 d): Large confident regions .................................................................60
Figure 27 e): Large confident regions seed borders ..........................................60
Figure 27 f): MSBR labels ..................................................................................60
Figure 27 g): High Confidence MSBR regions ....................................................60
Figure 27 h): A-priori information after border refinement .................................60
Figure 28: Euclidean space representation of L*a*b* ........................................63
Figure 29 a): Multiresolution representation of: original RGB ‘Star Fish’ image ....72
Figure 29 b): Multiresolution representation of: Color converted ‘Star Fish’ image72
Figure 29 c): Multiresolution representation of: Color gradient ..........................72
Figure 29 d): Multiresolution representation of: Seeds maps at the end of progressive region growth .................................................................72
Figure 29 e): Multiresolution representation of: Entropy based texture maps .......72
Figure 29 f): Multiresolution representation of: Interim and final segmentation outputs ........................................................................................................72
Figure 30 a): Interim Segmentation at: Level2 .....................................................73
Figure 30 b): Interim Segmentation at: Upconverted to Level1 ..............................73
Figure 30 c): A-priori information at level1 .................................................................73
Figure 30 d): Interim Segmentation at: Level1 ...............................................................73
Figure 30 e): Interim Segmentation at: Upconverted to Level0 .................................73
Figure 30 f): A-priori information at level0 .................................................................73
Figure 30 g): MAPGSEG final segmentation output ..................................................73
Figure 31 a): Church Results: Original .......................................................................75
Figure 31 b): Church Results: GRF .............................................................................75
Figure 31 c): Church Results: JSEG ...........................................................................75
Figure 31 d): Church Results: DCGT ...............................................................................75
Figure 31 e): Church Results: GSEG-V2.2 .................................................................75
Figure 31 f): Church Results: MAPGSEG .................................................................75
Figure 32 a): Parachute Results: Original .................................................................75
Figure 32 b): Parachute Results: GRF .......................................................................75
Figure 32 c): Parachute Results: JSEG .......................................................................75
Figure 32 d): Parachute Results: DCGT .................................................................75
Figure 32 e): Parachute Results: GSEG-V2.2 ...............................................................75
Figure 32 f): Parachute Results: MAPGSEG ...............................................................75
Figure 33 a): Cheetah Results: Original ....................................................................76
Figure 33 b): Cheetah Results: GRF ...........................................................................76
Figure 33 c): Cheetah Results: JSEG ........................................................................76
Figure 33 d): Cheetah Results: DCGT .......................................................................76
Figure 33 e): Cheetah Results: GSEG-V2.2 ...............................................................76
Figure 33 f): Cheetah Results: MAPGSEG ...............................................................76
Figure 34 a): Nature Results: Original .................................................................77
Figure 34 b): Nature Results: GRF .................................................................77
Figure 34 c): Nature Results: JSEG .................................................................77
Figure 34 d): Nature Results: DCGT .................................................................77
Figure 34 e): Nature Results: GSEG-V2.2 .........................................................77
Figure 34 f): Nature Results: MAPGSEG .........................................................77
Figure 35 a): Cars Results: Original .................................................................77
Figure 35 b): Cars Results: GRF .................................................................77
Figure 35 c): Cars Results: JSEG .................................................................77
Figure 35 d): Cars Results: DCGT .................................................................77
Figure 35 e): Cars Results: GSEG-V2.2 .........................................................77
Figure 35 f): Cars Results: MAPGSEG .........................................................77
Figure 36 a): Island Results: Original ..............................................................78
Figure 36 b): Island Results: GRF .................................................................78
Figure 36 c): Island Results: JSEG .................................................................78
Figure 36 d): Island Results: DCGT .................................................................78
Figure 36 e): Island Results: GSEG-V2.2 .........................................................78
Figure 36 f): Island Results: MAPGSEG .........................................................78
Figure 37: Human segmentations for the ‘Island’ image provided by University of California, at Berkeley .........................................................78
Figure 38 a): Asians Results: Original ..............................................................79
Figure 38 b): Asians Results: GRF .................................................................79
Figure 38 c): Asians Results: JSEG .................................................................79
XIV
Figure 38 d): Asians Results: MAPGSEG Level2...............................................................79
Figure 38 e): Asians Results: MAPGSEG Level1 ............................................................79
Figure 38 f): Asians Results: MAPGSEG Level0............................................................79
Figure 39: Human segmentations for the ‘Asians’ image provided by University of California, at Berkeley.................................................................79
Figure 40 a): Tree Results: Original ..............................................................................80
Figure 40 b): Tree Results: GRF ....................................................................................80
Figure 40 c): Tree Results: JSEG ...................................................................................80
Figure 40 d): Tree Results: MAPGSEG Level2 ............................................................80
Figure 40 e): Tree Results: MAPGSEG Level1 ............................................................80
Figure 40 f): Tree Results: MAPGSEG Level0 ............................................................80
Figure 41 a): Road Results: Original .............................................................................80
Figure 41 b): Road Results: GRF ...................................................................................80
Figure 41 c): Road Results: JSEG ..................................................................................80
Figure 41 d): Road Results: MAPGSEG Level2 ...........................................................80
Figure 41 e): Road Results: MAPGSEG Level1 ............................................................80
Figure 41 f): Road Results: MAPGSEG Level0 ............................................................80
Figure 42 a): NPR scores distribution for 300 images of the Berkeley database: GRF .........................................................................................................................80
Figure 42 b): NPR scores distribution for 300 images of the Berkeley database: JSEG .........................................................................................................................80
Figure 42 c): NPR scores distribution for 300 images of the Berkeley database: DCGT .........................................................................................................................81
Figure 42 d): scores distribution for 300 images of the Berkeley database: GSEG .81
Figure 42 e): NPR scores distribution for 300 images of the Berkeley database: MAPGSEG .......................................................................................................................................................................................81
Figure 42 f): NPR scores distribution for 300 images of the Berkeley database: All algorithms distributions superimposed..................................................................81
Figure 43 a): Computational time comparison utilizing Berkeley database (321X421): MAPGSEG, GSEG and DCGT ..........................................................................................................................83
Figure 43 b): Computational time comparison utilizing Berkeley database (321X421): Various levels of MAPGSEG ..........................................................................................................................83
Figure 43 c): Computational time comparison utilizing large resolution image database (750X1200): MAPGSEG, GSEG ..........................................................................................................................83
Figure 43 d): Computational time comparison utilizing large resolution image database (750X1200): Various levels of MAPGSEG ..........................................................................................................................83
Figure 44 a): Segmentation results: Original ................................................................84
Figure 44 b): Segmentation results: GSEG ................................................................84
Figure 44 c): Segmentation results: MAPGSEG (Level0)........................................84
LIST OF TABLES

Table 1: Daubechies 9/7 analysis filter coefficients.......................................................19

Table 2: MAPGSEG threshold selection for a two level decomposition (‘Cars’ image)................................................................................................................................51

Table 3: Evaluation of MAPGSEG using 300 images of the Berkeley database in comparison to published work........................................................................................82

Table 4: Evaluation of various levels of MAPGSEG using 300 images of the Berkeley database ............................................................................................................82

Table 5: Evaluation of various levels of MAPGSEG using 445 large resolution images in comparison to GSEG.................................................................82
Chapter 1: INTRODUCTION

1.1 Objectives and Motivations

Unsupervised image segmentation is a long standing problem in many computer vision and image understanding applications. Segmentation is defined as the meaningful partitioning of images into non-overlapping homogenous regions exhibiting similar features or image content. It finds a place in many important applications such as image rendering/indexing, object classification, content based image retrieval, medical imaging, image/video compression, image/video surveillance and multi-media applications. Few segmentation algorithms have been developed that efficiently facilitate: 1) selective access and manipulation of individual content in images based on desired level of detail, 2) handling sub sampled versions of the input images and decently robust to scalability, 3) a good compromise between quality and speed, laying the foundation for fast and intelligent object/region based real-world applications of color imagery.

1.2 Literature Review

Many grayscale/color domain methodologies have been adopted in the past to tackle this ill-defined problem (see [1, 2] for comprehensive surveys). Initial multiscale research was aimed to overcome drawbacks being faced by Bayesian approaches for segmentation/classification, using Markov Random Fields (MRF’s) and Gibbs Random Field’s (GRF’s) estimation techniques. Derin et al. [3] proposed a method of segmenting images by comparing the Gibbs distribution results to a predefined set of textures using a Maximum a posteriori (MAP) criterion. Pappas et al. [4] generalized the k-means
clustering algorithm using adaptive and spatial constraints, and the Gibbs Random Field (GRF) model to achieve segmentation in the gray scale domain. Chang et al. [5] extended this to color images by assuming conditional independence of each color channel. Improved segmentation and edge linking was achieved by Saber et al. [6] who combined spatial edge information and the regions resulting from a GRF model of the segmentation field. Bouman et al. [7] proposed an algorithm for segmenting textured images consisting of regions with varied statistical profiles using a causal Gaussian autoregressive model and a MRF representing the classification of each pixel at various scales. However most of the aforementioned methods suffered from the fact that the obtained estimates could not be calculated exactly and were computationally prohibitive. To overcome these problems, Bouman et al. [8] extended his work by incorporating a multiscale random field model (MSRF) and a sequential MAP (SMAP) estimator. The MSRF model was used to capture the characteristics of image behavior at various scales. However, the work in [7,8] had either used single scale versions of the input image, or multiscale versions of the image with the underlying hypothesis that the random variables at a given level of the image data pyramid were independent from the ones at other levels.

Comer et al. [9] used a multiresolution Gaussian autoregressive model (MGAR) for a pyramid representation of the input image and “maximization of posterior marginals” (MPM) for pixel label estimates. He established correlations for these estimates at different levels using the interim segmentations corresponding to each level. He extended his work in [10] by using a multiresolution MPM model for class estimates and a multiscale MRF to establish interlevel correlations into the class pyramid model. Liu et al. [11] proposed a relaxation process that converged to a MAP estimate of the eventual
segmentation of the input image using MRF’s in a quad-tree structure. An MRF model in combination with the discrete wavelet transform was proposed by Tab et al. [12] for effective segmentations with spatial scalability, producing similar patterns at different resolutions. Cheng et al. [13] incorporated Hidden Markov Models (HMM’s) for developing complex contextual structure, capturing textural information, and correlating among image features at different scales unlike previously mentioned MRF models. The methods usefulness was illustrated on the problem of document segmentation where intra scale contextual dependencies can be imperative. A similar principle was applied by Won et al. [14] who combined HMM and Hidden Markov Tree (HMT) forming a hybrid HMM-HMT model to establish local and global correlations for efficient block-based segmentations.

Watershed and wavelet-driven segmentation methods has been of interest for many researchers. Vanhamel et al. [15] proposed a scheme constituting a non-linear anisotropic scale space and vector value gradient watersheds in a hierarchical frame work for multiresolution analysis. In a similar framework Makrogiannis et al. [16] proposed watershed based segmentations utilizing a fuzzy dissimilarity measure and connectivity graphs for region merging. Jung et al. [17] combined orthogonal wavelet decomposition with the watershed transform for multiscale image segmentation.

Edge, contour and region structure are other features that have been adopted in various approaches for effective segmentations. Tabb et al. [18] instituted a multiscale approach where the concept of scale represented image structures at different resolutions rather than the image itself. The work involved performing a Gestalt analysis facilitating detection of edges and regions without any smoothing required at lower scales. On the
other hand, Gui et al. [19] obtained multiscale representations of the image using weighted TV flow and used active contours for segmentation. The contours at one level were given as input to the next higher level to refine the segmentation outcome at that level. Munoz et al. [20] applied fusion of region and boundary information, where the later was used for initializing a set of active regions which in turn would compete for pixels in the image in manner that would eventually minimize a region-boundary based energy function. Sumengen et al. [21] showed through his work that multiscale approaches are very effective for edge detection and segmentation of natural images. Mean shift clustering followed by a minimum description length (MDL) criterion was used by Luo et al. [22] for the same purpose.

Fusion of color and texture information is an eminent methodology in multiresolution image understanding/analysis research. Deng et al. [23] proposed a method prominently known as JSEG that performed color quantization and spatial segmentation in combination of a multiscale growth procedure for segmenting color-texture regions in images and video. Pappas et al. [24] utilized spatially adaptive features pertaining to color and texture in a multiresolution structure to develop perceptually tuned segmentations, validated using photographic targets. Dominant color and homogenous texture features (HTF) integrated with an adaptive region merging technique were employed by Wan et al. [25] to achieve multiscale color-texture segmentations.

The task of segmenting images in perceptually uniform color spaces is an ongoing area of research in image processing. Paschos et al. [26] proposed an evaluation methodology for analyzing the performance of various color spaces for color texture analysis methods such as segmentation and classification. The work showed that
uniform/approximately uniform color spaces such as L*a*b*, L*u*v* and HSV possess a performance advantage over RGB, a non uniform color space traditionally used for color representation. The use of these color spaces was found to be suited for the calculation of color difference using the Euclidean distance, employed in many segmentation algorithms. Yoon et al. [27] utilized this principle to propose a Color Complexity Measure (CCM) for generalizing the K-means clustering algorithm, in the CIE L*a*b* space. Chen et al. [28] employed color difference in the CIE L*a*b* space to propose directional color contrast segmentations. Contrast generation as a function of the minimum and maximum value of the euclidean distance in the CIE L*a*b* space, was seen in the work of Chang et al. [29]. This contrast map, subjected to noise removal and edge enhancement to generate an Improved Contrast Map (ICMap), was the proposed solution to the problem of over-segmentation in the JSEG algorithm. More recently, Gao et al. [30] introduced a ‘narrow-band’ scheme for multiresolution processing of images by utilizing the MRF expectations-maximization principle in the L*u*v* space. This technique was found to be competent especially for segmenting dermatoscopic images. Lefevre et al. [31] performed multiresolution image segmentation in the HSV space, applied to the problem of background extraction in outdoor images.

