

ABSTRACT

Title of dissertation: TEXTURE-BASED SEGMENTATION
AND FINITE ELEMENT MESH
GENERATION FOR HETEROGENEOUS
BIOLOGICAL IMAGE DATA

Prabhakar Reddy, Gudla, Ph.D., 2005

Dissertation directed by: Dr. Hubert J. Montas
Biological Resources Engineering Department

The design, analysis, and control of bio-systems remain an engineering challenge. This is mainly due to the material heterogeneity, boundary irregularity, and nonlinear dynamics associated with these systems. The recent developments in imaging techniques and stochastic upscaling methods provides a window of opportunity to more accurately assess these bio-systems than ever before. However, the use of image data directly in upscaled stochastic framework can only be realized by the development of certain intermediate steps. The goal of the research presented in this dissertation is to develop a texture-segmentation method and a unstructured mesh generation for heterogeneous image data.

The following two new techniques are described and evaluated in this dissertation.

1. A new texture-based segmentation method, using the stochastic continuum concepts and wavelet multi-resolution analysis, is developed for characterization of heterogeneous materials in image data. The feature descriptors are

developed to efficiently capture the micro-scale heterogeneity of macro-scale entities. The materials are then segmented at a representative elementary scale at which the statistics of the feature descriptor stabilize.

2. A new unstructured mesh generation technique for image data is developed using a hierarchical data structure. This representation allows for generating quality guaranteed finite element meshes.

The framework for both the methods presented in this dissertation, as such, allows them for extending to higher dimensions. The experimental results using these methods conclude them to be promising tools for unifying data processing concepts within the upscaled stochastic framework across biological systems. These are targeted for inclusion in decision support systems where biological image data, simulation techniques and artificial intelligence will be used conjunctively and uniformly to assess bio-system quality and design effective and appropriate treatments that restore system health.

TEXTURE-BASED SEGMENTATION AND FINITE ELEMENT
MESH GENERATION FOR HETEROGENEOUS BIOLOGICAL
IMAGE DATA

by

Prabhakar Reddy, Gudla

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2005

Advisory Committee:

Hubert J. Montas, Ph.D., Chair/Advisor

Adel Shirmohammadi, Ph.D., Co-Advisor

Yang Tao, Ph.D.

Dennis Healy, Ph.D.

Andrew S. Kane, Ph.D.

©Copyright by
Prabhakar Reddy, Gudla
2005

ACKNOWLEDGMENTS

This dissertation and the research that it describes was made possible thanks to the enthusiastic support of many people.

First and foremost, my sincere gratitude goes to my major advisor Dr. Hubert J. Montas. I thank him for the depth and breadth of his guidance, for his constant willingness to help, for his kind and supportive words at the uncertain times, and for the example that he sets in his commitment to integrity. It has been a great pleasure to work with and learn from such an exemplary individual.

I would like to thank my co-advisor, Dr. Adel Shirmohammadi, without his ideas and continuous guidance this dissertation would have been a distant dream. To only thank Dr. Yang Tao, Dr. Dennis Healy, and Dr. Andrew Kane for their roles as members of my dissertation committee is to not give them enough credit. They have helped me enormously along the way by involving themselves in my research and collaborating on research projects.

I extend my gratitude to the faculty, students, and staff of the Department of Biological Resources Engineering at the University of Maryland at College Park, who over the years have helped me in many ways. I am especially thankful for the organizational support and assistance from the staff members.

I am indebted to the people and institutions whose support has made my research possible. I would like to thank the tax payers of both the Republic of

India and the United States of America for bearing much of the financial burden for most of my higher education. I bow my head to all those who have made and continue to make sacrifices in order to make it possible for people like me to pursue their dreams. My work has been funded by National Science Foundation (Grant No. 0134424), College of Agriculture and Natural Resources through Maryland Cooperative Extension Service (MCES) and Maryland Agricultural Experimentation Station (MAES).

I warmly thank my friends and family for their kindness, understanding and support. I would not be here without my parents and their relentless patience. Thanks also to both my elder siblings for their constant encouragement and belief in my career goals.

My roommates have been a crucial factor in my finishing this dissertation smoothly. I would like to express my gratitude to Pavan, Anil, Kiran, Raju, and Sasi for their friendship and support.

Lastly, it's almost impossible for me to remember all those who have helped me through my doctoral studies. I apologize to those that I might have inadvertently left out.

TABLE OF CONTENTS

List of Tables	viii
List of Figures	xi
1 Introduction	1
1.1 Motivation	1
1.1.1 Image Segmentation	6
1.1.2 Unstructured Mesh Generation	10
1.2 Thesis Objectives	12
1.3 Contribution to Knowledge	13
1.4 Overview of Chapters	14
2 Texture-Based Image Segmentation	15
2.1 Introduction	15
2.2 Literature Review	17
2.2.1 Patterns, Features, and Feature Space	21
2.2.1.1 Structural-Based Texture Descriptors	23
2.2.1.2 Statistics-Based Texture Descriptors	24
2.2.1.3 Model-Based Texture Descriptors	24
2.2.1.4 Spatial/Spatial-Frequency Texture Descriptors	26
2.2.2 Classification Schemes	31
2.2.3 Summary	34
2.3 Objectives	37

2.4	Materials and Methods	38
2.4.1	Development	38
2.4.1.1	Pre-Processing and Sample Selection	42
2.4.1.2	Feature Descriptors	43
2.4.1.3	Identification of RES support	51
2.4.1.4	Classification Schemes	52
2.4.1.5	2-D Real Images	59
2.4.1.6	2-D Synthetic Images	61
2.4.2	Evaluation	64
2.4.2.1	3-D Real CT Head Data	67
2.4.2.2	3-D Simulated MR Brain Data	69
2.4.2.3	3-D Real MR Brain Data	73
2.5	Results and Discussions	77
2.5.1	Development	77
2.5.1.1	Feature Descriptors	81
2.5.1.2	Identifying RES Support and Classification	83
2.5.1.3	2-D Real Images	85
2.5.1.4	2-D Synthetic Images	93
2.5.2	Evaluation	97
2.5.2.1	3-D Real CT Brain Data	101
2.5.2.2	3-D Simulated MR Brain Data	107
2.5.2.3	3-D Real MR Brain Data	114
2.6	Conclusions and Future Research	120

3	Finite Element Mesh Generation Of Image Data	128
3.1	Introduction	128
3.2	Literature Review	131
3.2.1	Structured and Unstructured Meshes	132
3.2.2	Mesh quality for numerical methods	133
3.2.3	Unstructured Mesh Generation Techniques	140
3.2.3.1	Conforming Delaunay Meshing	141
3.2.3.2	Advancing Front Meshing	150
3.2.3.3	Constrained Delaunay Meshing	153
3.2.3.4	Hierarchical Meshing	157
3.2.3.5	Energy Minimization Image Meshing	163
3.2.3.6	Interval Volume Tetrahedralization	165
3.2.4	Summary	169
3.3	Objectives	170
3.4	Materials and Methods	171
3.4.1	Development	171
3.4.2	Analysis	178
3.5	Results	184
3.5.1	Development	184
3.5.2	Analysis	198
3.6	Conclusions and Future Work	211
4	Conclusion	216

A Effect of RES on SWA Classification Accuracy	220
Bibliography	221

LIST OF TABLES

2.1	Combination of feature vectors and their corresponding dimensional- ity at scale j	52
2.2	Summary of three test suites (T-3, T-4 and T-5) used for evaluating the new segmentation method.	66
2.3	Fresh CT slices from VWP categorized into different sections.	68
2.4	Details of the eleven coronal brain-only T1-weighted MR brain scans from IBSR.	76
2.5	Combination of feature vectors and their corresponding dimensional- ity at scale j	80
2.6	Average LOOE, ϵ_l and modified LOOE, ϵ_m expressed in percent for the soil image in Figure 2.6(a) at RES support of 3 using GM, MD, and LVQ classifiers.	91
2.7	Average LOOE, ϵ_l and modified LOOE, ϵ_m for the muscle image in Figure 2.6(b) at RES support of 4 using GM, MD, and LVQ classifiers.	92
2.8	Average LOOE, ϵ_l and modified LOOE, ϵ_m for the landscape image in Figure 2.6(c) at RES support of 4 using GM, MD, and LVQ classifiers.	93
2.9	SWA segmentation results for mosaics T-2(a)–T-2(c) expressed in average LOOE (modified LOOE).	96
2.10	SWA segmentation results for mosaics T-2(d)–T-2(f) expressed in average LOOE (modified LOOE).	98

2.11	SWA segmentation results for mosaics T-2(g)–T-2(i) expressed in average LOOE (modified LOOE).	99
2.12	SWA segmentation results for mosaics T-2(j)–T-2(l) expressed in average LOOE (modified LOOE).	100
2.13	Statistical characteristics of the five materials for the fresh CT scans of the head section from visible woman data set, T-3, at RES of 2. . .	103
2.14	Statistical characteristics of the four brain materials for data set T-4(g) with SNR = 3% and INU = 40% from MNI BrainWeb simulated brain MR scans at RES of 1.	111
2.15	Segmentation results of CSF, grey-matter, and white-matter brain tissues in T-4(a)–T-4(e) (INU = 0% and SNR = 1% – –9%) using SWA method at their corresponding RES support.	113
2.16	Segmentation results of CSF, grey-matter, and white-matter brain tissues in T-4(f)–T-4(j) (INU = 40% and SNR = 1% – –9%) using SWA method at their corresponding RES support.	114
2.17	Segmentation results of the brain MR volumes T-5, expressed in terms of Tanimoto similarity coefficient (s_t), into CSF, grey-matter, and white-matter using SWA approach at their respective RES support. .	118
2.18	Mean overlap ratio, expressed in terms of Tanimoto similarity coefficient (s_t), for eleven brain MR volumes in test suite T-5 using different segmentation methods.	121

3.1	Comparison of constructing balanced 2^d -tree using Sivans algorithm (BQT1) and BalancedTree algorithm (BQT2).	197
3.2	Summary of implementation results for IMESH algorithm.	199
3.3	Exhaustive listing of the number of possible non-isomorphic configurations, up to 4-D, for a leaf node that has its $(d - 1)$ -dimensional face(s) split.	203
3.4	Summary of the properties of the template triangles in the 2-D mesh from IMESH method.	207
3.5	Summary of the properties of the template tetrahedras in the 3-D mesh from IMESH method.	210

LIST OF FIGURES

1.1	The flow diagram describing the various steps of a integrated model analysis strategy for simulating the dynamics in heterogeneous bio-systems using image data.	4
2.1	The main steps involved in a generic pattern recognition process . . .	19
2.2	Illustration of the equivalence between the multi-scale representation and wavelet transform. The tiles in the multi-scale representation at different scales and their corresponding wavelet coefficients in all the sub-band filter outputs are shown with same pattern.	44
2.3	The analysis and synthesis filter bank schemes for computation of the wavelet coefficients using quadrature mirror filters. * denotes convolution and $\uparrow 2$ ($\downarrow 2$) down-sampling (up-sampling) by a factor of 2.	46
2.4	(a) sample 2-D image of size 128×128 pixels; (b) typical organization of the approximation and detail images for a 2-D standard wavelet transform up to depth 3; and (c) the corresponding pyramidal representation of the standard wavelet decomposition	48
2.5	Plot of a material property as a function of sampling volume, adapted from Bear (1972).	53
2.6	The three images in the first test suite (T-1) used for testing the SWA approach: (a) digital scan of soils image; (b) microscope image of muscle; and (c) digital air-photo of landscape.	61

2.7	The ground truth images, G-1, obtained from expert hand-guided segmentation of T-1.	62
2.8	Twelve texture mosaics in T-2 test suite used for analyzing the proposed segmentation approach: (a-c) mosaics with two textures and simple boundaries; (d-e) mosaics with ten textures and simple boundaries; (f-j) mosaics with five textures and a complex boundaries; and (k-l) mosaics with sixteen textures and very complicated boundaries.	63
2.9	The ground truth mosaics, G-2, corresponding to T-2 test suite shown in Figure 2.8.	65
2.10	Bias field of 20% intensity inhomogeneity imposed on the T1-weighted MRI scans by the MNI-BrainWeb MRI simulator. (a)-(f): 20% INU field maps at axial locations $z=30mm$, $60mm$, $90mm$, $120mm$, $150mm$, and $180mm$, respectively.	70
2.11	Example axial slice at $z = 30mm$ for the ten (10) T1-weighted simulated MRI volumes in T-4. (a)-(e): simulated MRI scans with 1%, 3%, 5%, 7%, and 9% signal-to-noise and 0% spatial inhomogeneity. (f)-(j): simulated MRI scans 1%, 3%, 5%, 7%, and 9% noise and 40% spatial inhomogeneity.	71
2.12	Axial ground truth slice at $z = 30mm$ from the discrete anatomical model from MNI-BrainWeb. (a)-(c): shows CSF, grey-matter and white-matter tissue membership, respectively.	72
2.13	Flow diagram showing the various components of the proposed texture segmentation method.	79

2.14	Illustration of RES support identification: (a) Sample 2-D image of size 256×256 with two textures; (b) plot showing the statistics of the first four-central moments as a function of scale.	86
2.15	The three images in the first test suite (T-1) used for analysis of the SWA segmentation method. The images in the top (a-c) and the bottom (d-f) are the input and the ground truth images from manual expert segmentation, respectively. The boxes indicate the training sample locations used for estimating the RES support and classifier design.	87
2.16	Spatial statistics used for identifying RES support of horizons A, B, and C in the soils image.	88
2.17	Spatial-statistics used for identifying the RES support of smooth, ganglia, and outer tissues in the muscle image.	89
2.18	Spatial-statistics used for identifying the RES support of water, woods, wetland, pasture, planted and bare regions in the landscape image.	90
2.19	The source images of the textures present in the mosaic T-2(f). These source images are also used as for identifying RES support as well as classifier design.	94
2.20	Identification of RES support scale using SWA method for the head section of VWD.	104

2.21	Top: original head section CT scans (512×512 pixels) from VWD at axial locations 1 (0.5mm), 40 (20mm), 80 (40mm), 120 (60mm), 160 (80mm), and 200 (100mm). Middle: raw segmentation results (128×128 pixels) of the corresponding head section CT scans from VWD into soft-tissue, skin, bone, noise and background materials using SWA method at RES support of 2. Bottom: processed segmentation results (128×128 pixels) of the corresponding head section CT scans from VWD into soft-tissue, skin, bone, noise and background materials using SWA method at RES support of 2.	106
2.22	Raw segmentation of head section CT scans from VWD using SWA method at RES support of 2. (a)–(e): surface renderings of the raw segmented volumes of soft tissue, skin, bone, noise, and background, respectively.	107
2.23	Segmented volumes of different materials after removing the high-frequency noise from the raw results of T-3 using SWA method. The first, second and third column shows the post-processed results for soft tissue, skin, and bone, respectively, in three different perspective views.	108
2.24	Cut-sections of volumes from the SWA segmentation method after removing the high-frequency noise from the raw results. The first, second and third column shows the cut-sections of post-processed results for soft tissue, skin, and bone, respectively, in three different perspective views.	109

2.25	Statistics of mean ($\hat{\nu}$), variance ($\hat{\nu}$), skewness ($\hat{\chi}$) and kurtosis ($\hat{\kappa}$) of background, CSF, grey-matter, and white-matter materials used in RES identification step for T-4(g), with SNR = 3% and INU = 40%.	112
2.26	The first, second, third, and fourth column shows the input, ground truth, segmentation and misclassification results of an axial slice at $z = 30\text{mm}$ with 1%, 3%, 5%, 7%, and 9% SNR and INU = 0% into CSF, grey-matter, and white-matter using SWA method, respectively.	115
2.27	The first, second, third, and fourth column shows the input, ground truth, segmentation and misclassification results of an axial slice at $z = 30\text{mm}$ with 1%, 3%, 5%, 7%, and 9% SNR and INU = 40% into CSF, grey-matter, and white-matter using SWA method, respectively.	116
2.28	Segmentation results of (a) grey-matter and (b) white-matter for T-4 at RES support using SWA approach. KVL refers to the EM segmentation algorithm with bias estimation (Leemput et al., 1999).	117
2.29	Segmentation results of: (a) CSF; (b) grey-matter; (c) white-matter for T-5 using SWA approach and various segmentation methods from IBSR. (d) Tanimoto coefficient averaged over all the brain materials for MR scans in T-5 test suite; (e) bar plot showing the mean Tanimoto coefficient over the entire T-5 test suite.	127
3.1	Types of meshes: (a) structured, (b) unstructured, (c) hybrid for the simple polygonal domains.	133

3.2	Skinny triangles have circumcircles larger than their smallest edges. Each skinny triangle may be classified as a <i>needle</i> , whose longest edge is much longer than its shortest edge, or a <i>cap</i> , which has an angle close to 180°	135
3.3	The five types of bad tetrahedral elements.	136
3.4	Illustration of radius-to-edge ratio defined for (a) triangular element in 2-D and (b) tetrahedral element in 3-D, and (c) definition of aspect-ratio for a triangle and a convex body.	137
3.5	Geometric construction to prove the relation between radius-to-edge ratio and the minimum internal angle for a two-dimensional triangular element.	138
3.6	Every triangle in a Delaunay triangulation satisfies the <i>empty-circle-condition</i>	143
3.7	Illustration of Delaunay triangulation of a vertex set with empty circle for each triangle.	144
3.8	(a) If no four vertices are co-circular, two crossing edges cannot both be Delaunay, (b) case where e is locally Delaunay, and (c) case where e is not locally Delaunay. The edge created if e is flipped is locally Delaunay, (d) If the triangle t is not Delaunay, then at least one of its edges (in this case, e) is not Delaunay, and (e) Two triangulations of a vertex set: at left, e is locally Delaunay; at right, e is not.	145

3.9	Three ways to define the Delaunay diagram in the presence of co-circular vertices: (a) include all Delaunay edges, even if they cross, (b) exclude all crossing Delaunay edges, and (c) a subset of Delaunay edges that forms a triangulation is selected.	146
3.10	On the left is a Delaunay triangulation of vertex set and the right-side figure shows reversed/flipped quadrilateral along the edge e	148
3.11	A Sweep-line algorithm for computing an initial triangulation of a vertex set.	148
3.12	In three dimensions, an edge flip exchanges three tetrahedra sharing an edge for two tetrahedra sharing a triangle, or vice versa.	150
3.13	A Sweep-line algorithm for computing an initial triangulation of vertex set.	152
3.14	The surface of a tetrahedral mesh computed by an advancing front generator.	153
3.15	The constrained Delaunay triangulation of a polygon with holes.	154
3.16	(a) A quadtree decomposition, and (b) A quadtree-based triangulation of a vertex set.	158
3.17	Meshes generated for the Lake Superior as PSLG input by the: (a) Chews first constrained Delaunay algorithm, (b) Rupperts Delaunay refinement algorithm, and (c) Bern-Eppstein-Gilbert quadtree-based algorithm.	161
3.18	The surface of a tetrahedral mesh derived from an octree-based mesh generator.	163

3.19	An example of (a) 2-D input image of size 8×8 pixels, (b), its maximal block decomposition, and (c) the corresponding pointer-based region quadtree structure.	175
3.20	(a) Maximal block decomposition and (b) quadtree structure after imposing the 2:1 balancing rule on the quadtree in Figure 3.19.	177
3.21	An example of (a) two-dimensional image with its coordinate system and (b) relationship between a 2-D block's four quadrants and its boundaries.	179
3.22	An example of (a) three-dimensional image with its coordinate system, (b) the labelling of faces, edges, and vertices for a 3-D block, and (c) labelling of octants associated with a 3-D block.	179
3.23	Example showing the equivalence of representing a 2-D node and a 3-D node as a d -partite graph. The vertices of the d -partite graphs (on the right) represent the edges and faces of the 2-D block and 3-D block, respectively.	181
3.24	Illustration showing a 2-D node (λ) in a e -balanced quadtree along with its possible e -neighbors. The configuration on the left shows λ having an edge neighbor of half its size. In the middle, λ shares its edge with an equal size e -neighbor. The rightmost configuration shows λ have an e -neighbor of twice its size.	182

3.25	Example showing the equivalence of representing a 2-D node and a 3-D node as a d -partite graph when its e -neighbors are split. The vertices of the d -partite graphs, shown on the rightmost column, with "solid black" color corresponds to the 2-D node's edges and 3-D node's faces that are split.	183
3.26	An example of applying IMESH algorithm in 2-D for: (a) input image of size 8×8 pixels, (b) its quadtree maximal block decomposition, (c) its $2 : 1$ e -balanced quadtree maximal block decomposition, (d) unique vertices of the leaf nodes, and (e) output Delaunay mesh of the input image.	188
3.27	(a) Input segmented landscape image, and (b) output mesh generated by IMESH algorithm.	189
3.28	(a) Input segmented MR brain volume with CSF, grey-matter and white-matter tissues, and (b) tetrahedral mesh generated by IMESH algorithm.	190
3.29	Test images used for quantitative comparisons of the e -balanced 2^d -tree construction using Sivan's algorithm and BalancedTree algorithm: (a) raw MR scan from MNI-BrainWeb, and (b) corresponding segmented MR scan with ten different tissues.	196
3.30	The d -partite graph representations for the six possible non-isomorphic splitting of a leaf node in two-dimensions.	203
3.31	The d -partite graph representations for the ten possible non-isomorphic splitting of a leaf node in three-dimensions.	204

3.32	Illustration showing the vertices of a 2-D leaf node (black circles) in a e -balanced quadtree and the extra vertices (gray circles) needed to match its split e -neighbors in each of the six possible non-isomorphic 2-D leaf node configurations.	204
3.33	Illustration showing the vertices of a 3-D leaf node (black circles) in a e -balanced octree and the extra vertices (gray circles) needed to match its split e -neighbors in each of the ten possible non-isomorphic 3-D leaf node configurations.	205
3.34	Delaunay triangulation of six possible quadnode configurations in a balanced quadtree decomposition. The triangles with same color in the decomposition are similar.	206
3.35	Four prototype triangles deduced from the set of Delaunay triangulation of nodes in a balanced quadtree shown in Figure 3.34.	207
3.36	Eleven unique prototype tetrahedra obtained from the Delaunay tetrahedralization of all non-isomorphic 3-D node configurations shown in Figures 3.33(k)–3.33(t).	209
4.1	Schematic of an integrated tool for analysis of complex bio-systems.	219
A.1	Effect of RES support on the SWA classification accuracy for the two-textured mosaic shown in Figure 2.14.	220

CHAPTER 1: INTRODUCTION

1.1 Motivation

The design, analysis, and control of bio-systems remain an engineering challenge mainly because of their heterogeneity, boundary irregularity, and nonlinear dynamics associated. A classical approach to addressing this problem has been to break down these (bio)systems hierarchically into functional sub-systems, units, sub-units, and finer and finer sub-components and analyzing them separately at each scale/level in order to better understand the system. For example, an organism (system) has various organs (sub-systems) which are made from specific tissue (units) and tissues in turn are made from aggregation of specific cells (sub-units). Similarly, a geographic region (system) can be divided into a group of watersheds (sub-systems) which can have several categories of soils (units) and each soil type is composed of the aggregation of different soil-layers (sub-units). Each of these components at different hierarchical levels (scales) are characterized by specific structure, behavior, function, and role affecting the overall dynamics of the system (organism or geographic region). In order to accurately assess the *dynamics* of these (bio)systems, it is important from a numerical modeling perspective to take into account the pertinent information from different scales (units and sub-units). In practice, however, the analysis usually stops at a pre-selected coarser scale (e.g. unit) and the sub-scale heterogeneity is either considered implicitly (as uncertainty) or not at all (pure determinism). The prime reasons for this are two fold: (1) the non-availability of

heterogeneous material properties for the finest scale, and (2) even when such information is available, considering the heterogeneity explicitly down to the finest scales is usually prohibitive from a computational standpoint.

Developments in imaging sensors and techniques, such as, magnetic resonance (MR), computed tomography (CT), ultrasound (US), positron emission tomography (PET), infra-red (IR), confocal laser microscopy, scanning electron microscopy (SEM), satellite/air-borne imagery, and ground penetrating radar (GPR), with their improved spatial and spectral resolutions are now able to provide highly detailed measures of internal structure and various critical biological, physical, and chemical properties of different materials within these bio-systems (Leggett et al., 1996; McKinlay et al., 2004). The information captured by the image sensors is helpful for the scientific community to non-invasively analyze these heterogeneous bio-systems for better diagnosis, treatment strategies, and prognosis more safely and effectively than ever before. The concurrent developments in up-scaled stochastic numerical analysis suggest that it may now be possible to efficiently incorporate sub-scale heterogeneity into the stochastic framework (Bastian et al., 2000; E. and Engquist, 2003; Montas et al., 2000b). Results ensuing from such analysis promise to more accurately reflect the non-linear dynamics of bio-systems than the approaches based on classical continuum mechanics (pure determinism). By providing the mathematical framework to directly incorporate the up-scaled material properties at a coarser scale, the computational requirements are significantly reduced. Yet, the stochastic results are more representative than produced by conventional numerical techniques. The coarser scale at which the up-scaled stochastic continuum tech-

niques work is commonly known in the literature as *representative elementary scale* (RES) (Bear, 1972; Pachepsy et al., 2004; Woods et al., 1995). More specifically, in two and three-dimensions it is known as *representative elementary area* (REA) and *representative elementary volume* (REV), respectively.

The impetus for of this research was the realization that the inputs required for simulating the dynamics of bio-systems using up-scaled numerical techniques can be derived from the images of the material properties. The corresponding approach for analyzing heterogenous bio-systems consists of four main steps: (i) information acquired from the sensors in the form of images; (ii) identify the boundaries of different heterogeneous materials within which the properties of these materials become homogeneous; (iii) generate a finite element mesh of the coarser scale material boundaries; and (iv) apply up-scaled numerical methods to simulate the dynamics. Figure 1.1 shows the flow diagram of such an integrated system. Its development will help the scientific community to non-invasively analyze these heterogeneous bio-systems for better diagnosis, treatment strategies, and prognosis more safely and effectively.

The motivation for carrying out this dissertation was further strengthened while studying two applications of numerical modeling: (a) effects of chemotherapy on glioma growth dynamics (Tracqui et al., 1995) and (b) nutrient movement in watershed using GIS-based models (Montas et al., 2000a). Tracqui et al. (1995) used numerical simulation of a simple mathematical model for describing the proliferation and infiltration of glioma *anaplastic astrocytoma* based in part on quantitative image analysis of histological sections of human brain, especially the cross-sectional

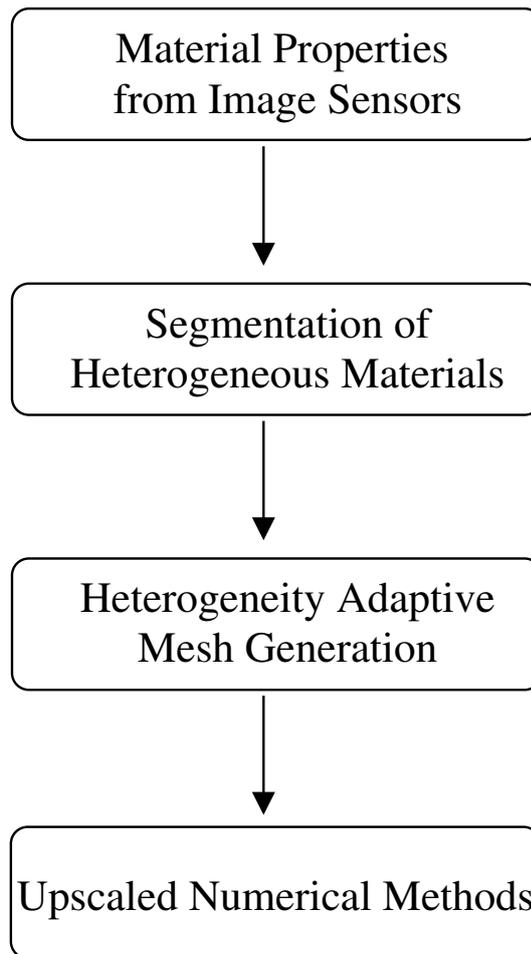


Figure 1.1: *The flow diagram describing the various steps of a integrated model analysis strategy for simulating the dynamics in heterogeneous bio-systems using image data.*

area/volume measurements of serial CT images. They modeled the spatial and the temporal tumor growth dynamics of two types of cancerous cells when the patient was undergoing two treatments of chemotherapy and neutron irradiation. The first course of chemotherapy treatment consisted of 5 cycles of 6 drugs (UW protocol) and the second treatment consisted of *cis*-platinum dosage. Two types of cells were considered, the first type being sensitive to both chemotherapy agents, whereas the second were assumed to be resistant to the first course of chemotherapy, but possibly

sensitive to the second course. In order to account for the effect of brain geometry and its natural barriers, they segmented the slice of MR scan into homogeneous brain material and estimated the diffusivity parameter. This information was then used in a finite difference framework to model the two-cell population dynamics.