Color gradient-based segmentation is a new contemporary methodology in the segmentation realm. Dynamic color gradient thresholding (DCGT) was first seen in the work by Balasubramanian et al. [32]. The DCGT technique was primarily used to guide the region growth procedure, laying emphasis on color homogenous and color transition regions without generating edges. However this algorithm faced problems of over segmentation due to lack of a texture descriptor and proved to be computationally
expensive. Ugarriza et al. [33] proposed a Gradient SEGmentation (GSEG) algorithm that was an enhanced version of the DCGT technique, by incorporating an entropic texture descriptor and a multiresolution merging procedure. The method brought significant improvement in the segmentation quality and computational costs, but was not fast enough to meet real time practical applications.

1.3 Contributions

In this thesis we propose a new unsupervised Multiresolution Adaptive and Progressive Gradient SEGmentation (MAPGSEG) algorithm, facilitating: 1) robust handling of sub-sampled versions of the original input image, 2) multiple segmentation outputs representing distinct levels of detail, desired by the user, 3) a potential solution that computationally measures up to the demands of most practical applications involving segmentation, 4) an effective compromise between quality and speed.

![Fig.1. Overview of the proposed approach](image-url)
An overview of the proposed approach is shown in Fig. 1. The algorithm begins with a vector gradient computation [34] in CIE L*a*b* color space on the input image at full resolution, followed by a wavelet decomposition to obtain a pyramid representation of it. Starting at the smallest resolution, the functionality of the CIE L*a*b* space includes, but is not limited to, automatically and adaptively generating thresholds required for initial clustering, as well as carrying out a computationally efficient region growth procedure. The resultant classification is combined with an entropy-based texture model and statistical procedure to obtain an interim segmentation representing a certain degree of detail, in comparison to the original input. The up scaled version of this segmentation map is utilized as the a-priori knowledge for segmenting the next higher resolution. Furthermore, this up scaled segmentation is put through confidence computation utilizing the gradient map of the current resolution and non-linear spatial filtering techniques. Regions of high confidence are passed to a fresh run of the algorithm, at the current resolution, subjecting it to lesser work in comparison to its previous stage. However, the thresholds for region growth and distributed dynamic seed generation at higher resolutions are selected in a progressive manner based on a histogram analysis of the gradient values of the image at the current resolution and the unsegmented ‘low confidence’ regions. The aforementioned procedure takes into account the fact that low gradient regions in images can be segmented at relatively small resolutions in comparison to the size of the original, and to this effect, only when more detail is required do we need to perform segmentation at subsequent bigger resolutions. Our algorithm is entirely implemented in MATLAB and tested on a large database of ~745 images. Its performance was benchmarked against popular segmentation techniques utilizing the
NPR index on the same test bed of images in Berkeley database [42] comprising of 300 images (inclusive in the testing database). Furthermore, a comprehensive runtime evaluation was performed on all 745 images with varied resolutions (from 321X481 to 768X1024), and the two evaluations combined show that the MAPGSEG is significantly less computationally intensive, maintaining benchmark segmentation quality with the capabilities of facilitating real time performance.

1.4 Potential Applications

Image segmentation has wide spread medical, military and commercial interests. Our algorithm is designed from a commercial standpoint with an enormous emphasis on performance. Here we illustrate few applications that can take advantage of the capabilities our algorithm.

1.4.1 Image Rendering

Rendering is often utilized in cameras and printers to acquire images with superior visual or print quality. This application is a tool that comes closest to transmuting reality to a photograph or printer output. A typical region/object oriented rendering algorithm, designed for better print quality is shown in Fig. 2. The rendering procedure illustrated is commenced by segmenting the input image using the MAPGSEG algorithm. As can be seen, the output of the MAPGSEG consists of multiple interim results and one final segmentation. Interim output1 obtained at the lowest resolution, represents a coarse segmentation where only the low gradient regions such as the sky and mountain are well represented. Interim output2 is the segmentation result at the next higher resolution where
we see more detail associated with vegetation and manmade structures. The final result shows fine detail with well defined edges for all regions. This hierarchy of detail and corresponding computational performance can be utilized for efficient and intelligent rendering.

Fig. 2 Image rendering utilizing MAPGSEG

If the rendering objective is just limited to the low gradient regions then customized rendering intents are applied to these regions extracted from the up scaled coarse segmentation, to achieve better print quality. The advantage is that the coarse result achieved is much faster than its higher resolution counterparts. Furthermore, the up scaling operation is performed to acquire a coarse segmentation at the same resolution of the input image. As the scope of the rendering intentions are increased, higher resolution segmentations are utilized at which are more computationally expensive. This multiscale
segmentation-integrated rendering approach is much more flexible and computationally inexpensive than utilizing an approach that operates only on a single scale.

1.4.2 Content Based Image Retrieval (CBIR)

Content based image retrieval also known as Query By Image Content (QBIC) is defined as the process of sifting through large archives of digital images based on color, texture, orientation features, and other image content such as objects and shapes.

Fig. 3 illustrates the advantage of incorporating the MAPGSEG algorithm for region-based image retrieval. Here again, if the objective of the retrieval procedure is to acquire images with low gradient regions such as sky then a lower resolution of the input query image would suffice. The query image at the lower resolution and its corresponding segmentation are then given as inputs to a region classification algorithm which identifies
sky without much hindrance, owed to its low gradient content. Moreover, the aforementioned inputs along with the classification output can be used for an effective retrieval procedure. The computational costs are significantly reduced because all operations are performed at a lower resolution of the query image. Regions of higher gradient densities (such as text in Fig. 3) can be similarly used for retrieval at bigger resolutions.

1.5 THESIS OUTLINE

The remainder of this report is organized as follows. In Chapter2, a review of the necessary background required to effectively implement our algorithm is presented. The proposed algorithm, presented in Chapter3, is subdivided into five Sections: 3.1 introduces the adaptive gradient thresholding module, 3.2 explains the dyadic wavelet decomposition scheme, 3.3 illustrates the multiresolution region growth and distributed dynamic seed addition procedure, and Sections 3.4 and 3.5 recap the texture modeling and statistical merging procedure used in the GSEG (V2.2) algorithm. The NPR technique used for evaluating various segmentation results is discussed in Chapter4. Results obtained in comparison to popular segmentation methods and human segmentations are provided in Chapter5 and conclusions drawn in Chapter6.
Chapter 2: BACKGROUND

This section familiarizes some technical concepts that are required for the optimal implementation and understanding of our algorithm. Firstly, we provide a mathematical insight into the Wavelet Transform, the foundation on which the wavelet theory has been established. Secondly, we provide a brief discussion involving the extension of the wavelet transform for pyramidal image representations and its practical implementation using filter banks, imperative from a multiresolution analysis standpoint. Thirdly, we give a brief description of the CIE L*a*b* color space and its characteristics that helped us develop this efficient algorithm.

2.1 WAVELET TRANSFORM

Wavelets are powerful tools capable of dividing data into various frequency bands describing, in general, the horizontal, vertical, and diagonal spatial frequency characteristics of the data. A detailed mathematical analysis of initial multiresolution image representation models and its relation to the Wavelet Transform (WT) can be seen in the work of Mallat et al. [35]. Let $L^2(R)$ denote the Hilbert space of square integrable 1-D functions $f(x)$. The dilation of this function by a scaling component $s$ can be represented as:

$$f_s(x) = \sqrt{s} f(sx)$$  \hspace{1cm} (1)

The WT can be defined by decomposing a signal into a class of functions obtained by the
translation and dilation of a function $\psi(x)$. Here, $\psi(x)$ is called a wavelet and the class of functions is defined, using (1), by $(\sqrt{s}\psi(s(x-u)))_{(s,u)\in \mathbb{R}^2}$. To this effect, the WT is defined as:

$$Wf(s,u) = \int_{-\infty}^{\infty} f(x)\sqrt{s}\psi(s(x-u))dx$$  \hspace{1cm} (2)

An inner product representation of Eq. (2) can be written as:

$$Wf(s,u) = \langle f(x),\psi_s(x-u) \rangle$$  \hspace{1cm} (3)

To enable the reconstruction of $f(x)$ from $Wf(s,u)$ the Fourier transform of $\psi(x)$ must comply with:

$$C_{\psi} = \int_{0}^{+\infty} |\hat{\psi}(\omega)|^2 d\omega < +\infty$$  \hspace{1cm} (4)

Eq. (4) signifies that $\hat{\psi}(0) = 0$, and $\psi(x)$ is small in the vicinity of $\omega = 0$. Therefore, $\psi(x)$ can be construed as the impulse response of a Band Pass Filter (BPF). WT can be now written as a convolution product given as:

$$Wf(s,u) = f * \hat{\psi}_s(u)$$  \hspace{1cm} (5)

where $\hat{\psi}_s(x) = \psi_s(-x)$. Thus, a WT can be interpreted as a filtering of $f(x)$ with a BPF whose impulse response is $\hat{\psi}_s(x)$. Furthermore from the aforementioned discussion we see that the resolution of a WT varies with scale parameter $s$. Sampling $s$, $u$ and selecting a sequence of scales $(\alpha^j)_{j\in \mathbb{Z}}$, can be utilized to discretize the WT. Thus Eq. (5) can be rewritten as:

$$Wf(\alpha^j,u) = f * \hat{\psi}_{\alpha^j}(u)$$  \hspace{1cm} (6)
2.2 Multiresolution Image Decomposition/Representation

A signal $f(x)$ at resolution $r$ can be acquired by filtering $f(x)$ with a Low Pass Filter (LPF) whose bandwidth is proportional to the desired uniform sampling rate $r$, of the filtered result [35]. To negate the possibility of inconsistency with resolution variation these LPF’s are obtained from a function $\theta(x)$ dilated by the resolution parameter $r$ and can be represented in form identical to that of Eq. (1), given below:

$$\theta_r = \sqrt{r} \theta(rx)$$  \hspace{1cm} (8)

Likewise to Eq. (7) the discrete approximation of a function $f(x)$ on a dyadic array of resolutions $(2^j)_{j \in \mathbb{Z}}$ can be represented as:

$$A_{2^j} f = \left( f \ast \theta_{2^j} \left(2^{-j}n \right) \right)_{n \in \mathbb{Z}}$$  \hspace{1cm} (9)

Eq. (9) represents an important category of the DWT known as orthogonal wavelets. Consequently, a wavelet orthonormal basis corresponds to the DWT for $\alpha = 2$ and $\beta = 1$. Although orthonormal basis can be constructed for scale sequences other than $(2^j)_{j \in \mathbb{Z}}$, in general dyadic scales are used because they result in simple decomposition algorithms. For pyramidal multiresolution image representations, $\theta(x)$ is chosen with a Fourier transform defined by:

$$\hat{\theta}(\omega) = \prod_{p=1}^{\infty} U(e^{-i2^{-p}\omega})$$  \hspace{1cm} (10)

where $U(e^{-i\omega})$ represents the transfer function of a discrete filter $U = (u_n)_{n \in \mathbb{Z}}$. 

14
Subsequently, the approximation of a function \( f(x) \) at a scale \((2^j)_{j \geq 0}\) is obtained by filtering \( A_{2^j} f_{2^j} \) with \( U \) and restoring every alternate sample in the resultant convolution, written as:

\[
\Lambda = A_{2^j} f * U = (\lambda_{2^n})_{n \in \mathbb{Z}} \quad (11)
\]

\[
A_{2^j} f = (\lambda_{2^n})_{n \in \mathbb{Z}} \quad (12)
\]

where \( A_{2^j} f = (f * \theta_{2^j n} (n2^{-j} x))_{n \in \mathbb{Z}} \). Eq. (11) and (12) can be utilized iteratively to find the approximation of the signal \( f(x) \) at any dyadic resolution \((2^{-j}, j > 0 \text{ where } 0 \geq j \geq -J)\).

Furthermore the apart from an estimate, the details of a signal at a particular resolution can be also obtained. From Eq. (11) and (12) we see that \( A_{2^j} f \) has double the number of samples in \( A_{2^j} f \). Thus the details \( D_{2^j} f \) at a resolution \( 2^j \) is given by:

\[
D_{2^j} f = A_{2^j} f - A_{2^j}^e \quad (13)
\]

where \( A_{2^j}^e \) is the expanded version of \( A_{2^j} f \) acquired by inserting a zero between each of its samples followed by filtering the resultant signal with an LPF.