Montas et al. (2000a) used a GIS-integrated, physically-based, distributed parameter model for simulating subsurface flow and transport of nitrate-N in a watershed. The model used discretized mass balance equations (control volume scheme) to simulate steady-state, two-dimensional flow and transport using material properties from the region's GIS data. The predictions from this integrated model were used to demonstrate that the conjunction of soil and land use variability can cause complex nitrate transport patterns which can locally saturate or bypass riparian buffers and prevent them from achieving their full filtration potential. Such an analysis can be useful in evaluating the impacts of riparian buffers on water quality and for identifying zones where restoration or implantation of new buffers may be needed.

The shortcomings of the first application are two-fold: (1) the model does not account for the heterogeneity of brain tissues; and (2) the uniform grid used in the finite difference schemes makes it computationally expensive, especially while simulating the tumor dynamics in three-dimensions. The latter is also true for the control volume method used for evaluating the fate of non-point source pollutants in watershed. These problems can be alleviated by using the integrated model analysis strategy outlined in Figure 1.1. However, in order to successfully apply this approach, there is a need for development of appropriate quantitative image

analysis and domain decomposition methods adapted specifically to heterogeneous image data. Among the research areas that compose the field of image analysis lies *texture segmentation*, which studies the recognition and isolation of textural patterns found in images. Similarly, *unstructured mesh generation* is a sub-topic in computational geometry that deals with domain decomposition. It is in these area where the contributions reported in this dissertation fall. The following sub-sections briefly describe the current state of knowledge available in these two areas and the driving problems within them.

1.1.1 Image Segmentation

Image segmentation is a sub-category of study under a much broader area known as *pattern recognition* (Vapnik, 1998). In general, a pattern recognition system involves two main steps: (a) identification and extraction of pattern descriptors, called *features*; and (b) supervised or unsupervised *classification* of the patterns using the features. These patterns can be a set of recurring objects of any nature, like, faces of people in a crowd, mosaic of bricks in a wall, and fluctuations of financial markets. The features can either be an abstract or quantitative measurement that describes the patterns. For example, if the objective is to analyze the patterns of faces in a crowd, abstract descriptors like "roundness" and "sharpness" can be used. Similarly, for identifying different types of bricks in a mosaic their geometry can be one of the useful descriptors for discrimination.

Within the realm of image segmentation, *texture* is one of the most important cues used for segmentation of objects. Research on the human visual cortex has

shown the relevance of texture in identifying different patterns observed by the eye (Rosenfeld and Kak, 1982). Although texture is an intuitive concept, many attempts to formally define the term have been evasive (Bovik et al., 1990; Cross and Jain, 1983; Haralick et al., 1973; Jain and Karu, 1996). In spite of this, all researchers agree on two counts. Firstly, there is significant variation in the intensity levels (greyness) between nearby image elements, i.e., at the limit of resolution there is non-homogeneity. Secondly, texture is a homogeneous property (statistically) at some spatial scale larger than the resolution of the image. The latter resonates with the RES concept used in the up-scaled stochastic continuum theory. Hence, it is one of the main components of this dissertation to use texture-based segmentation for characterizing heterogeneous materials in image data. It is notable that, the use of material texture in image segmentation requires that the spatial resolution of imaged material properties data be fine enough that small scale heterogeneity is accurately represented (so that its statistics can be computed).

According to the reviews by Pal and Pal (1993) and Reed and du Buf (1993), texture segmentation methods can be categorized into: structural-based, statistics-based, model-based and transform-based methods. Structural-based methods uses the concept of primitives or *texels* to describe textural objects (Haralick, 1979; Levine, 1985). The advantage of the structural approach is that it provides a good symbolic description of the texture objects in an image; however, this feature is more useful for synthesis than analysis tasks. Statistics-based methods, instead, represents the texture indirectly by the non-deterministic properties that govern the distributions and relationships between the grey levels of an image. The most

popular second-order statistical features for texture analysis are derived from the well-known co-occurrence matrix (Haralick, 1979). Methods based on the features from co-occurrence matrix have been shown to achieve higher discrimination rates than the structural methods (Weszka et al., 1976). A major limitation with these methods is that they operate on the statistics computed from a single scale. The model-based methods uses fractal and stochastic models for interpreting an image texture (Chellappa and Chatterjee, 1985; Manjunath and Chellappa, 1991; Zhang et al., 2001). The fractal model has been shown to be useful for modelling some natural textures. However, the computational complexity associated with the estimation of model parameters is major drawback with these methods.

Although the above methods can be successful for specific applications, they suffer from *localization* issues commonly associated with textures. This is chiefly due to their analysis of texture objects at a single scale. Transform-based methods overcome this weakness by transforming the image into a new space whose co-ordinate system has an interpretation that is closely related to the characteristics of a texture, such as frequency or size. This approach is analogous to the manner in which the human visual system functions (Daugman, 1990). In particular, methods such as Fourier transforms (Zhu et al., 1993), Gabor transforms (Bovik et al., 1990), and wavelet transforms (Mallat, 1989) have shown promising results for texture analysis because of their capability to represent the image in the transformed space at multiple scales. Among these methods, Gabor and wavelet transforms have been the most widely studied methods in the past fifteen years. Gabor decomposition provides better spatial localization than Fourier transform, but it only solves the

problem of localization partially. The wavelet transforms, in contrast, offers better variable spatial resolution and a wide range of wavelet functions to use for texture analysis. These properties make the wavelet transform attractive for texture segmentation and has now become a de facto standard.

In the past decade, several authors have used a wide variety of wavelet functions, wavelet decomposition schemes, and classification methods for texture analysis of both artificial and natural textures (Chang and Kuo, 1993; Fatemi-Ghomi et al., 1996; Laine and Fan, 1993; Lu et al., 1997; Unser, 1995; Zhu and Yang, 1998). However, in spite of the vast literature available on these methods, specific performance results for heterogeneous biological image data are sparse. Furthermore, all these methods use feature descriptors (like entropy, energy and mean-deviation) from multiple scales simultaneously which do not convey quantitatively the heterogeneity within different samples. These methods, thus far, have not been successful at identifying the optimal scale for texture characterization. Most of the studies, use feature descriptors estimated from 3 to 4 scales, without any quantitative justification.

An alternative approach to achieve image segmentation, instead of defining features and then classifying them, is to isolate the desired patterns using some local operations (like, gradient) and then build a mathematical or geometrical model of the evolving boundary. This is of particular interest in bio-medical applications (e.g., surgical planning). In this context, algorithms based on curve and surface evolution methods, such as snakes (Kass et al., 1987; Liang et al., 1999) and level-set (Malladi and Sethian, 1996a,b,c; Sethian, 1999) have shown impressive potential

for bio-medical image data. However, these methods do not perform well when applied to texture-predominant images because they work on local operations.

1.1.2 Unstructured Mesh Generation

A mesh is a discretization of a geometric domain Ω in \mathbb{R}^2 or \mathbb{R}^3 into small simple shapes or simplices, typically triangles or quadrilaterals in \mathbb{R}^2 and tetrahedra or hexahedra in \mathbb{R}^3 . Meshes composed of these simple shapes are used in many applications, such as computer graphics, numerical interpolation, surveying, geographic information systems, and terrain modelling. Their most significant application is in the numerical simulation of process dynamics using domain-based numerical techniques. For this purpose, the mesh should satisfy the following conditions: (a) it must conform to the boundary of the domain, which may have more than one connected component; (b) it must be fine enough to produce an acceptable approximation to the original problem (parts of the domain where the solution is complicated or changes rapidly may require much smaller elements than others); (c) the number of elements in the mesh should be small (optimal), because large number of elements could increase the complexity of solving the problem; and (d) individual elements must be *well-shaped* and *round*.

Three basic types of meshes are used in practice: (a) structured mesh, (b) unstructured mesh and (c) hybrid or block-structured mesh. Structured meshes offer simplicity and easy data access while unstructured meshes offer better elemental adaptivity, refinement and de-refinement capability, and geometric conformity to complicated domains. High-quality hybrid meshes enjoy the advantages of both ap-

proaches, but hybrid meshing is not yet fully automatic. Because of their advantages, unstructured meshes have received wider acceptance for numerical simulations.

In the past two decades, several automatic, quality guaranteed unstructured mesh generation techniques for polygonal and polyhedral domains have been proposed. The two most notable algorithms are: (i) the quadtree-based mesh generation due to Bern et al. (1990), and (ii) the *constrained Delaunay triangulation* method by Ruppert (1993). The current state of research in computational geometry can generate quality meshes on a wide variety of two and three dimensional domains. However, most of these methods are targeted towards for polygonal/polyhedral domains and not image data.

Although methods for discretizing image data, for FEA, are rare in the literature, the need for unstructured mesh generation techniques for image data is receiving increased attention. This is mostly, due to the non-invasive property associated with image sensors. The current methods available for meshing image data can be grouped into two categories: (a) interval volume decomposition methods and (b) integrated image-analysis methods. The first category of methods works with the polygonal/polyhedral surfaces bounding the interval image data (Bajaj et al., 1999; Frey et al., 1994; Fujishiro et al., 1996; Nielson and Sung, 1997; Wang et al., 2004; Zhang et al., 2003). Though these methods preserve the topology of the domain in the image data, they come with no theoretical guarantees on elemental quality. Moreover, these methods need an extra image-processing step to generate the polygonal/polyhedral surfaces bounding the interval volume. On the other hand, the integrated image-analysis based techniques (Hale, 2001; Langer et al., 2001),

work directly on the input image and generate the finite element mesh by aligning elemental edges with domain boundaries in the image data. These methods minimize an objective function for alignment of the mesh with the boundaries. Image segmentation is an indirect outcome of these methods. Both methods are computationally expensive because they require repeated computations (like *Monte Carlo* simulations) for minimizing the objective function and its partial derivatives. Additionally, these methods manipulate the input image information such that original material boundaries cannot be recovered from the meshes.

1.2 Thesis Objectives

The main goal of this study is to develop necessary quantitative tools that can be used for analyzing the dynamics of heterogeneous bio-systems. The specific research objectives addressed in this dissertation are:

Objective 1: Develop a supervised texture-based segmentation technique that can quantitatively characterize the heterogeneity of a material at a coarser scale by accounting for the heterogeneity from sub-scales.

Objective 2: Evaluate the developed technique for segmenting heterogeneous materials in biological image data.

Objective 3: Develop and implement a quality guaranteed unstructured simplicial mesh generation technique for (segmented) image data.

Objective 4: Analyze the developed unstructured simplicial mesh generation technique for elemental quality and output size.

These specific objectives correspond to steps 2 and 3 of the bio-system analysis strategy presented earlier in Figure 1.1.

1.3 Contribution to Knowledge

The following are the key contributions made by this dissertation in the fields of texture-based image segmentation and unstructured mesh generation of heterogeneous biological image data.

1. Multiresolution analysis, using Haar wavelets, can be used to efficiently identify the representative elementary scale of the materials in a heterogeneous biological image.
2. Spatial moments of pixel intensities, computed at a pre-identified representative elementary scale, form efficient feature vectors for the segmentation of heterogeneous biological images in 2-D.
3. The combination of RES identification by Haar MRA with spatial moments as feature vectors produces a segmentation method (named SWA) that can efficiently and accurately classify heterogeneous biological images in 3-D.
4. The 2-D meshing algorithm of (Bern et al., 1990) can be extended to produce efficient heterogeneity adaptive, boundary preserving multi-dimensional meshes on segmented bio-image data. (This novel method is named IMESH).
5. The developed multi-dimensional unstructured mesh generation technique, IMESH, is guaranteed to produce quality elements in both 2-D and 3-D.

1.4 Overview of Chapters

The remainder of this dissertation is organized into three chapters:

Chapter 2 is focused on texture segmentation of heterogeneous image data. It starts with a literature review on pattern recognition with emphasis on multi-resolution based texture analysis. Then, it describes the framework of the novel texture segmentation method for heterogeneous bio-materials developed in this dissertation, analyzes the role of texture descriptors using two-dimensional real and synthetic textures, and evaluates the new method using three-dimensional real and simulated bio-medical data.

Chapter 3 is focused on unstructured mesh generation. It presents a review of the literature on this topic. It then describes the algorithm for generating unstructured meshes in arbitrary dimensions developed in this dissertation and analyzes the developed method for elemental quality.

Chapter 4 summarizes the contributions made in this thesis and highlights the future research work that remains to be done in order to combine the research accomplishments reported in this dissertation into an unified cohesive framework for the analysis of heterogeneous biological systems.

In this chapter, a new supervised texture-based segmentation technique, stochastic-wavelet-analysis (SWA), for characterizing heterogeneous materials in biological image data is developed. The SWA method uses multi-scale stochastic texture descriptors estimated from wavelet analysis for characterizing heterogeneous materials. A rigorous analysis of the developed method on both natural and synthetic two-dimensional texture-predominant images is performed to identify the optimal combination of feature descriptors and classification scheme. The novel texture segmentation method is evaluated on three biological image data sets from different imaging modalities. The improved accuracy of the segmentation results produced by the proposed method, relative to contemporary techniques, demonstrates the strength of its conceptual underpinnings. The main contributions of the work described in this chapter are as follows: (1) the necessary mathematical framework to directly estimate the multi-scale stochastic texture descriptors from Haar wavelet coefficients for arbitrary dimensions is presented and (2) a quantitative approach for identifying the characteristic length (RES length) at which heterogeneous materials (textures) become continua is also provided.

2.1 Introduction

In the past three decades, identifying texture segmentation techniques has been one of the main focus of the pattern recognition and image processing community. Im-

provements in the texture descriptors, texture extraction techniques, and classifier designs led to the development of various image segmentation techniques. These improved techniques provided results, which had far reaching impacts in several areas such as signal compression, scene analysis, object identification and retrieval, and character and speech recognition. In spite of such advancements, a common denominator missing from these systems is that they fail to identify the characteristic length at which the definition of texture starts to emerge. Hence, if the heterogeneous image data is to be used in up-scaled transport models, it is necessary to develop an appropriate quantitative image processing tool that can address this issue.

The main aim of the research described in this chapter is to develop a quantitative image segmentation framework rooted in up-scaled stochastic continuum framework for identifying the representative elementary scale (RES) support in heterogeneous image data. This is achieved, first, by developing appropriate statistical texture descriptors that quantify the sub-scale heterogeneity of a material at coarser scale and then, by developing a technique for simultaneously calculating these descriptors and identifying the scale at which they stabilize. The availability of such descriptors and RES identification strategy can then be used in a conventional pattern recognition framework to segment the heterogeneous materials in biological image data.

The remaining part of this chapter is organized as follows. Section §2.2 reviews the research literature on pattern recognition with a focus on texture-based segmentation. Following a careful review on the shortcomings of existing techniques, the

main research objectives are proposed in Section §2.3. The materials and methods needed for achieving these objectives are presented in Section §2.4. The main results of the research are then presented and discussed in Section §2.5. Section §2.6 concludes by highlighting the important contributions and the future directions of the research described in this chapter.

2.2 Literature Review

The domain of image segmentation encompasses several key concepts from pattern recognition. According to Vapnik (1998), a pattern recognition system is:

A person (the instructor) observes occurring situations/events and determines to which of c classes each one of them belong. It is required to construct a device (the classifier) which, after observing the instructors procedure, will carry out the classification approximately in the same manner as the instructor.

The pattern recognition process, generally, consists of the following eight stages (Webb, 2002):

- Problem formulation: gaining a clear understanding of the aim of the investigation and planning the remaining stages.
- Data collection: making measurements on appropriate variables and recording details of the data collection procedure (ground truth).
- Initial examination of the data: checking the data, calculating summary statistics and plotting the data in order to get a feel of the structure underlying

the data.

- Feature selection or feature extraction: selecting variables from the measured set that are appropriate for the task. These new variables may be obtained by a linear or nonlinear transformation of the original data (feature extraction).
- Pattern classification or clustering: This may be viewed as exploratory data analysis and it may provide a successful conclusion to a study. On the other hand, it may be a means of pre-processing the data for a supervised classification. To some extent, there is a thin line demarcating the feature extraction and classification process.
- Designing classifier: applying discrimination or regression procedures, as appropriate to, design a classifier using a training set of exemplar patterns.
- Assessment of results: may involve applying the trained classifier to an independent test set of labelled patterns to independently estimate classifier accuracy.
- Interpretation: analyzing the results to see if the desired objectives are attained.

Note that not all the stages described earlier may be present in all pattern recognition processes. Some steps be merged together so that the distinction between two operations may not be clear even if both are carried out. There may also be some application specific data processing that may be regarded as one of the stages listed. The application of these steps can be an iterative process: the analysis

of the results may generate new hypotheses that require further data collection; the cycle may be terminated at different stages if the question posed is answered by an initial examination of the data or it may be discovered that the data cannot answer the problem and hence must be reformulated.

However, irrespective of these variations, the stages described above are fairly typical in a pattern recognition process. When the data to be studied by a pattern recognition system comes from digital images, the topic of pattern recognition converges with the concepts of *image processing and analysis*. Furthermore, when the "patterns" studied in digital image data match the definition of a texture, the investigation becomes texture-based image segmentation. Figure 2.1 illustrates the schematic of a generic pattern recognition process.

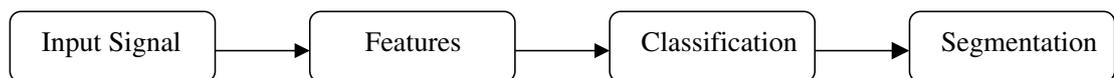


Figure 2.1: *The main steps involved in a generic pattern recognition process*

An image is a spatial (sometimes also temporal) measurement of a parameter that depends on the type of imaging modality and the type of imaging sensors used. For instance, these measurements can be: (a) radiation absorption in X-ray imaging, (b) acoustic pressure in ultrasound, (c) range or depth image in laser scans, (d) radio frequency signal amplitude in MR, (e) temperature in IR imaging, and/or (f) light reflectance in satellite imagery. Of these, light intensity images are the most common type of images. A generic multi-dimensional, multi-channel analog image

can be described mathematically as,

$$f'(x'_1, \dots, x'_d, y_1, \dots, y_n) \quad (2.1)$$

where x'_1, \dots, x'_d denote the real-valued spatial coordinates in \mathbb{R}^d and y_1, \dots, y_n denotes the parameter measured in n channels and f' is the image function that associates measurements y_i to spatial coordinates x'_i . For instance, hyper-spectral imaging measures electromagnetic intensity in different spectral bands, thematic images from LANDSAT satellites and color photography captures this measurement at multiple locations in the color spectrum, and MRI is capable of measuring proton density (PD) and relaxation times of the protons both in parallel (T1-weighted) and perpendicular (T2-weighted) directions with respect to the original magnetic field. When the image sensor measures the parameter in just one spectral band (single channel) it results in a monochrome, multi-dimensional image and equation 2.1 reduces to $f'(x'_1, \dots, x'_d)$, where f' now represents the magnitude of the measure entity. The digital version of the monochrome multi-dimensional image is a discrete approximation of the both spatial coordinates and magnitude of measured parameter in \mathbb{R}^d and is represented mathematically by the function, $f(x_1, \dots, x_d)$. The coordinates x_1, \dots, x_d frequently form a uniform grid and the image element (value of f) at any such location is known as a *pixel* or *voxel* in two and three dimensions, respectively. For further discussions, it is assumed that image data is monochrome multi-dimensional digital image data.

2.2.1 Patterns, Features, and Feature Space

Much of the information that surrounds us manifests itself in the form of patterns. The ease with which humans classify and describe patterns often leads to the incorrect assumption that this capability is easy to automate. Sometimes the similarity between two patterns is immediately apparent, whereas in other instances it is not. Recognizing characters and faces is an example of the former; economic forecasting based on trends in the stock market illustrates the latter.

Pattern recognition, naturally, is based on *patterns*. A pattern can be as basic as a set of measurements or observations, perhaps represented mathematically in a vector or a matrix notation. These measurements could be entities such as blood pressure, tumor diffusion rates, or soil permeability. These measurements can also be represented in many forms, for instance, two or three dimensional images, drawings, waveforms, set of measurements, temporal or spatial sequence of events. Furthermore, patterns may be converted from one representation to another for easier identification or better understandability. This is one of the main applications of signal transforms like Fourier transform and wavelet analysis.

The important step, after data acquisition from sensors, is to identify a set of descriptors known as *features* (denoted by F_i , that can best be used to identify and characterize the different patterns in the data. Broadly speaking, features are any extractable measurements that are characteristics of a pattern. An example of low-level feature is the raw intensity value of individual pixels in a digital image. An example of higher-level feature is a statistic calculated over several neighboring

pixels. Each feature can be numerically continuous, discrete, or even binary. *Binary features* are used to represent the presence or absence of a particular feature in a pattern or object. For ease of analysis, the set of features, F_i , to be used for segmentation are arranged as a d -dimensional *feature vector*, denoted by:

$$\vec{F} = \{F_1, F_2, \dots, F_d\} \quad (2.2)$$

which defines a multi-dimensional measurement space or *feature space*, $\vec{F} \in \mathbb{R}^d$. The descriptors of each pattern to be extracted from an image is expected to correspond to a different point (or neighborhood) in the feature space. It is common in practice to associate a set of feature vectors to describe a pattern (class), i.e., $\vec{F}_n = \{\vec{F}_1, \dots, \vec{F}_n\}$ and if a signal has c classes, denoted by w_1, \dots, w_c , then a set of feature vectors for each class, w_i , is expressed as \vec{F}_{n_i, w_i} . In order to effectively segment the different classes in the signal, the feature descriptors should be characteristic of their corresponding patterns, i.e., they should have enough discriminatory information for identifying the pattern.

Texture is one of the important concepts used in image processing for describing patterns in image data. Although there is no strict definition of the image texture, it is easily perceived by humans and is believed to be a rich source of visual information about the nature and three-dimensional shape of physical objects. Generally speaking, textures are complex visual patterns composed of entities, or sub-patterns, that have characteristic brightness, color, slope, or size. Thus, texture can be regarded as a similarity grouping in an image (Rosenfeld and Kak, 1982). The local subpattern properties give rise to the perceived lightness, uniformity, density,

roughness, regularity, linearity, frequency, phase, directionality, coarseness, randomness, fineness, smoothness, and/or granulation, of the texture as a whole (Levine, 1985). In the past three decades, several diverse approaches for extracting texture properties have been proposed in the pattern recognition literature. These feature descriptors used for texture segmentation can be categorized into: (a) structural; (b) statistically-based; (c) model-based; and (d) spatial/spatial-frequency descriptors.

2.2.1.1 Structural-Based Texture Descriptors

Structural-based descriptors represent texture by well defined primitives (microtexture) and a hierarchy of spatial arrangements (macrotexture) of those primitives (Haralick, 1979; Levine, 1985). To describe the texture, the primitives (*texels*) are defined along with syntactic rules for their placement (Fu, 1982). The choice of a primitive (from a set of primitives) and the probability of the chosen primitive to be placed at a particular location depends on the location and the neighboring primitives near the location.

The advantage of the structural approach is that it provides a good symbolic description of the image; however, this feature is more useful for synthesis than analysis tasks. The abstract descriptions can be ill defined for natural textures because of the variability of both micro and macrostructure and no clear distinction between them. A powerful tool for structural texture analysis is provided by mathematical morphology (Dougherty et al., 1992; Serra, 1982) and has been useful for bone image analysis, e.g. for the detection of changes in bone microstructure.

2.2.1.2 Statistics-Based Texture Descriptors

In contrast to structural methods, statistical approaches do not attempt to understand explicitly the hierarchical structure of the texture. Instead, they represent the texture indirectly by the non-deterministic properties that govern the distributions and relationships between the grey levels of an image. Methods based on second-order statistics (i.e. statistics given by pairs of pixels) have been shown to achieve higher discrimination rates than the power spectrum (transform-based) and structural methods (Weszka et al., 1976). Human texture discrimination in terms of texture statistical properties was first investigated by Julesz (1975), who suggested that textures in grey-level images are discriminated spontaneously only if they differ in second order moments. Equal second-order moments, but different third-order moments require deliberate cognitive effort. This may be an indication that also for automatic processing, statistics up to the second order may be most important (Niemann, 1990). The most popular second-order statistical features for texture analysis are derived from the well-known co-occurrence matrix (Haralick, 1979). They were demonstrated to feature a potential for effective texture discrimination in bio-medical images (Freeborough and Fox, 1998; Lerski et al., 1993).

2.2.1.3 Model-Based Texture Descriptors

Model based texture segmentation methods commonly use stochastic models (Chellappa and Chatterjee, 1985; Chellappa et al., 1992; Kervrann and Heitz, 1995; Manjunath and Chellappa, 1991) and fractal models (Chaudhuri and Sarkar, 1995a;

Kaplan, 1999; Kasparis et al., 2001) for interpreting the textures within an image data. Chellappa with his collaborators (Chellappa and Chatterjee, 1985; Chellappa et al., 1992; Manjunath and Chellappa, 1991) used second-order Gauss Markov random field (GMRF) for modeling the conditional probability density of the grey-level intensity of a given texture in an image data. The parameters of the GMRF model of the textures are estimated using a least-square estimates within non-overlapping tiles. A clustering method, based on a heuristic parameter, is then used to merge these regions to get a coarser segmentation. Finally, the parameters of the model estimated from the clustered segments are then used in two different schemes, one being an approximation to the maximum *a posteriori* estimate of the labels and the other minimizing the percentage misclassification error. Manjunath and Chellappa (1991) tested their approach on, both hand drawn images and texture mosaics containing a combinations of Grass, Raffia, Leather, and Wood materials. In all their experiments, the sub-image size was chosen to be 32×32 . The value of the clustering parameter was chosen to be inversely proportional to the approximate number of classes within the test image. By modeling the textures using GMRF models, they were able to achieve an average of 97% segmentation for most of their test images.

Kervrann and Heitz (1995) presented an unsupervised texture extraction technique that does not require knowledge about the different texture regions, their parameters, or the number of available texture classes. Their proposed algorithm relies on the analysis of local and global second and higher order spatial statistics of the original images. The segmentation map is modeled using an augmented-state Markov random field, including an outlier class that enables dynamic creation of

new regions during the optimization process. A Bayesian estimate of this map is computed using a deterministic relaxation algorithm.

In practice, the computational complexity arising in the estimation of stochastic model parameters is the primary problem. The fractal model has been shown to be useful for modeling some natural textures. It can be used also for texture analysis and discrimination (Chaudhuri and Sarkar, 1995b; Kaplan and Kuo, 1995). However, it lacks orientation selectivity and is not suitable for describing local image structures.

2.2.1.4 Spatial/Spatial-Frequency Texture Descriptors

The texture descriptors discussed so far are either structural, statistics-based or model-based in nature. Approaches used for textured feature extraction can also be grouped loosely into those based on statistical methods and those using spatial-frequency or spatial/spatial-frequency methods. The statistics-based methods discussed earlier have in the past proven superior to the traditional frequency domain techniques, like Fourier transform. This was largely due to lack of locality in in these early frequency-based texture analysis methods. This problem caught the attention of both mathematicians and the signal processing community and led to the developments of transform-based method (Bovik et al., 1990; Burt and Adelson, 1983; Greenspan et al., 1994; Jain and Karu, 1996; Mallat, 1989; Unser and Eden, 1989). Especially, the multi-channel and multi-resolution analysis signal processing tools, like: (pseudo) Wigner distributions (Reed and Wechsler, 1990, 1991; Zhu et al., 1993), STFT or Gabor transform (Chen and Chen., 1996; Daugman, 1985; Dunn

and Higgins, 1995; Fogel and Sagi, 1989; Haley and Manjunath, 1995; J. Bigün and J. M. H. du Buf, 1994; Jain and Farrokhnia, 1991; Lee, 1996; Manjunath et al., 1996; Teuner et al., 1995), wavelet and wavelet packet transforms (Chang and Kuo, 1993; Fatemi-Ghomi et al., 1996; Laine and Fan, 1993; Lu et al., 1997; Unser, 1995; Zhu and Yang, 1998), have been promising additions for texture analysis. Joint spatial/spatial-frequency methods are inherently local in nature, and have characteristics that compare favorably with those of statistics-based methods.

The spatial/spatial-frequency methods are based on image representation that indicate the frequency content in spatially localized regions and closely relates to the characteristics of a texture. Additionally, these methods also achieve high resolution in both the spatial and spatial-frequency domains and are consistent with the recent theories on human vision—the visual cortex can be modeled as a set of independent channels, each with a particular orientation and spatial frequency tuning (Beck et al., 1987; Daugman, 1990; Farrokhnia, 1990; Lee, 1996).

The Wigner distribution (WD) is a spatial/spatial-frequency representation which was first introduced (in its 1-D form) in quantum mechanics, to characterize the positions and momenta of particles. The two-dimensional pseudo-Wigner distribution (PWD), a discrete approximation of the continuous WD, was first used for texture segmentation by Reed and Wechsler (1990, 1991). Their approach was to determine the high energy frequency components in the PWD of the image to be segmented and to select a small set of those components (“frequency planes”) as features. The segmentation results of PWD-based features for texture characterization were qualitatively superior to other spatial/spatial-frequency representations,

namely Gabor transform, spectrogram (Fourier transform), and Difference of Gaussians (DOGs). The experiments were based on textured image data from four different sources: synthetic textures constructed from products of complex exponentials; Beck-type textures (Beck et al., 1987); Brodatz textures (Brodatz, 1966).