Altogether, the previously mentioned discussion can be utilized to develop a multiresolution wavelet model. Earlier, Eq. (9) represented the estimate of \( f(x) \) at a scale of \( 2^j \), utilizing Eq. (3) and (5) this estimate can be re-written as:

\[
A_{2^j} f = \left( f(x), \tilde{\theta}_{2^j} \left( x - 2^{-j} n \right) \right)_{n \in \mathbb{Z}} \quad (14)
\]

In addition, the best estimate of \( f(x) \) at a resolution \( 2^j \) can be derived to be the
orthogonal projection of the signal on the array of all possible estimates designated by a vector space $V_{2^j}$, a proposition of the projection theorem. The array $(V_{2^j})_{j \in Z}$ is known as the multiresolution approximation of $L^2(R)$, requires an orthonormal basis for its computation. An orthonormal basis can be acquired by dilating and translating a scaling function $\phi(x)$, denoted at any $2^j$ resolution (from Eq. (1) or (8)) as $\phi_{2^j}(x) = \sqrt{2^j} \phi(2^j x)$. Thus from the initial definition of the WT the class of functions $(\phi_{2^j}(x - 2^{-j} n))_{n \in Z}$ can be called the orthonormal basis of the vector space $V_{2^j}$. From Eq. (10) we have:

$$\hat{\phi}(\omega) = \prod_{p=1}^{\infty} H\left(e^{\pi i 2^{-j} \omega}\right)$$

(15)

Here $H(e^{\pi i \omega})$ is the transfer function of a discrete filter. Furthermore if:

$$|H(e^{\pi i \omega})|^2 + |H(-e^{\pi i \omega})|^2 = 1$$

(16)

Then the discrete filters represented by $H = (h_n)_{n \in Z}$ are called as quadrature mirror filters. In addition, the orthogonal projection of $f(x)$ on $V_{2^j}$ is given by:

$$P_{V_{2^j}}(f)(x) = \sum_{n \in Z} \langle f(u), \phi_{2^j}(u - 2^{-j} n) \rangle \phi_{2^j}(x - 2^{-j} n)$$

(17)

represents the best estimate of $f(x)$. We now express $A_{2^j} f$ in terms of $\tilde{\phi}(x)$ instead of $\phi(x)$, $\phi(x)$ being an LPF. Thus Eq. (9) becomes:

$$A_{2^j} f = \left(f * \tilde{\phi}_{2^j}(2^{-j} n)\right)_{n \in Z} = \left(\langle f(x), \phi_{2^j}(x - 2^{-j} n) \rangle\right)_{n \in Z}$$

(18)
Utilizing Eq. (15), (18) in conjunction with Eq. (11) and (12) the discrete approximations $A_{2^j} f$ of a signal $f(x)$ at a resolution $2^j$ can be obtained. In addition, the approximation of a signal at a resolution $2^{j+1}$ in $V_{2^{j+1}}$, can be considered to be better than it counterpart at a resolution $2^j$ in $V_{2^j}$. The difference in detail between the two resolutions is given by the orthogonal projection of $f(x)$ on the orthogonal complement of $V_{2^{j+1}}$ in $V_{2^j}$, denoted as $O_{2^j}$. Hence, $O_{2^j}$ orthogonal to $V_{2^j}$ is given by:

$$O_{2^j} \oplus V_{2^j} = V_{2^{j+1}}$$

(19)

The orthogonal projection of $f(x)$ onto $O_{2^j}$ can be obtained in a manner similar to orthogonal projection of $f(x)$ onto $V_{2^j}$. However, if we denote $\varphi_{2^j}(x) = \sqrt{2} \psi(2^j x)$ to be the scaling function and $(\psi_{2^j}(x - 2^{-j} n))_{n \in Z}$ be the orthonormal basis in this case, the Fourier transform of $\varphi(x)$ is given by:

$$\hat{\varphi}(2\omega) = G(e^{-i\omega}) \hat{\psi}(\omega) \quad \text{and} \quad G(e^{i\omega}) = e^{-i\omega} \overline{H(e^{-i\omega})}$$

(20)

where $G(e^{-i\omega})$ is the transfer function of a discrete filter $G = (g_n)_{n \in Z}$. From Eq. (17) and (18) we have:

$$P_{O_{2^j}} (f)(x) = \sum_{n \in Z} \langle f(u), \varphi_{2^j}(u - 2^{-j} n) \rangle \varphi_{2^j}(x - 2^{-j} n)$$

(21)

$$D_{2^j} f = \langle f(x), \psi_{2^j}(x - 2^{-j} n) \rangle_{n \in Z}$$

(22)

Here $D_{2^j} f$ represents the difference in details between successive dyadic resolutions.

Consequently, from the aforementioned mathematical discussion of the wavelet theory,
it can be concluded that the notion of multiscale-resolution and quadrature mirror filters are directly allied to a wavelet orthonormal basis. Without any loss of generalization, this theory can be extended to 2-D signals $f(x, y)$.

$$
\begin{align*}
A_{2^j} f &= \left( f(x, y) \ast \varphi_{2^j}(x - 2^{-j}n, y - 2^{-j}m) \right)_{(n,m) \in \mathbb{Z}^2} \\
D_{2^j}^1 f &= \left( f(x, y), \psi_{2^j}^1(x - 2^{-j}n, y - 2^{-j}m) \right)_{(n,m) \in \mathbb{Z}^2} \\
D_{2^j}^2 f &= \left( f(x, y), \psi_{2^j}^2(x - 2^{-j}n, y - 2^{-j}m) \right)_{(n,m) \in \mathbb{Z}^2} \\
D_{2^j}^3 f &= \left( f(x, y), \psi_{2^j}^3(x - 2^{-j}n, y - 2^{-j}m) \right)_{(n,m) \in \mathbb{Z}^2}
\end{align*}
$$

**Fig. 4.** (a) Multiresolution image representation, (b) Analysis filter bank.

In 2-D the orthonormal basis is acquired using three wavelets $\psi_{1}^{1}(x), \psi_{2}^{2}(x), \psi_{3}^{3}(x)$, where each of these can be considered to be the impulse response of a BPF with a certain orientation preference. Thus the approximation $A_{2^j} f$ of a signal $f(x, y)$ at a scale $2^j$ and its information difference with $A_{2^j} f$ are given as:

$$
\begin{align*}
A_{2^j} f &= \left( f(x, y) \ast \varphi_{2^j}(x - 2^{-j}n, y - 2^{-j}m) \right)_{(n,m) \in \mathbb{Z}^2} \\
D_{2^j}^1 f &= \left( f(x, y), \psi_{2^j}^1(x - 2^{-j}n, y - 2^{-j}m) \right)_{(n,m) \in \mathbb{Z}^2} \\
D_{2^j}^2 f &= \left( f(x, y), \psi_{2^j}^2(x - 2^{-j}n, y - 2^{-j}m) \right)_{(n,m) \in \mathbb{Z}^2} \\
D_{2^j}^3 f &= \left( f(x, y), \psi_{2^j}^3(x - 2^{-j}n, y - 2^{-j}m) \right)_{(n,m) \in \mathbb{Z}^2}
\end{align*}
$$
Here \( D^1_{2j} f \) (HL) and \( D^2_{2j} f \) (LH) correspond to the vertical and horizontal high frequencies respectively, while \( D^3_{2j} f \) (HH) corresponds to high frequency components in both directions, represented in Fig. 4(a). However it must be noted that in Fig. 4 the scales are in terms of \( 2^{-j}, J > 0 \) where \( 0 \geq j \geq -J \).

Practical implementation of multiscale image decomposition has been done effectively using filter banks. A filter bank is defined as an array of filters utilized to separate a signal into various sub bands, generally designed in a manner to facilitate reconstruction of the signal by simply combining the acquired sub bands. The decomposition and reconstruction procedures are better known as analysis and synthesis respectively. Fig. 4(b) and Table 1(below) portray the analysis filter bank and the Daubechies 9/7 analysis coefficients (rounded to 16 digits) in the JPEG2000 compression scheme [36], employed for multiscale analysis in the MAPGSEG algorithm.

<table>
<thead>
<tr>
<th>( i )</th>
<th>Low Pass Filter ( h_L (i) )</th>
<th>High Pass Filter ( h_H (i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.6029490182363570</td>
<td>1.1150870524569900</td>
</tr>
<tr>
<td>±1</td>
<td>0.2668641184428720</td>
<td>-0.5912717631142470</td>
</tr>
<tr>
<td>±2</td>
<td>-0.0782232665289878</td>
<td>-0.0575435262284995</td>
</tr>
<tr>
<td>±3</td>
<td>-0.0168641184428749</td>
<td>0.0912717631142494</td>
</tr>
<tr>
<td>±4</td>
<td>0.0267487574108097</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE 1:** Daubechies 9/7 analysis filter coefficients

2.3 **CIE 1976 L*a*b* Color Space**

In 1976 the Commission International de l’Eclairage (CIE) proposed two device independent approximately uniform color spaces, \( L^*a^*b^* \) and \( L^*u^*v^* \), for different industrial applications with the aim to model the human perception of color. One
important objective these color spaces were able to achieve with reasonable consistency was that, given two colors, the magnitude difference of the numerical values between them was proportional to the perceived difference as seen by the human eye [37]. Experimental data was used to model the response of a person through tristimulus values X, Y and Z, which are linear transformations from R, G and B. Using these tristimulus values the CIE L*a*b* was defined as:

\[
L^* = 116 f \left( \frac{Y}{Y_n} \right) - 16
\]

\[
a^* = 500 \left( f \left( \frac{X}{X_n} \right) - f \left( \frac{Y}{Y_n} \right) \right)
\]

\[
b^* = 200 \left( f \left( \frac{Y}{Y_n} \right) - f \left( \frac{Z}{Z_n} \right) \right)
\]

where \( f(x) = \begin{cases} x^{1/3} & x \geq \alpha \\ 7.787 x + 0.1379310344 & 0 \leq x \leq \alpha \end{cases} \), and \( X_n, Y_n \) and \( Z_n \) are the tristimulus values of a reference white.
Chapter 3: PROPOSED ALGORITHM

The MAPGSEG algorithm embodied in six modules is shown in Fig. 5. The first module (M1) is utilized to adaptively generate thresholds required for initial clustering and region growth at varied levels of the input image pyramid. The second module (M2) performs dyadic wavelet decomposition for multiresolution or pyramidal representation of the input image. The third module (M3) carries out a progressively thresholded growth procedure involving distributed dynamic seed addition. Module4 (M4) is responsible for identifying transferable regions from one resolution to another by exploiting the interim results as a-priori information. Texture modeling utilizing color quantization and entropy computation, is performed in Module5 (M5). The proposed algorithm culminates in a region merging module (M6) fusing the texture characterization channel and current fully grown seed map, to give interim segmentations at low resolutions, and the final segmentation map at a dyadic scale equal to that of the original input image. Furthermore, it is imperative to note that the algorithm does not employ all modules at every scale of the input image pyramid (observe the color coding legend in Fig. 5). The following sub- sections elucidate each of these modules in detail.
Fig. 5. Block diagram of MAPGSEG
3.1 Adaptive Gradient Thresholding

The GSEG algorithm Version2.2 (V2.2) developed by Ugarriza et al. [33] utilized fixed thresholds for segmentation, in the RGB color space. Initial clustering was performed using a threshold value of 10, followed by a region growth procedure carried out at thresholds intervals of 15, 20, 30, 50, 85, and 120. These fixed thresholds were utilized for any image irrespective of its content, and intuitively can be deemed non-ideal, owed to the varied gradient composition present in natural images. This intuitive notion was substantiated as the fixed thresholds intervals were found to consistently pose major problems that hindered the performance of the algorithm, clearly demonstrated by the images in Fig. 6.

Fig. 6. Gradient histogram of: (a) Parachute, (b) Cheetah, (c) Cars.
3.1.1 Effects of Static Threshold Interval Selection

In Fig. 6 three natural scene images with their corresponding enhanced gradient map histograms, are shown. In addition marked in green and red along each of the histograms are the fixed threshold intervals utilized for initial clustering and region growth respectively. Enhanced gradient maps are obtained by computing the gradient utilizing the algorithm in [34] on the increased and decreased contrast versions of the original RGB inputs and finding the pixel by pixel maximum among the two. Increased contrast enhances dark regions and exposes edges present in these regions. On the contrary decreased contrast exposes edge information present in bright areas of the image. Thus, the maximum of the two yields a gradient map consisting of most edge information present in the image. Although gradient map enhancement (employed in V2.2) is useful it comes at the expense of increased computation, especially for large resolution images.

In Fig. 6 it can be observed that the varying shape of gradient histograms from image to image causes the fixed thresholds to be distributed erratically without following a uniform pattern, resulting in contrasting segmentation results. One way of analyzing the effects of static threshold interval selection is by comparing the gradient content of the images in each interval. Considering the first two intervals for region growth, from 10 to 15 and 15 to 20, we see large gradient content in the ‘Cheetah’ and ‘Cars’ images within these intervals, and in contrast for the ‘Parachute’ image the content is small which may result in over segmentation of flat regions with higher computational costs. In addition, V2.2 was designed such that, only seeds (a labeled collection of pixels corresponding to a particular region) which satisfy a certain minimum size criterion based on the current stage of algorithm be considered for further processing. In such a scenario few minute
seeds generated in these low gradient intervals, in the parachute image may be discarded, rendering the thresholds constituting this interval to have negligible contribution to the final segmentation result. Conversely, if an interval is very large in comparison to the extent or span of the histogram, it causes regions with significantly different gradient detail to be merged together, providing a segmentation that is incoherent with the original input image (under segmentation).

Moreover, in Fig. 6(a) and 6(b) the span of the histogram both cases is smaller than the final region growth threshold (120) resulting in wasted computational costs, significantly effecting the overall performance of the algorithm. Thus, overcoming these problems necessitated an adaptive thresholding approach based on image content.

3.1.2 ADVANTAGES OF CIE L*a*b* OVER RGB

The MAPGSEG employs adaptive gradient thresholding in the CIE 1976 L*a*b* color space. The algorithm begins with a conversion from RGB to CIE L*a*b* for correct color differentiation, owed to the fact that the latter is better modeled for human perception and is more uniform in comparison to the RGB space. The L*a*b* data is 8-bit encoded to values ranging from 0-255 for convenient color interpretation and to overcome viewing and display limitations. In addition it has also been widely used for commercial applications. The resultant color converted data is utilized for computing the vector color gradient utilizing the previously mentioned algorithm described in [34], without any enhancement methodology. In general for an image, 8-bit L*a*b* values were found to span over a much smaller range than 8-bit RGB, consequently resulting in a relatively compact histogram than its enhanced RGB counterpart.
In Fig. 7(a), 7(b) and 7(c), shown are the histogram comparisons of RGB (in blue) and L*a*b* (in red), along with the color converted equivalents for the three images in Fig. 6. It can be observed that the red curves are squeezed versions of the blue curves, where the span of the red curves are significantly smaller than the ones in blue but the amplitude of the red are much larger in comparison. To this effect, if we limit our thresholds to the span of the histogram, this squeezed property is an advantage as the region growth procedure is now confined to a significantly smaller range and for any arbitrary threshold interval in this reduced range a higher number of pixels are worked upon, in comparison to RGB. In addition, observe that color space changeover to L*a*b*
has enabled distinct differentiation between the chromatic and achromatic regions, presenting the algorithm with this additional piece of information.

### 3.1.3 Adaptive Threshold Generation

The MAPGSEG algorithm is initiated with a color space conversion of the input image from RGB to CIE L*a*b* for reasons specified in Sections 2.3 and 3.1.2. Using the resultant L*a*b* data, the magnitude of the gradient $G(i, j)$ of the full resolution color image field is calculated. The threshold values required for segmentation are determined utilizing the histogram of the color converted gradient map.

At first, the objective is to select a threshold for the initiation of the seed generation process. Preferably, a threshold value should be selected to expose most edges while ignoring the noise present in images. However, accomplishing this task is precluded by the unique disposition of natural scene images, where a threshold that correctly demarcates the periphery of a given region may unify other regions. Due to this factor, we initiate our thresholding algorithm by estimating a value $\lambda$ that aids in selecting the regions without any edges or with extremely weak and imperceptible edges. We estimate this threshold primarily based on the span of the histogram in combination with empirical data. Given an image, we propose choosing one of two empirically determined threshold values for initiating the seed generation process, by validating how far apart the low and high gradient content in the image are, in its corresponding histogram. The idea is that a high initial threshold be used for images in which a large percentage of gradient values spread over a narrow range and a low initial threshold value be used for images in which a large percentage of gradient values spread over a wide range, in comparison to the span.
of the histogram. The choice of $\lambda$ made in such a manner ensures that significant low gradient regions are acquired as initial seeds.

![Histogram based adaptive gradient thresholding](image)

**Fig. 8.** Histogram based adaptive gradient thresholding

From a practical implementation standpoint, we made this decision of selecting the initial threshold by obtaining the percentage ratio of the gradient values corresponding to 80% and 100% area under the histogram curve, as shown in Fig. 8. If 80% area under the histogram curve corresponds to a gradient value that is less than 10% of the maximum gradient value in the input image, a high threshold value is chosen else a low initial threshold value is chosen. Keeping in view the problems posed by over and under-segmentation, the low and high threshold values were empirically chosen to be 5 and 10 respectively. The former case was used for images where background and foreground have largely indistinguishable gradient detail from each other. The latter case was used for images consisting of a slowly varying background with less gradient detail, well distinguished from prominent foreground content. Having obtained $\lambda$, all significant flat regions and its neighboring areas are generated at threshold intervals of $\lambda$ and $\lambda+5$ with
varied size criterions, to form the initial seeds map. Once the threshold for initiating the segmentation process is determined, we proceed to calculate thresholds intervals for the dynamic seed addition portion of the region growth procedure.

Dynamic seed generation is that portion of the growth process where additional seeds are added to the initial seeds at the lowest resolution or existing high confidence seeds at subsequent higher resolutions. These threshold limits constituting various intervals selected for region growth are determined utilizing the area under the gradient histogram that does not fall within the gradient range of the initial thresholds. The first threshold for dynamic seed addition ($T_i$, $i=1$) is determined by adding 10% of the histogram area greater than the maximum gradient value of the initial seeds ($T_{i-1} = \lambda +5$), to the cumulative area detected by $\lambda+5$ and obtaining the corresponding gradient value. This process is continued for each new stage of the dynamic seed addition procedure where a 10% increment of the histogram area greater than the upper limit of the threshold interval of its corresponding previous stage ($100-A_{i-1}$) is added to the cumulative image area detected at the end of that (previous) stage ($A_{i-1}$), as illustrated in Fig. 8 and 9. Generating the threshold values in such a manner always ensures that: 1) they are adjusted to account for the exponential decay of gradient values (as seen in Fig. 8), 2) regions of significant size are added to the segmentation map at each interval, 3) they lie within the span of the histogram, avoiding the possibility of wasted computational efficiency.
Fig. 9. Flow chart of adaptive gradient thresholding
Fig. 10. (a) Static Vs Adaptive thresholds, (b) Multiresolution gradient histograms.

The effect of utilizing the aforementioned threshold generation procedure is clearly illustrated in Fig. 10(a). In this figure, shown is the comparison of static and adaptively generated thresholds for the ‘cheetah’ image. Here the magenta and yellow markers signify thresholds intervals utilized for initial clustering ($\lambda, \lambda+5$) and region growth respectively. It can be observed that these intervals are distributed along the histogram curve in a manner that they can meaningfully contribute to the segmentation result. This is clearly seen by comparing the gradient content in the last few intervals, where the adaptive thresholds cover more significant areas (on the red curve) than the static thresholds which include much less gradient content (on the blue curve). Also observe that the adaptive thresholds are all located within the span of the histogram thus avoiding wasted computational costs. In the MAPGSEG the adaptive thresholds were generated on the full resolution image (mentioned previously). Once these thresholds were acquired, the same thresholds were utilized in a progressive framework for faster segmentation at various resolutions. This was possible as given any threshold interval, the gradient content in this interval increases from lower to higher resolutions with the overall shape
of the gradient histogram being the same, as can be observed in Fig. 10(b). Therefore in
the MAPGSEG the threshold generation scheme was performed only once and the same
thresholds were utilized for segmenting the input image at all resolutions.

3.2 Dyadic Wavelet Decomposition

The MAPGSEG algorithm employs a dyadic wavelet decomposition scheme for
multiscale image representation, as described in Sections 2.1 and Section 2.2. In order to
ensure good approximations of the 2-D input signal the analysis coefficients utilized are
the same as the ones used in the JPEG2000 compression scheme, which is considered to
be one amongst the state of the art compression standards. Since segmentation can be
used in multiple applications we decided to make the number of decomposition levels
dynamic, for an arbitrary image. However in order to be able to achieve this objective we
introduce a user or application defined variable called ‘Desired dimension’. Desired
dimension (D) is defined as the smallest workable dimension desired by a user or
constrained by an application. Often applications are restricted by the smallest size of an
image that they can handle. The MAPGSEG is designed such that its gives the
application or user the option to set the smallest workable dimension for segmentation.
Once D is initialized based on the resolution of the input image, the algorithm
automatically determines the number of dyadic decomposition levels that will result in
the input image resolution being in the vicinity DXD, since DXD may or may not be a
dyadic scale of the original input. In the case of images that are of the form m by n where
m≠n (rectangular image) we find the number of decomposition levels by working with
the maximum of m and n and find the number of levels that will take this maximum value
in the vicinity of D (see Fig. 11(a)). This will result in the smaller dimension to be automatically mapped such that aspect ratio is constant. In the MAPGSEG algorithm we set D=128 and the number of levels are counted till the maximum dimension of the input is in the range $0.8 \times D \leq \text{maximum} (m, n) \leq 1.2 \times D$, as shown in Fig. 11(a).

![Diagram of decomposition levels](image)

**Fig.11.** (a) Determination of number of decomposition levels, (b) Two level decomposition with corresponding designations.

Having obtained the number of decomposition levels (L) based on desired dimension D the input image (L=0) is decomposed to the smallest resolution (L=k). In doing so all the channel (L*, a*, b*) information acquired from the LL sub band and corresponding size information pertaining to the intermediate levels (L=k-1, k-2, ..., 1) are stored. To this effect the decomposition scheme is performed only once without having to be repeated for every level. In Fig. 11(b) shown is a typical dyadic scale image pyramid with designated levels.
3.3 **MULTIRESOLUTION REGION GROWING**

In V2.2 the region growth and seed addition process were interlaced with each other, where every growth cycle corresponded with a seed addition stage. However in the MAPGSEG we propose a progressively thresholded growth procedure where region growth cycles do not have an exclusive one-to-one relationship with the seed addition procedure. The following subsections discuss our unique multiresolution region growing procedure involving distributed dynamic seed addition and its performance advantages in a multiscale framework. A flow chart of the entire module is shown in Fig. 12.

### 3.3.1 Initial Clustering

The initial positioning of seeds utilizing $\lambda$ (either 5 or 10 for a particular image) and $\lambda+5$ is done only at the lowest resolution of the image pyramid, as shown in Fig. 16. All regions in the image, whose gradient value fall below these thresholds, are classified as initial seeds or Parent Seeds (PS). The parent seeds map signifies the starting points for region formation and the seeds that are part of this map are constrained by varying multiplicative products of a minimum seed size (MSS) criterion. The MSS at a level $L$ is a function of the down sampling rate $2^L$ employed during decomposition, is computed as

$$\text{MSS}=2^L \times 0.01\% \times m \times n$$  \hspace{1cm} (30)

where $m$ and $n$ are the dimensions of the original input image. These varying size criterions obtained as $50\times$MSS and $25\times$MSS for $\lambda$ and $\lambda+5$ respectively, are imperative for proper region formation as illustrated in Fig. 13.
Fig. 12. Progressive region growth involving distributed dynamic seed addition
Fig. 13. (a) Cars-L*a*b* (81 X121), (b) Corresponding color gradient, (c) Initial clusters at $\lambda$ (5), 50*MSS, (d) Logical seed map, (e) Logical seed map after dilation, (f) Padded seeds in the gradient map, (g) Initial clusters at $\lambda+5$ (10), 25*MSS, (h) Parent Seeds.

The ‘Cars’ image in Fig. 13(a) contains a lot of gradient detail as displayed in Fig. 13(b). Thus for initial clustering the threshold $\lambda$ was determined to be 5. It can be seen that $\lambda$ constrained with size criterion of 50*MSS utilizing connected component analysis, detects large flat regions pertaining to the motorway and sky (Fig. 13(c)). Other low gradient regions that are not detected are acquired by padding the existent seeds generated at $\lambda$, so that a threshold increment and size constraint reduction results in detection of smaller seeds at locations other than the pre-existing ones. In V2.2 seed padding was performed using nonlinear spatial filtering techniques. However, the MAPGSEG adopts a two step morphological method for seed padding. Firstly, a logical map is obtained, consisting of 0’s and 1’s where a pixel having a value of 1 signifies that it is a part of an existing seed and 0 indicated an unassigned pixel location. This is followed by the dilation of the logical map by a 3 by 3 structuring element. The two step seed padding procedure is represented in Figs. 13(d) and 13(e) respectively. The padded seeds are marked in the gradient map signifying all locations that are available for seed generation and vice versa (Fig. 13(f)). The threshold is incremented to $\lambda+5$ and the size constraint is reduced to 25*MSS, resulting in smaller gradient areas being detected, as
portrayed in Fig. 13(g). Thus we see that varying size constraints play a major part in proper region formation. The agglomeration of all seeds detected, forming the parent seeds map is shown in Fig. 13(h).

### 3.3.2 Seed Saturation

The parent seeds map, prior to region growth, is subjected to a seed saturation process where all isolated and small unassigned pixel regions encompassed within seed boundaries, are assigned the labels of corresponding parent seeds. However, contiguous unassigned pixel locations larger than the current size criterion (25*MSS) are left unassigned as these are potential locations for new seeds during region growth. The seed saturation procedure for the parent seeds map shown in Fig. 13(h) is illustrated in Fig. 14.

![Figure 14. (a) Logical PS map, (b) Unassigned pixels, (c) Large unassigned regions, Small and isolated unassigned pixel: (d) Locations, (e) Map after dilation, (f) Borders, (g) Neighborhood labels, (h) Label assignment, (i) Seed saturation.](image)

A logical map for Fig. 13(h) is portrayed in Fig. 14 (a). The image negative of this logical map is shown in Fig. 14(b). This represents all unsegmented pixel locations in the
image. In order to find all the unassigned pixel locations larger than the current size criterion (25*MSS) we employ connected component analysis to the map in Fig. 14(b). The result is shown in Fig. 14(c) and it these large unsegmented regions that are passed on to the region growth procedure. In addition, the large regions are removed from the map consisting of all unsegmented pixel locations (Fig. 14(b)), to give a map of isolated and small contiguous pixel regions that can be directly assigned to the labels of their corresponding encompassing parent, shown in Fig. 14(d). However, in order to achieve this objective we need to know the labels of parent surrounding the small pixel regions. This is done by first obtaining borders of these pixel regions which are morphologically extracted. Seed borders are obtained by first dilating all seeds by a 3 by 3 structuring element in a manner similar to the aforementioned seed padding process, and eliminating the original seeds from their dilated versions to obtain corresponding seed borders. The dilated version of Fig. 14(d) seed map is shown in Fig. 14(e). All the nonzero pixels in Fig. 14(d) are removed from its dilated counterpart yielding isolated and small seed borders shown in Fig. 14(f). This seed border map is then point wise multiplied with the parent seeds map to obtain the parent labels in the proximity of the isolated pixels, as shown in Fig. 14(g). Having obtained all surrounding parent labels the small and isolated unsegmented pixels are assigned appropriate labels to complete the seed saturation process, as presented in Figs. 14(h) and 14(i) respectively. The advantage of the seed saturation procedure can be analyzed by comparing Fig. 13(h) and 14(i). It can be observed that a decently large portion of isolated pixels have been assigned labels without having to be processed during region growing. This results in a more efficient growth procedure where computational costs are channelized to segmented meaningful
regions of the image rather than working on small isolated and insignificant regions that will visually not have any impact or show up as distinct segments in the final segmentation result.

### 3.3.3 Sequential Region Growing and Dynamic Seed Addition

The adaptive gradient thresholding algorithm discussed in Section 3.1 generates dissimilar values of growth intervals for most natural scene images. However, in the case of images with less gradient detail or foreground content, a situation may arise where identical thresholds are generated, causing the region growth and seed addition procedure to be inefficient. To overcome this problem, at the very beginning of the region growth procedure, all the thresholds demarcating the growth intervals are checked for similarity with one another. The ‘check’ is designed such that the growth procedure is performed only if the two thresholds constituting the current interval are different from each other, else it is forcibly existed and the processing of the next interval begins. This adds an additional dimension to the algorithm, as not only are the thresholds generated adaptively but also their number may vary from image to image.

Once the updated parent seeds map after seed saturation is obtained (Fig. 15(a)), the MAPGSEG algorithm proceeds to the growth procedure in a manner similar to V2.2, by increasing the threshold to detect new areas, referred to as Child Seeds (CS), shown in Fig. 15(b). However, at this point, only the child seeds that are adjacent to previously generated parent seeds are classified. The adjacent child seeds are found by obtaining the seeds that share pixels with parent seed borders. In V2.2 parent seed borders are found using a non-linear spatial filter that operates in a 3 by 3 neighborhood such that, the
output of the filter is zero if all elements in the neighborhood are exclusively zero or non-zero, and the gives a nonzero output if the neighborhood elements are a mixture of zero and non-zero values. However, the MAPGSEG adopts a morphological method to acquire parent seed borders (Fig. 15(c)) with identical results to V2.2, aforementioned in the discussion on seed saturation. Morphological extraction of seed borders was found to be computationally more efficient especially for large resolution images, in comparison to non-linear spatial filtering.