Gabor filters produce space-frequency decompositions that achieve the theoretical lower bound of the uncertainty principle and attain minimum possible joint resolution in space and frequency. This is highly significant in the process of texture extraction in which the conflicting objectives of accuracy in texture representation and texture spatial localization are both important. A more practical way to gain in the trade-off between space and frequency resolution without using Gabor functions is with a dyadic or wavelet filter bank. The wavelet filter bank produces octave bandwidth segmentations in space-frequency (s-f). It allows simultaneously for high spatial resolution at high s-f's and high s-f resolution at low s-f's. A two-band quadrature mirror (QMF) filter bank utilizes orthogonal analysis filters to decompose data into low-pass and high-pass frequency bands. When filtering is recursively applied to the low-pass frequency bands the QMF filter bank produces an octave band split or wavelet decomposition. Separable QMF filters reduce the computational complexity of the filter banks and make them very attractive for implementation. Entropy (H), energy or variance (E), and mean deviation (md) of the sub-band filtered signals are the most commonly used feature descriptors. If $Y(x_1, \dots, x_d)$ denotes the output from the sub-band filters then these above three

texture descriptors are estimated mathematically using the following relations:

$$H = -\frac{1}{\#} \sum_{x_1} \dots \sum_{x_d} |Y(x_1, \dots, x_d)| \log [|Y(x_1, \dots, x_d)|] \quad (2.3)$$

$$E = \frac{1}{\#} \sum_{x_1} \dots \sum_{x_d} (Y(x_1, \dots, x_d))^2 \quad (2.4)$$

$$md = \frac{1}{\#} \sum_{x_1} \dots \sum_{x_d} |Y(x_1, \dots, x_d)| \quad (2.5)$$

Chang and Kuo (1993) approached the problem of supervised texture-image classification by considering tree-structure wavelet transform for constructing the energy maps. Their experimental studies produced classification errors less than 5% on texture images from the Brodatz album (Brodatz, 1966). Fatemi-Ghomi et al. (1996) used the similar feature descriptors as Chang and Kuo (1993) but performed a larger number of experiments (some 800 in total) using ten different wavelet filters (Haar, Daubechies, and Coiflets). Their testing bed also comprised of a variety of composite image taken from the Brodatz database.

Laine and Fan (1993) exhibited the reliability of using texture signatures obtained from both energy and entropy metrics based on wavelet packets analysis. Their study used features computed from from Daubechies wavelets. Their experimental results showed that energy signatures computed from redundant wavelet packet representation produces more classification error than standard wavelet decomposition. And between energy-based and entropy-based texture signatures, the segmentation results from the former performed were slightly better when tested on Brodatz textures.

Unser (1995) described a new approach for characterization of texture properties at multiple scales using the overcomplete wavelet frame transform that was

translation invariant. Unser showed that this wavelet representation constitutes a tight frame in \mathbb{L}^2 and also proposed a fast iterative algorithm for computing the transform. He used the channel variance (energy) from various orthogonal wavelets (Battle-Lemarié functions) and bi-orthogonal wavelet (B-spline and D-spline functions) for textural characterization. Additionally, Unser also studied the effect of energy features from different scales (1, 2, and 3) on the segmentation. The experimental results on twelve images from Brodatz database showed that choice of wavelet functions had little impact on the overall performance. However, there was considerable improvement in the classification accuracy by including energy information from more number of scales: approximately, 93.88%, 98.7%, and 99.22% for scales 1, 2, and 3, respectively.

Lu et al. (1997) proposed a texture segmentation method based on wavelet-based approach using a set of high-frequency channel energy function as features and a multi-level thresholding technique. The experimental results showed that accurate results can be obtained from within two to four scales of wavelet decompositions for the images. The number of texture classes from Brodatz texture album and digital aerial photographs were correctly estimated with error rates as low as 5%.

Zhu and Yang (1998) performed texture classification at multiple scales using different mother wavelet functions, wavelet decomposition models and different levels of wavelet decomposition. They showed that the accuracy was dependent on the choice of wavelet filter length used for classification for a specific number of features in the images. The test images for all this method were also chosen from the Brodatz texture album (Brodatz, 1966).

2.2.2 Classification Schemes

The next stage in image segmentation is the classification step. Given a feature vector \vec{F}_i of a pattern or an object, the desired goal is to assign the \vec{F}_i to one of the possible classes, w_i . A *decision rule* partitions the feature space into regions S_i belonging to c number of classes, $\{1, 2, \dots, c\}$. There are two main divisions of classification: *supervised classification* (or discrimination) and *unsupervised classification* (sometimes referred in the statistics literature simply as classification or clustering). The first category assumes that the class label of each texture is known (i.e. an "instructor" is available). This information is readily used as exemplars during the classifier design. But in some practical applications like identifying different soil horizons from a satellite remotely without human interaction, the different classes might not be available. This is addressed by the second category of techniques. In unsupervised techniques, the data are not labelled and the goal is to find groups in the data and the features descriptors that distinguish one group from another.

There are two main categories for supervised classification. First category deals with supervised classification via *Bayes' rule* and estimation of the class-conditional densities. The second category take a *discriminant function* approach to supervised classification. The former can be further divided into two subcategories. First, namely, *parametric methods*, assumes a knowledge of the underlying class-conditional probability density function, i.e., the probability density functions of the feature vectors for a given class. In many applications these will usually be unknown and must be estimated from a set of correctly classified samples termed the *design* or

training set. The second, namely, *non-parametric methods*, develops decision rules that use the data estimate the decision boundaries directly, without the explicit calculation of the probability density functions. A detailed review on all types of classification schemes is beyond the scope of this research. Most of commonly used methods are well documented in several pattern recognition textbooks (Duda et al., 2001; Fu, 1982; Jain and Dubes, 1988; Kendall and Stuart, 1977; Lillesand and Kiefer, 1987; Pitas, 2000; Schalkoff, 1992; Vapnik, 1998; Webb, 2002).

Chang and Kuo (1993) approached the problem of supervised texture-image classification by considering tree-structure wavelet transform and constructing the energy maps. They illustrated the efficiency of their technique by comparing different types of minimum-distance classifiers, viz. Euclidean, Mahalanobis and Bayes decision rule. Their experimental studies using the three classifiers gave an classification accuracy of 95.3%, 99.7%, and 99.6%, respectively, on texture images from the Brodatz album (Brodatz, 1966).

Laine and Fan (1993) exhibited the reliability of using texture signatures obtained from both energy and entropy metrics based on wavelet packets analysis. Their study used features computed from from Daubechies wavelets and the classification was done using two-layer neural network classifier (ANNs) and minimum-distance (MD) classifier. Their experimental results showed that energy signatures computed from redundant wavelet packet representation gave more classification error than standard wavelet decomposition. And between energy-based and entropy-based texture signatures, the segmentation results from the former performed were slightly better when tested on Brodatz textures.

Unser (1995) proposed an overcomplete wavelet transform representation and studied the effect of energy features computed from various orthogonal (Battle-Lemariè) and bi-orthogonal wavelet (B-spline and D-spline) functions for textural characterization based on minimum-distance Bayes' classifier. Additionally, he also studied the effect of energy features from different scales (1, 2, and 3) on the segmentation. The experimental results on twelve images from Brodatz database showed that choice of wavelet functions had little impact on the overall performance. However, there was considerable improvement in the classification accuracy by including energy information from more number of scales: approximately, 93.88%, 98.7%, and 99.22% for scales 1, 2, and 3, respectively.

Fatemi-Ghomi et al. (1996) performed large number of experiments (some 800 in total) using ten different wavelet filters (Haar, Daubechies, and Coiflets) on a variety of composite image taken from the Brodatz database. The test images were segmented using a *c*-means fuzzy classifier based on energy features computed from the sub-band filtered images. They also introduced a two-point correlation function for measuring the performance of the segmentation results and showed that it can quantify performance in a way that correlates well with ground truth measures. Most of the texture mosaics were segmented with less than 2% error, irrespective of the wavelet filter, suggesting that the choice of wavelet filter plays an insignificant role for texture analysis.

Lu et al. (1997) proposed an unsupervised texture segmentation using wavelet-based approach using a set of high-frequency channel energy function as features and a multi-level thresholding technique. The experimental results showed that accurate

results can be obtained from within two to four scales of wavelet decompositions for the images. The number of texture classes from Brodatz texture album and digital aerial photographs were correctly estimated with error rates as low as 5%.

(Kervrann and Heitz, 1995) proposed an algorithm for unsupervised texture segmentation technique that does not require knowledge about the different texture regions, their parameters, or the number of available texture classes. Their method relies on analysis of local and global second and higher order spatial statistics of the original images and the Bayesian estimate of segmentation map is modeled using an augmented-state Markov random field and is computed using a deterministic relaxation algorithm.

2.2.3 Summary

The vast majority of research on texture segmentation has been targeted towards development of features for textural characterization which can be divided into four categories. Structural descriptors can provides a good symbolic description of the image useful for certain applications, like detection of changes in bone microstructure. However, these texture features are more useful for synthesis than analysis tasks.

Statistics-based texture descriptors are the most widely used descriptors because they represent the texture indirectly by the non-deterministic properties that govern the distributions and relationships between the grey levels of an image. Commonly used statistics-based features for texture analysis are derived from the first-order central moments and co-occurrence matrix. However, the problem with these

descriptors is that they are derived from pixels within tiles at a single scale.

Model-based texture descriptors attempt to interpret an image texture by using fractals and stochastic models. Both these methods have been shown to be useful for modeling some natural textures. However, in practice, the computational complexity arising in the estimation of stochastic model parameters is the primary problem. Similarly, the lack of the orientation selectivity and incapability to describe local image structures limits the application of fractal-based descriptors.

Space-frequency based features derived from Fourier analysis based descriptors, like power spectrum performs poorly in practice due to its lack of spatial localization. In contrast, the pseudo-Wigner distribution (PWD) allows feature descriptors to have superior joint spatial/spatial-frequency resolution and also encodes the phase information. These properties, especially the latter, makes PWD-based descriptors useful for textures that differ only in phase. However, PWD-based features perform poorly when used for characterization of complex textures due to aliasing. Furthermore, the PWD yields a set of potential texture features of very large dimensionality (30). Feature descriptors computed from Gabor filters provide better spatial localization than Fourier transforms. However, these descriptors have practical limitations because there is usually no single filter resolution at which the spatial structure in natural textures can be localized. In comparison to the feature descriptors estimated from Gabor filters, the wavelet analysis allows for varying the spatial resolution to represent textures at the most suitable scale. It also provides a wide range of wavelet functions tailored for specific texture analysis of image data. These properties make the wavelet transform attractive for texture segmentation.

The main shortcomings of the texture segmentation methods reviewed earlier are as follows. Although, image segmentation techniques using texture signatures from multiple scales have shown promising results for texture-predominant image from sources like Brodatz album and synthesized textured images, the performance of these methods on heterogeneous biological image data is sparse. Hence, there is a need to develop a texture segmentation method that can perform equally well for characterization of heterogeneous materials in multi-dimensional images obtained from different imaging sensors.

Additionally, most of the multi-scale and multi-channel methods only uses few descriptors, like entropy, energy and mean-deviation, for texture characterization. Although, these features have shown great potential to discriminate textured patterns in images they are insufficient to quantitatively represent the heterogeneity within the materials in image data sets. In this context, statistics-based features offer a wide range of texture measures estimated from pixels within a tile. However, these descriptors have limited application as they primarily estimated from tiles tiles at only one scale. Extending these single-scale statistical feature descriptors into multi-scale texture analysis framework will therefore expand the repository of the available descriptors for texture characterization.

Lastly, the features descriptors from single and multiple scales uses an arbitrary tile size and pre-determined number of scales, respectively, for characterization of textures without any quantitative justification. For instance, the single scale statistical measures are computed from a sufficiently *large* sample material. Similarly, most of the multi-resolution texture analysis methods use features estimated from

sub-band filtered images up to 3 or 4 scales. This clearly demonstrates the need for developing a quantitative strategy for identification of the optimal tile size.

2.3 Objectives

The overall goal of this chapter is to develop a texture-based segmentation technique for characterization of heterogeneous bio-materials in image data. The specific objectives of the research addressed in this chapter are:

Objective 1: Develop a supervised texture-based segmentation technique that can quantitatively characterize the heterogeneity of a material at a coarser scale by accounting for the heterogeneity from sub-scales.

Objective 2: Evaluate the developed technique for segmenting heterogeneous materials in biological image data.

The texture-based segmentation technique should be able to identify the characteristic length (representative elementary scale) at which the properties of the heterogeneous materials become homogeneous. It should be able to identify the optimal feature descriptors and classifier combination that can be used for segmenting real biological image data. The feature descriptors should also be the parameters that can be used directly in stochastic up-scaling numerical techniques. It should be easily extendable to image data in arbitrary dimensions and applicable to different imaging modalities so that it accommodates both 2-D images (e.g., air photos) and 3-D images (MR and CT scans).

2.4 Materials and Methods

In this section, the materials and methods required for accomplishing the main objectives of the research are described. Section §2.4.1 provides the discussion on methods required for the developing the supervised segmentation strategy with emphasis on different components, like feature descriptors and classification methods, for successful characterization and identification of heterogeneous materials in image data. Additionally, the two test suites needed for analyzing the various aspects of the developed segmentation strategy are also described. Section §2.4.2 describes the biological image data sets from different imaging modalities and the methods used for evaluating the developed segmentation.

2.4.1 Development

The approach used in this work to develop the segmentation procedure consists of modifying the appropriate steps of existing multi-resolution based texture segmentation methods, such that: (1) it can characterize macroscopic entities by accounting for their heterogeneity from sub-scales; (2) it can identify the support scale at which heterogeneous materials can be treated as continua; and (3) it can be applied without any modifications to multi-dimensional image data acquired from different modalities. The strategy of multi-resolution analysis (MRA) for texture segmentation is used as a starting point because the definition of texture used in image segmentation literature is conceptually similar to the definition of representative elementary scale (RES) used in continuum mechanics. Moreover, research on human visual system

for textural characterization indicates a strong resemblance with multi-channel and multi-scale approaches. Specifically, the developed method will use Haar wavelet multi-resolution analysis (MRA) due to the computational simplicity that it offers because of its orthogonal and symmetric properties.

According to the literature review, most, if not all, of the multi-resolution based texture segmentation methods can be broken down into three main steps: (1) extract or select feature descriptors that characterizes different materials (for example, multi-channel entropy, mean deviation, and energy); (2) use classification method to segment the materials; and (3) error assessment of the segmentation results. The modifications used to adapt these steps for accurate characterization of heterogeneous materials in multi-dimensional images from different modalities will require: (1) development and analysis of additional feature descriptors (besides entropy) that are based on up-scaled stochastic continuum concept; (2) development of a strategy for identifying the representative elementary scale at which the heterogeneous materials becomes continua; and (3) selection of the most optimal combination of feature descriptors and classification method based on rigorous analysis.

The first modification is needed to capture micro-scale heterogeneity properties of macro-scale entities if such image data is to be used in up-scaled transport models. In this context, conventionally used measures (like entropy and energy), although useful, are not sufficient to successfully characterize heterogeneous materials. This shortcoming is overcome by developing and analyzing the potential of three types of feature descriptors for textural characterization: first-order central moments, \vec{M} , entropy computed from multiple scales, \vec{H} , and directional correlation lengths, \vec{Z} .

The first order central moments and directional correlation lengths are included because they can be directly used as input parameters in up-scaled transport models, whereas the inclusion of entropy is based on its ability cited in literature for analyzing texture predominant signals. The feature descriptors will be calculated either by using existing formulae or by extending them using Haar wavelet multi-resolution analysis (explained below).

The second modification is required to ensure that the macro-scale material signatures, to be identified, is based on a quantitative or objective procedure rather than intuitive or subjective selection procedure. For example, the multi-resolution methods used for texture segmentation method uses arbitrary number of scales for textural characterization without any quantitative justification. In this study, a strategy will be developed to identify the characteristic length of a textured material (representative elementary scale, RES) using stochastic continuum mechanics concept.

The third modification is necessary because the developed method when applied for analyzing heterogeneous bio-images should be optimal both in terms of computational efficiency and accuracy. This will be accomplished by selecting an optimal combination of feature descriptors (developed in step 2) and three classification methods (explained below), namely, Gaussian maximum likelihood classifier (LVQ), minimum distance classifier (MD), and learning vector quantization (LVQ) classifiers, using two test suites: (1) three real 2-D images–T-1 and (2) twelve synthetic texture predominant mosaics–T-2; will be used to analysis. The latter, specifically, is selected owing to its extensive use in comparative studies of several texture seg-

mentation methods (Mäenpää et al., 2000; Randen and Husøy, 1999). The analysis experiments will study the effects of combinations of feature descriptors and classification methods on these test suites using a modified leave-one-out-error which accounts for size of feature vector in the segmentation results.

With the modifications listed above the segmentation method developed in this chapter will consist of the following steps:

1. pre-processing the input data and extracting samples for each material in the image;
2. calculating spatial-statistics from wavelet MRA and identifying characteristic length of texture (RES);
3. calculating the material signatures (features) at the identified RES support; and
4. classifying the heterogeneous materials in the image at RES support and evaluating the segmentation results.

The remaining part of this section is organized as follows. Section §2.4.1.1 describes the method(s) used for pre-processing the input multi-dimensional image data for further analysis and also describes the technique to be used for sample selection. Section §2.4.1.2 introduces Haar wavelet multi-resolution analysis and various feature descriptors that will be used and developed for segmenting heterogeneous materials in images. Section §2.4.1.3 describes the method that will be used to identify the representative elementary scale using stochastic continuum concept. Section

§2.4.1.4 explains the classification schemes that will be analyzed for incorporating into the developed segmentation method along with the performance index that will be used for evaluating the segmentation results. Sections §2.4.1.5 and §2.4.1.6, describe the two-dimensional test suites that will be used for rigorous analysis of the developed texture segmentation technique prior to applying the method on real heterogeneous bio-images.

2.4.1.1 Pre-Processing and Sample Selection

Let the input data for the segmentation method be a d -dimensional image, such that

$$f(x_1, x_2, \dots, x_d \mid x_i \in [0, 2^n - 1]), \forall n \in \mathbb{Z}^+ \quad (2.6)$$

The pre-processing step in the developed segmentation technique will (mostly) involve checking the input data for the dyadic requirement needed for Haar wavelet analysis. Additionally, it can also include conventional image processing steps, like, smoothing for noise removal and re-sampling anisotropic data.

The second step of the developed supervised segmentation method consists of selecting samples corresponding to the each of the different textured materials present in the image data. In most instances, the allowable size of the sample materials will be dependent on the input image data. For example, images with low resolution can prevent selecting sufficiently large samples for further analysis. Caution should be practiced during supervised selection of representative samples because of two reasons: (1) they should truly reflect the material being characterized, hence, should not be selected from two or more materials; (2) they should also be of

sufficient size to enable the estimation of the RES support. This supervised selection method can be automated when ground truth data is available by computing the 2^d -tree representation of the d -dimensional ground truth data (see Section §3.4.1) and then selecting samples for each material present from the corresponding location in the raw input data.

2.4.1.2 Feature Descriptors

The three feature descriptors: (1) moments-based feature vector (\vec{M}); (2) entropy-based feature vector (\vec{H}); and (c) correlation lengths estimated from the directional autocorrelation functions (\vec{Z}), used and developed in this study should capture micro-scale heterogeneity characteristics of a macro-scale entity. That is, if $\tau_j(x_1, x_2, \dots, x_d)$ represents a non-overlapping tile at scale j , such that,

$$f = \{ \tau_j(x_1, x_2, \dots, x_d) \mid x_i \in [0, 2^{n-j} - 1] \}, \quad (2.7)$$

then the goal is to be able to describe heterogeneity of the material within $\tau_{j,i}$ (at location x_i) using feature descriptors calculated over tiles at lower scales, $\tau_{j-1,i}, \dots, \tau_{1,i}$. For example, in Figure 2.2 the goal is identify the feature descriptors of tile $\tau_2(0, 0)$ at scale 2 in terms of $\tau_1(0 \dots 1, 0 \dots 1)$ at scale 1. This multi-scale representation with non-overlapping tiles given by equation 2.7 and the multi-resolution Haar wavelet decomposition are two closely related. Figure 2.2 illustrates the equivalence between both the representations for a two-dimensional case. Hence, any analysis that needs to be done on tiles can now be performed directly in wavelet space.

In one dimension, Haar wavelet analysis involves convolution of the input

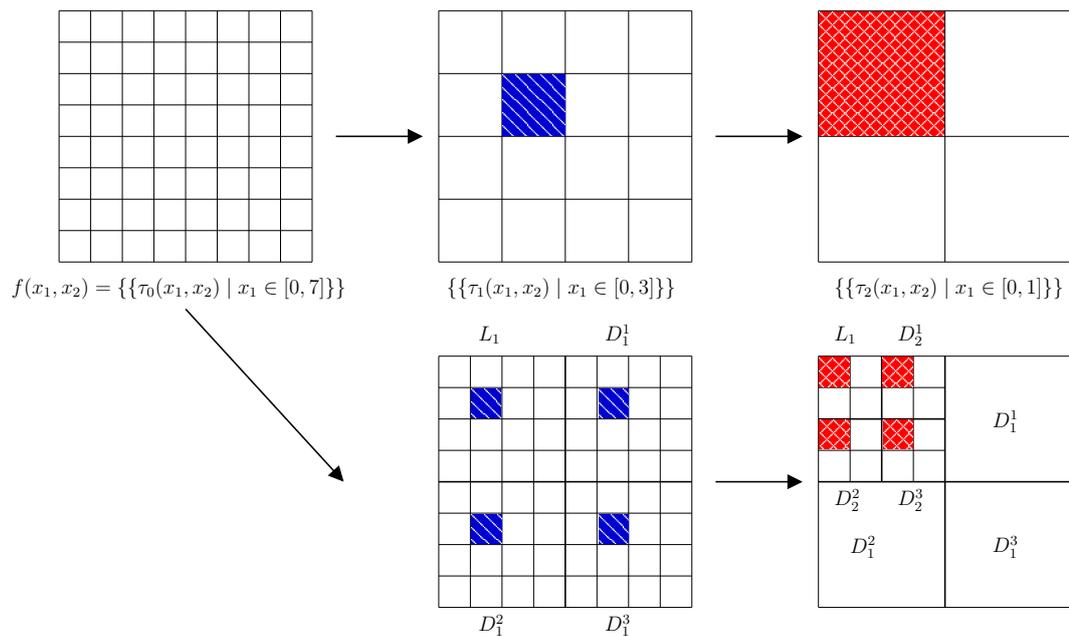


Figure 2.2: *Illustration of the equivalence between the multi-scale representation and wavelet transform. The tiles in the multi-scale representation at different scales and their corresponding wavelet coefficients in all the sub-band filter outputs are shown with same pattern.*

signal, $f(x_1)$, with a low-pass filter (smoothing function, $h(x_1) = \frac{1}{\sqrt{2}} [1 \ 1]$) and a high-pass filter (detail function, $g(x_1) = \frac{1}{\sqrt{2}} [1 \ -1]$). This process is recursively applied on the low-pass filter output from previous scale, $k - 1$, such that, the wavelet representation of the input signal at scale k follows the relation

$$f(x_1) = \sum_n l_k(n)h(x_1) + \sum_{j=1}^k \sum_n d_j(n)g(x) \quad (2.8)$$

The coefficients of first term in this expansion represents low resolution *approximate signal*, $L_k = \{l_k(n)\}$, whereas the coefficients $D_j = \{d_j(n)\}$ constitute the *detail signal* at scale j . The complete set of wavelet coefficients, $\{L_k, \{D_j\}_{1 \leq j \leq k}\}$, is called the *wavelet representation* of the signal f at *depth* k .

In terms of matrix notation, the approximate and the detail low resolution signals at scale k from the Haar wavelet analysis can also be computed using the scalar product,

$$\{l_{k+1}(n), d_{k+1}(n)\} = \langle \mathcal{H}_{2 \times 2}, \{l_k(2n - 1), l_k(2n)\} \rangle \quad (2.9)$$

where $\mathcal{H}_{2 \times 2}$ is the Haar block matrix given by:

$$\mathcal{H}_{2 \times 2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.10)$$

According to filter-bank theory, the h and g filters are only the analysis part and in order to restore the original signal, the orthogonal and symmetric properties of the Haar wavelet function can be used to construct the synthesis filter-bank, namely, $\tilde{h}(x)$ and $\tilde{g}(x)$. These filters are defined as conjugate to the $h(x)$ and $g(x)$, such that

$$\tilde{h}(x) = h(-x) \quad \text{and} \quad \tilde{g}(x) = g(-x) \quad (2.11)$$

Using these synthesis part of the filter bank the signal at scale k can be reconstructed either using,

$$l_k(n) = \sum_n \left[l_{k+1}(n) \tilde{h}(x_1 + 2n) + d_{k+1}(n) \tilde{g}(x_1 + 2n) \right] \quad (2.12)$$

or using the block matrix notation

$$\{l_k(2n - 1), l_k(2n)\} = \langle \mathcal{H}_{2 \times 2}, \{l_{k+1}(n), d_{k+1}(n)\} \rangle \quad (2.13)$$

The analysis and synthesis parts for one-dimensional case are illustrated in Figure 2.3. The one-dimensional wavelet analysis takes the input signal, filters and sub-samples the outputs from analysis filter-bank, thus producing a approximate signal and detail signal at higher scales. This sub-sampling is necessary to maintain equal number of coefficients in the wavelet space as that in the input signal.

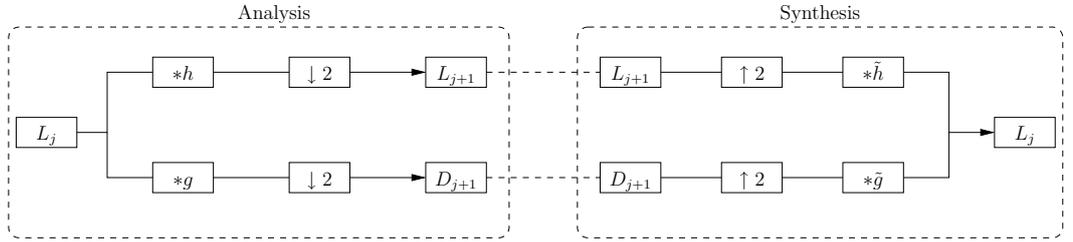


Figure 2.3: *The analysis and synthesis filter bank schemes for computation of the wavelet coefficients using quadrature mirror filters. * denotes convolution and $\uparrow 2$ ($\downarrow 2$) down-sampling (up-sampling) by a factor of 2.*

The extension of the 1-D DWT to 2-D images, $f(x_1, x_2)$, using sub-band filtering scheme is achieved by taking convolution of the input signal with 2-D Haar sub-band matrix, $\mathcal{H}_{4 \times 4}$ (Strang and Nguyen, 1997). The 2-D Haar sub-band matrix

is obtained by taking the tensor product¹ of $\mathcal{H}_{2 \times 2}$, that is,

$$\begin{aligned} \mathcal{H}_{4 \times 4} &= \mathcal{H}_{2 \times 2} \otimes \mathcal{H}_{2 \times 2} \\ &= \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \end{aligned} \quad (2.14)$$

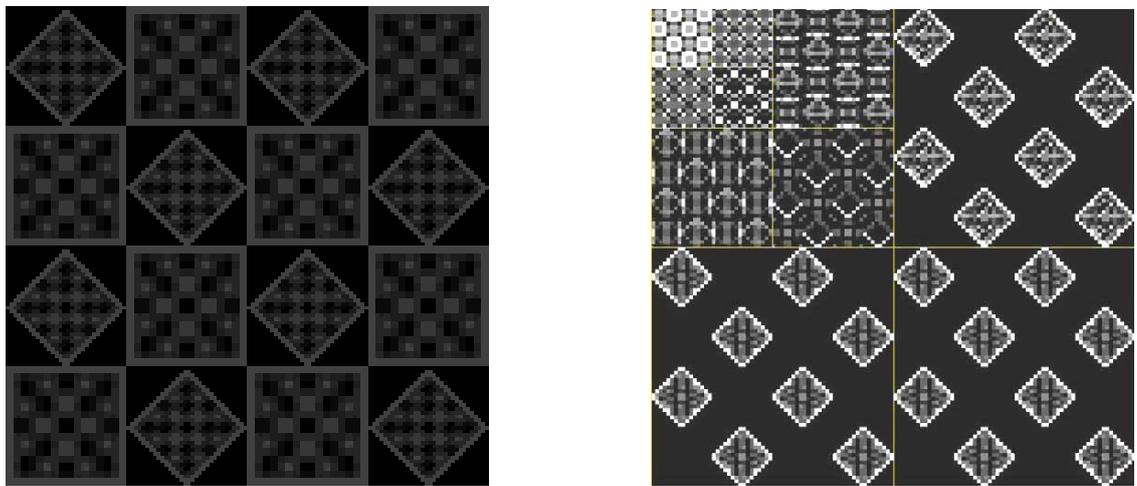
The 2-D Haar DWT generates four low resolution (or coarse scale) images: an approximate image, L_j and three detail images, D_1^j , D_2^j and D_3^j capturing the vertical, horizontal, and diagonal details, respectively, of the original image. Similar to the one-dimensional case, the four sub-images at coarser scale, j , are generated by recursively applying the analysis Haar filter-bank on the approximate (smooth) sub-image L_{j-1} and are calculated using

$$\begin{pmatrix} L_j(n_1, n_2) \\ D_j^1(n_1, n_2) \\ D_j^2(n_1, n_2) \\ D_j^3(n_1, n_2) \end{pmatrix} = \mathcal{H}_{4 \times 4} \begin{pmatrix} L_{j-1}(2n_1 - 1, 2n_2 - 1) \\ L_{j-1}(2n_1, 2n_2 - 1) \\ L_{j-1}(2n_1 - 1, 2n_2) \\ L_{j-1}(2n_1, 2n_2) \end{pmatrix} \quad (2.15)$$

Figure 2.4(b) illustrates the standard wavelet decomposition up to depth 3 for the input image shown in Figure 2.4(a) and the corresponding pyramidal representation for the decomposition is shown in Figure 2.4(c).