![Image: Fig. 15. (a) Parent Seeds map after seed saturation, (b) New seeds after threshold increment, (c) Parent seed borders, (d) Adjacent child seeds map, (e) Seed map after one interval of the region growth procedure, (f) Seeds obtained during the first stage dynamic seed addition procedure, (g) Parent Seeds for the next region growth interval.]

Having obtained the adjacent child seeds (Fig. 15(d)) utilizing the parent seed borders map, the MSS criterion is now employed to differentiate between child seeds that can directly be merged with corresponding parents and those that have to be further processed. Incorporation of the MSS criterion at this point reduces the number of child seeds. The child seeds greater than the MSS constraint are checked for luminance and chrominance ($L^*, a^*, b^*$) similarity with their parents, using the euclidean distance measure between their mean channel information. The reason for choosing this color
space and distance metric combination is that: 1) it ensures that comparison of various regions is similar to the distinction made by the human eye, 2) the increased complexity of a different distance metric like the Mahalanobis distance does not improve the results, due to the small variance of the regions being compared, owed to their spatial proximity. On the other hand V2.2 employed the euclidean distance metric in the RGB color space which is non-uniform in nature. Thus the euclidean distance measure in a non-uniform color space, employed earlier, was not a true indication of similarity of between regions, resulting in V2.2 yielding many oversegmented results. However the use of the CIE L*a*b* which is more uniform in comparison to RGB, helped reducing the over segmentation problem to a great extent. The maximum color distance to allow the integration of a child seed to its parent was empirically chosen to be 60 in the MAPGSEG algorithm.

The dynamic seed addition portion of the region growth procedure is responsible for the detection of new areas with higher gradient densities, where each stage corresponds to a different threshold validated by performing a similarity check for the thresholds generated at the very beginning of the growth procedure. The seeds added due to dynamic seed addition process may consist of adjacent and non-adjacent seeds, and obtained at varying size criterions (10*MSS, 5*MSS, and a criterion equivalent to MSS for all remaining seed addition thresholds) in a manner similar to initial clustering (shown in Fig. 15 (f)). The non-adjacent seeds that are larger than the corresponding seed size criterion, based on the interval of operation, are added as parent seeds to the current seed map and the all the adjacent seeds are processed in the previously explained procedure. The seed map obtained at the end of each interval of the region growth and dynamic seed
addition process, becomes the parent seed map for the next interval, displayed in Fig. 15 (g). The seed tracking algorithm (of V2.2) is employed in the growth procedure for growth rate feedback, preventing seeds to overflow into regions of similar $L^*a^*b^*$ values but different textures. When the last growth interval has been reached, all the significantly identifiable regions would have been given a label and all remaining unsegmented areas are close to the edges of the segmented regions.

3.3.4 **PROGRESSIVE REGION GROWING UTILIZING DISTRIBUTED DYNAMIC SEED ADDITION (DDSA)**

In the region growth process discussed so far, there exists an exclusive one-to-one relationship with the seed addition procedure, which is the methodology adopted by the MAPGSEG algorithm only at the smallest resolution in the image pyramid (see red arrows in Fig. 12). However the true progressive and cost-effective nature of the growth procedure is accentuated at subsequent higher resolutions where Distributed Dynamic Seed Addition takes place. The DDSA procedure commences at $(k-1)^{th}$ level in a $k$ level decomposition, after the interim segmentation of the $k^{th}$ level is passed through the seed transfer module (M4 in Fig. 12). The seed transfer module is responsible for acquiring regions of high confidence from the $k^{th}$ level segmentation at the resolution of the $(k-1)^{th}$ level.

The significance of the DDSA can be intuitively derived from the images in Fig. 16. In Figs. 16 (a) and (b), shown is the ‘Cars’ image and its corresponding gradient map at a resolution of 161X241 (level k-1 where k=2). The image in Fig. 16(a) is obtained by garnering all channel and size information corresponding to the $(k-1)^{th}$ level, fusing them together to give the Current Dyadic Scale (CDS) image, a process aforementioned in
Section 3.2. Thus, we see no additional computational expenses in trying to acquire the CDS image at the (k-1)th level. The MAPGSEG output at the smallest resolution (81X121, level k where k=2) is shown in Fig. 16(c). This is achieved after the output of module3 (region growing module) is combined with a texture map (module4) in a statistical merging procedure (module5), as represented in Fig. 12. The output of the seed transfer module is shown in Fig. 16(d). The seeds in Fig. 16(d) represent all regions of high confidence at the CDS which can directly be incorporated from the interim segmentation of the previous level (Fig. 16(c)), and can be considered as a-priori information for processing at the current scale. The entire protocol from the end of region growing to the segmentation output, followed by seed transfer, is present in subsequent sections.

Fig. 16. (a) Cars-L*a*b*(161X241), (b) Corresponding color gradient, (c) Interim segmentation (81X121), (d) High confidence seeds (161X241), (e) Padded high confidence seeds in the gradient map.

It can be observed that the a-priori information at the CDS, shown in Fig. 16(d), consists of seeds mostly in low gradient regions. Due to this reason, initial clustering discussed earlier in this section, is not employed at the commencement of processing, at this level. Moreover if we consider the a-priori information as parent seeds for the current
level, intuitively it can be observed that all the growth intervals generated for this image will not be required to segment the remaining regions using our region growing methodology, since these unsegmented regions occupy a relatively smaller area in the image. It is based on this intuitive notion we designed the DDSA. The essence of the DDSA is to explore the possibility of utilizing some or all of the adaptively generated growth intervals directly for seed addition without having to actually having to perform region growth. In other words we aim to identify the intervals that can be used for addition of seeds by bypassing the region growth protocol and the ones in which region growth is indispensable before any seed addition can be performed. To this effect, where seed addition is done in a dynamic and distributed framework, we call the procedure as ‘Distributed Dynamic Seed Addition’.

Practically, we achieve this objective by a histogram analysis of gradient information of the CDS image (Fig. 16(b)) and the gradient values of all unsegmented regions that are derived after padding the high confidence seeds in the CDS gradient map (shown in Fig. 16(e)). In Fig. 17(a) shown are the gradient histogram plots of Fig. 16(b) (blue curve) versus the histogram of unpadded pixels in Fig. 16(e), along with the generated threshold intervals, shown as magenta and yellow markers.
Fig. 17. Gradient histogram comparison of ‘Cars’ image Vs unsegmented pixels, at the CDS (161X241). (a) Entire histogram, (b) Zoomed view 1, (c) Zoomed view 2.

Fig. 17(a) at each gradient value, the drop in number of pixels indicated by the difference of a point on the blue curve to its counterpart on the green curve corresponds to the number high confidence pixels possessing that gradient value. Thus the shaded region in red signifies gradient values of all high confidence pixels, and the gradient value range of all a-priori seeds are from zero to the point of intersection of the two curves. Furthermore observe that the behavior of the two curves is the same in the latter half of the histogram suggesting that all strong gradient regions have remained unclassified. Since most pixels with low gradient values are already assigned labels, performing the region growth procedure in the low gradient threshold intervals is bound not to bring about any significant change in the area covered by the existent seeds.
yielding extravagant computations with little contribution towards the final segmentation result. Consequently, the point of intersection of the two curves can be utilized as a decision boundary for classifying intervals that will be of any significance during region growth. The threshold intervals below the intersection point were considered for seed addition without region growth and the intervals above the intersection point were subjected to the region growth procedure followed by seed addition.

Fig. 18. (a) Classifying threshold intervals for DDSA. (b) Zero crossing curve between red and green curves in (a).
On the other hand, due to the diverse nature of natural images this consideration can yield contrasting results, illustrated by Figs. 17(b) and (c) which are the zoomed versions of the histograms shown in Fig. 17 (a). We see that the curves do not intersect with each other until a gradient value of 108 towards the end of the histogram (0% difference line in Fig. 18(a)), is reached. In such a scenario utilizing the exact intersection point as the decision boundary for classifying seed addition thresholds prior to region growth, will result in a large number of minute seeds, increasing the computational overhead for region growing and merging. Thus instead of searching for an exact intersecting point to classify these threshold intervals, we search for a decision threshold confined to the interval ranging from a gradient value 0 to a gradient value corresponding to 0.01% of the maximum difference value of the two histogram curves as seen in Fig. 18(a). In addition we utilize the difference curve (a-priori information gradient histogram) between the blue and green histogram curves to a find a suitable decision boundary for classifying thresholds. Note that the value of 0.01% of maximum difference is considered as a simulated point of intersection for the two curves.

The zero crossing point between the histogram curve of the segmented (red) and unsegmented pixels (green), was chosen to be a suitable threshold to distinguish among intervals which can be used for seed addition with and without region growing. To ensure that the correct decision threshold is being used we also checked for the consistency in zero-crossing, as can be seen in Fig. 18(b). From Fig. 18(a) it can be seen that the intersection point between the red and green curves (shown as a black marker at a gradient value of 22) determines the maximum range within which there is a significant change in the number of pixels per gradient value, or the range of gradient values which
contain a large portion of a-priori seeds, and is the ideal range that can be used for adding seeds of significant sizes without loss in visible gradient detail. For images where this decision boundary (Decision boundary 1 in Fig. 18(a)) yields no significant seeds, the threshold corresponding to 0.01% of maximum difference is utilized to find any significant seeds without merging discernible gradient information (Decision boundary 2 in Fig. 18(a)), not used at the current level for this image). All threshold intervals beyond 0.01% of the maximum difference gradient value, are utilized for the previously mentioned sequential region growth procedure. Observe that though the number of unsegmented pixels in the intervals of high gradient is lesser in comparison to ones in low gradient ranges, the gradual increment of seed area due to region growth in the high gradient range is required to be able to segment regions without merging edge information, which is not requisite at low gradient ranges.

The performance advantage of the DDSA can be seen in Fig. 19. The region growth intervals for the ‘Cars’ image constitute threshold values of 12, 15, 21, 36 and 54 (yellow markers in Fig. 19(a)). The point of intersection of segmented and unsegmented pixels, or the decision boundary for classifying thresholds as mentioned previously was obtained to be at a gradient value of 22. Therefore, for this image at level 1 the thresholds suitable for seed addition without the need for region growth were chosen to be 12, 15 and 21, while the intervals from 21-36 and 36-54 were chosen for region growth. The agglomeration of seed generated at 12, 15, and 21 is shown Fig. 19(a). In this seed map it can be seen that all seeds are of decent size and cover a significant portion of the unsegmented image area in low gradient regions. On the other hand observe that in Fig. 19(c) the seeds generated at all growth intervals (12, 15, 21, 36 and 54) consist of a whole number of minute seeds
which in addition to the growth process will hinder the performance of the region merging module, also designed in an iterative format. Incidentally, seeds of 19(a) are generated as a result of our controlled decision making, and Fig. 19(c) would result if the decision boundary was chosen to be the exact point of intersection of the two curves. This illustrates the advantage of choosing Decision boundary 1 over the exact point of intersection, as the former is a good compromise between the regions that are directly assigned labels and the ones that are grown, such that the overall computational effort is minimal. Figs. 19(b) and (d) represent the seed map after the initial phase of seed addition and prior to region growing, corresponding to Figs. 19(a) and (c) respectively.

Fig. 19. At decision boundary 1 (gradient value 22): (a) Agglomeration of seeds obtained, (b) Overall seed map prior to region growth. At exact point of intersection (gradient value 108): (c) Agglomeration of seeds obtained, (d) Overall seed map prior to region growth.

Clear advantages of our controlled threshold section for progressive region growing can be seen by observing the images presented in Fig. 20. In Figs. 20 (a) and (b), shown is the ‘Cars’ image and its corresponding gradient map at a resolution of 321X481 (level k-2 where k=2). The MAPGSEG interim output at level1 (161X241) is shown in Fig. 20(c). The a-priori information for the CDS is shown in Fig. 20(d). It can be observed that most areas of the image have been assigned a region and the ones close to strong gradient content are unassigned. Here again our previous discussed histogram analysis in performed utilizing Figs. 20(b) and unpadded areas of Fig. 20(e). A histogram comparison of the two is portrayed in Fig. 21. We see that the histogram curve in green
reflects a gradient map in which most areas have already been segmented which results in our decision criterion (Decision boundary 1 at gradient value 60) to choose all the growth intervals (12, 15, 21, 36 and 54) only of pre-growth seed addition. Thus for the CDS no region growth is performed bringing about significant improvement in runtime of the algorithm, owing to the iterative nature of the growth procedure. The seed generated due to all growth intervals in the unassigned regions are shown in Fig. 20(f). Fig. 20(g) represents the seed map after the pre-growth seed addition process which is directly led to the merging module. Thus we see that as the MAPGSEG traverses from one resolution to another, the region growth procedure is performed in progressively increasing threshold intervals. In addition, as the algorithm navigates across resolutions, the DDSA procedure is dispensed with more responsibility while the growth procedure becomes discretionary, to the extent that it may be completely bypassed as seen in the example of the ‘Cars’ image. Moreover this controlled mechanism of thresholding enables the our algorithm to work efficiently without a seed tracking algorithm, thus compensating for it at all dyadic scales other than the smallest one.

Fig. 20. (a) Cars-L*a*b* (321X481). (b) Corresponding color gradient. (c) Interim segmentation (161X241). (d) High confidence seeds (321X481). (e) Padded high confidence seeds in the gradient map. (f) Agglomeration of seeds obtained at various thresholds lower than the decision gradient value.
Fig. 21. Gradient histogram comparison of ‘Cars’ image Vs unsegmented pixels, at the CDS (321X481).

**TABLE 2:** MAPGSEG threshold selection for a two level decomposition (‘Cars’ image)

<table>
<thead>
<tr>
<th>Level (Resolution)</th>
<th>MAPGSEG Thresholds/Intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2  (81X121)</td>
</tr>
<tr>
<td>Initial Clustering</td>
<td>5, 10</td>
</tr>
<tr>
<td>Pre-Growth Seed Addition</td>
<td>Nil</td>
</tr>
<tr>
<td>Region Growth Intervals</td>
<td>12, 15,21,36,54</td>
</tr>
<tr>
<td>Post-Growth Seed Addition</td>
<td>12, 15,21,36,54</td>
</tr>
</tbody>
</table>

Table 2 summarizes the functionality of the adaptively generated thresholds at various scales of the ‘Cars’ image pyramid. The progressive nature of the region growth procedure can be clearly observed in this table, where sequential growth takes place at level 2 and in doing so employing all growth intervals. At level 1 the growth procedure shift to the higher gradient content and finally at the highest resolution is not employed at all because of the absence of any significant unsegmented regions so as to take full advantage of region growing. However for most images the MAPGSEG operates in a threshold range that covers regions of significant area in comparison to the image resolution, thus leaving all strong gradient regions unsegmented, as shown Fig. 22.
In Fig. 22 (a) and (b), the seed maps at the end of the region growth procedure for level1 and level0 respectively, are shown. Observe that high gradient or strong edge regions are consistently unsegmented due to the previously mentioned reasons. Since strong gradient regions occupy a very small area of the image we assign these regions with labels pre-existing in the seed map at the end of the growth procedure. In order to achieve this objective with minimal computational costs the MAPGSEG performs a combination of neighborhood and iterative morphological label assignment. Neighborhood label assignment is the process by which unassigned pixels are assigned to the label having the maximum count in its 3X3 neighborhood using a non-linear spatial filtering technique discussed in the seed transfer module subsection. The results of this procedure are shown in Figs. 22 (c) and (d), corresponding to Figs. 22 (a) and (b) respectively. However neighborhood label assignment alone is not sufficient to label pixels encompassed by 0’s (all zero neighborhoods). Thus the results of Figs. 22(c) and (d) are subjected to iterative morphological label assignment, where the all seeds are
dilated in an iterative fashion using a 3X3 a structuring element until there exists no unassigned pixel. The result of this operation for level1 and level0 are shown in Figs. 22 (e) and (f). Thus at the end of morphological label assignment all pixels in the seed map would have received a label. This post region growth processing is much more computationally efficient than method utilized by V2.2 where a major portion of the sequential growth procedure is repeatedly carried out assigning each unsegmented pixel to the most occurring parent in its neighborhood. In addition since the region growth map of V2.2 consists of much larger number of segments than the MAPGSEG for most images, the computational costs required to perform parent based label assignment is huge. V2.2 region growth map before and after unassigned residual pixel label assignment for the ‘Cars’ image is shown in Figs. 22(g) and (h). In these two images we see that the results of the region growth procedure in V2.2 is much more oversegmented that in case of the MAPGSEG where the output of the growth procedure is close representation of the eventual segmentation. This is primarily due to the use of uniform L*a*b* and a flexible and efficient adaptive threshold generation scheme.

3.4 INTER-RESOLUTION INFORMATION TRANSFER

The seed transfer module can be deemed as an interface for information transfer from one resolution to another in the MAPGSEG algorithm. This module (M4 as seen in Figs. 5 and 12) is responsible for identifying transferable regions between resolutions by exploiting the interim segmentation outputs as a-priori information, and acquiring regions of high confidence from the k\textsuperscript{th} level segmentation at the resolution of the (k-1)\textsuperscript{th} level. A block diagram illustrating all constituents requisite for multiresolution seed transfer is
shown in Fig. 23. The interface functionality of the module can be observed from the input/output relationship, where the input is at the $k^{th}$ level and the processing culminates in the $(k-1)^{th}$ level, as seen in Fig. 23.

![Seed Transfer Module Diagram](image)

**Fig. 23. Seed Transfer Module**

![Interim Output Images](image)

**Fig. 24. Interim output: (a) Level2, (b) Level1. Zero insertion yielding: (c) Level1, (d) Level0. Neighborhood label assignment: (e) Level1, (f) Level0.**

This module is initiated by a seed map up conversion of the $k^{th}$ level segmentation to the resolution of the $(k-1)^{th}$ level. This step is necessary to ensure that the data
transferred is in perfect alliance with the next higher resolution. To this effect, we first up sample the interim segmentation at k\textsuperscript{th} level by a factor of two along each dimension thus transmuting it to the subsequent higher dyadic resolution. The up conversion process is a two step method consisting of zero’s insertion followed by neighborhood pixel assignment. Zero’s insertion involves inserting zero’s between every pixel along both dimensions such that an MXN scale is transmuted to a (2*M) X (2*N) scale. In Figs. 24 (a) and (b) the interim segmentations acquired at level 2 and level 1, are respectively shown. The results at resolutions of 81X121 and 161X241 when subjected to zero insertion are transformed to resolutions 161X241(level 1) and 321X481(level 0), displayed in Figs. 24 (c) and (d). To overcome viewing limitations an encircled portion of the result obtained after zero insertion has been zoomed in and shown in Fig. 24 (g).

Having obtained the zero inserted images these are now subjected to neighborhood pixel assignment utilizing a non-linear spatial filter defined as:

\[
F(i, j) = \begin{cases} 
    a & \text{if } \max(\text{count( nonzero( } \beta)) ) = a \land \text{’}a\text{' is unique} \\
    \max( a) & \text{if } \max(\text{count( nonzero( } \beta)) ) = a \land \text{’}a\text{' is not unique}
\end{cases}
\]

(31)

where $\beta$ is the 3X3 neighborhood being operated. The filter $F(i, j)$ operates such that, given a 3X3 neighborhood, it assigns a nonzero label to the center pixel equivalent to the one with the maximum pixel count in that neighborhood, provided the maximum count is unique. If the maximum count is not unique then the largest of the maximum count value of the neighborhood in consideration is assigned to the center. In addition this filter is applied only to the neighborhood’s whose center pixel is zero, which from the aforementioned discussion is MXN numbered in a (2*M) X (2*N) scale image. The result of this non-linear spatial filtering operation on the images present in Figs. 24 (c),
(d), shown in Figs. 24 (e) and (f) respectively. These two images represent the a-priori information for their corresponding dyadic scales. In addition the a-priori information can be considered to be the estimates of the segmentation output at the current scale.

### 3.4.1 Gradient Quantization

Gradient quantization is required to determine the pixels in the estimated seed map that are acceptable with high confidence, and be passed on as a-priori information for an arbitrary decomposition level. In general when we decompose an image to certain number of levels, flat regions can be segmented with relative ease even at lowest scale in comparison to strong gradient regions. This is due to the fact they have not undergone much change in gradient content, but it is just their size that has decreased. However, in case of strong gradient regions decomposition results in loss of information content of these regions and so cannot be segmented with the same ease as done on the full resolution image. The MAPGSEG algorithm is designed to exploit this gradient characteristic for facilitating seed transfer. Thus we quantize the gradient map at every dyadic scale to differentiate between high and low confidence pixels at that scale. We choose the gradient quantization levels to be the adaptively generated threshold intervals, obtained at the commencement of the MAPGSEG algorithm. The quantized gradient map combined with varying size criterions (discussed later in the seed map cleaning procedure) is utilized to derive a-priori information at a certain decomposition level.
A quantized gradient map utilizing the initial threshold ($\lambda=5$), growth intervals at (12, 15, 21, 34, 56), as well the maximum gradient value in the histogram (111), is shown in Fig 25. Based on our previous discussion, for low scales (e.g. level1 for the ‘Cars’ image), pixels within low gradient quantization levels (5, 12) are chosen as potential high confidence regions. When we move to the next higher scale (level0), higher quantization intervals (15, 21) including the ones utilized in prior scales (5, 12) are chosen as confident a-priori information. In addition since the number of decomposition levels may not be equal to the number of quantization levels, we vary the increment in the number high confidence intervals depending on the number of decomposition levels, such that, apart from strong gradient regions, most low gradient regions shown up as a-priori
information at the highest scale. For the gradient confidence map with 7 quantization levels shown in Fig. 25 we increment the quantization levels in steps of 2 so as to distribute low gradient intervals evenly for levels1 and level0. (Note the level 2 the smallest resolution has no a-priori information associated with it). In Fig. 26 (a) and (b) a logical map and corresponding labeled color map portraying high confidence pixel locations obtained at level1 utilizing the two lowest quantization intervals (5 and 12), are shown respectively. The labeled color map was obtained after point wise multiplication of the estimated segmentation map (Fig. 24(e)) and the logical map shown (Fig. 26 (a)). Clearly, it can be observed that these quantization intervals have covered signification image area. Since the quantization intervals are low valued, the regions shown in Fig. 26(b) can be deemed as information can be passed on to the processing at the current resolution with no loss in gradient detail, thus reducing the computational requirements for segmentation at the current level. However observe that pixel based confidence was results in numerous minute seeds which are isolated as well as mutually adjacent to larger existent seeds as shown in Fig. 26(c). These minute seeds cannot be passed on as a-priori information as they would result in high computation requirements. Due to this reason they are eliminated from the labeled color map of high confidence pixels, a process referred to as seed map cleaning.

3.4.2 MUTUAL SEED BORDER REGIONS (MSBR)

The removal of minute seeds cannot be done by connected component analysis as it would only result in partial elimination of these seeds and simultaneously merge mutually adjacent ones, giving an undesired result. Therefore in order to be able to efficiently clean
up all isolated as well as mutually adjacent seeds we proceed to determine the Mutual adjacent Seed Border Regions (MSBR). MSBR is defined as all those pixels that are common to two regions labeled differently. These regions are obtained through non-linear spatial filtering in the MAPGSEG. The advantage of using nonlinear spatial filters is that it gives information in the image without actually manipulating individual pixel values.

Given a labeled seed map for facilitating the calculation of MSBR we first identify all pixel neighborhoods containing having multiple labels including 0. This is done by differencing each pixel in a neighborhood from its adjacent value and finding the total difference. If this value is 0 then all pixels have the same value (in the neighborhood) else their labels differ. Having obtained all such neighborhoods a validation matrix \( V \) is generated, given by

\[
V = \sum_{(i, j) \in \beta} \left| h(i, j) - \sqrt{h(i, j) \ast \text{mean}(\beta)} \right| \tag{32}
\]

where \( \beta \) is the 3X3 neighborhood being operated and \( h \) is the map consisting high confidence pixel locations. This validation matrix is required to segregate neighborhood’s consisting of multiple labels but having unique nonzero labels and the ones having multiple nonzero labels. Assume that we are computing \( V \) in a unique nonzero neighborhood \( \beta_1 \). In such a scenario the mean of \( \beta_1 \) will be equivalent to the nonzero label itself resulting in \( V \) for \( \beta_1 \) being 0. Similarly for multiple nonzero labels we obtain \( V > 0 \). We thus define MSBR as

\[
MSBR = \begin{cases} 
1 & \text{if } V > 0 \\
0 & \text{otherwise}
\end{cases} \tag{33}
\]
The MSBR for the high confidence pixels map at level 1 (Fig. 27 (a)) is shown in Fig. 27 (b). This logical map consists of 0’s and 1’s where a pixel having a value of 1 signifies that it is a part of an MSBR vice versa.

![Fig. 27. (a) High confidence pixel locations color map. (b) MSBR. (c) High confidence pixel locations color map after MSBR removal. (d) Large confident regions. (e) Large confident regions seed borders. (f) MSBR labels. (g) High Confidence MSBR regions. (h) A-priori information after border refinement.](image)

### 3.4.3 Seed Map Cleaning and Border Refinement

The MSBR computation is followed by its elimination from high confidence pixels map, resulting in all seeds being independent, sharing no common border, as shown in Fig. 27(c). This map with all independent seeds is subjected to connected component analysis to find all large seeds. A size criterion is placed to achieve this objective, and is unique for every scale. Starting from the (k-1)th level, the size criterions for connected component analysis is varied as 10*MSS, 5*MSS, and a criterion equivalent to MSS (dynamic seed addition size criterions) for all remaining scales, where each scale corresponds to certain gradient quantization levels to determine high confidence seeds. The result of employing connected component analysis is shown in Fig. 27 (d). From this figure we see all minute isolated and adjacent seeds that were earlier part of the seed map.
are no longer present. Although the aforementioned seed map cleaning procedure help eliminating all minute seeds, the large seeds that are present in the seed map have borders that are coarse in nature, due to MSBR removal.

The border refinement procedure is responsible for finding all MSBR that have labels present in the map consisting of large seeds, after subjecting it through the seed map cleaning protocol. These borders in turn are added back to large seeds map (Fig. 27(d)) to acquire seeds with smoother borders. Border refinement is a four step procedure. Initially all large seeds borders are extracted morphologically, as discussed earlier (shown in Fig. 27(e)). In addition all MSBR labels are obtained by performing a point wise multiplication of the MSBR map with the map consisting of all high confidence pixels. (Fig. 27(a)). The resultant MSBR labels are shown in Fig. 27(f). The labels in the Fig.27(f) that are adjacent members of the large seeds (Fig. 27(d)) are obtained by a point wise multiplication of large seed parent borders and MSBR labels, and the result is presented in Fig. 27(g). These labels adjacent to large seeds are now added to the large seeds map to acquire smooth region borders, displayed in Fig. 27(h). The border refinement procedure is the culmination point of the seed transfer module (see Fig. 23). The map obtained at the end of border refinement is considered to the a-priory information for the current dyadic scale at which most seed transfer processing is done. The following section will briefly discuss module 5 and module 6.