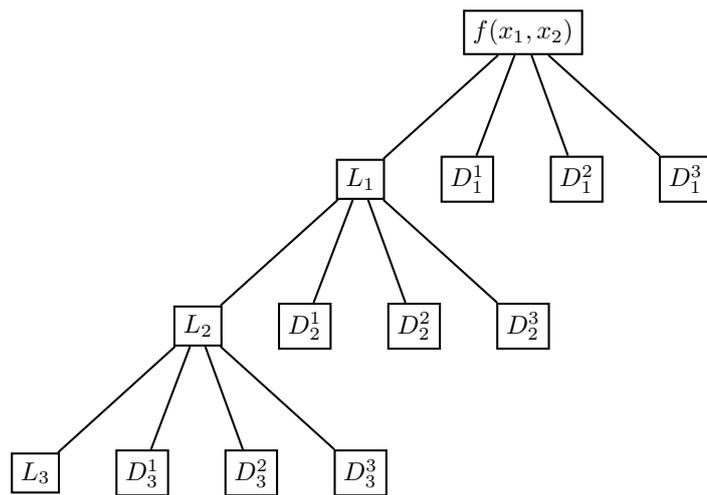
Similarly, for a three-dimensional image data, $f(x_1, x_2, x_3)$, the separable 3-D

¹A tensor product has order $N_1 N_2$ when the matrices have orders N_1 and N_2 . The matrix of order N_1 appears N_2 times on each block row of the tensor product, multiplied by entries from the matrix of order N_2 .



(a)

(b)



(c)

Figure 2.4: (a) sample 2-D image of size 128×128 pixels; (b) typical organization of the approximation and detail images for a 2-D standard wavelet transform up to depth 3; and (c) the corresponding pyramidal representation of the standard wavelet decomposition

DWT can be computed by using $\mathcal{H}_{8 \times 8}$, which will generate eight sub-band signals consisting of one average sub-band signal, L_1 , and seven detail sub-band signals, $D_1^i, \forall i \in [1, 7]$. The subsequent coarser representations of the original signal at scale j is obtained by recursively applying Haar filter-bank matrix on the average sub-band signal, L_{j-1} and are computed using,

$$\begin{pmatrix} L_j(n_1, n_2, n_3) \\ D_j^1(n_1, n_2, n_3) \\ D_j^2(n_1, n_2, n_3) \\ D_j^3(n_1, n_2, n_3) \\ D_j^4(n_1, n_2, n_3) \\ D_j^5(n_1, n_2, n_3) \\ D_j^6(n_1, n_2, n_3) \\ D_j^7(n_1, n_2, n_3) \end{pmatrix} = \mathcal{H}_{8 \times 8} \begin{pmatrix} L_{j-1}(2n_1 - 1, 2n_2 - 1, 2n_3 - 1) \\ L_{j-1}(2n_1, 2n_2 - 1, 2n_3 - 1) \\ L_{j-1}(2n_1 - 1, 2n_2, 2n_3 - 1) \\ L_{j-1}(2n_1, 2n_2, 2n_3 - 1) \\ L_{j-1}(2n_1 - 1, 2n_2 - 1, 2n_3) \\ L_{j-1}(2n_1, 2n_2 - 1, 2n_3) \\ L_{j-1}(2n_1 - 1, 2n_2, 2n_3) \\ L_{j-1}(2n_1, 2n_2, 2n_3) \end{pmatrix} \quad (2.16)$$

The moment-based features used in this study are the first-order four central moments, viz. mean ($\hat{\mu}$), variance ($\hat{\nu}$), skewness ($\hat{\chi}$), and kurtosis ($\hat{\kappa}$) and are computed from the elements in $\tau_{j,i}$, using the following relations:

$$\hat{\mu}(\tau_{j,i}) = \frac{\sum_{x_1} \dots \sum_{x_d} \{\tau_{j,i}\}}{\#(\tau_{j,i})} \quad (2.17)$$

$$\hat{\nu}(\tau_{j,i}) = \frac{\sum_{x_1} \dots \sum_{x_d} [\{\tau_{j,i}\} - \hat{\mu}(\tau_{j,i})]^2}{\#(\tau_{j,i})} \quad (2.18)$$

$$\begin{aligned} \hat{\chi}(\tau_{j,i}) &= \frac{\sum_{x_1} \dots \sum_{x_d} [\{\tau_{j,i}\} - \hat{\mu}(\tau_{j,i})]^3}{\#(\tau_{j,i}) [\hat{\nu}(\tau_{j,i})]^{1.5}} \\ &= \frac{\hat{\alpha}(\tau_{j,i})}{[\hat{\nu}(\tau_{j,i})]^{1.5}} \end{aligned} \quad (2.19)$$

$$\begin{aligned}
\hat{\kappa}(\tau_{j,i}) &= \frac{\sum_{x_1} \dots \sum_{x_d} [\{\tau_{j,i}\} - \hat{\mu}(\tau_{j,i})]^4}{\#(\tau_{j,i}) [\hat{\nu}(\tau_{j,i})]^2} \\
&= \frac{\hat{\beta}(\tau_{j,i})}{[\hat{\nu}(\tau_{j,i})]^{2.0}}
\end{aligned} \tag{2.20}$$

This makes the moment-based feature vector, \vec{M} , a point in \mathbb{R}^4 feature space. The mathematical framework for estimating the first order moments directly from wavelet MRA for multi-dimensional images will be developed and presented in Section §2.5.1.1.

In general, The entropy of any signal, $Y(x_1, \dots, x_d)$, is estimated using,

$$H(Y(x_1, \dots, x_d)) = -\frac{1}{\#} \sum_{x_1} \dots \sum_{x_d} |Y(x_1, \dots, x_d)| \log [|Y(x_1, \dots, x_d)|] \tag{2.21}$$

where the absolute value of the signal is used to overcome negative wavelet coefficients. The multi-scale entropy feature vector, \vec{H} , for tile $\tau_{j,i}$ is calculated from its corresponding wavelet coefficients, using the following relation,

$$\vec{H}(\tau_{j,i}) = \{H(L_{j,i}), H(D_{k,i}^n) \mid k = 1, \dots, j; n = 1, \dots, 2^d - 1\} \tag{2.22}$$

This makes the entropy feature vector, \vec{H} , a point in $\mathbb{R}^{(2^d-1)j+1}$ feature space.

In addition to the above features, ACF-based features will also be used to account for spectral characterization of textures within an image. The ACF-based features, \vec{Z} , to be used in this study are estimated in two steps: (1) calculating the auto-correlation function in each principal direction; and (2) estimating the correlation length by curve-fitting the calculated ACF to a theoretical ACF. First, the directional auto correlation function at lag, Δh , for the d -dimensional dyadic

image, $f(x_1, \dots, x_d)$, is computed using,

$$Z_i(h) = \frac{1}{Z_i(0) N_p} \sum_{x_1=0}^{2^n-1} \dots \sum_{x_i=0}^{2^n-1} \dots \sum_{x_d=0}^{2^n-1} f(x_1, \dots, x_i, \dots, x_d) f(x_1, \dots, x_i + \Delta h, \dots, x_d) \quad (2.23)$$

where, N_p is the total number of number of pairs at a lag distance of z , such that $0 \leq \Delta h \leq 2^n - 2$. The $Z_i(0)$ in the denominator is used to normalize the ACFs so that $Z_i(\Delta h)$ attains a maximum value of 1 at zero-lag. Since ACF is directly related to the *power spectral density* of the signal it is instead estimated using FFTs, which is much efficient from an implementation standpoint (Pitas, 2000). Second, the correlation lengths, λ_i , are estimated by *curve-fitting* an theoretical exponential ACF, $\exp^{-\frac{\Delta h}{\lambda_i}}$, to the ACF computed using FFT. Hence, the ACF-based entropy feature vector, $\vec{Z} = \{\lambda_i\}$, is a point in \mathbb{R}^d feature space.

The analysis experiments will use seven combinations of the above three feature vectors, shown in Table 2.1, for determining the optimal feature vector.

2.4.1.3 Identification of RES support

In this study, the characteristic length at which heterogeneous materials in the bio-image data become a continua will be identified using stochastic continuum concept. The method used for identification entails two steps: (1) calculating spatial-moments of the materials as a function of sampling tile size (using Haar wavelet coefficient) and (2) selecting the sample tile size at which the statistics stabilize as the representative elementary scale. Figure 2.5 illustrates the concept representative elementary scale where the plot shows the trend in a particular material property (for example, porosity) measured over sampling volume V , as $V \rightarrow 0$. For large sampling vol-

Table 2.1: *Combination of feature vectors and their corresponding dimensionality at scale j*

Feature Vector	Features	Feature Space Size
$\vec{F1}$	$\{\vec{M}\}$	4
$\vec{F2}$	$\{\vec{H}\}$	$3j + 1$
$\vec{F3}$	$\{\vec{Z}\}$	2
$\vec{F4}$	$\{\vec{M}, \vec{Z}\}$	6
$\vec{F5}$	$\{\vec{H}, \vec{M}\}$	$3j + 5$
$\vec{F6}$	$\{\vec{H}, \vec{Z}\}$	$3j + 3$
$\vec{F7}$	$\{\vec{H}, \vec{M}, \vec{Z}\}$	$3j + 7$

umes, the curve changes smoothly but distinctly, whereas for intermediate sampling volume, the curve becomes stable at some fairly precise value of material property (porosity) and then becomes erratic at an even smaller V where pore-scale variations become important. The representative elementary volume is the value of V above which the fluctuations are negligible. It is expected that, by analyzing trends of spatial-moments computed on different sizes of sampling tiles will be useful in RES support identification.

2.4.1.4 Classification Schemes

The developed segmentation method in this study uses three classification schemes: Gaussian maximum likelihood method (GM); Knn method with $K = 1$ (MD);

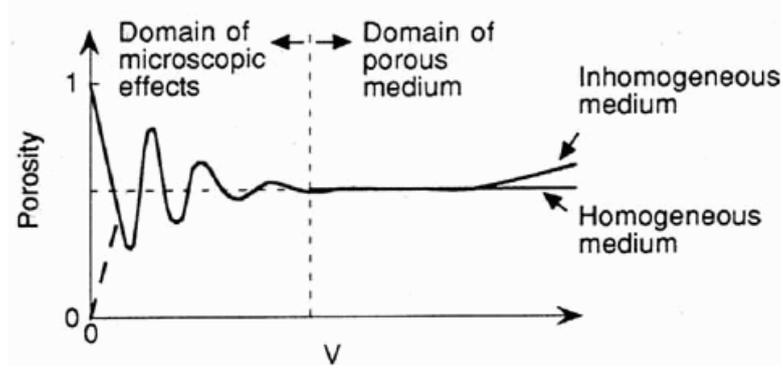


Figure 2.5: Plot of a material property as a function of sampling volume, adapted from Bear (1972).

and learning vector quantization (LVQ), for segmenting image data into different classes. The first two classifiers are based on Bayes' decision rule and belongs to the category of *parametric* and *non-parametric* methods, respectively, whereas LVQ classifier belongs to the category of *clustering* techniques.

According to the Bayes' decision rule, if a signal has c classes, w_1, w_2, \dots, w_c , with known *a priori* probabilities $p(w_1), p(w_2), \dots, p(w_c)$ and no information regarding an object other than the class conditional distribution then for minimizing the probability of making an error the object is assigned to class w_j if

$$p(w_j) > p(w_k) \quad \forall k = 1, \dots, c; \quad k \neq j \quad (2.24)$$

This classifies all objects as belonging to one class. For classes with equal probabilities patterns are assigned arbitrarily between those classes. However, if a feature vector \vec{F}^+ describing the object is available, then the decision rule based on probabilities is to assign \vec{F}^+ to class w_j if the probability of class w_j given the feature vector \vec{F}^+ , $p(w_j | \vec{F}^+)$, is greatest over all the classes w_1, w_2, \dots, w_c . That is, assign

\vec{F}^+ to class w_j if

$$p(w_j|\vec{F}^+) > p(w_k|\vec{F}^+) \quad \forall k = 1, \dots, c; k \neq j \quad (2.25)$$

The *a posteriori* probabilities $p(w_j|\vec{F}^+)$ can be expressed in terms of the *a priori* probabilities and the class-conditional density functions $p(\vec{F}^+|w_j)$ by using the Bayes' theorem,

$$p(w_j|\vec{F}^+) = \frac{p(\vec{F}^+|w_j)p(w_j)}{p(\vec{F}^+)} \quad (2.26)$$

and so the decision rule 2.25 can be rewritten as, assign \vec{F}^+ to w_j if

$$p(\vec{F}^+|w_j)p(w_j) > p(\vec{F}^+|w_k)p(w_k) \quad \forall k = 1, \dots, c; k \neq j \quad (2.27)$$

This is known as the Bayes' rule of *minimum error*.

Parametric classifiers assume that the class conditional probabilities have a known functional form and only depends on few parameters. Hence, parametric classifiers achieve the goal of segmentation by estimating those distribution parameters from a design set, \vec{F}^* , where

$$\vec{F}^* = \left\{ \{\vec{F}_{n_1, w_1}^*\}, \{\vec{F}_{n_2, w_2}^*\}, \dots, \{\vec{F}_{n_c, w_c}^*\} \right\}, \quad \forall \vec{F}_{n_i, w_i}^* \in \mathbb{R}^d,$$

and n_i denotes the number feature vectors for class w_i in the input signal. Many parametric classifiers can be designed using different forms or mixtures of parametric distributions. Among them, classifiers using Gaussian distribution is the most popular choice. The multi-variate *Gaussian classifier* assumes that class conditional probabilities follow a normal distribution, i.e.,

$$p(\vec{F}^+|w_i) = \frac{1}{\sqrt{(2\pi)^d |C_i|}} e^{\left[-\frac{1}{2}(\vec{F}^+ - \vec{\mu}_i)^T C_i^{-1} (\vec{F}^+ - \vec{\mu}_i) \right]} \quad (2.28)$$

with: $\vec{\mu}_i = E \left[\vec{F}_{n_i, w_i} \right]$ and $C_i = E \left[\left(\vec{F}_{n_i, w_i} - \vec{\mu}_i \right) \left(\vec{F}_{n_i, w_i} - \vec{\mu}_i \right)^T \right]$, where d is the dimensionality of the feature vector and $\vec{\mu}_i$ and C_i denote vector of means and covariance matrix of the design set \vec{F}_{n_i, w_i} , respectively. Classification is achieved by assigning the new pattern \vec{F}^+ for which the posterior probability, $p \left(w_j | \vec{F}^+ \right)$, is the greatest, or equivalently $\ln \left(p \left(w_j | \vec{F}^+ \right) \right)$. Using Bayes' rule and the normal assumption for the conditional densities, the posterior probability is given by,

$$p \left(w_i | \vec{F}^+ \right) = -\frac{1}{2} \left(\vec{F}^+ - \vec{\mu}_i \right)^T C_i^{-1} \left(\vec{F}^+ - \vec{\mu}_i \right) - \frac{1}{2} \ln (|C_i|) - \frac{d}{2} \ln (2\pi) + \ln (p(w_i)) - \ln \left(p \left(\vec{F}^+ \right) \right) \quad (2.29)$$

where the prior probability, $p(w_i)$ is estimated by $n_i / \sum_{i=1}^c n_i$. Since $p \left(\vec{F}^+ \right)$ is independent of class, the *quadratic discriminant rule* is: assign \vec{F}^+ to w_i if $g_i > g_j \forall j \neq i$, where

$$g_i \left(\vec{F}^+ \right) = -\ln p(w_i) + \frac{1}{2} \ln |C_i| + \frac{1}{2} \left[\left(\vec{F}^+ - \vec{\mu}_i \right)^T C_i^{-1} \left(\vec{F}^+ - \vec{\mu}_i \right) \right] \quad (2.30)$$

The last part in the right hand side of the quadratic discriminant rule represents the Mahalanobis distance between the \vec{F}^+ and the feature vectors of the design set (Webb, 2002, see page 167). Henceforth, the classifier given by equation 2.30 will be referred to as the Gaussian maximum likelihood with Mahalanobis distance metric (GM).

Non-parametric classifiers, on the other hand, do not assume any known parametric form for the class likelihoods distributions and estimate probabilities, $p \left(\vec{F}^+ | w_i \right)$, using proximity rules directly from the design set. The K -nearest-neighbor (Knn) method is one of the simplest, yet viable, method for density estimation. The Knn classifier uses *proximity* of \vec{F}^* to the K nearest samples from the

design set as the main criteria for determining the class membership of \vec{F}^* . If K is the number of samples, out of a total of n , falling within a volume V , then the Knn approach is to fix the probability K/n (or equivalently, for a given number of design samples $n = \sum_{i=1}^c n_i$, to fix K) and to determine the d -dimensional volume V which contains K design samples centered on the point \vec{F}^+ . Suppose that in the first K design samples there are K_i in class w_i (so that $\sum_{i=1}^c K_i = K$). Let the total number of samples in class w_i be n_i (so that $\sum_{i=1}^c n_i = n$). The class-conditional probability can be estimated based on the number of nearest neighbors that belong to class w_i by using the relation:

$$p(\vec{F}^+ | w_i) = \frac{K_i}{n_i V} \quad (2.31)$$

and since the prior probability can also be estimated from the design set using the relation:

$$p(w_i) = \frac{n_i}{n} \quad (2.32)$$

substituting equations 2.31 and 2.32 into equation 2.27 then produces the decision rule for the assignment of \vec{F}^+ to w_i , if

$$K_i \geq K_j \quad \forall i, j = 1, \dots, c; \quad i \neq j \quad (2.33)$$

This leads to a very simple classification procedure: the sample \vec{F} is assigned to the class that receives the largest vote amongst the K nearest neighbors from the design feature set. In practice, the Euclidean distance between \vec{F}^+ and $\{\vec{F}_{n_i, w_i}\}$ is used to find the nearest neighbor(s). Since this rule is derived from the Bayesian rule, it is optimal with respect to conditional risk. There are several ways of breaking ties. Ties can either be broken arbitrarily or by assigning \vec{F}^+ to a specific class out

of all the classes with tying values of K_i that has nearest the mean vector to \vec{F}^+ (with the mean vector calculated over the K_i samples). Another method is to assign \vec{F}^+ to the most compact class². Sometimes, distance-weighted rules are also used for class membership (Webb, 2002). When $K = 1$, i.e., only one nearest neighbor is used, the *Knn* classifier simplifies to the *minimum-distance* classifier (MD).

The learning vector quantization (LVQ) belongs to the category of artificial neural networks (ANNs) based classification schemes (Kohonen, 1990). In general, vector quantization or clustering (in the sense of partitioning a data set, not seeking meaningful groupings of objects) is often performed as a preprocessor for supervised classification. However, it can also be used in a supervised way to classify the data (Randen and Husøy, 1999). There are several ways in which vector quantizers (or self-organizing maps: SOMs) can be used for labelling design data. One approach that uses vector quantizers in a supervised way is to model the whole of the design data with a single vector quantizer rather than each class separately. Each sample from the design data is assigned to the nearest *codebook* vector (\vec{v}), which is then labelled with the class of the majority of the patterns assigned to it. The codebook entries are the prototype feature vectors for a particular class along with the class labels, that is,

$$\vec{v}_i = \left\{ \vec{F}_{n_i, w_i}, w_i \right\} \quad \forall i \in [1, c] \quad (2.34)$$

The LVQ strategy bears a strong resemblance to the *c*-means algorithm (Schalkoff, 1992). The basic algorithm described in the following paragraph, iteratively (in-

²Assign \vec{F}^+ to the one for which the distance to the K_i th member is the smallest. This can be achieved without any extra computation.

dexed by t) uses nearest-neighbor rule on the labelled codebook entries to achieve the classification.

1. Initialize cluster centers, z_1, \dots, z_c , and the cluster labels, w_1, \dots, w_c , for each codebook vector \vec{v}_i .
2. For a design sample \vec{F}_{n_i, w_i} find the *closest* codebook cluster with center z_i .
3. If \vec{F}_{n_i, w_i} is correctly classified, that is, it is labelled as class w_i , then update the nearest codebook cluster center, z_i , according to the rule:

$$z_i(t+1) = z_i(t) + \alpha(t) \left(\vec{F}^+(t+1) - z_i(t) \right).$$

4. If \vec{F}_{n_i, w_i} is not correctly classified then update the nearest codebook cluster center, z_j , using the relation:

$$z_i(t+1) = z_i(t) - \alpha(t) \left(\vec{F}^+(t+1) - z_i(t) \right).$$

Here $\alpha(t)$ is an iteration-dependent parameter used to control the convergence of the algorithm. For stability, $0 < \alpha^t < 1$, and α is constrained to decrease monotonically with t . The adjustment strategy of LVQ is intuitively appealing. Correct classifications lead to a refinement of the codebook cluster center in a direction towards the design sample, where as incorrect classification (or quantization) moves the codebook cluster center in the opposite direction. Because only the nearest neighbor codebook entries are adjusted, this kind of learning is often referred to as *competitive learning*.

The analysis experiments in this study will use two performance metrics, namely, average leave-one-out-error (LOOE, ϵ_l) and modified LOOE (ϵ_m), to quantitatively compare the segmentation results from the above three classifiers. The first metric is calculated by subdividing the total data samples, n , into k test samples, while the remaining $(n - k)$ samples to be used for classifier design and is repeated until each available sample is exactly used once for testing by drawing new sets of test and design samples after replacement. If k is set to 1, then the error estimation becomes *leave-one-out error*, ϵ_l (Vapnik, 1998). The main drawback of this estimate is that the classifier needs to be designed several times (n/k times for the leave-k-out error method). In practice, the upper bound or approximate value of ϵ_l is found by training the classifier on all the n samples (Cawley and Talbot, 2003; Webb, 2002).

The second metric modifies the LOOE metric so that it can also account for the size of feature vector used for classification and is defined as:

$$\epsilon_m = -\frac{\#(\vec{F})\epsilon_l}{100} \quad (2.35)$$

The optimal combination of feature-vector and classifier will be selected from seven feature vectors and three classifiers with maximum ϵ_m value.

2.4.1.5 2-D Real Images

Figure 2.6 shows the three images in first test suite, which will be used for the analysis of the proposed segmentation method. The three-layered soil image, T-1(a), shown in Figure 2.6(a), was obtained by digital scanning an image at 100dpi from Fitzpatrick (1983) and storing it in TIFF image format with 256 grey-levels.

The original digital scan resulted in a image of 117×234 pixels. This image was then converted to a 256×256 pixels image: (1) by reversing and duplicating the entire image once to the right, and (2) by reversing and duplicating a region of 21×21 pixels, one from the right side and the other from the bottom of the image. The final image resulted in a pixel size corresponding to approximately $5mm$ of physical resolution.

The second image in this test suite, T-1(b), is a 800×600 pixel muscle tissues obtained in JPEG image format from Young and Heath (1999). A 512×512 pixels cross-section was then: extracted, converted to monochrome, and stored as a grey-scale TIFF resulting in a pixel size corresponding to approximately $0.5\mu m$. The final image used for analysis is shown in Figure 2.6(b).

The third and final image in this test suite, T-1(c), is a landscape satellite image and is shown in Figure 2.6(c). This image is obtained from the USGS Digital OrthoPhoto Quarter Quad (DOQQ) landcover collection of the Dorchester County in the state of Maryland. The original three band thematic image was converted to a monochrome image by summing the pixel intensities over the three bands and scaling the result to lie in the interval of $[0, 255]$. A 1024×1024 pixels section was then extracted from the processed DOQQ to result in a image with pixel resolution of approximately equal to $1m$ of the physical system.

The main objective of using this test suite was: (a) to analyze the performance of SWA segmentation method as a function of feature descriptors membership in the complete feature vector and, (b) to study impact of different classifiers on the segmentation accuracy. To allow for quantitative assessment of the segmentation

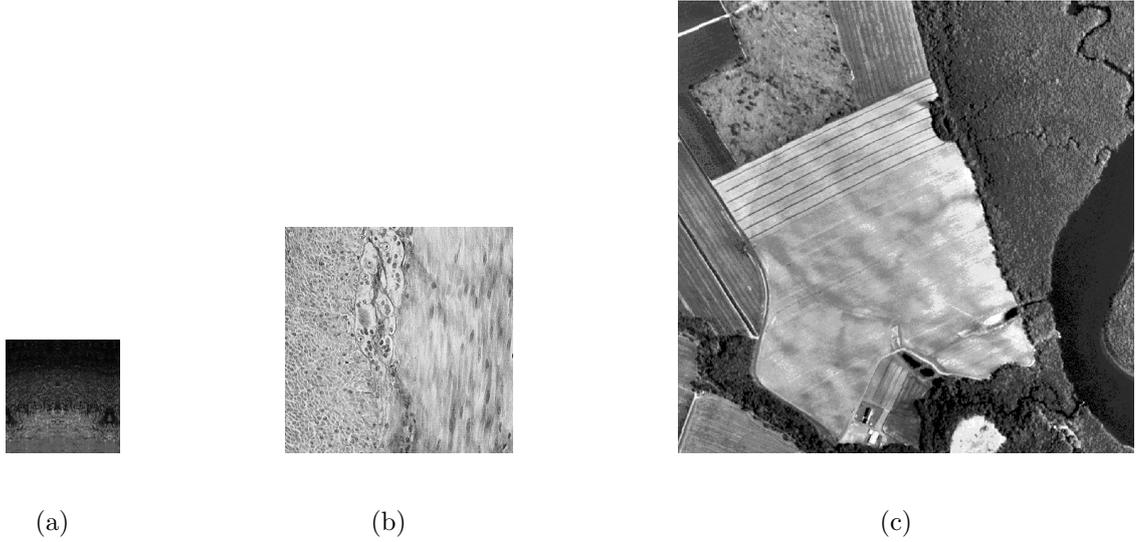


Figure 2.6: *The three images in the first test suite (T-1) used for testing the SWA approach: (a) digital scan of soils image; (b) microscope image of muscle; and (c) digital air-photo of landscape.*

accuracy, the three test images (Figures 2.6(a)–2.6(c)) were manually segmented. These manual hand-guided segmentations (refer to Figure 2.7) will be used as ground truth for quantitative assessment. The classification performance for this test suite will be measured using the average leave-one-out-error, ϵ_l .