3.5 Texture Channel Generation and Region Merging

This section largely recapitulates the texture modeling (M5) and statistical merging procedure (M6) employed in V2.2, most part of which, have been left unchanged in the
MAPGSEG algorithm. However, a few modifications have been discussed. Most problems in image segmentation algorithms are caused by the presence of regions that contain distinct patterns composed of multiple shades of colors, causing over-segmentation and misinterpretation of the edges surrounding these regions. Due to the extensive presence of distinct patterns in images, V2.2 utilized an entropy-based texture descriptor. The entropy of various image segments is calculated and the ones with similar entropy values are grouped together. However, in order to achieve computational efficiency by avoiding joint entropy calculation between channels, quantization is done by uniformly dividing the 8-bit encoded L*a*b* cube into small boxes, and mapping all information that fall within each box to the color and luminance value at the center of that box (see Fig. 28). The advantage of quantizing the L*a*b* cube over the RGB color cube is that, unlike uniform L*a*b* data, if nonuniform RGB data is uniformly quantized, a constant distance between any two quantization levels will result in large variation of perceptual color difference [38]. After the quantization process, each pixel of an image can be indexed to one of the 216 representative levels, effectively reducing the probability of each level occurring to a one-dimensional random variable. To create a texture channel, the local entropy is computed in a 9-by-9 neighborhood around each pixel of the indexed image, and the resulting value is assigned to the center pixel of the neighborhood. This model of texture is then utilized in the region merging process.

The merging module is utilized to merge regions as deemed necessary, which are over-segmented in the growth procedure due to occlusions and minor texture differences. A multivariate analysis of all independent regions utilizing L*a*b* and texture is carried out based on the procedure described in [39]. The essence of this method is to investigate
the possibility that multiple groups with various features are associated with a single factor that enables them to be merged together. The multivariate analysis involving the Mahalanobis distance calculation between groups, is carried out on a matrix of dimensions equivalent to the total number of pixels in the image and number of variables (L*, a*, b*, texture) per pixel, for convenient handling of groups. Similar regions are initially found based on the minimum distance measure corresponding to maximum similarity. These are merged by appropriate relabeling of regions and increasing the similarity value. The process is repeated until the similarity value exceeds a user defined threshold or the maximum number of acceptable groups is reached.

![Fig. 28. Euclidean space representation of L*a*b*.](image)

In V2.2 the similarity value and the maximum number of acceptable groups (MaxNg) were set to 2 and 50 respectively. In addition the increment of the similarity value was carried out in steps of 0.1. The merging limit was 2 was found to be a too low a value for many natural scene image. This resulted in the merging algorithm to be computationally expensive as the increment of 0.1 did not bring about any significant change in the output segmentation and in most cases the algorithm iterated many times to reach the desired
number of 50 groups in the output. However in the MAPGSEG we increase the similarity value to 4 and the similarity value increment is made a function of the maximum number of groups. Initially, when the region growth and texture maps are fed to the merging module the number of groups in the seed map is checked. If this number is large then a large increment in similarity value is utilized between merging iterations, and in due functioning of the module, as the number of groups in the seed map approached MaxNg the similarity increment is reduced. This adaptive merging methodology decreased the run time of the merging module. Overall the similarity increment was varied from 10% to 50% of MaxNg for a given image. In addition we modified the maximum number of acceptable groups to be 40. However all parameters discussed here could be varied depending on the application in which the algorithm is being used.
Chapter 4: QUANTITATIVE EVALUATION OF SEGMENTATION METHODS

In recent years, the introduction of new image segmentation techniques for handling diverse applications has motivated the need for evaluating these methodologies effectively. However segmentation being an ill defined problem with no unique/perfect solution, the evaluation of the obtained results necessitated a comparison to be made against all possible segmentations for an image. Multiple solutions primarily occur due to the fact that the level of detail at which an image is perceived is highly inconsistent from one individual to another. Thus a good evaluation metric for the segmentation problem must take into consideration some of the following requirements [40]:

- The metric should not yield cases where the evaluation produces a high value in spite of the automatic segmentation result being nowhere closely similar to any one of its corresponding human segmentation results.
- No assumptions should be made about the data involving labels assignment and region sizes.
- The measure should be designed such that it penalizes the final evaluation score when the automatic segmentation does not distinguish between regions that humans can distinctly identify. Conversely, the metric should also allow for fewer penalties on the evaluation score when the segmentation output is not favorable in regions which are visually ambiguous to humans. This is also known as adaptive accommodation of label refinement.
The metric should facilitate comparison amongst possible segmentations of the same image as well as segmentations of different images.

In this regard, to objectively measure the quality of our segmentation results, we have implemented a recently proposed measure of similarity, referred to as the Normalized Probabilistic Rand (NPR) index [40], which a generalization of the Rand Index originally proposed by William Rand [41].

3.1 RAND INDEX

The Rand Index facilitates the comparison of two segmentations utilizing pair wise label relationships. Let $S$ and $S'$ be two segmentations with corresponding label assignments $\{l_i\}$ and $\{l'_i\}$ for N points $X = \{x_i\}$ where $i = 1, 2, ..., N$. The Rand Index ($R$) used for comparing the two segmentations is defined as the ratio of number of pixel pairs that share the same label relationship in $S$ and $S'$. This is represented as:

$$R(S, S') = \frac{1}{\binom{N}{2}} \sum_{i,j \neq j} \left[ I(l_i = l_j \land l'_i = l'_j) + I(l_i \neq l_j \land l'_i = l'_j) \right]$$

Here $I$ is the identity function and the denominator represents all possible unique pixel pairs in a dataset of N points. It is important to note that the number of unique labels in $S$ and $S'$ may differ from each other, and having this quantity equal in both segmentations is just a special case. This measure varies from 0 to 1, where 0 represents complete dissimilarity and 1 symbolizes that $S$ and $S'$ are identical. In addition the Rand Index does accommodate label refinement during evaluation.
### 3.2 Probabilistic Rand (PR) Index

The Probabilistic Rand Index enables the evaluation between segmentation taking into consideration the statistical nature of the Rand Index and combining it with the competence of accommodating label refinement. The PR index allows comparison of a test segmentation result to a set of multiple ground-truth segmentation images (or human/manual segmentation images) through a soft non-uniform weighting of pixel pairs as a function of the variability in the ground-truth set [40].

Let \( \{S_1, S_2, \ldots, S_K\} \) be a set of ground truth segmentations of an image \( X = \{x_i\} \) with \( N \) points where \( i = 1, 2, \ldots, N \). Let the result of an unsupervised segmentation algorithm which is to be compared to the manually labeled set, be represented as \( S_{test} \). Further let \( \{l^{s_{test}}_i\} \) and \( \{l^{s_k}_i\} \) represent the label assignment of a pixel \( i \) in \( S_{test} \) and the \( K^{th} \) manual segmentation \( S_k \) respectively. Let \( \hat{l}_i \) denote the set of “true labels” for a pixel \( x_i \).

Utilizing the aforementioned data, the probability of a label relationship between a pair of pixels \( x_i \) and \( x_j \) is defined as:

\[
P(\hat{l}_i = \hat{l}_j) = \frac{1}{K} \sum_{k=1}^{K} I(l_i^{s_k} = l_j^{s_k}) = p_{ij}
\]

and

\[
P(\hat{l}_i \neq \hat{l}_j) = \frac{1}{K} \sum_{k=1}^{K} I(l_i^{s_k} \neq l_j^{s_k}) = 1 - P(\hat{l}_i = \hat{l}_j) = 1 - p_{ij}
\]

The Probabilistic Rand (PR) Index is now defined as:

\[
PR(S_{test}, \{S_K\}) = \frac{1}{\binom{N}{2}} \sum_{i,j \neq j} I(l_i^{s_{test}} = l_j^{s_{test}})P(\hat{l}_i = \hat{l}_j) + I(l_i^{s_{test}} \neq l_j^{s_{test}})P(\hat{l}_i \neq \hat{l}_j)
\]

Equation (37) can be rewritten as:
\[ PR(S_{\text{test}}, \{S_K\}) = \frac{1}{\binom{N}{2}} \sum_{i<j} c_{ij} p_{ij} + (1-c_{ij})(1-p_{ij}) \]  \hspace{1cm} (38)

where \( c_{ij} = I(l_i^{S_{\text{test}}} = l_j^{S_{\text{test}}}) \). The PR Index takes the same range of values as the Rand Index, from 0 to 1 where 0 signifies the most dissimilarity and 1 represents a perfect match to human segmentations. In addition since \( c_{ij} \in \{0,1\} \) Equation (38) takes the form:

\[ PR(S_{\text{test}}, \{S_K\}) = \frac{1}{\binom{N}{2}} \sum_{i<j} p_{ij}^{c_{ij}} (1-p_{ij})^{(1-c_{ij})} \]  \hspace{1cm} (39)

In Equation (39) \( p_{ij}^{c_{ij}} (1-p_{ij})^{(1-c_{ij})} \) represents the likelihood of pixel pairs \( x_i \) and \( x_j \) taking values \( l_i^{S_{\text{test}}} \) and \( l_j^{S_{\text{test}}} \) under the defined Bernoulli distribution. In addition, Unnikrishnan et al. [40] showed that the computational complexity of the PR index is \( O(KN + \sum L_k) \). In practice though the PR Index accommodates label refinement wherever required it suffers from little variation in its values over a diverse set of images. This is due to the small range of the PR Index and the variation in the maximum value over a set of images. In order to overcome this problem Unnikrishnan et al. [40] proposed the Normalized Probabilistic Rand (NPR) Index for the objective evaluation of segmentation outputs.

### 3.3 Normalized Probabilistic Rand (NPR) Index

The NPR evaluation method compares results obtained from a tested algorithm to a set of manually segmented ones, meeting all the requirements stated at the beginning of this section. The impact and effectiveness of any measure of similarity is primarily based on the reference to which it is measured. In segmentation this reference may be the
expected value of the similarity measure, computed utilizing the variation and randomness in the set of input images. The NPR metric is designed utilizing the aforementioned principle. The Normalized Probabilistic Rand (NPR) Index is given by:

\[
NPR = \frac{PR - \text{Expected index}}{\max[PR] - \text{Expected index}} = \frac{PR - E[PR]}{\max[PR] - E[PR]}
\]  

(40)

It can be observed from Equation (40), that the NPR Index is normalized with respect to the expected value of the PR Index. This results in the modified index which is the NPR to have a much higher range than the PR making it a much more sensitive evaluation metric. Here the maximum value of the PR Index is chosen to be 1 (\(\max[PR]=1\)). The expected value of the PR Index (\(E[PR]\)) is obtained utilizing Equations (37) and (38) as:

\[
E[PR(S_{test}, \{S_K\})] = \frac{1}{N \choose 2} \sum_{i<j} \left[ E[I(l_i^{s_{test}} = l_j^{s_{test}})]p_{ij} + E[I(l_i^{s_{test}} \neq l_j^{s_{test}})](1-p_{ij}) \right]
\]  

(41)

\[
E[PR(S_{test}, \{S_K\})] = \frac{1}{N \choose 2} \sum_{i<j} \left[ p_{ij}p_{ij} + (1-p_{ij})(1-p_{ij}) \right]
\]  

(42)

To make the computation of \(p_{ij} = E[I(l_i^{s_{test}} = l_j^{s_{test}})]\) meaningful Unnikrishnan et al. [40] proposed computing it from segmentations of all images for all unordered pixel pairs \((i, j)\). If \(\Phi\) is the number of images in the database and \(K_\Phi\) is the number of ground truths per image then the value of \(p_{ij}\) can be computed by the following equation:

\[
p_{ij} = E[I(l_i^{s_{test}} = l_j^{s_{test}})] = \frac{1}{\Phi} \sum_{\Phi} \frac{1}{K_\Phi} \sum_{k=1}^{K_\Phi} I(l_i^{s_k} = l_j^{s_k})
\]  

(43)

Here \(p_{ij}\) signifies that \(E[PR(S_{test}, \{S_K\})]\) is a weighted sum of \(PR(S_{test}, \{S_K\})\). Prior to the evaluation, our results were re-labeled such that each independent segment had a
different label, owed to our algorithm’s capability of handling occlusions, which may result in disconnected regions being uniquely labeled. In addition the segmentation results at various resolutions were up-scaled to the size of the input original utilizing the methodology explained in the Section 3.4, and then were evaluated using the NPR Index.
Chapter 5. RESULTS AND DISCUSSIONS

The MAPGSEG results were benchmarked qualitatively and quantitatively - using the Normalized Probabilistic Rand index (NPR) [40] - against several popular algorithms on the same test bed of manually segmented images (ground truth). Our results are compared against those from a spectrum of published segmentation algorithms such as GRF [6], JSEG [23], DCGT [32], GSEG-V2.2 [33], and a computational time analysis was also performed to furnish a fair indication of the overall performance of the MAPGSEG algorithm. The NPR index requires a set of images each having multiple manual segmentations, for evaluation. Such a set, consisting of 1633 manual segmentations for 300 images of dimension ~321X481, created by 30 human subjects, has been made publicly available by the University of California at Berkeley [42]. An additional (randomly selected) 445 images with dimension ~750X1200 were also utilized for accessing the performance of the MAPGSEG against its single scale version. The entire testing database (745 images) was segmented on the same machine having a Pentium® 4 CPU 3.20GHz, and 3.00 GB of RAM. The GRF, DCGT and GSEG-V2.2 algorithms are run from the executable files and MATLAB code provided by the Rochester Institute of Technology, while the JSEG algorithm was run from a different executable file provided by the University of California at Santa Barbara. The proposed method was implemented using MATLAB version R2007a.

The results of the MAPGSEG algorithm at different stages are presented in Figs. 29(a)-29(f). The original RGB input image pyramid and its CIE L*a*b* counterpart, are shown in Fig. 29(a) and (b). The outcome of gradient computation on the color
converted input images at various resolutions, is shown in Fig. 29(c). The seed maps at the end of the region growth procedure, obtained utilizing thresholds that are generated adaptively, are displayed in Fig. 29(d). Observe that these region growth maps are oversegmented, due to reasons specified in Section IIIE. The texture channels generated (at various scales) using color quantization and local entropy calculation are depicted in Fig. 29(e). Finally, the interim and final segmentation maps at the end of the region merging algorithm are shown in Fig. 29(f).