2.4.1.6 2-D Synthetic Images

The second test suite used for analyzing the proposed SWA approach has twelve texture mosaics created from three different texture sources; the Brodatz album Brodatz (1966), the MIT Vision Texture database Vision Textures (2004), and the MeasTex Image Texture database MeasTex (2004). Based on the number of textures within each mosaic and the size of these test mosaics, they can be grouped into four

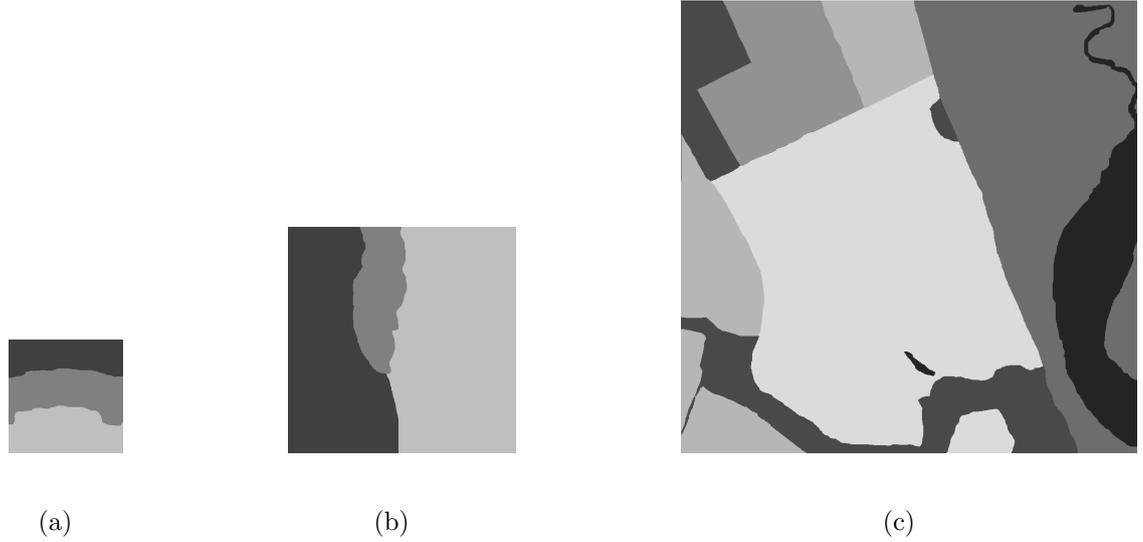


Figure 2.7: *The ground truth images, $G-1$, obtained from expert hand-guided segmentation of $T-1$.*

categories: (1) simple two-texture mosaics of size 256×512 pixels, Figures 2.8(a)-2.8(c); (2) ten-texture mosaics with simple borders of size 256×640 pixels, Figures 2.8(d)–2.8(e); (3) complicated five-texture mosaics of size 512×512 pixels, Figures 2.8(f)–2.8(j); and (4) mosaics of size 512×512 pixels with very complex borders and as many as sixteen textures, Figures 2.8(k)–2.8(l).

Since the *source* texture images, used for creating each of these test mosaics, were captured using different equipment and under different conditions, it renders this test suite extremely complex. All the mosaics in this test suite are grey-scale with 8-bits per sample, when presented to the segmentation method. In order to make the test images nondiscriminable for the mean grey level and variance, the source images and also the test mosaics are globally histogram equalized prior to any analysis.

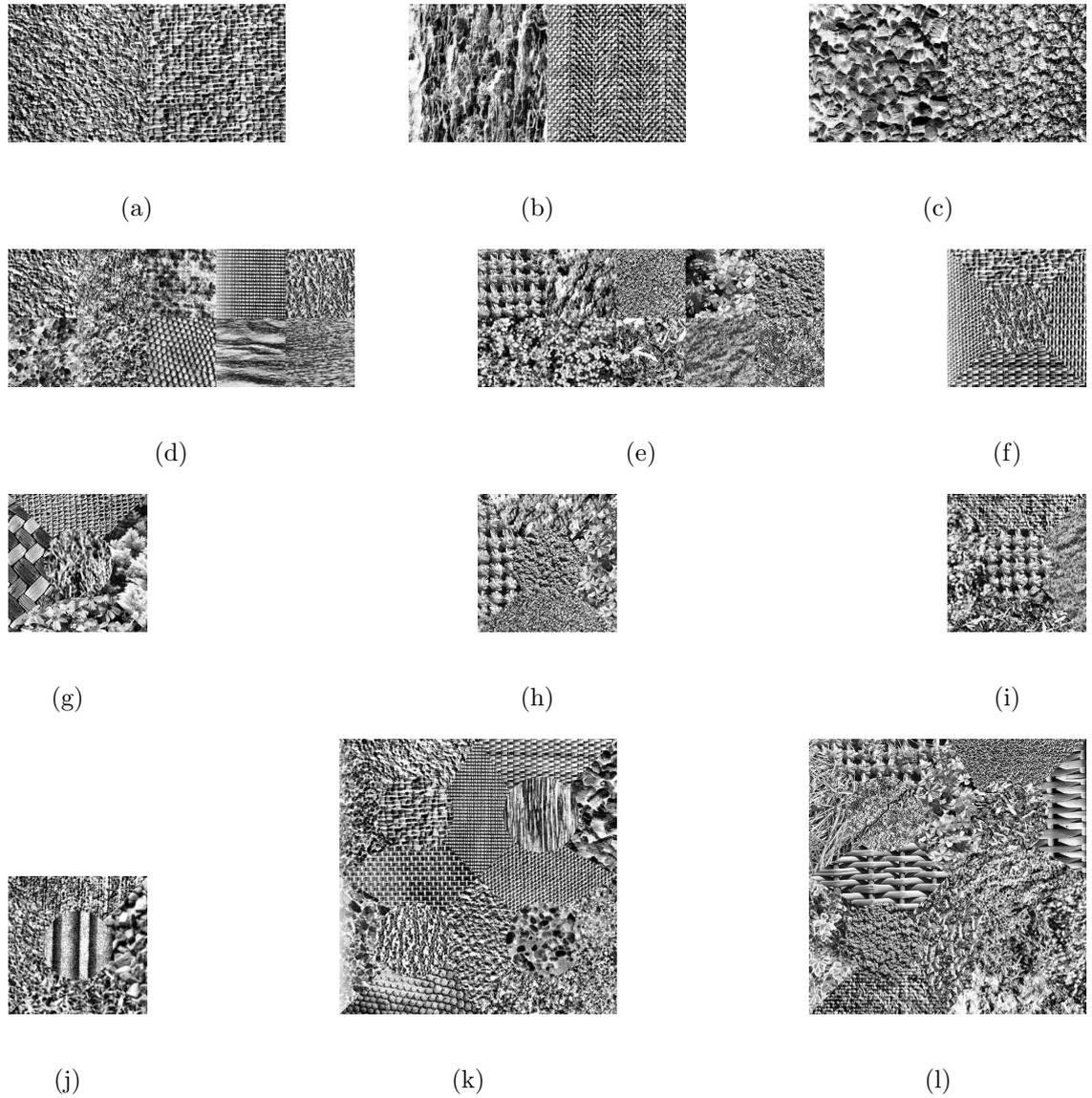


Figure 2.8: *Twelve texture mosaics in T-2 test suite used for analyzing the proposed segmentation approach: (a-c) mosaics with two textures and simple boundaries; (d-e) mosaics with ten textures and simple boundaries; (f-j) mosaics with five textures and a complex boundaries; and (k-l) mosaics with sixteen textures and very complicated boundaries.*

The objectives of using this test suite for analysis is identical to the previous analysis experiment on T-1. Since the texture classes within each of the test mosaic were known *a priori*, the construction of ground truth (G-2) for this test suite was relatively straight forward. The twelve ground truth mosaics corresponding to T-2(a)–T-2(k) are shown in Figure 2.9.

2.4.2 Evaluation

Three widely used biological test suites are selected for evaluating the proposed texture-based segmentation method: (1) fresh CT scans of a female cadaver; (2) simulated brain MR scans; (3) real 3-D MR brain scans. These test-suites vary from each other with respect to: imaging modality; data source (simulated or real); imaging artifacts, like intensity inhomogeneity (INU) and signal-to-noise ratio (SNR) included in the data sets; and spatial resolution including anisotropic. They are summarized in Table 2.2.

The evaluation procedure for each test suite consists of the following steps: (1) sample selection of the different materials; (2) identification of the RES support based on stochastic continuum concept; (3) segmentation of the materials using a feature vector and classification scheme combination identified from analyzing different combinations of feature descriptors and classifiers; (4) error estimation and comparison of the segmentation results to methods in literature, whenever applicable. The nature of experiments performed on these test suites include both qualitative and quantitative assessments. For notational convenience, the three test suites will denoted by T-3, T-4, and T-5 and their corresponding ground truth data, where

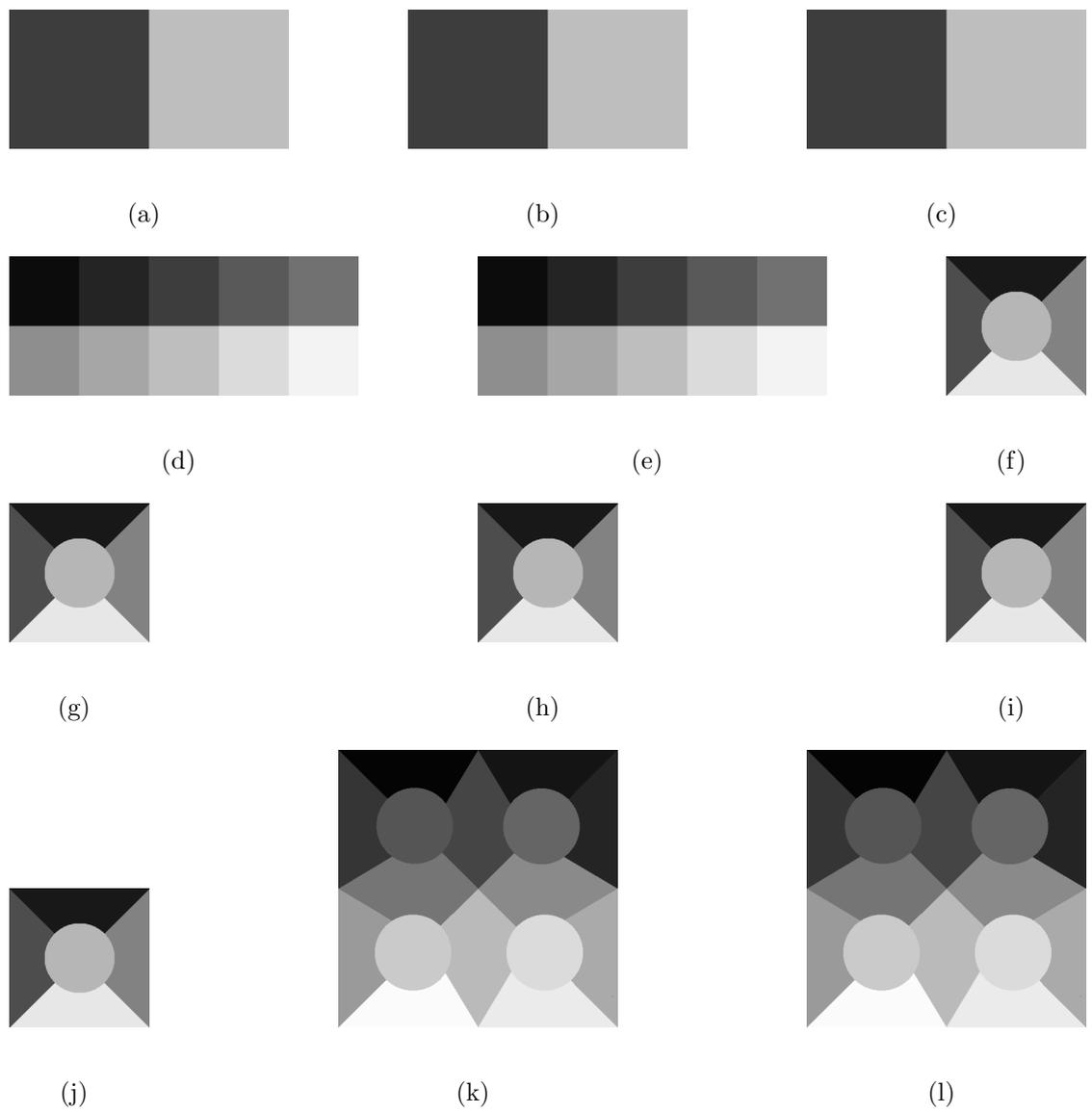


Figure 2.9: *The ground truth mosaics, G_2 , corresponding to T_2 test suite shown in Figure 2.8.*

Table 2.2: Summary of three test suites (T-3, T-4 and T-5) used for evaluating the new segmentation method.

Test Suite No.	T-3	T-4	T-5
Test Suite Name	VWP	MNI-BrainWeb	IBSR
Body Part	Head	Brain	Brain
Modality	CT	MR	MR
Voxel Size	$0.5 \times 0.5 \times 0.5mm^3$	$1 \times 1 \times 1mm^3$	$1 \times 1 \times 3mm^3$
Classes	Soft tissue Skin with Fat Hard tissue Noise Background	CSF Grey-matter White-matter	CSF Grey-matter White-matter
No. of Data sets	1	10	11
Variables	–	SNR, INU	SNR, INU
Ground Truth	–	yes	yes
Performance Index	–	Dice coefficient, s_d	Tanimoto s_t coefficient, s_t
Methods Compared	–	1	5

ever applicable, will be referred as G- i , for all $i \in [3, 5]$.

2.4.2.1 3-D Real CT Head Data

The first test suite, T-3, used for evaluating the segmentation technique is a three-dimensional bio-medical image data from the Visible Woman Project (VWP), performed under the aegis of the National Library of Medicine, National Institute of Health (National Library of Medicine, 2002). The data is public domain and users can acquire it by signing a license agreement with the National Library of Medicine. The VWP provides data sets which consists of CT scans of both fresh and frozen cadaver, MR scans of fresh cadaver, and high resolution digital 24-bit photographs of cross-sections. The test data for qualitative assessment uses the fresh cadaver CT-scans. The fresh CT data was selected for the evaluation purposes because CT scans give a better contrast between soft tissues, hard tissues and air boundaries. Additionally, among the different modalities, the fresh CT data had the smallest slice spacing producing nearly isotropic voxels.

The fresh CT data set is made of axial scans taken over the entire body while the cadaver was fresh and warm. These scans were acquired using GE Genesis System (General Electric, Milwaukee, WI). The CT acquired the axial scans with varying in-slice resolution and a uniform axial slice spacing of $1mm$. The data is stored one slice per file and the files are available in UNIX compressed format. A total of 1734 slices were available under the fresh CT data set using approximately 480 megabytes of disk storage (compressed). Uncompressed, each slice was a 16-bit image of 512×512 pixels with a 3416-bytes of header information. The format of the

header for General Electric Genesis is described in the Medical Image Format FAQ Clunie (2002). The image header contains, among other things, the table position and Field of View (FOV). Such header information is vital when working with CT data, because various parameters (for instance, the spacing between the slices and the pixel size) vary throughout the data set.

The fresh CT slices from the VWP are named $c_vfxxxx.fre$, where $xxxx$ represents the slicing location expressed in millimeters in the direction of head-to-toe. The data set was broken into seven sections, based on at least one of three following criteria: (a) change in FOV; (b) change in slice spacing; and (c) gap in data acquisition. Sections are summarized in Table 2.3.

Table 2.3: *Fresh CT slices from VWP categorized into different sections.*

Section	Slice Range	No. of Slices	FOV	Pixel Size (mm)	Aspect Ratio	Anatomical Part
0	1001–1209	209	250	0.48828	2.0480	Head
1	1210–1227	18	370	0.72266	1.3838	Chin
2	1228–1249	22	440	0.85937	1.1636	Neck
3	1250–2106	857	480	0.93750	1.0677	Torso
4	2107–2110	4	370	0.72266	1.3838	Thighs-1
5	2111–2117	7	480	0.93750	1.0677	Thighs-2
6	2118–2734	617	370	0.72266	1.3838	Lower Limbs

The objective of using this test data is to qualitatively evaluate the performance of segmenting the CT data from VWP into five different materials (classes): (1) soft tissues, (2) outer skin with fat, (3) hard tissue, like bones and skull, (4) noise embedded in the data, and (5) background and air boundaries. Section-0, capturing the head portion of the female cadaver will be used in the evaluation of the developed segmentation procedure. The head section of the female cadaver comprises of 209 CT slices at $1mm$ axial spacing and approximately $0.5mm$ in slice resolution. Since no ground truth data is available for comparisons, the segmentation results from this evaluation experiment will be assessed qualitatively by visually inspecting the surfaces and the cut-sections corresponding to each segmented material.

2.4.2.2 3-D Simulated MR Brain Data

The second bio-medical data for evaluating the proposed segmentation method contains, simulated MR brain scans from the Montreal Neurological Institute, BrainWeb database (MNI-BrainWeb). This data set was selected for the evaluation experiments because of its popularity in the bio-medical image processing research (Archibald et al., 2003; Leemput et al., 1999; Marroquin et al., 2002). The MRI simulator (Cocosco et al., 1997; Collins et al., 1998; Kwan et al., 1999, 1996), generates synthetic brain MR volumes that are spatially normalized (coregistered) to the *colin27* brain MRI template (Collins et al., 1994). These simulated brain MR volumes are public domain and can be downloaded in various data formats (like, MINC, raw-byte, and compressed raw-byte) from the MNI-BrainWeb world wide web interface. The interface also allows to select databases of varying: (a) MR

modalities, i.e., T1-weighted, T2-weighted, or PD-weighted; (b) axial slice thickness of $1mm$, $3mm$, $5mm$, $7mm$, and $9mm$; (c) SNR of 0%, 1%, 3%, 5%, 7%, and 9%; and (d) INU percentages of 0, 20, and 40. The simulator generates different SNR levels for the T1 modality using white-matter tissue intensity as reference, Similarly, for the T2 and PD modality it uses CSF tissue as the reference. The different INUs are generated by linearly re-scaling the 20% INU field. Figure 2.10 shows the 20% known T1-weighted bias field at different axial locations.

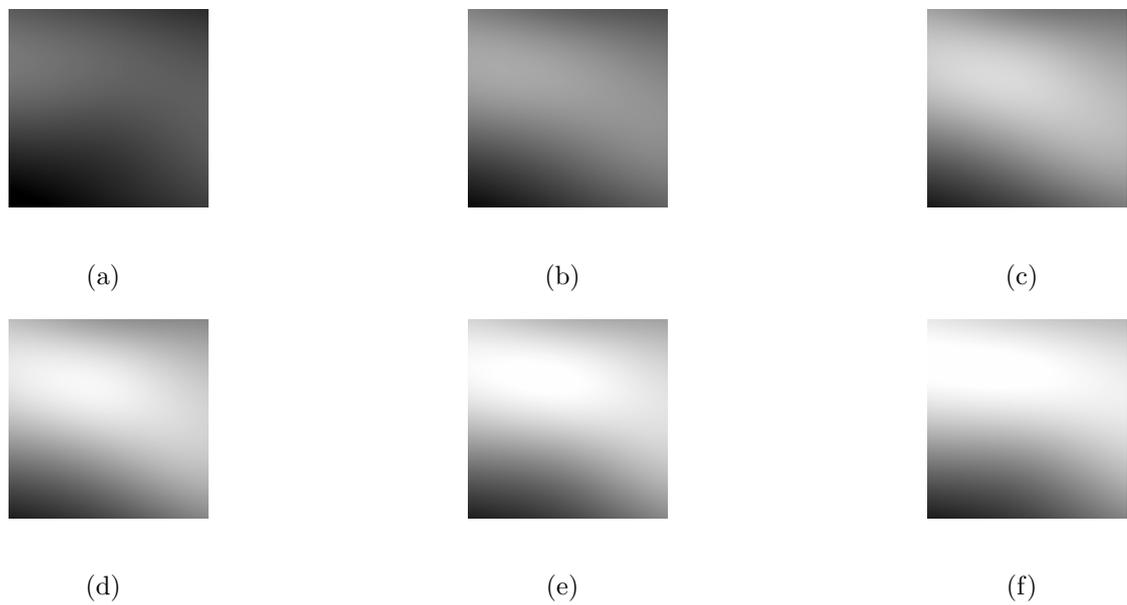


Figure 2.10: *Bias field of 20% intensity inhomogeneity imposed on the T1-weighted MRI scans by the MNI-BrainWeb MRI simulator. (a)-(f): 20% INU field maps at axial locations $z=30mm$, $60mm$, $90mm$, $120mm$, $150mm$, and $180mm$, respectively.*

For evaluation purposes, T1-weighted simulated MRI data with an isotropic voxel size of $1mm^3$ will be used to study the influence of signal-to-noise ratio (SNR) and spatial intensity inhomogeneities (INU) on the performance of the new segmen-

tation technique. The following two cases of bias field corruption will be used in the evaluation experiment: (a) MR images with 1%-9% noise levels and no intensity inhomogeneity, T-4(a)–T-4(e); and (b) MR images with 1%-9% noise levels and 40% intensity inhomogeneity, T-4(f)–T-4(j). These ten data sets, T-4(a)–T-4(j), were downloaded from MNI-BrainWeb in UNIX compressed raw-byte format. The volumetric data is stored in these compressed files as contiguous blocks using approximately 6.8 megabytes of disk storage. Uncompressed, each data set was a 8-bit volume of $181 \times 217 \times 181$ voxels with zero offset. Figure 2.11 shows an axial slice at $z = 30\text{mm}$ for each of the ten simulated data sets in T-4.

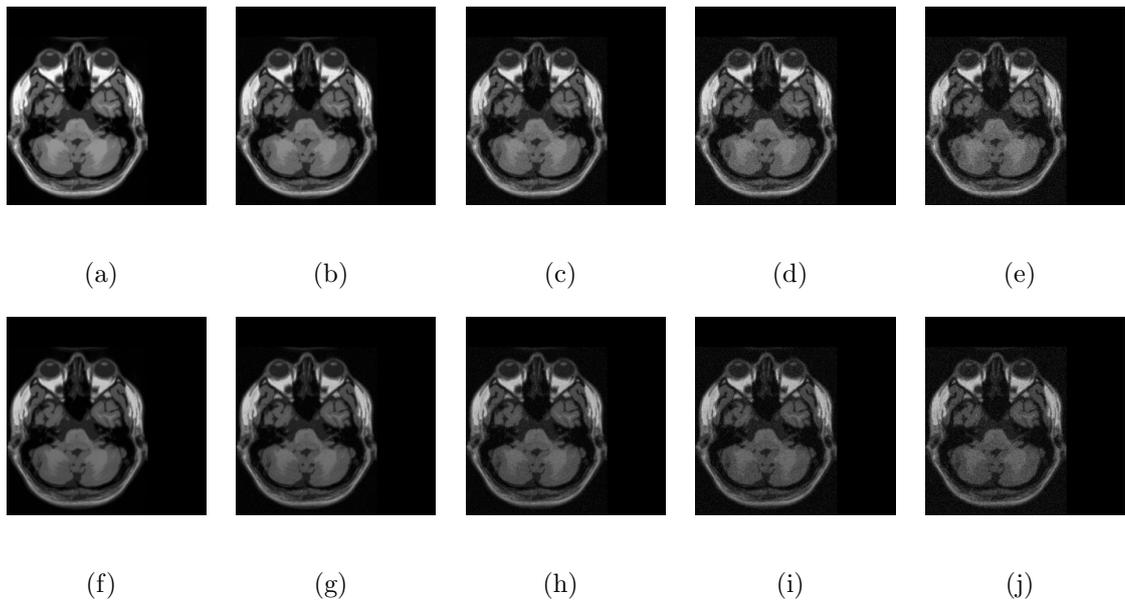


Figure 2.11: *Example axial slice at $z = 30\text{mm}$ for the ten (10) T1-weighted simulated MRI volumes in T-4. (a)-(e): simulated MRI scans with 1%, 3%, 5%, 7%, and 9% signal-to-noise and 0% spatial inhomogeneity. (f)-(j): simulated MRI scans 1%, 3%, 5%, 7%, and 9% noise and 40% spatial inhomogeneity.*

In addition to the simulated brain database, MNI-BrainWeb also provides the anatomical model used by the simulator as a set of three-dimensional *fuzzy* and *discrete* tissue membership volumes, one for each tissue class. The voxel values in the fuzzy volumes reflects the proportion of tissue present in that voxel, in the range of $[0, 1]$. On contrary, the discrete anatomical model provides a class label (integer) at each voxel, representing the tissue which contributes the most to that voxel (0=Background, 1=CSF, 2=Grey Matter, 3=White Matter, 4=Fat, 5=Muscle/Skin, 6=Skin, 7=Skull, 8=Glial Matter, 9=Connective). Figure 2.12 shows the CSF, grey-matter, and white-matter brain tissues from the discrete anatomical model at an axial plane of $z = 30mm$.

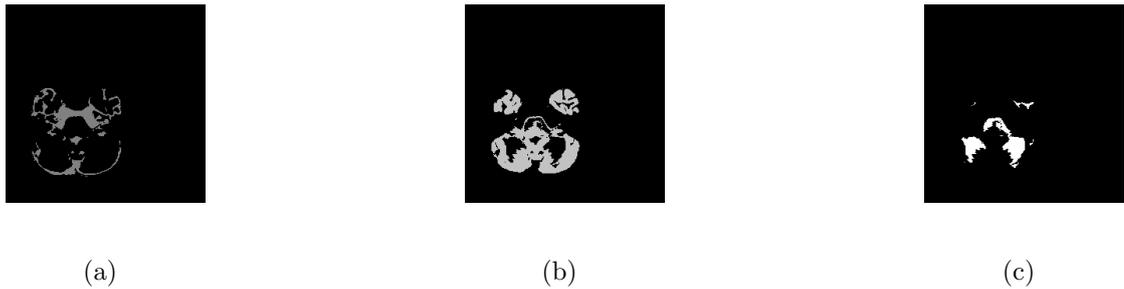


Figure 2.12: Axial ground truth slice at $z = 30mm$ from the discrete anatomical model from MNI-BrainWeb. (a)-(c): shows CSF, grey-matter and white-matter tissue membership, respectively.

The main objective of using this test suite is to quantify the segmentation of the simulated brain MR scans into: CSF, grey-matter, and white-matter tissues, using the developed segmentation method. For quantitative assessment of the segmentation results, the discrete anatomical model from the MNI-BrainWeb will be

used as the ground truth data. The performance index used for measuring the segmentation results will be the *Dice coefficient*, s_d . This similarity index measures the amount of overlap between classification results and ground truth and for a particular class (material) k , and is given by:

$$s_d(k) = 2 \frac{V_{S \cap G}(k)}{V_S(k) + V_G(k)} \quad (2.36)$$

where $V_{S \cap G}(k)$ denotes the number of elements classified by both the proposed method and the ground truth as class k , and $V_S(k)$ and $V_G(k)$ represents the number of elements classified as class k individually by the proposed method and by the ground truth, respectively. This performance metric, first described by Dice (1945) and more recently reintroduced by Zijdenbos et al. (1994) for bio-medical image data, attains a value of 1 if the proposed segmentation method coincides with the ground truth and decreases as the quality of the segmentation deteriorates. A value of 0 indicates no overlap at all. Additionally, this performance index was chosen for this particular data set to allow meaningful comparison with the results reported in literature (Leemput et al., 1999). The Dice coefficient, s_d , is inversely proportional to false negatives (type-I error metric) used in common statistical analysis (Desco et al., 2001; Zeng et al., 1999).

2.4.2.3 3-D Real MR Brain Data

The third, and final, data set that will be used for evaluating the segmentation approach consists of real brain MR volumes provided by the International Brain Segmentation Repository (IBSR), Center for Morphometric Analysis (2004). This

test suite, T-5, contains eleven, coronal, T1-weighted, normal brain-only MR volumes available from IBSR. These MR brain volumes are skull-stripped MR scans and have been categorized into levels of difficulty according to their degree of contrast and presence of spatial inhomogeneities. The brain MR data sets with their corresponding IBSR reference scan number are given in the first column of Table 2.4, in decreasing order of difficulty. In addition to the real MR brain scans, the repository also provides expert hand-guided segmentations of CSF, grey-matter, and white-matter tissues. Both the normal brain scans, T-5(a)–T-5(j), and their corresponding segmented data files, G-5(a)–G-5(j), are 8-bit scaled MR volumes, but vary in sizes. The number of coronal slices in each scan is shown in the third column and fourth column for the normal and segmented data, respectively. Note that the segmented volumes have fewer coronal slices than the T1 image data. The matching between the two volumes can be found using the offsets given in the last column of Table 2.4. Performance results from five automatic segmentation methods are also available from IBSR. Thus, permitting a standardized mechanism for testing the performance of novel segmentation methods against existing ones on the basis of signal-to-noise ratio, intensity inhomogeneities, and shape complexity. The six IBSR methods are: maximum *a posteriori* (MAP) probability, adaptive maximum *a posteriori* probability (AMAP), biased maximum *a posteriori* probability (BMAP), fuzzy k-means (FkMN), maximum likelihood classifier (MLC), and tree-structured k-means (TSKM). These methods are computationally intensive, iterative, unsupervised clustering algorithms. The k-means methods assume initially that voxels can belong to multiple classes with the proportion of classes adding to one. They

proceed to identify the best class means iteratively and then perform image segmentation using these means. These methods are purely local (no voxel neighborhood is considered). The MAP methods assume that image materials form Markovian random fields characterized by mean class values and a class-based spatial heterogeneity functional computed over an 18 voxel neighborhood. The best mean class values and heterogeneity functional parameters are determined iteratively for a given image and then used to segment the image. A major difference with the SWA is that MAP uses a single set of heterogeneity functional parameters for the entire image (irrespective of materials) whereas SWA uses individual, independent, spatial statistics computed separately on each material in the image.

The objective of evaluating the proposed segmentation method with this test suite is to quantitatively assess the segmentation results of different brain tissues obtained from the new segmentation method. For this test suite, the expert segmented data from IBSR will be used as the ground truth. To allow for comparison of the results from IBSR, *Tanimoto* or *Jaccard* coefficient, s_t will be used as the performance index. This index, also used for measuring the amount of overlap between segmentation results and ground truth, is given by the following relation (Duda et al., 2001; Jain and Dubes, 1988):

$$s_t(k) = \frac{V_{S \cap G}(k)}{V_{S \cup G}(k)}, \quad (2.37)$$

where $V_{S \cap G}(k)$ denotes the number of elements classified correctly as class k by both the segmentation method and the ground truth data. The denominator, $V_{S \cup G}(k)$, represents the number of elements classified as class k either by the proposed method

Table 2.4: *Details of the eleven coronal brain-only T1-weighted MR brain scans from IBSR.*

Test Data No.	IBSR Scan No.	No. of Slices	No. of Segmented Slices	Offset
T-5(a)	5_8	58	55	1
T-5(b)	4_8	59	56	2
T-5(c)	6_10	61	57	0
T-5(d)	15_3	58	57	0
T-5(e)	8_4	61	65	2
T-5(f)	112_2	61	58	1
T-5(g)	202_3	61	57	3
T-5(h)	12_3	61	58	0
T-5(i)	1_24	63	56	-2
T-5(j)	205_3	61	58	2
T-5(k)	11_3	61	55	1

or the ground truth data. This performance metric also attains a value of 1 if the proposed segmentation method coincides with the ground truth and decreases as the quality of the segmentation deteriorates. A value of 0 indicates no overlap at all. It should be noted that for a given segmentation method and a ground truth, $s_t(k) \leq s_d(k)$, although both these real-valued indices vary in the range of $[0, 1]$. The Tanimoto coefficient, s_t , is also closely related to the type-I error metric used in common statistical analysis. Except for the type of test suite data and the performance metric, this experiment is identical to previous experiment involving test-suite T-4.

2.5 Results and Discussions

In this section, the results of research described in this chapter are presented. First, the new texture-based supervised segmentation technique, referred as SWA, is presented in Section §2.5.1 along with the necessary mathematical framework for: (a) calculating the moment-based feature descriptors and (b) identifying the representative elementary scale. This section also elaborates the results of analyzing the SWA method on two different data sets using different combinations of feature descriptors and classification schemes. The results of the evaluation experiments on three bio-medical data sets are discussed in Section §2.5.2.

2.5.1 Development

The segmentation method (referred to as SWA in the following) was developed and analyzed in accordance with the 4-step algorithm presented in Section §2.4.1. The

flow diagram for segmenting heterogeneous materials in multi-dimensional image data is shown in Figure 2.13. As discussed previously, the four main steps of the SWA method are: (1) pre-process and select samples for each material in the image data; (2) compute spatial-statistics (\vec{M}) as a function of scale using Haar wavelet coefficients; (3) identify the RES support using stochastic continuum concept and extract material signatures (\vec{M} , \vec{H} , and, \vec{Z}) at RES; (4) classify the input image at RES support and assess the segmentation results.

The developed SWA method was implemented in MATLAB using its *Signal and Image Processing* toolbox and *WaveLab* wavelet toolbox, and applied to two test suites: (1) three real texture predominant images (test suite, T-1), and (2) twelve texture mosaics from different sources (test suite, T-2). These test suites are used to quantitatively analyze the effects of seven combinations of different feature descriptors (refer to Figure 2.5) and the performance of three different classifiers: (a) Gaussian maximum likelihood classifier with Mahalanobis distance metric (GM), minimum-distance classifier (MD), and learning vector quantization (LVQ). The dimensionality of the feature space corresponding to each combination is dependent on the RES support and varies from in the range of $[2, 3j + 7]$, with $\vec{F3}$ and $\vec{F7}$ being the smallest and largest dimensional feature space

The objective of both these analysis experiments is to identify the best combination of feature descriptors and classifier for segmenting heterogeneous bio-image data. The steps used for achieving this objective are based on the concepts of the developed SWA method: (1) selection of sample materials corresponding to each material within the data, (2) estimation of different multi-scale feature descriptors,

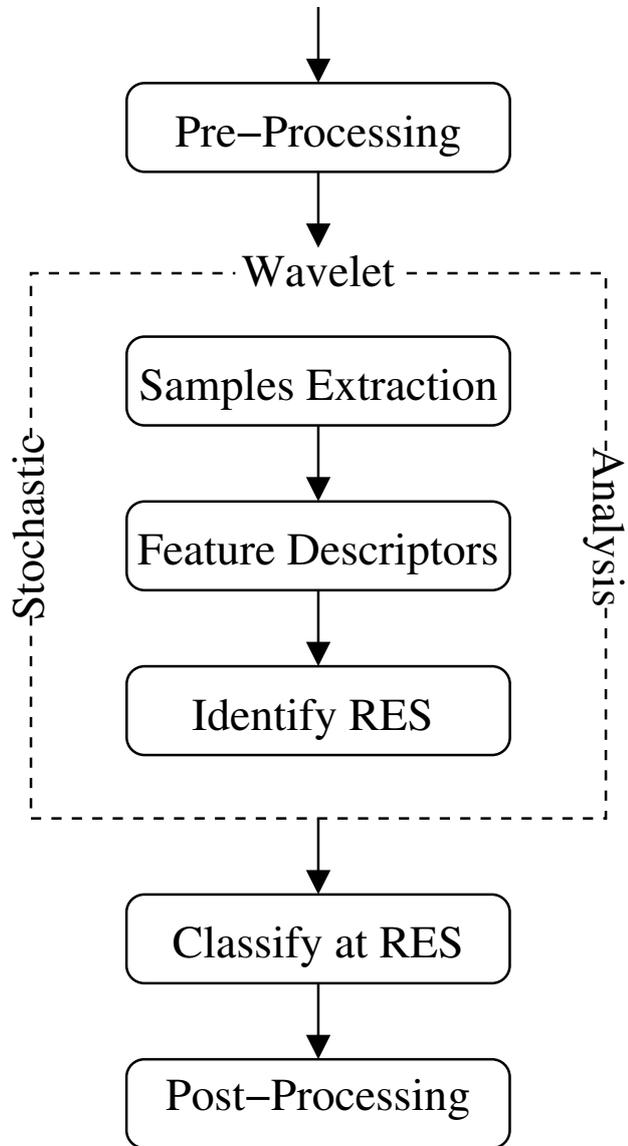


Figure 2.13: *Flow diagram showing the various components of the proposed texture segmentation method.*

Table 2.5: *Combination of feature vectors and their corresponding dimensionality at scale j*

Feature Vector	Features	Feature Space Size
$\vec{F1}$	$\{\vec{M}\}$	4
$\vec{F2}$	$\{\vec{H}\}$	$3j + 1$
$\vec{F3}$	$\{\vec{Z}\}$	2
$\vec{F4}$	$\{\vec{M}, \vec{Z}\}$	6
$\vec{F5}$	$\{\vec{H}, \vec{M}\}$	$3j + 5$
$\vec{F6}$	$\{\vec{H}, \vec{Z}\}$	$3j + 3$
$\vec{F7}$	$\{\vec{H}, \vec{M}, \vec{Z}\}$	$3j + 7$

namely, four central moments (\vec{M}), entropy (\vec{H}), and directional correlation lengths (\vec{Z}), (3) identification of RES from the statistics of the feature descriptors, and (4) error assessment using leave-one-out error (LOOE, ϵ_l) and modified LOOE index (ϵ_m). Additionally, the segmentation results of the test suite T-2 (twelve texture mosaics) will be compared to the best results reported in literature (Randen and Husøy, 1999) based on 40-dimensional entropy feature vector, RH16d, computed from a 16-tap FIR filter with wavelet packet decomposition up to depth 5. The following sub-sections describes the developmental results and the segmentation results obtained from analyzing test suites, T-1 and T-2.

2.5.1.1 Feature Descriptors

The spatial-statistics (\vec{M}) consisting of the first-order four central moments: mean, $\hat{\mu}$; variance, $\hat{\nu}$; skewness, $\hat{\chi}$; and kurtosis, $\hat{\kappa}$; are computed from Haar wavelet coefficients as follows.

The first central moment, i.e., mean ($\hat{\mu}$), for a d -dimensional tile can be, given by equation 2.17, can be rewritten in terms of mean value estimated over tiles from lower scales,

$$\hat{\mu}(\tau_{j,i}) = \frac{\sum_{k=1}^{2^d} \hat{\mu}(\tau_{j-1,k})}{2^d} = \overline{\hat{\mu}}(\tau_{j-1,i}) \quad (2.38)$$

where, *overbar* denotes the average taken over all the tiles from scale $j - 1$ that lie within $\tau_{j,i}$. Similarly, the multi-scale variance, skewness, and kurtosis for τ_2 can be expressed, using the following relations:

$$\begin{aligned} \hat{\nu}(\tau_{j,i}) &= \frac{\sum_{k=1}^{2^d} \hat{\nu}(\tau_{j-1,k})}{2^d} + \frac{\sum_{k=1}^{2^d} [\hat{\mu}(\tau_{j-1,k}) - \hat{\mu}(\tau_{j,i})]^2}{2^d} \\ &= \overline{\hat{\nu}}(\tau_{j-1,i}) + \nu(\tau_{j,i}) \end{aligned} \quad (2.39)$$

$$\begin{aligned} \hat{\alpha}(\tau_{j,i}) &= \frac{\sum_{k=1}^{2^d} \hat{\alpha}(\tau_{j-1,k})}{2^d} + \frac{\sum_{k=1}^{2^d} [\hat{\mu}(\tau_{j-1,k}) - \hat{\mu}(\tau_{j,i})]^3}{2^d} \\ &+ 3(2^d) \sum_{k=1}^{2^d} \left[\left\{ \frac{\sum_{x_1} \dots \sum_{x_d} [\{\tau_{j-1,k}\} - \hat{\mu}(\tau_{j-1,k})]^2}{2^d} \right\} \{ [\hat{\mu}(\tau_{j-1,k}) - \hat{\mu}(\tau_{j,i})] \} \right] \\ &= \overline{\hat{\alpha}}(\tau_{j-1,i}) + \alpha(\tau_{j,i}) + 3(2^d) \left[\{ \hat{\nu}(\tau_{j-1,k}) \} \times \{ \Delta \hat{\mu}_{j-1,k} \}' \right] \end{aligned} \quad (2.40)$$

$$\begin{aligned}
\widehat{\beta}(\tau_{j,i}) &= \frac{\sum_{k=1}^{2^d} \widehat{\beta}(\tau_{j-1,k})}{2^d} + \frac{\sum_{k=1}^{2^d} [\widehat{\mu}(\tau_{j-1,k}) - \widehat{\mu}(\tau_{j,i})]^4}{2^d} \\
&+ 6(2^d) \sum_{k=1}^{2^d} \left[\left\{ \frac{\sum_{x_1} \dots \sum_{x_d} [\{\tau_{j-1,k}\} - \widehat{\mu}(\tau_{j-1,k})]^2}{2^d} \right\} \{[\widehat{\mu}(\tau_{j-1,k}) - \widehat{\mu}(\tau_{j,i})]^2\} \right] \\
&+ \sum_{k=1}^{2^d} \left[\left\{ \frac{\sum_{x_1} \dots \sum_{x_d} [\{\tau_{j-1,k}\} - \widehat{\mu}(\tau_{j-1,k})]^3}{2^d} \right\} \{[\widehat{\mu}(\tau_{j-1,k}) - \widehat{\mu}(\tau_{j,i})]\} \right] \\
&= \widetilde{\beta}(\tau_{j-1,i}) + \beta(\tau_{j,i}) + 6(2^d) \left[\{\widehat{\nu}(\tau_{j-1,k})\} \times \{(\Delta \widehat{\mu}_{j-1,k})^2\}' \right] \\
&+ \left[\{\widehat{\nu}(\tau_{j-1,k})\} \times \{\Delta \widehat{\alpha}_{j-1,k}\}' \right] \tag{2.41}
\end{aligned}$$

Since the image elements (pixels in 2-D and voxels in 3-D images) in a tile at scale j are related to their corresponding Haar coefficients by the following relation,

$$\{\tau_{j,i}\}' = \mathcal{H}_{2^d \times 2^d}^{-1} \left\{ \begin{array}{c} L_{j,i} \\ D_{j,i}^1 \\ D_{j,i}^2 \\ \vdots \\ D_{j,i}^{2^d-1} \end{array} \right\} \tag{2.42}$$

where, $\mathcal{H}_{2^d \times 2^d}$ represents the Haar DWT in block matrix notation.

In 2-D, using equation 2.15, the mean statistic of the tile at scale $j - 1$ can be calculated as

$$\begin{aligned}
\widehat{\mu}(\tau_{j-1,i}) &= \widehat{\mu} \left(H_{4 \times 4} \times \{L_{j-1,i}, D_{j-1,i}^1, D_{j-1,i}^2, D_{j-1,i}^3\}' \right) \\
&= \frac{L_1}{(\sqrt{2})^{2 \times 1 \times 1}} \tag{2.43}
\end{aligned}$$

The three indices of the denominator correspond to dimension, scale, and order of the moment. Similarly, variance, skewness, and kurtosis for a 2-D tile $\tau_{j,i}$ using

corresponding wavelet coefficients can be computed with the following relations,

$$\widehat{\nu}(\tau_{j-1,i}) = \frac{(D_{j-1,i}^1)^2 + (D_{j-1,i}^2)^2 + (D_{j-1,i}^3)^2}{(\sqrt{2})^{2 \times 1 \times 2}} \quad (2.44)$$

$$\widehat{\alpha}(\tau_{j-1,i}) = \frac{(D_{j-1,i}^1)(D_{j-1,i}^2)(D_{j-1,i}^3)}{(\sqrt{2})^{2 \times 1 \times 3}} \quad (2.45)$$

$$\begin{aligned} \widehat{\beta}(\tau_{j-1,i}) = & \frac{(D_{j-1,i}^1)^4 + (D_{j-1,i}^2)^4 + (D_{j-1,i}^3)^4}{(\sqrt{2})^{2 \times 1 \times 4}} + \frac{6(D_{j-1,i}^1)^2(D_{j-1,i}^2)^2}{(\sqrt{2})^{2 \times 1 \times 4}} + \\ & \frac{6(D_{j-1,i}^2)^2(D_{j-1,i}^3)^2}{(\sqrt{2})^{2 \times 1 \times 4}} + \frac{6(D_{j-1,i}^1)^2(D_{j-1,i}^3)^2}{(\sqrt{2})^{2 \times 1 \times 4}} \end{aligned} \quad (2.46)$$

The parameters, $\nu(\tau_{j,i})$, $\alpha(\tau_{j,i})$, and $\beta(\tau_{j,i})$ in equations 2.38–2.41 are computed from the wavelet coefficients of the sub-band filter outputs at scale j . For the 2-D example, these are given by,

$$\nu(\tau_{j,i}) = \frac{(D_{j,i}^1)^2 + (D_{j,i}^2)^2 + (D_{j,i}^3)^2}{(\sqrt{2})^{2 \times 1 \times 2}} \quad (2.47)$$

$$\alpha(\tau_{j,i}) = \frac{(D_{j,i}^1)(D_{j,i}^2)(D_{j,i}^3)}{(\sqrt{2})^{2 \times 1 \times 3}} \quad (2.48)$$

$$\begin{aligned} \beta(\tau_{j,i}) = & \frac{(D_{j,i}^1)^4 + (D_{j,i}^2)^4 + (D_{j,i}^3)^4}{(\sqrt{2})^{2 \times 1 \times 4}} + \\ & 6 \frac{[(D_{j,i}^1)^2(D_{j,i}^2)^2 + (D_{j,i}^2)^2(D_{j,i}^3)^2 + (D_{j,i}^1)^2(D_{j,i}^3)^2]}{(\sqrt{2})^{2 \times 1 \times 4}} \end{aligned} \quad (2.49)$$

Substituting equations 2.43–2.49 into 2.38–2.41 will directly estimate the multi-scale first-order central moments in terms of wavelet coefficients.

2.5.1.2 Identifying RES Support and Classification

The RES support is identified using the procedure described in Section §2.4.1.3. Since the process of determining the RES support uses both the stochastic continuum concept and wavelet multi-resolution analysis, the segmentation method is

named *Stochastic-Wavelet Analysis*, SWA. The identification of optimal RES support demands proper judgement. Usually, selecting either a smaller or a larger scale than the optimal RES support should be avoided, because this could lead to spatially varying material signatures. For instance, if the scale selected is larger than the optimum RES support then the material signatures will capture properties of more than one material. On contrary, if the appropriate optimal scale is selected then the statistics will be independent of the sampling location and will be consistent across the entire material.

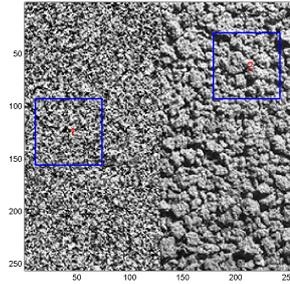
A 2-D mosaic (refer to Figure 2.14(a)) with two synthetic textures is used to describe the RES identification process from the spatial-statistics. First, sufficiently large samples that are representative of the two texture are selected from the mosaic and are shown as boxes overlaid on top of the input image. Next, the spatial-statistics, \vec{M} , are computed as a function of scale for both the samples using Haar wavelet coefficients. This is followed by the estimation of the statistics (mean, max-min, and ± 1 -standard deviation) of the four central moments for all the non-overlapping tiles at a particular scale and are plotted as a function of scale (shown in Figure 2.14(b)). In this plot, the first row shows the sample regions over which the statistics of the feature descriptors are computed. The last four rows show the distributions of $\hat{\mu}$, $\hat{\nu}$, $\hat{\chi}$, and $\hat{\kappa}$ for each of the sample material as a function of scale. In all these graphs, the solid line is the mean statistic of the corresponding feature parameter and the shaded zone represents ± 1 -standard deviation interval. The top and bottom dotted lines are the maximum and minimum of the feature descriptor(s) calculated over all the tiles at the given scale, respectively. It is clearly

evident that, the statistical distributions of all the texture descriptors, for both the sample materials, stabilize at scale 4. Hence, the RES support is selected as 4, i.e., the sampling tiles are of size $2^4 \times 2^4$ pixels, for both the textures in the example input image. (Segmentation results presented in Appendix A further supports this process by demonstrating the decrease in SWA classification error with increasing sample size up to the RES scale.)

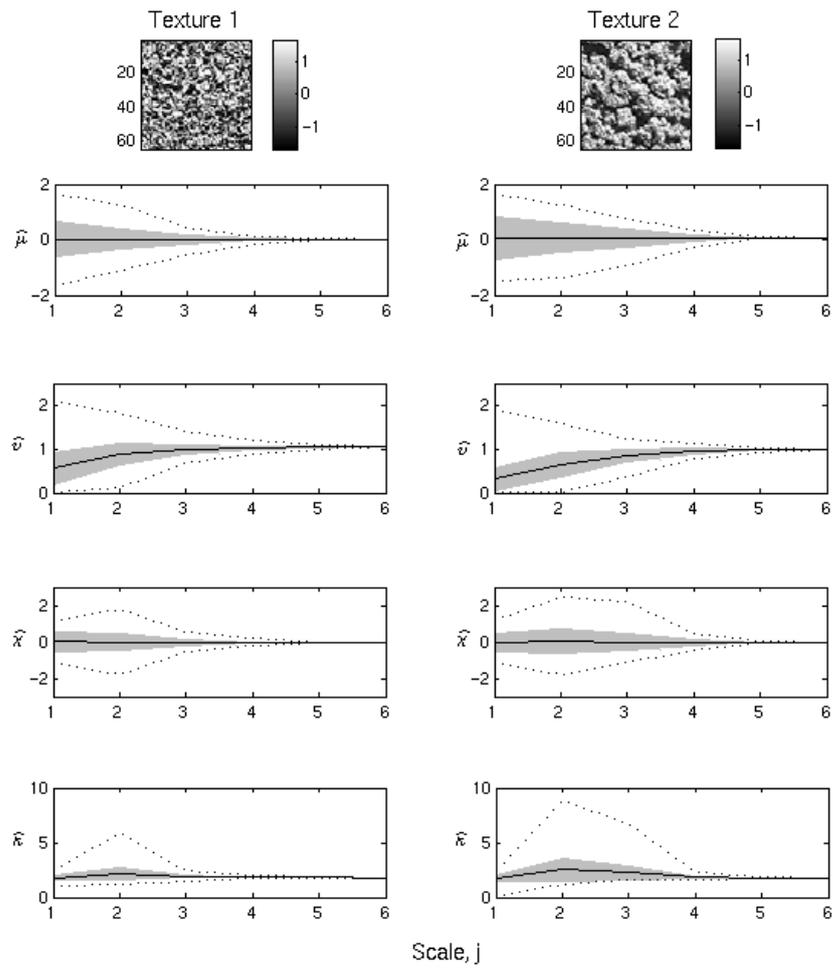
2.5.1.3 2-D Real Images

The developed SWA method was applied to test suite T-1 containing three 2-D images: (1) digital scan of a three layered soil photo; (2) microscope image of three muscles; and (3) satellite (airphoto) of landscape and wetland. Samples of sufficiently large sizes for each material within these images were selected manually after carefully inspecting both the input and the ground truth data. The locations of the samples for different materials within the three images are shown in Figure 2.15. In this figure, the top row shows the sample locations overlaid on top of the test images, whereas the bottom row shows the sample locations overlaid on the ground truth. The latter helps to confirm that the selected samples comes from the corresponding material and not from multiple materials.

The selected samples were used to compute statistics of the pixel intensity moments at successively large scale and the procedure outline in Section §2.5.1.2 was used to identify the RES support for each test image. The spatial-statistics for the three test images in T-1 are shown in Figures 2.16, 2.17, and 2.18, and the RES support was identified as 3, 4, and 4. This corresponds to representative elementary



(a)



(b)

Figure 2.14: *Illustration of RES support identification: (a) Sample 2-D image of size 256×256 with two textures; (b) plot showing the statistics of the first four-central moments as a function of scale.*

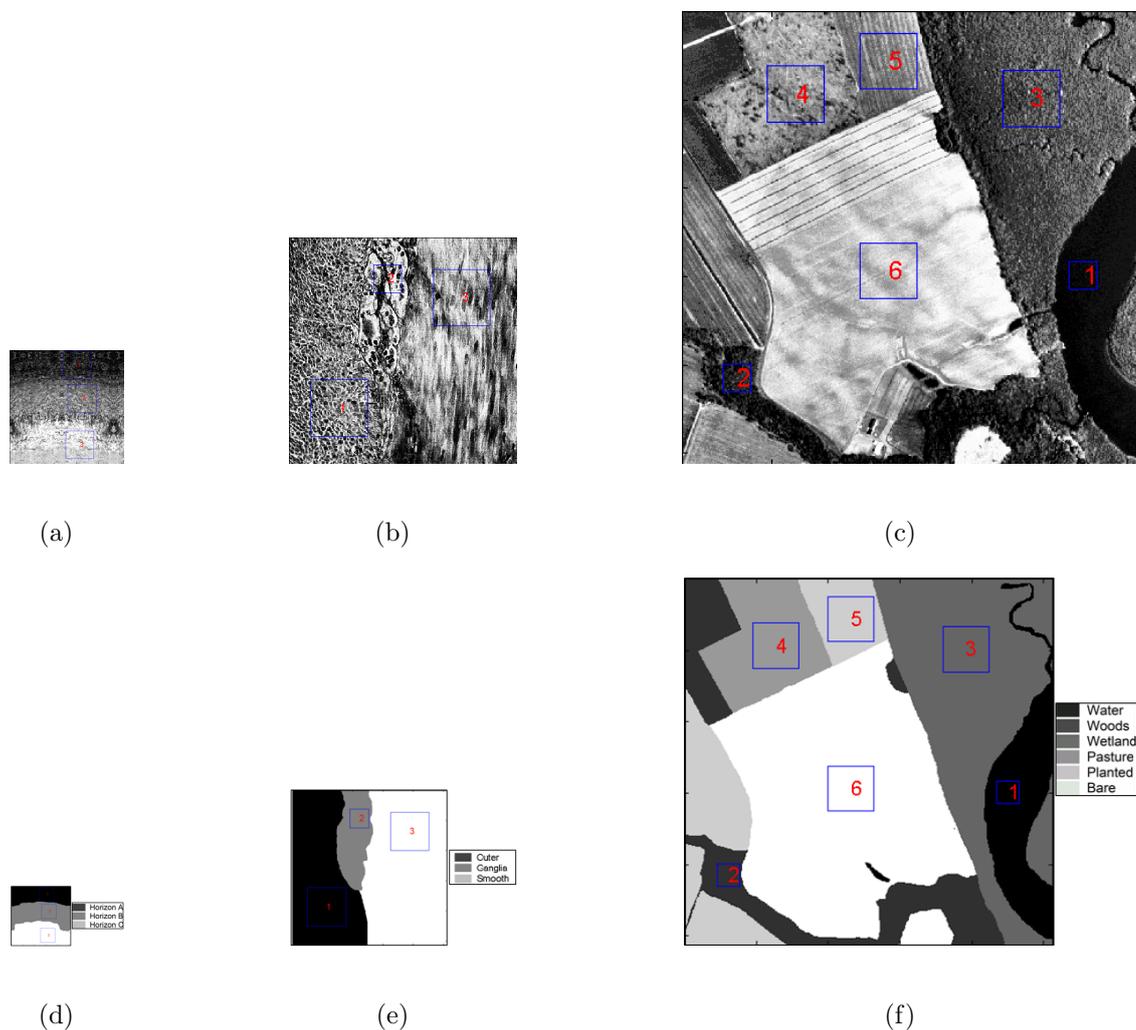


Figure 2.15: *The three images in the first test suite (T-1) used for analysis of the SWA segmentation method. The images in the top (a-c) and the bottom (d-f) are the input and the ground truth images from manual expert segmentation, respectively. The boxes indicate the training sample locations used for estimating the RES support and classifier design.*

area (REA) of size 8×8 pixels, 16×16 pixels, and 16×16 pixels.

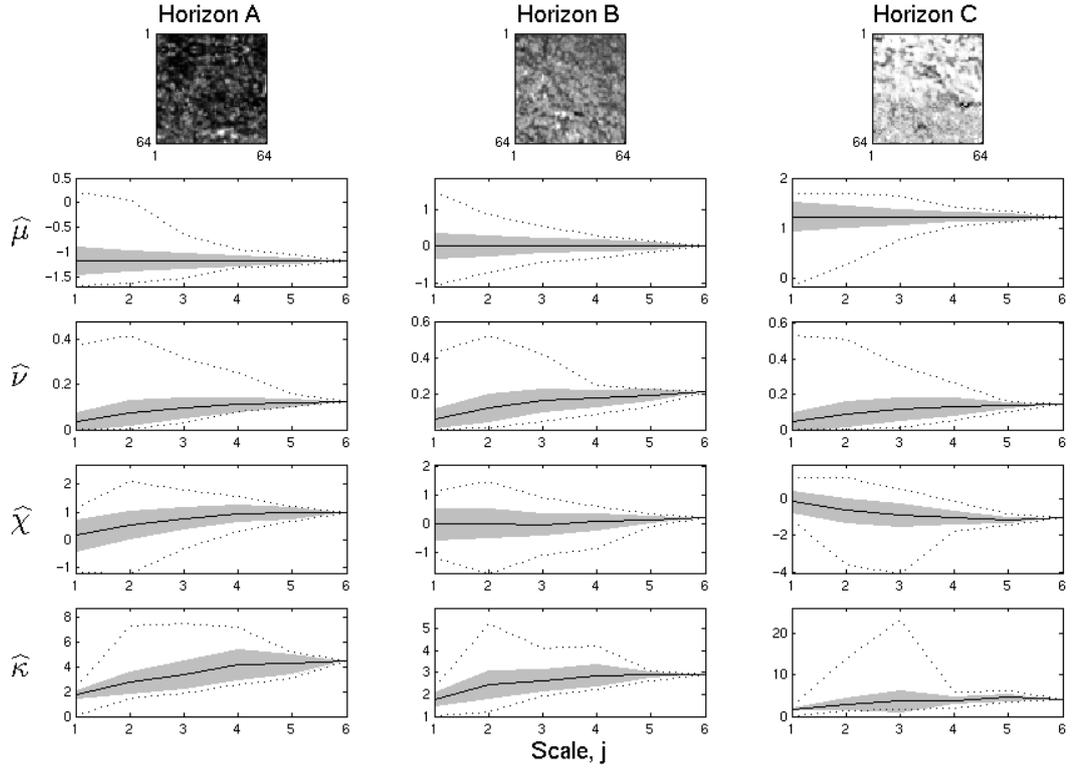


Figure 2.16: *Spatial statistics used for identifying RES support of horizons A, B, and C in the soils image.*

The seven combinations of feature descriptors computed at the identified RES support were used to segment each test image using GM, MD, and LVQ classification methods. The segmentation results were compared with the ground truth images average LOOE (ϵ_l) and modified LOOE (ϵ_m) for soils, muscle, and landscape test images using seven feature combinations and three classifiers are summarized numerically in Tables 2.6, 2.7, and 2.8, respectively.