Fig. 29. Multiresolution representation of: (a) Original RGB ‘Star Fish’ image, (b) Color converted ‘Star Fish’ image, (c) Color gradient, (d) Seeds maps at the end of progressive region growth, (e) Entropy based texture maps, (f) Interim and final segmentation outputs.
Fig. 30. Interim Segmentation at: (a) Level2, (b) Upconverted to Level1. (c) A-priori information at level1. Interim Segmentation at: (d) Level1, (e) Upconverted to Level0. (f) A-priori information at level0. (g) MAPGSEG final segmentation output.

In addition, Fig. 30 demonstrates our multiresolution seed transfer procedure in the MAPGSEG framework. The level2 segmentation result of the ‘Star fish’ image and its up converted version to level1 are shown in Figs. 30(a) and (b) respectively. This unconverted seed map is the estimate of the segmentation result at level1. This estimate is passed through the seed transfer module to give the a-priori information for level1, as shown in Fig. 30(c). Utilizing this a-priori information the algorithm arrives at an interim
result at level1 (shown in Fig. 30 (d)). The aforementioned procedure is repeated at level0 as shown in Figs. 30(e), (f), (g).

Clear performance advantages of the MAPGSEG algorithm can be viewed in Figs. 29 and 30. In Fig. 29 (b) the increase in gradient detail from the lowest to the highest resolution is visible, which supports our hypothesis of selecting flat regions at low gradient quantization levels and vice versa. As a result large flat regions can be segmented at the lowest resolution, up scaled to the size of the input image, and in turn be used for various applications. The ability of the MAPGSEG to fulfill all these functionalities projects it as a potential performance enhancement tool in any application it is used. In addition, it can be observed that the seed maps obtained at end of the region growth procedure improves with each higher scale to the extent that at the highest resolution it is a close representation of the eventual segmentation. This signifies lesser work to the region merging algorithm at successive scales rendering the algorithm to be more computational efficient than its single scale version GSEG-V2.2.

Results obtained from the MAPGSEG in comparison to the previously mentioned segmentation methods, are shown in Figs. 31 - 35. The ‘Church’ image in Fig. 31(a) represents a moderately complex image. Observed that in Figs. 31(b), (c), (d), (e) the GRF, JSEG, DCGT and GSEG algorithms over segment this image (sky and dome regions) due to illumination disparity seen in various regions. However, our algorithm employs the CIE L*a*b* color space where the L* channel contains the luminance information in the image, incapacitates the illumination problem. Similar results can be seen in the ‘Parachute’ image. All algorithms apart from the MAPSEG, over segment the sky and mountain regions, as seen in Figs. 32(b), (c), (d), and (e).
Segmenting textured regions becomes a hard challenge when regions with diverse textures are extremely similar in color. Here a good texture descriptor is indispensable. Fig. 33(a) represents an image of a Cheetah which has a skin tone that almost matches its background making it extremely difficult to segment it based on just color information. The GRF, JSEG, DCGT results shown in Figs. 33(b), (c) and (d) illustrates the effect of an indistinct texture descriptor for segmentation. The GSEG-V2.2 (Fig. 33(e)) algorithm in comparison has been able to achieve a good segmentation. However, the use of the RGB space for color similarity has yielded incoherence in the segmentation of the
background. This problem has been overcome in the MAPGSEG due to the use of the L*a*b* color space, shown in Fig. 33(f). The same anomalies spotted in the parachute and cheetah image can be seen in Fig. 34. Observe in Fig. 34 (b), (c), and (e) that lake region is segmented into two regions due to illumination variation. In addition the DCGT algorithm merged the tree bark region due to lack of a proper texture descriptor. Here again, the MAPGSEG is successful in overcoming illumination and color space non-uniformity problems. Our algorithm like the GSEG has the ability to segment fine details such as text with great efficiency unlike the GRF, JSEG, and DCGT as illustrated by the results in Fig. 35. Observe that the word ‘Castrol’ as seen in Fig. 35(a) is segmented out at multiple locations with near perfection by the MAPGSEG algorithm as seen in Fig. 35(f). The GRF, JSEG and GSEG cause over segmentation in regions representing the motorway due to varying illumination and occlusion by the foreground objects, as see in Figs. 35(b), (c), and (e). Thus, the efficiency of the MAPGSEG algorithm in handling the background occlusion problem is emphasized in the ‘Cars’ results.

![Fig. 33. Cheetah Results: (a) Original, (b) GRF, (c) JSEG, (d) DCGT, (e) GSEG-V2.2, (f) MAPGSEG.](image-url)
In the following figures, shown are the interim and final segmentation outputs of our algorithm in comparison to the DCGT, GSEG and human segmentations provided by the University of California at Berkeley. In Fig. 36 (b), (c) the results of the DCGT and GSEG from the ‘Island’ image have been oversegmented in the lake region due to illumination variation. Conversely the MAPGSEG is able to segment this region as one, even at the smallest resolution (see Fig. 36(f)). It is imperative to remember that the segmentations at lower resolutions other than the original are being displayed after up
scaling them to the size of the original input utilizing our up scaling methodology in Section 3.4.

In addition the human segmentations for the island image are shown in Fig. 37. Observe the closeness of the up-scaled segmentations of all levels of the MAPGSEG to the human segmentations. This signifies the algorithms effectiveness and robustness to scalability.

Fig. 36. Island Results: (a) Original, (b) DCGT, (c) GSEG. MAPGSEG: (d) Level2, (e) Level1, (f) Level0.

Fig. 37. Human segmentation for the ‘Island’ image provided by University of California, at Berkeley.
In Fig. 38 the ‘Asian’ image is portrayed. The DCGT fails on this image due the lack of a texture descriptor. Though this problem is overcome in the GSEG, it can be observed that the back ground is oversegmented which is not a favorable result when compared to the human segmentations shown in Fig. 39. However the MAPGSEG has been successful in segmenting the background as one region to a large extent as can be seen in Fig. 38(f) when compared to the images in Fig. 39. In addition the level of detail in Fig. 38 (f) can be observed to be similar to most of the human segmented image unlike the GSEG.
algorithm which over segments the robes of the two people and in one case merges the hand of the person with the background. Furthermore, the closeness of level1 and level0 results can be observed, signifying faster processing time for segmentation.

In the NPR evaluation, the normalization factor was computed by evaluating the Probabilistic Rand (PR) for all available manual segmentations, and the expected index (E [PR]) obtained was 0.6064. A distributional comparison of our evaluation, of the segmentation results for 300 images (of size 321X481) in the Berkeley database, obtained from the GRF, JSEG, DCGT, GSEG and MAPGSEG is displayed in Fig. 42. In Fig. 42 (a), it can be observed that the distribution for the GRF is weighted more towards the lower half of the distribution with a minimal NPR value going as low -0.9. A similar observation can made with the NPR distribution of the DCGT algorithm in Fig. 42 (c). An improvement over the previous two algorithms is the JSEG in Fig. 42 (b) were the values are weighted more towards the higher end of NPR score distribution. More favorable NPR scores can be observed in the case of the GSEG and MAPGSEG in Figs. 42(d) and (e).

Fig. 42. NPR scores distribution for 300 images of the Berkeley database: (a) GRF, (b) JSEG.
The actual improvement can be seen by superimposing all these distributions (as seen in Fig. 42 (f)), and observing the number of segmentation scores that fall within the range of very good segmentation results \([0.7 < \text{NPR} < 1]\). These numbers for the GRF, JSEG, DCGT, GSEG, and MAPGSEG were computed as 38, 65, 62, 79 and 85 respectively (see Table 3). This indicates that approximately a third of the images segmented using our algorithm match closely to the segmentations performed by humans.
TABLE 3: Evaluation of MAPGSEG using 300 images of the Berkeley database in comparison to published work

<table>
<thead>
<tr>
<th></th>
<th>GRF</th>
<th>JSSEG</th>
<th>DCGM</th>
<th>GSIG</th>
<th>MAPGSEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Time (sec)</td>
<td>240.0</td>
<td>162</td>
<td>86.9</td>
<td>35.1</td>
<td>11.1</td>
</tr>
<tr>
<td>Avg. NPR</td>
<td>0.357</td>
<td>0.439</td>
<td>0.295</td>
<td>0.487</td>
<td>0.496</td>
</tr>
<tr>
<td>NPR&lt;0.7</td>
<td>38</td>
<td>65</td>
<td>62</td>
<td>79</td>
<td>85</td>
</tr>
<tr>
<td>Environment</td>
<td>C</td>
<td>C</td>
<td>MATLAB</td>
<td>MATLAB</td>
<td>MATLAB</td>
</tr>
</tbody>
</table>

TABLE 4: Evaluation of various levels of MAPGSEG using 300 images of the Berkeley database

<table>
<thead>
<tr>
<th></th>
<th>Level0</th>
<th>Level1</th>
<th>Level2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Time (sec)</td>
<td>11.1</td>
<td>4.6</td>
<td>2.3</td>
</tr>
<tr>
<td>Avg. NPR</td>
<td>0.496</td>
<td>0.495</td>
<td>0.487</td>
</tr>
<tr>
<td>NPR current level</td>
<td>-</td>
<td>99.7%</td>
<td>98.09%</td>
</tr>
<tr>
<td>Vs. NPR Level 10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE 5: Evaluation of various levels of MAPGSEG using 445 large resolution images in comparison to GSEG

<table>
<thead>
<tr>
<th></th>
<th>Level0</th>
<th>Level1</th>
<th>Level2</th>
<th>Level3</th>
<th>GSIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Time (sec)</td>
<td>35.7</td>
<td>15.6</td>
<td>11.2</td>
<td>8.7</td>
<td>177.2</td>
</tr>
</tbody>
</table>

A comparison of our evaluation, for the segmentation results obtained from the five methods, is displayed in Table 3. This table shows that our algorithm has the highest average NPR score, and the lowest average run time per image, showing that our algorithm is achieving quality segmentations with the least computational complexity, considering the different environments in which they were developed. Table 4 exhibits qualitative and quantitative comparison of various levels of the MAPGSEG, after all interim outputs are up scaled to the size of original input. Comparing the average level2 NPR score to that that of level0 we see that even at level2, the outputs obtained are more than 98% of segmentation quality at the highest resolution(level0), and acquired as fast as
2.3 seconds an image. Further more from Table 3 and 4 it can be seen that the MAPGSEG is three times faster than the GSEG with marginal improvement in segmentation quality. Table 5 shows the computational time comparison of various levels of the MAPGSEG to the GSEG for 445 large resolution images (~750X1200). Here it is seen that the GSEG has an average runtime in minutes (177.2 sec ≈ 2.9 minutes) in comparison to our algorithm with an overall runtime of 35.7 seconds, almost 5 times faster than its single scale version. In Fig. 43 shown are graphical representations of the computational efficiency of the MAPGSEG in comparison to other algorithms. Additional results of the MAPGSEG in comparison to the GSEG are shown in Fig.44.

Fig. 43. Computational time comparison utilizing Berkeley database (321X421): (a) MAPGSEG, GSEG and DCGT, (b) Various levels of MAPGSEG. Computational time
comparison utilizing large resolution image database (750X1200): (c) MAPGSEG, GSEG, (d) Various levels of MAPGSEG.

Fig. 44. Additional segmentation results: (a) Original, (b) GSEG, (c) MAPGSEG (Level0).
Chapter 6: CONCLUSIONS AND FUTURE WORK

This work presents a computationally efficient method designed for fast unsupervised segmentation of color images with varied complexities in a multiresolution framework. This Multiresolution Adaptive and Progressive Gradient SEGmentation (MAPGSEG) algorithm is primarily based on adaptive gradient thresholding, progressive region growth involving distributed dynamic seed addition, multiresolution seed transfer and culminates in a unique region merging procedure. The algorithm has been tested on a large database of images including the publicly available Berkeley database [42], and the quality of results show that our algorithm is robust to various image scenarios at different scales and is superior to the results obtained on the same image when segmented by other methods, as can been seen in the results displayed. The accomplishments of the proposed work are summarized below.

1) The MAPGSEG is an efficient method designed for fast segmentation of color images at various resolutions.

2) A low level image understanding tool with an efficient break up of detail present in the image.

3) An overall improvement factor of 3X (Berkeley database) and 5X (randomly selected images-RIT database) over its single scale version.

4) The significant improvement in computational complexity has been achieved maintaining benchmark segmentation quality.

5) A potential solution to for fast and intelligent object/region based rendering, with a good balance between quality and speed.
The objectives of future research are to:

1) Investigate the potential failure modes of the existing algorithm in the single/multiscale framework, and design effective solutions/methodologies to overcome them. Some identified drawbacks are:
   - Loss of perceptually important content such as manmade structures due to its existence in low gradient regions or insignificant size in comparison to the image resolution.
   - Primitive texture characterization.
   - CIE L*a*b* suffers from non uniformity in shadow regions and contrasting illumination conditions. To overcome this we plan to investigate the effects of utilizing uniform color difference formulae such as the CIE94, CIEDE2000 for image segmentation, in comparison to approximately uniform CIEL*a*b and non-uniform RGB, for color representation.

2) Extend our existing state of the art image segmentation algorithm to effectively handle video streams, utilizing spatial as well as temporal information by taking advantage of temporal dependencies between frames.

3) The later stages of algorithm development will also involving developing an unsupervised evaluation methodology for our results, eventually integrated within the framework of our algorithm in an adaptive feedback mechanism. This will enable a user/application to analyze the performance of the algorithm as well as have control on the quality of expected segmentation results.

4) Natural objects including humans tend to be of different types. These can be segmented on the basis of brightness, and spectral signature. Different imaging
modalities (hyperspectral/multispectral) provide different levels of spatial and spectral resolution, and no two modalities provide identical segmentations. It is our intention to extend our highly effective GSEG/MAPGSEG algorithm to perform segmentation using this enhanced format.

5) Utilizing the GSEG/MAPGSEG segmentation results as an initial estimate for developing a probabilistic model for the current approach utilizing Bayesian Networks.
REFERENCES