For the soils image, segmentation results expressed in terms of average LOOE, ϵ_l , vary between 8.3% and 67.4%, and when assessed in terms of modified LOOE, ϵ_m , the segmentation results vary between -0.33 and -5.47. In general, it can be

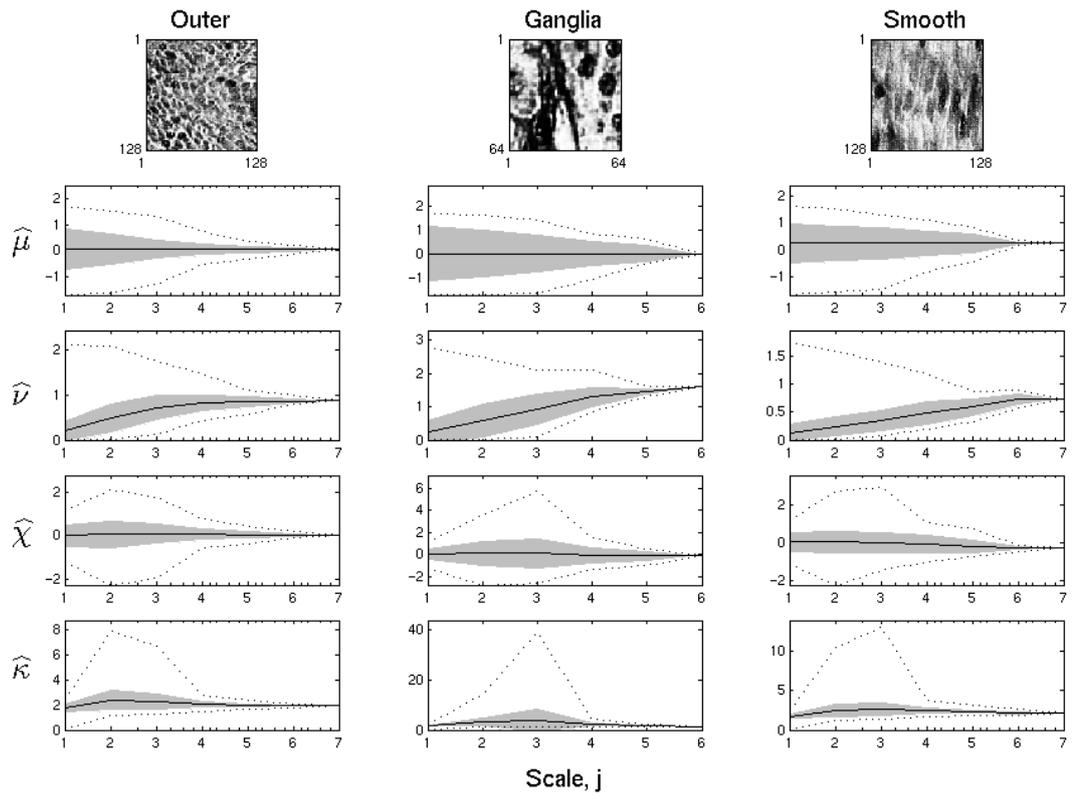


Figure 2.17: *Spatial-statistics used for identifying the RES support of smooth, ganglia, and outer tissues in the muscle image.*

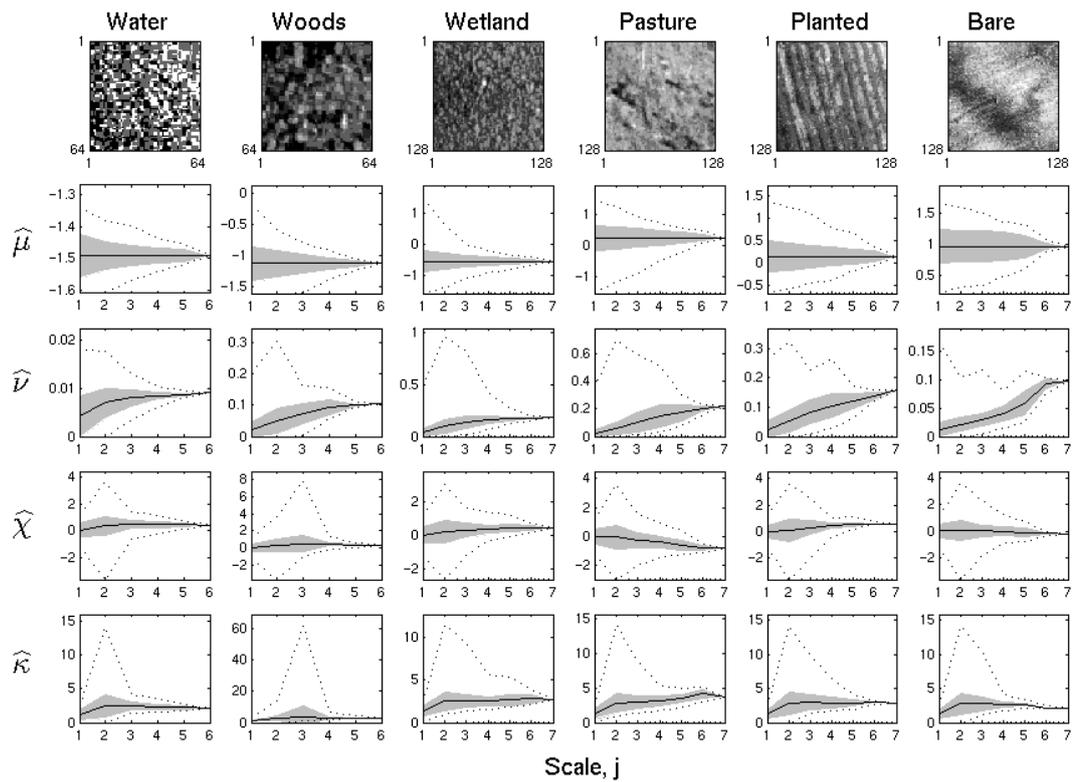


Figure 2.18: *Spatial-statistics used for identifying the RES support of water, woods, wetland, pasture, planted and bare regions in the landscape image.*

observed that \vec{M} produces highly accurate and computationally optimal results. Furthermore, the results also indicate that adding extra feature descriptors does not improve the classification accuracy. Based on ϵ_m metric it can be concluded that $\vec{F1}$ -GM and $\vec{F6}$ -LVQ gives the best and the worst segmentation results.

Table 2.6: Average LOOE, ϵ_l and modified LOOE, ϵ_m expressed in percent for the soil image in Figure 2.6(a) at RES support of 3 using GM, MD, and LVQ classifiers.

Feature Vector	RES	Classification Error ϵ_l (ϵ_m)		
		GM	MD	LVQ
$\vec{F1}$	4	8.30 (-0.33)	8.90 (-0.36)	8.30 (-0.33)
$\vec{F2}$	10	26.80 (-2.68)	38.40 (-3.84)	41.60 (-4.16)
$\vec{F3}$	2	66.20 (-1.32)	67.40 (-1.35)	64.20 (-1.28)
$\vec{F4}$	6	8.40 (-0.50)	9.20 (-0.55)	10.00 (-0.60)
$\vec{F5}$	14	9.70 (-1.36)	10.60 (-1.48)	10.00 (-1.40)
$\vec{F6}$	12	28.90 (-3.47)	39.20 (-4.70)	45.60 (-5.47)
$\vec{F7}$	16	9.60 (-1.54)	10.70 (-1.71)	12.60 (-2.02)

The segmentation results for the muscle image, expressed in terms of ϵ_l , vary between 6.0% and 88.5% and are produced by $\vec{F5}$ feature vector when used with GM and MD classifiers. However, based on ϵ_m , $\vec{F1}$ -GM combination produces the best optimal segmentation results with a value of -0.40. In general, GM classifier performs better than LVQ classifier which is better than MD classifier.

Table 2.7: Average LOOE, ϵ_l and modified LOOE, ϵ_m for the muscle image in Figure 2.6(b) at RES support of 4 using GM, MD, and LVQ classifiers.

Feature Vector	RES	Classification Error ϵ_l (ϵ_m)		
		GM	MD	LVQ
$\vec{F1}$	4	9.90 (-0.40)	23.70 (-0.95)	27.00 (-1.08)
$\vec{F2}$	13	16.00 (-2.08)	17.10 (-2.22)	20.30 (-2.64)
$\vec{F3}$	2	49.20 (-0.98)	30.80 (-0.62)	35.20 (-0.70)
$\vec{F4}$	6	15.10 (-0.91)	11.80 (-0.71)	18.80 (-1.13)
$\vec{F5}$	17	6.00 (-1.02)	88.50 (-15.05)	14.10 (-2.40)
$\vec{F6}$	15	15.70 (-2.36)	16.60 (-2.49)	19.80 (-2.97)
$\vec{F7}$	19	6.30 (-1.20)	57.30 (-10.89)	13.00 (-2.47)

Similar to the previous case, $\vec{F5}$ feature vector with GM and MD classifiers produces the best and the worst segmentation results with 5.9% and 97.9% of average LOOE for the landscape image. Unlike the previous case, MD classifier performs better than LVQ classifier, except when used with $\vec{F1}$. Even in this case, $\vec{F1}$ -GM gives the best optimal and $\vec{F5}$ -MD gives the worst optimal segmentation results in terms of ϵ_m .

Overall, the SWA segmentation results for the three test images in this test suite produced excellent classification accuracy with as few as four texture descriptors (the first four central moments, \vec{M}) using the GM classifier. The increase in

Table 2.8: Average LOOE, ϵ_l and modified LOOE, ϵ_m for the landscape image in

Figure 2.6(c) at RES support of 4 using GM, MD, and LVQ classifiers.

Feature Vector	RES	Classification Error ϵ_l (ϵ_m)		
		GM	MD	LVQ
$\vec{F1}$	4	8.00 (-0.32)	9.50 (-0.38)	32.90 (-1.32)
$\vec{F2}$	13	17.20 (-2.24)	19.00 (-2.47)	49.10 (-6.38)
$\vec{F3}$	2	40.80 (-0.82)	34.40 (-0.69)	61.80 (-1.24)
$\vec{F4}$	6	7.90 (-0.47)	9.50 (-0.57)	38.50 (-2.31)
$\vec{F5}$	17	5.90 (-1.00)	97.90 (-16.64)	33.70 (-5.73)
$\vec{F6}$	15	19.80 (-2.97)	19.60 (-2.94)	49.70 (-7.46)
$\vec{F7}$	19	6.30 (-1.20)	39.90 (-7.58)	38.30 (-7.28)

the feature descriptors did not have much impact on the classification accuracy.

Hence, the moment-based feature descriptors $\vec{F1}$ with GM classifier is concluded as the *optimal feature vector and classifier combination*.

2.5.1.4 2-D Synthetic Images

The developed SWA segmentation method was applied to second test suite consisting four texture mosaics categories, viz. mosaics with two textures and simple boundaries; mosaics with ten textures and simple boundaries; mosaics with five textures and complicated boundaries; and mosaics with sixteen textures and com-

plicated boundaries. For the sample selection step, the complete 256×256 pixels source image for each texture material within each mosaic was used for estimating sample statistics. (Note that each of the texture mosaic in second data set were created from individual 256×256 *source* textures corresponding to each texture material present in the mosaic.) For example, the five training images corresponding to textures in T-2(f) (refer to Figure 2.8(f)) are shown in Figure 2.19.

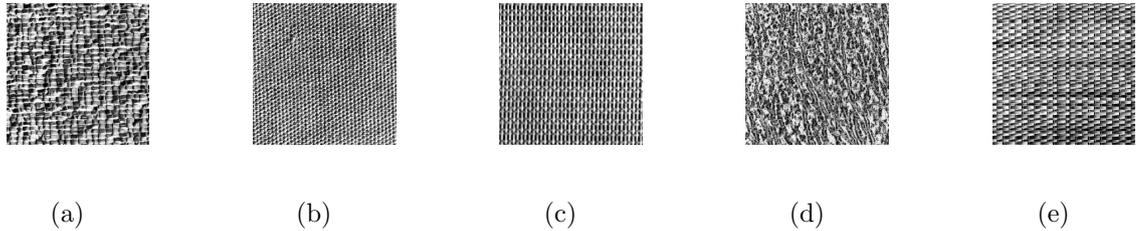


Figure 2.19: *The source images of the textures present in the mosaic T-2(f). These source images are also used as for identifying RES support as well as classifier design.*

The RES support for each of these sample materials was identified, using the procedure described in Section §2.5.1.2, as the scale at which spatial-statistics of the feature descriptors stabilized. The RES support for the twelve texture mosaics, presented in the top row of Tables 2.9, 2.9, 2.9, and 2.9, ranges from scales 3 to 5 that corresponds to representative elementary area of size 8×8 pixels and 32×32 pixels. Among the four categories, larger representative elementary areas were needed for the last two categories, i.e., mosaics with 5 textures, T-2(f)–T-2(j), and mosaics with 16 textures, T-2(k)–T-2(l), suggesting the need for larger sampling areas for the corresponding textures to become homogeneous materials.

The signatures at RES support were used to segment the test mosaics using

three classifiers (GM, MD, and LVQ) and seven feature combinations ($\vec{F1}$ through $\vec{F7}$). The segmentation results obtained from SWA method for were compared to ground truth data for computing the average LOOE (ϵ_l) and the modified LOOE (ϵ_m) metrics and are numerically summarized in Tables 2.9–2.9 along with the best classification result obtained by Randen and Husøy (1999) using RH16d feature vector.

The overall range of ϵ_l error metric for this test suite varied in the range of 0.2% (for T-2(b) using $\vec{F3}$ -MD) and 90.4% (for T-2(i) using $\vec{F1}$ -MD). In terms of average LOOE on individual mosaics, segmentation results from GM and MD classifiers achieve better performance for 9 (75%) and 3 (25%) mosaics, respectively and the ϵ_l varies between 0.2% (for T-2(b) using $\vec{F3}$ -MD) and 37.5% (for T-2(i) using $\vec{F6}$ -MD). In terms of feature vectors, $\vec{F5}$ and $\vec{F7}$ contribute for 5 and 4 mosaics, respectively, with the remaining three mosaics shared by $\vec{F3}$ and $\vec{F6}$. In general, the results based on ϵ_l indicate that GM classifier and entropy-based feature descriptors (it is a common descriptor in $\vec{F5}$, $\vec{F6}$, and $\vec{F7}$) produces the best segmentation results.

Similarly, the overall range of ϵ_m error metric varied between 0 (for T-2(b) using $\vec{F3}$ -MD) and -12.25% (for T-2(l) using $\vec{F7}$ -LVQ). In terms of maximum ϵ_m for each individual mosaics, GM classifier performed better than MD and MD better LVQ on 7, 3, and 2 mosaics, respectively, with the values ranging between 0 (for T-2(b) using $\vec{F3}$ -MD) and -1.64 (for T-2(l) using $\vec{F7}$ -MD). In terms of feature vectors, $\vec{F3}$, i.e., directional ACF-based features, accounted for 75% of the total test suite and the remaining 25% was shared between $\vec{F5}$ (2 mosaics, T-2(b) and T-2(j)) and

Table 2.9: SWA segmentation results for mosaics $T\text{-}2(a)\text{-}T\text{-}2(c)$ expressed in average LOOE (modified LOOE).

Feature Vector	T-2(a), RES = 3				T-2(b), RES = 4				T-2(c), RES = 5			
	Size	GM	MD	LVQ	Size	GM	MD	LVQ	Size	GM	MD	LVQ
\vec{F}_1	4	43.60 (-1.74)	45.30 (-1.81)	48.20 (-1.93)	4	7.20 (-0.29)	44.30 (-1.77)	17.20 (-0.69)	4	8.40 (-0.34)	45.30 (-1.81)	30.10 (-1.20)
\vec{F}_2	10	5.70 (-0.57)	4.30 (-0.43)	16.00 (-1.60)	13	3.10 (-0.40)	3.90 (-0.51)	3.90 (-0.51)	16	0.80 (-0.13)	1.40 (-0.22)	3.50 (-0.56)
\vec{F}_3	2	10.90 (-0.22)	7.00 (-0.14)	25.40 (-0.51)	2	0.60 (-0.01)	0.20 (-0.00)	5.90 (-0.12)	2	31.10 (-0.62)	12.70 (-0.25)	39.80 (-0.80)
\vec{F}_4	6	7.20 (-0.43)	23.20 (-1.39)	23.80 (-1.43)	6	5.70 (-0.34)	15.80 (-0.95)	5.70 (-0.34)	6	3.30 (-0.20)	34.60 (-2.08)	27.10 (-1.63)
\vec{F}_5	14	6.40 (-0.90)	6.60 (-0.92)	16.20 (-2.27)	17	4.50 (-0.77)	5.70 (-0.97)	5.10 (-0.87)	20	0.60 (-0.12)	0.60 (-0.12)	2.90 (-0.58)
\vec{F}_6	12	0.80 (-0.10)	6.60 (-0.79)	14.10 (-1.69)	15	2.50 (-0.38)	3.90 (-0.58)	2.00 (-0.30)	18	1.40 (-0.25)	1.40 (-0.25)	4.50 (-0.81)
\vec{F}_7	16	2.00 (-0.32)	8.20 (-1.31)	15.00 (-2.40)	19	4.50 (-0.85)	5.90 (-1.12)	3.30 (-0.63)	22	0.60 (-0.13)	0.60 (-0.13)	3.70 (-0.81)
RH16d	40	- (-)	- (-)	8.10 (-3.24)	40	- (-)	- (-)	0.80 (-0.32)	40	- (-)	- (-)	8.20 (-3.28)

$\vec{F6}$ (1 mosaic, T-2(c)). In general, results based on ϵ_l indicate that GM classifier and ACF-based feature descriptors produces optimal segmentation results.

Comparison between SWA and RH16d results indicates that SWA outperforms the latter in the vast majority of cases. The average LOOE varies between 0.8% and 41.7% and it outperforms SWA results for only two mosaics, T-2(f) and T-2(i). Results based on modified LOOE further indicate that the approach of SWA method is more optimal than that of RH16d which uses a 40-dimensional feature vector for classification.

In addition to the results from previous test suite, it can be concluded that the SWA method gives superior segmentation results and clearly demonstrates the conceptual underpinnings of segmenting heterogeneous materials at RES support. The GM classifier gives better results than MD and LVQ classifiers. In contrast to the analysis on T-1, entropy-based and ACF-based features were found to be the most useful descriptors for artificial texture characterization in T-2. In spite of this, evaluation experiments (described in next section) will use \vec{M} because these material signatures can also be used as input parameters up-scaled transport models. The entropy feature vector, \vec{H} , is not selected for these evaluation experiments in order to minimize the computational complexity (entropy feature is associated with high feature space dimensionality).

2.5.2 Evaluation

In this section, the evaluation results of the SWA method on three bio-medical image data sets are presented. The first data used for evaluation is CT scans whereas the

Table 2.10: SWA segmentation results for mosaics $T-2(d)$ – $T-2(f)$ expressed in average LOOE (modified LOOE).

Feature Vector	T-2(d), RES = 3				T-2(e), RES = 5				T-2(f), RES = 3			
	Size	GM	MD	LVQ	Size	GM	MD	LVQ	Size	GM	MD	LVQ
$\vec{F1}$	4	81.90 (-3.28)	84.20 (-3.37)	84.50 (-3.38)	4	67.20 (-2.69)	78.10 (-3.12)	81.10 (-3.24)	4	61.30 (-2.45)	62.90 (-2.52)	72.70 (-2.91)
$\vec{F2}$	10	32.80 (-3.28)	41.40 (-4.14)	48.10 (-4.81)	16	43.80 (-7.01)	31.30 (-5.01)	48.00 (-7.68)	10	18.00 (-1.80)	23.00 (-2.30)	17.20 (-1.72)
$\vec{F3}$	2	55.80 (-1.12)	64.80 (-1.30)	60.50 (-1.21)	2	51.90 (-1.04)	54.40 (-1.09)	71.90 (-1.44)	2	27.30 (-0.55)	27.00 (-0.54)	25.80 (-0.52)
$\vec{F4}$	6	56.40 (-3.38)	71.30 (-4.28)	54.70 (-3.28)	6	60.00 (-3.60)	59.40 (-3.56)	63.40 (-3.80)	6	25.80 (-1.55)	44.10 (-2.65)	26.60 (-1.60)
$\vec{F5}$	14	30.20 (-4.23)	39.80 (-5.57)	47.00 (-6.58)	20	26.30 (-5.26)	33.10 (-6.62)	49.80 (-9.96)	14	13.30 (-1.86)	19.90 (-2.79)	17.20 (-2.41)
$\vec{F6}$	12	32.70 (-3.92)	35.00 (-4.20)	42.80 (-5.14)	18	26.90 (-4.84)	32.50 (-5.85)	46.40 (-8.35)	12	18.40 (-2.21)	19.50 (-2.34)	14.50 (-1.74)
$\vec{F7}$	16	25.50 (-4.08)	35.60 (-5.70)	38.90 (-6.22)	22	25.00 (-5.50)	29.40 (-6.47)	46.90 (-10.32)	16	14.80 (-2.37)	18.40 (-2.94)	14.50 (-2.32)
RH16d	40	- (-)	- (-)	39.80 (-15.92)	40	- (-)	- (-)	28.50 (-11.40)	40	- (-)	- (-)	8.70 (-3.48)

Table 2.11: SWA segmentation results for mosaics $T-2(g)-T-2(i)$ expressed in average LOOE (modified LOOE).

Feature Vector	T-2(g), RES = 4				T-2(h), RES = 4				T-2(i), RES = 5			
	Size	GM	MD	LVQ	Size	GM	MD	LVQ	Size	GM	MD	LVQ
$\vec{F1}$	4	54.30 (-2.17)	66.00 (-2.64)	59.80 (-2.39)	4	57.80 (-2.31)	72.30 (-2.89)	61.70 (-2.47)	4	64.10 (-2.56)	78.10 (-3.12)	74.20 (-2.97)
$\vec{F2}$	13	13.70 (-1.78)	26.60 (-3.46)	13.70 (-1.78)	13	25.80 (-3.35)	29.30 (-3.81)	42.60 (-5.54)	16	42.20 (-6.75)	45.30 (-7.25)	48.00 (-7.68)
$\vec{F3}$	2	51.20 (-1.02)	53.90 (-1.08)	49.60 (-0.99)	2	48.80 (-0.98)	49.20 (-0.98)	50.80 (-1.02)	2	68.80 (-1.38)	43.80 (-0.88)	73.40 (-1.47)
$\vec{F4}$	6	33.20 (-1.99)	47.30 (-2.84)	41.40 (-2.48)	6	33.60 (-2.02)	63.70 (-3.82)	40.20 (-2.41)	6	56.30 (-3.38)	59.40 (-3.56)	70.70 (-4.24)
$\vec{F5}$	17	12.50 (-2.13)	23.80 (-4.05)	16.00 (-2.72)	17	21.90 (-3.72)	21.50 (-3.65)	37.50 (-6.38)	20	46.90 (-9.38)	50.00 (-10.00)	47.70 (-9.54)
$\vec{F6}$	15	14.50 (-2.17)	25.00 (-3.75)	22.30 (-3.35)	15	25.40 (-3.81)	32.40 (-4.86)	34.40 (-5.16)	18	39.10 (-7.04)	37.50 (-6.75)	47.70 (-8.59)
$\vec{F7}$	19	12.10 (-2.30)	19.50 (-3.71)	20.30 (-3.86)	19	23.00 (-4.37)	24.60 (-4.67)	31.30 (-5.95)	22	40.60 (-8.93)	48.40 (-10.65)	46.90 (-10.32)
RH16d	40	- (-)	- (-)	18.90 (-7.56)	40	- (-)	- (-)	23.30 (-9.32)	40	- (-)	- (-)	18.40 (-7.36)

Table 2.12: SWA segmentation results for mosaics $T-2(j)-T-2(l)$ expressed in average LOOE (modified LOOE).

Feature Vector	T-2(j), RES = 4				T-2(k), RES = 4				T-2(l), RES = 5			
	Size	GM	MD	LVQ	Size	GM	MD	LVQ	Size	GM	MD	LVQ
\vec{F}_1	4	57.00 (-2.28)	71.50 (-2.86)	68.00 (-2.72)	4	79.10 (-3.16)	90.40 (-3.62)	88.90 (-3.56)	4	81.00 (-3.24)	89.50 (-3.58)	89.30 (-3.57)
\vec{F}_2	13	7.00 (-0.91)	7.40 (-0.96)	20.30 (-2.64)	13	41.70 (-5.42)	42.70 (-5.55)	59.20 (-7.70)	16	37.00 (-5.92)	43.50 (-6.96)	55.20 (-8.83)
\vec{F}_3	2	48.40 (-0.97)	43.80 (-0.88)	52.30 (-1.05)	2	63.70 (-1.27)	67.60 (-1.35)	74.40 (-1.49)	2	74.70 (-1.49)	81.90 (-1.64)	80.00 (-1.60)
\vec{F}_4	6	41.00 (-2.46)	59.80 (-3.59)	53.90 (-3.23)	6	54.10 (-3.25)	83.70 (-5.02)	66.30 (-3.98)	6	73.50 (-4.41)	82.30 (-4.94)	75.50 (-4.53)
\vec{F}_5	17	6.30 (-1.07)	5.10 (-0.87)	20.70 (-3.52)	17	36.80 (-6.26)	45.60 (-7.75)	55.00 (-9.35)	20	35.00 (-7.00)	49.70 (-9.94)	55.60 (-11.12)
\vec{F}_6	15	6.60 (-0.99)	7.40 (-1.11)	18.40 (-2.76)	15	40.00 (-6.00)	41.60 (-6.24)	52.20 (-7.83)	18	39.90 (-7.18)	44.60 (-8.03)	57.10 (-10.28)
\vec{F}_7	19	5.90 (-1.12)	5.50 (-1.04)	20.70 (-3.93)	19	36.70 (-6.97)	44.70 (-8.49)	50.60 (-9.61)	22	36.70 (-8.07)	49.90 (-10.98)	55.70 (-12.25)
RH16d	40	- (-)	- (-)	17.20 (-6.88)	40	- (-)	- (-)	36.40 (-14.56)	40	- (-)	- (-)	41.70 (-16.68)

remaining two data sets are MR scans. Between the two MR data sets, the first data set is a simulated MR brain scans with isotropic voxels and the other is a real MR scans with anisotropic voxels. The experiments on these data sets also differ in terms of the qualitative and quantitative nature of assessment.

Based on the results from analyzing the SWA method on two-dimensional real and synthetic image data sets, the evaluation experiments in this section will use GM method for classification and will use the moments-based feature descriptors for both RES identification and classification. In the following sub-sections, the results of the evaluation experiment on each of the three heterogeneous bio-medical image data are presented.

2.5.2.1 3-D Real CT Brain Data

The 2-D stack of CT scans were first re-sampled using bilinear interpolation to result in a new axial slice spacing of $0.5mm$. This ensured that the data set is (approximately) made of isotropic voxels ($0.48828mm \times 0.48828mm \times 0.5mm$). During the re-sampling process care was taken to retain the original 209 slices. Following this operation, ninety-five slices of size 512×512 pixels with zero-intensity were appended to the re-sampled data in order to generate a dyadic, isotropic data set (512^3 voxels). This was done to meet the dyadic requirement of the SWA approach.

After re-sampling the CT data, sample regions corresponding to each of the five materials were manually selected from the CT voxel data set. The first row in Figure 2.20 shows the samples selected for classification along with their size: 32^3 voxels for soft tissue, 16^3 voxels of skin, 8^3 voxels for bone, 64^3 voxels for noise, and

16^3 voxels for background. Following the sample selection procedure, RES support for this test data was identified from the statistics of \vec{M} . The statistical plot of all the four central moments at different scales is shown in Figure 2.20. In the second row, the distribution of $\hat{\mu}$ as a function of scale is shown for each sample material. The third and fourth rows show the statistics of $\hat{\nu}$ and $\hat{\chi}$ as a function of scale. The last row shows the statistics of $\hat{\kappa}$ computed over different tile sizes. In all these plots: the solid line is the mean value of the corresponding statistics; the grey shaded zone represents ± 1 -standard deviation interval; and the top and bottom dashed-lines are the maximum and minimum of the statistics calculated over all the tiles at a given scale.

Results indicate that the statistics for soft tissue and background sample materials stabilize at scale 3 (2^3 voxels). For bone and skin samples it is difficult to arrive at any such conclusion, as the sample sizes are too small (16^3 voxels and 8^3 voxels, respectively) for the statistics to stabilize. For noise, as expected, the statistics do not stabilize with the increase in the sample volume. Results however indicate that the mean statistics of the five materials are relatively well separated as early as scale 2. Based on these observations and data restrictions the RES was selected as 2, meaning the tile size for segmentation is $2^2 \times 2^2 \times 2^2$ voxels. This support scale physically corresponds to a sampling cube of size $2 \times 2 \times 2 \text{ mm}^3$ and appears reasonable in as much as it may represent the characteristics of the major components within a human head.

The signatures of the five materials are presented in Table 2.13. The values within the parenthesis correspond to the standard deviations of their respective

statistic computed at RES. The signatures in the table show that the overall mean intensity ($\hat{\mu}$) between soft tissue and bone sample materials is not distinct enough to discriminate both the materials. However, the standard deviation ($\hat{\nu}$) and skewness ($\hat{\chi}$) for these materials do differ and are expected to help segmenting these materials accurately. It can also be noticed that noise and background have similar statistics suggesting the possibility of belonging to single material. Since these materials will never be used for any further analysis, it is not of a major concern. The $\hat{\nu}$ statistic increases as from soft-tissue to background, which indicates the increase in the heterogeneity of the corresponding materials. Furthermore, the kurtosis statistic ($\hat{\kappa}$) for most of the materials is approximately 3 suggesting that those sample materials with near-zero skewness may be normally distributed. Hence, it is likely that using the GM method with the material signatures that are relatively well separated will yield good segmentation results for the T-3 data set.

Table 2.13: *Statistical characteristics of the five materials for the fresh CT scans of the head section from visible woman data set, T-3, at RES of 2.*

Material	$\hat{\mu}$	$\hat{\nu}$	$\hat{\chi}$	$\hat{\kappa}$
Soft Tissue	68.10 (0.25)	0.01 (0.001)	-0.57 (0.31)	2.92 (0.70)
Skin	65.24 (2.49)	0.02 (0.008)	-0.04 (0.47)	2.52 (0.66)
Bone	155.86 (5.58)	0.05 (0.02)	-0.69 (0.43)	3.10 (1.21)
Noise	6.7 (1.80)	0.08 (0.06)	0.04 (0.61)	3.1 (1.00)
Background	4.1 (0.23)	0.1 (0.03)	0.05 (0.73)	3.89 (1.87)

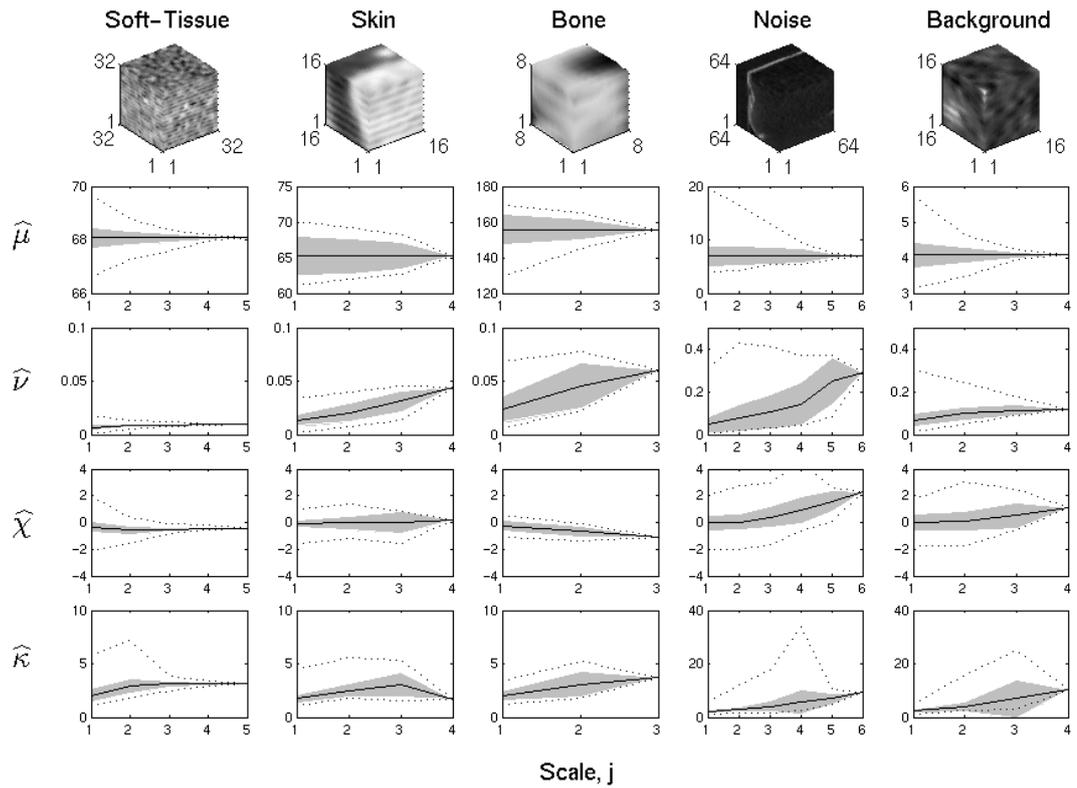


Figure 2.20: Identification of RES support scale using SWA method for the head section of VWD.

The raw classification results from the SWA method at different axial locations (1, 40, 80, 120, 160, and 200) and the raw segmented surfaces of the five materials are shown in Figures and 2.21 2.22, respectively. These results capture the general features of the head (skin, bones and brain) quite well. However, the raw segmentation results contains scattered high frequency noise mostly at the material boundaries. For instance, interface between soft tissue and skin. This high frequency noise is removed by sieving the raw classification results (Pitas, 2000). The output from this post-processing steps is presented in Figure 2.23 and the cut-sections of the processed results are shown in Figure 2.24. The segmented soft tissue is shown in the leftmost column in three different orthographic views. Similarly, the middle and the rightmost columns display the results for skin and bone materials.

Visual inspection of these results clearly indicates that the SWA segmentation method, based on material signatures at RES support, performed very well as skin and bone are completely separated, with skin entirely outside of the skull. Most of the soft tissue (muscle and brain) is found inside and near the skull as expected. An attempt was made to further classify soft tissues separately into muscle and brain matter but the sample of these materials showed identical statistical signatures. This was not unexpected because CT uses X-rays to distinguish between soft and hard tissues and does not provide sufficient detail to further classify within these tissues. Alternative imaging modalities, like MR scans or high resolution color cross-sections would be required to further differentiate the soft tissue. The latter, especially, contains detailed information, but would involve excessive computational resources to process. Overall, the quality of the segmentation results from SWA method can

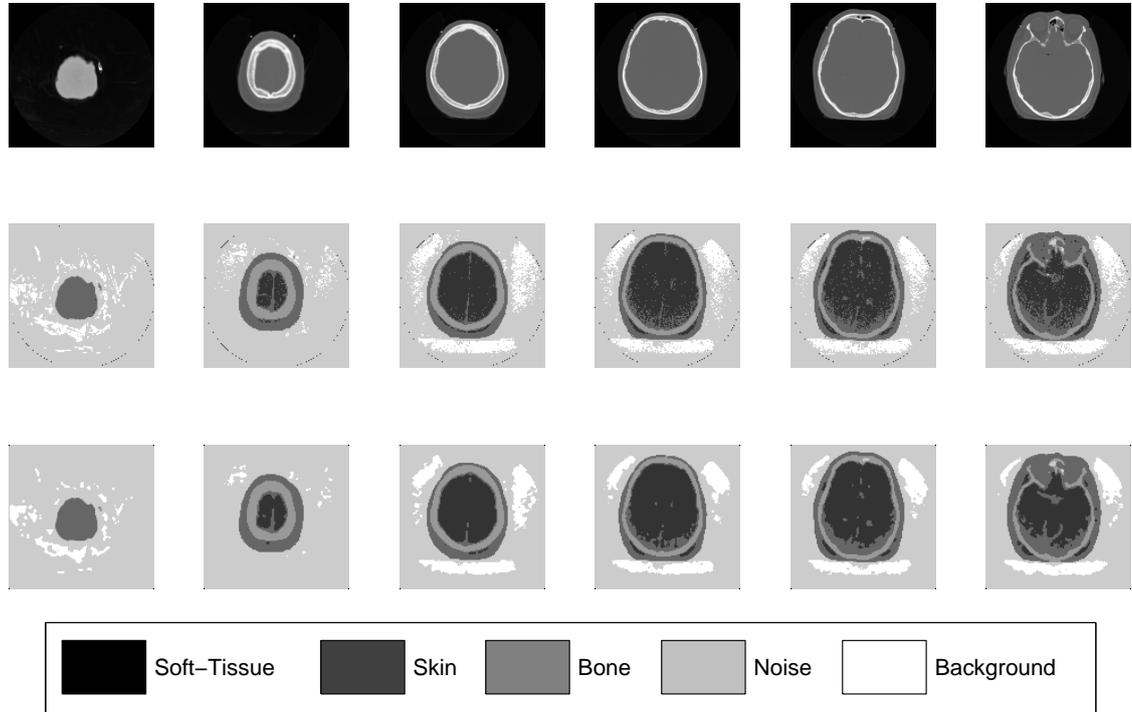


Figure 2.21: *Top: original head section CT scans (512×512 pixels) from VWD at axial locations 1 (0.5mm), 40 (20mm), 80 (40mm), 120 (60mm), 160 (80mm), and 200 (100mm). Middle: raw segmentation results (128×128 pixels) of the corresponding head section CT scans from VWD into soft-tissue, skin, bone, noise and background materials using SWA method at RES support of 2. Bottom: processed segmentation results (128×128 pixels) of the corresponding head section CT scans from VWD into soft-tissue, skin, bone, noise and background materials using SWA method at RES support of 2.*

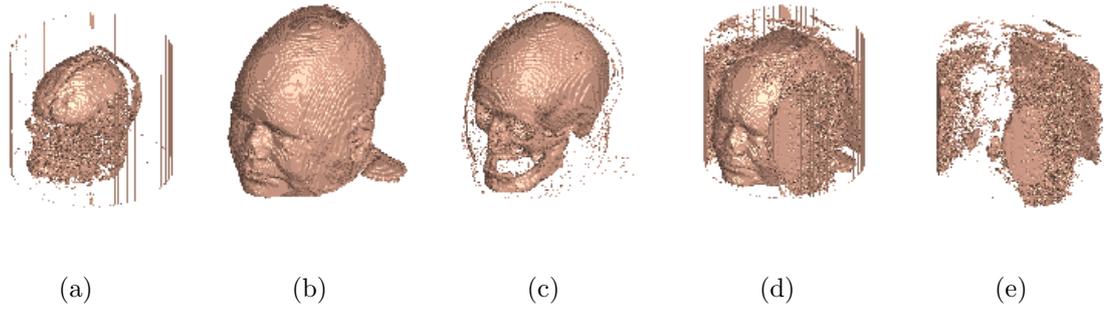


Figure 2.22: *Raw segmentation of head section CT scans from VWD using SWA method at RES support of 2. (a)–(e): surface renderings of the raw segmented volumes of soft tissue, skin, bone, noise, and background, respectively.*

be concluded to be acceptable for heterogeneous material characterization.

2.5.2.2 3-D Simulated MR Brain Data

In this validation experiment, since the ground truth data was available, the samples for each of the material (CSF, grey-matter, white-matter) were selected for each MR brain scans, T-4(a)–T-4(j), by extracting the maximal octants from a octree decomposition.

After selecting the samples for each material, the statistics of the the four central-moments were computed as a function of scale to identify the RES support. The results of the manually identified RES support for the ten brain MR scans are reported in the second column of Tables 2.15 and 2.16 for data sets with INU = 0% and INU = 40 %, respectively. The RES support for all the ten simulated brain MR scans in this test suite was selected as $2 \times 2 \times 2$ voxels, keeping in view of sample size available for CSF material (4^3 voxels). For example, it can be observed from

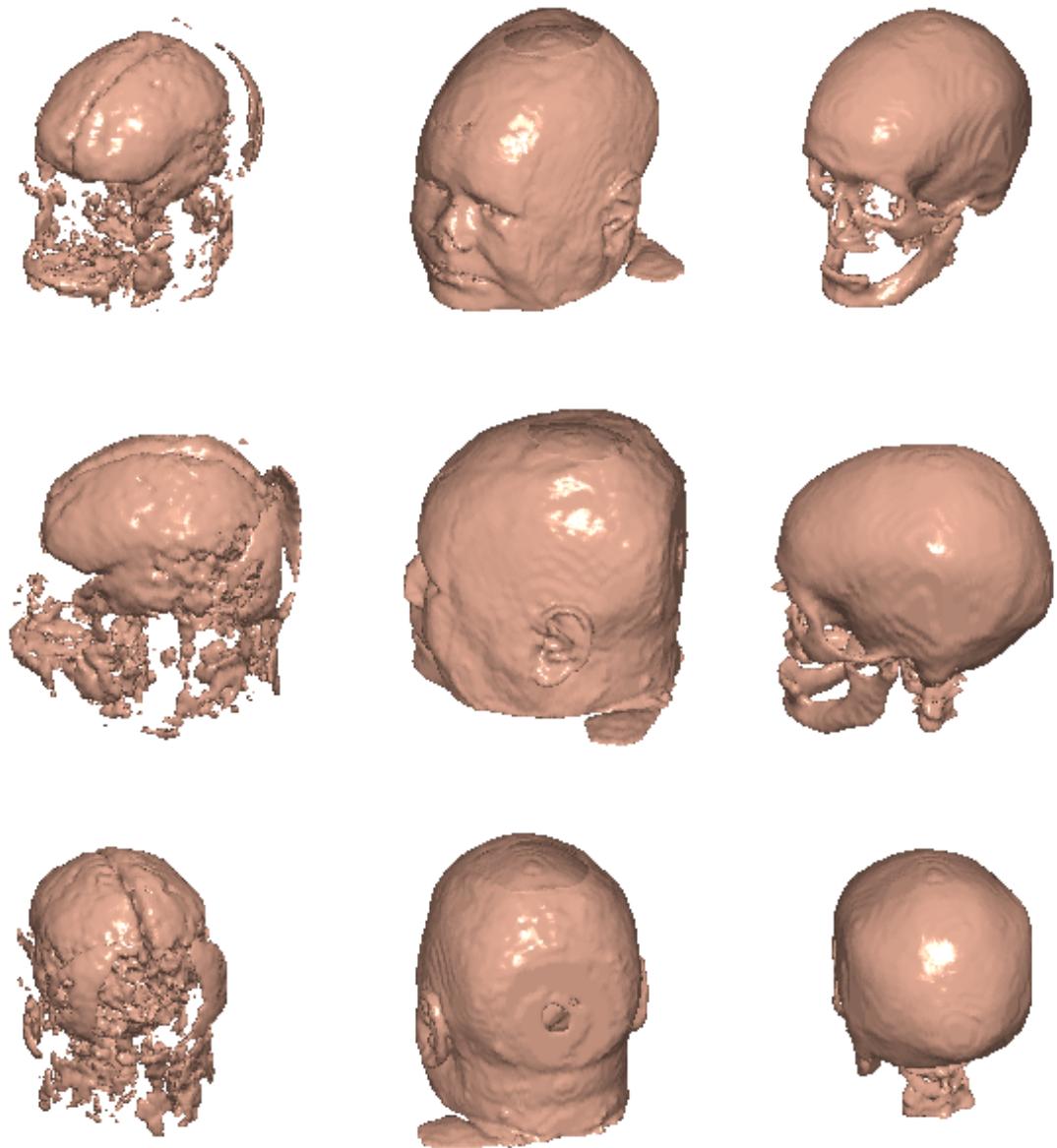


Figure 2.23: *Segmented volumes of different materials after removing the high-frequency noise from the raw results of T-3 using SWA method. The first, second and third column shows the post-processed results for soft tissue, skin, and bone, respectively, in three different perspective views.*

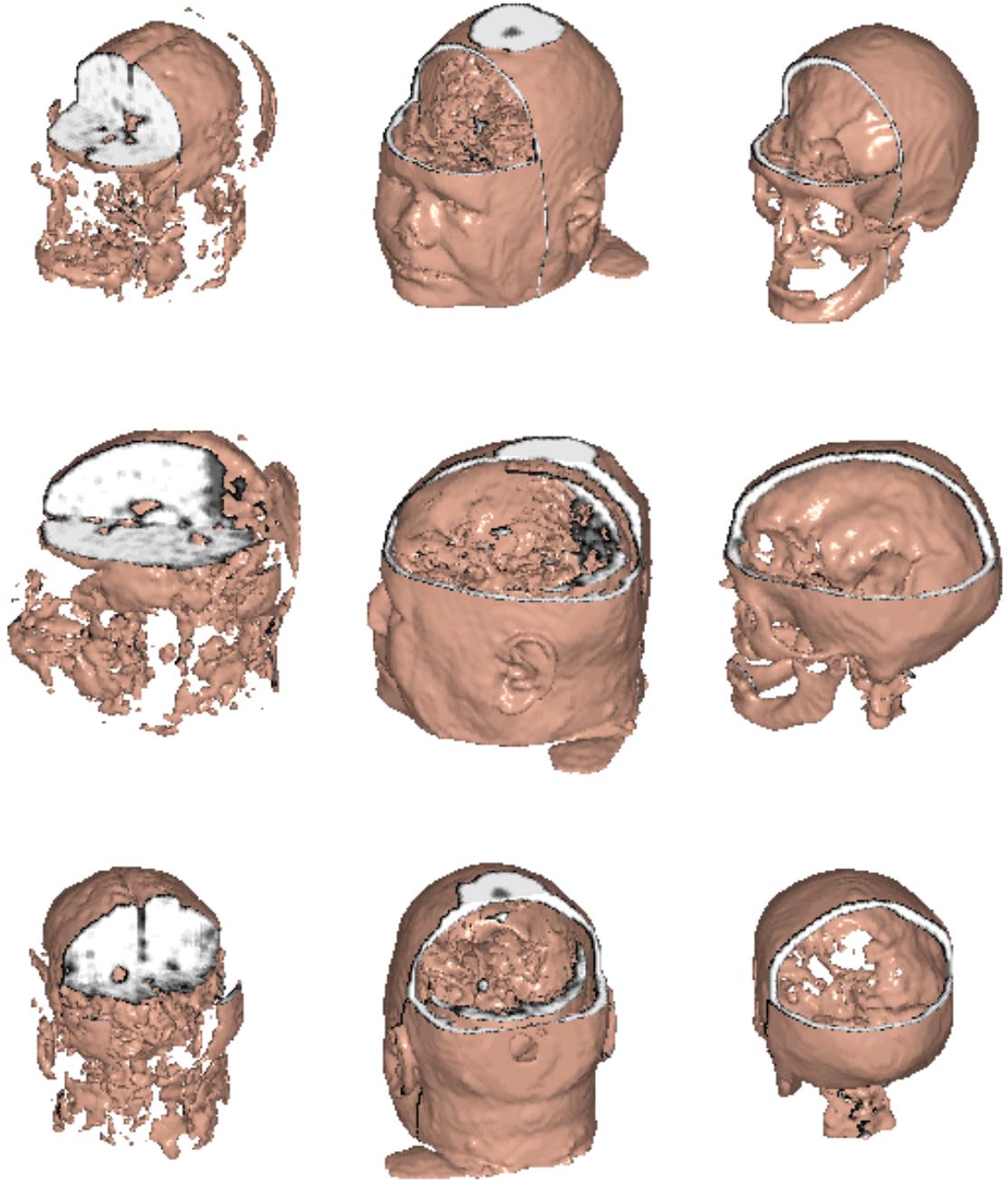


Figure 2.24: *Cut-sections of volumes from the SWA segmentation method after removing the high-frequency noise from the raw results. The first, second and third column shows the cut-sections of post-processed results for soft tissue, skin, and bone, respectively, in three different perspective views.*

Figure 2.25 that, statistics of the feature descriptors for background, grey-matter, and white-matter samples stabilizes at a sampling volume of size $4 \times 4 \times 4$ voxels (i.e., RES support of 2). Based on this observation, if the RES is selected as scale 2, then it becomes too large for characterizing CSF material. Hence, support volume for all the data sets in this test suite was selected as $2 \times 2 \times 2$ voxels which corresponds to a $2 \times 2 \times 2 \text{mm}^3$ of physical brain volume. Furthermore, representative elementary volume (REV) of $2 \times 2 \times 2$ voxels is justified because the mean statistic of the four central moments at this scale are close to normal distribution (refer Table 2.14). Thus, satisfying the assumptions of a GM classifier. However, this is not a limitation of the SWA method because it uses stochastic continuum concept for characterizing different materials and expects different materials to be available in enough sizes for characterization. Similar to the bone sample restriction in the previous evaluation experiment using fresh CT scans, this experiment also suggests the need for high resolution imaging techniques for detailed classification of heterogeneous materials.

Segmentation results expressed in terms of Dice coefficient, s_d , are summarized numerically in Tables 2.15–2.16 and are illustrated in Figures 2.26– 2.27 for an example axial slice at $z = 30\text{mm}$ for the ten simulated MR volumes (i.e., MR brain scans obtained at $\text{INU} = 0\%$ and $\text{INU} = 40\%$, each with $\text{SNR} = 1\%, 3\%, 5\%, 7\%, 9\%$) along with the locations of misclassified voxels for CSF, grey-matter and white-matter. The performance of the developed segmentation method is shown graphically for grey-matter and white matter in Figures 2.28(a) and 2.28(b), respectively. The graphs also show the results obtained by Leemput et al. (1999),

Table 2.14: *Statistical characteristics of the four brain materials for data set T-4(g) with SNR = 3% and INU = 40% from MNI BrainWeb simulated brain MR scans at RES of 1.*

Material	$\hat{\mu}$	$\hat{\nu}$	$\hat{\chi}$	$\hat{\kappa}$
Background	0.59 (2.14)	0.47 (0.04)	0.24 (0.37)	2.31 (0.59)
CSF	1.92 (0.01)	0.05 (0.01)	0.23 (0.59)	2.60 (0.71)
Grey-Matter	4.63 (0.01)	0.02 (0.00)	0.06 (0.42)	2.40 (0.70)
White-Matter	6.25 (0.00)	0.01 (0.00)	-0.21 (0.27)	2.2 (0.38)

who used an Expectation-Maximization method with a extra image processing step to correct the bias field (denoted as KVL). In general, a trend for the grey-matter segmentation results from the SWA method can be observed and is mostly stable across increasing SNR for simulated brain scans with and without intensity inhomogeneities. Surprisingly the performance index of white-matter segmentation increases as the SNR increases from 1%-9% for data sets with and without bias field corruption. Off course, the white-matter segmentation using SWA method under performs for T-5(f) and T-5(g). This was largely due to the white-matter voxels being classified as noise. The overlap ratio produced by SWA method range from 0.72 to 0.96 with the vast majority of values above 0.85. Grey-matter is generally better classified than white-matter which is better classified than CSF. The segmentation results from SWA method are found to be consistent across SNR and INU values expect for a mild trend towards increased accuracy with increasing noise level for

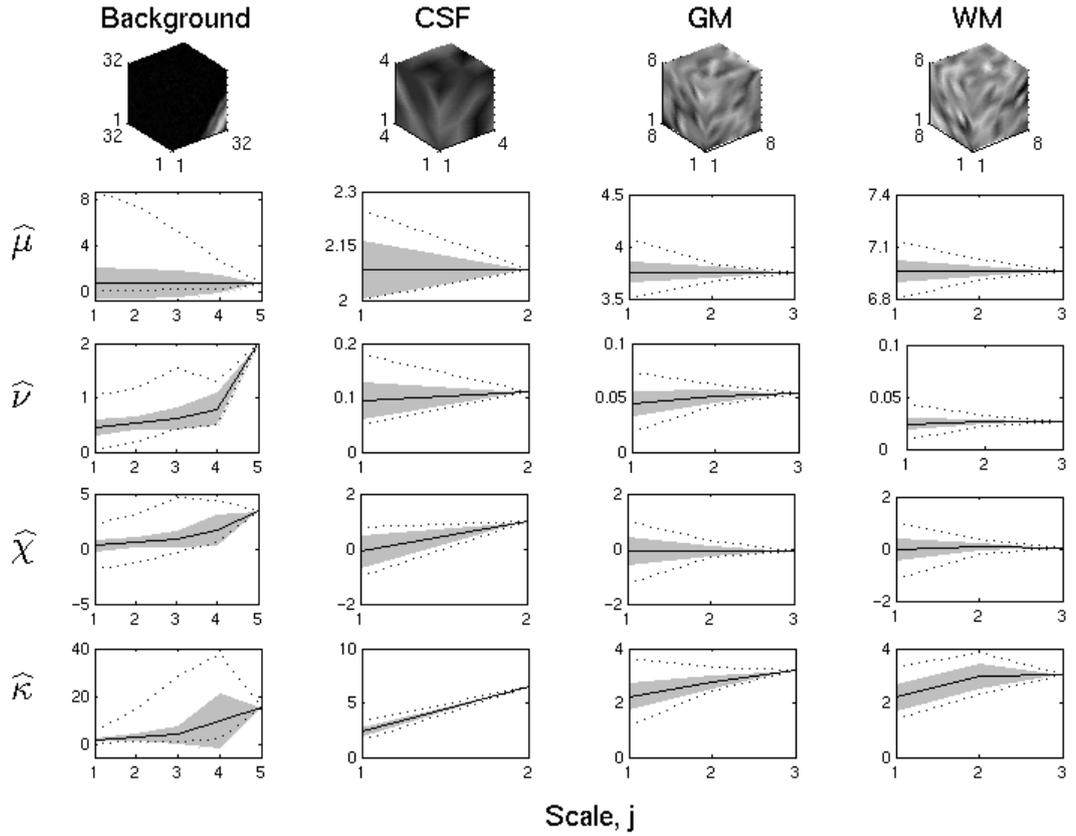


Figure 2.25: Statistics of mean ($\hat{\mu}$), variance ($\hat{\nu}$), skewness ($\hat{\chi}$) and kurtosis ($\hat{\kappa}$) of background, CSF, grey-matter, and white-matter materials used in RES identification step for $T-4(g)$, with $SNR = 3\%$ and $INU = 40\%$.

white matter at $INU = 40\%$.

Comparison between SWA and KVL results indicates that SWA outperforms KVL in the vast majority of cases. The mean overlap ratios obtained for data sets with $INU = 0\%$ and SNR in the range of 1% – 9% using SWA and KVL are: 0.96 and 0.93; and 0.90 and 0.90, for grey-matter and white-matter, respectively. Similarly, the mean overlap ratio for data sets with $INU = 40\%$ and SNR in the range of 1% – 9% between the methods are: 0.94 and 0.90; and 0.91 and 0.90, for grey-matter and white-matter respectively. Results presented in Figure 2.28 further indicate

Table 2.15: Segmentation results of CSF, grey-matter, and white-matter brain tissues in T-4(a)–T-4(e) ($INU = 0\%$ and $SNR = 1\% - -9\%$) using SWA method at their corresponding RES support.

Test Data	RES	Overlap Ratio, $s_d(k)$		
		CSF	Grey-Matter	White-Matter
SNR = 1%, T-4(a)	1	0.90	0.95	0.90
SNR = 3%, T-4(b)	1	0.95	0.94	0.92
SNR = 5%, T-4(c)	1	0.89	0.96	0.94
SNR = 7%, T-4(d)	1	0.86	0.96	0.94
SNR = 9%, T-4(e)	1	0.88	0.96	0.92
SWA Mean $s_d(k)$		0.90 (± 0.03)	0.96 (± 0.001)	0.93 (± 0.02)
KVL Mean $s_d(k)$		–	0.90 (± 0.02)	0.90 (± 0.03)

that the accuracy of SWA classification is more stable over SNR and INU than that of KVL which exhibits a marked decrease in accuracy with increasing SNR. The only case in which SWA underperforms KVL is for white matter classification at 1% and 3% SNR when $INU = 40\%$ which represents less than 5% of tested cases and still produces an overlap ratio of 0.82 or more. The overall results from the SWA method for grey-matter and white-matter segmentation shows an excellent consistent performance along with outperforming the best results reported in the literature (Leemput et al., 1999).

Table 2.16: Segmentation results of CSF, grey-matter, and white-matter brain tissues in $T-4(f)$ – $T-4(j)$ ($INU = 40\%$ and $SNR = 1\% - -9\%$) using SWA method at their corresponding RES support.

Test Data	RES	Overlap Ratio, $s_d(k)$		
		CSF	Grey-Matter	White-Matter
SNR = 1%, T-4(f)	1	0.83	0.96	0.83
SNR = 3%, T-4(g)	1	0.89	0.96	0.87
SNR = 5%, T-4(h)	1	0.72	0.95	0.92
SNR = 7%, T-4(i)	1	0.83	0.95	0.91
SNR = 9%, T-4(j)	1	0.86	0.89	0.96
SWA Mean $s_d(k)$		0.83 (± 0.06)	0.94 (± 0.03)	0.90 (± 0.05)
KVL Mean $s_d(k)$		–	0.91 (± 0.02)	0.90 (± 0.03)

2.5.2.3 3-D Real MR Brain Data

In this experiment also, since the ground truth data was available, the samples for the three brain materials were extracted from the eleven brain MR volumes (T-5(a)–T-5(k)) by first constructing an octree on their respective ground truth volumes (G-5(a)–G-5(k)) and then selecting the largest possible blocks from the octree that correspond to the brain materials. After the sample materials were selected, the statistics of the four central moments were estimated as the function of scale in the usual manner. The plots of statistics as a function were then used as visual

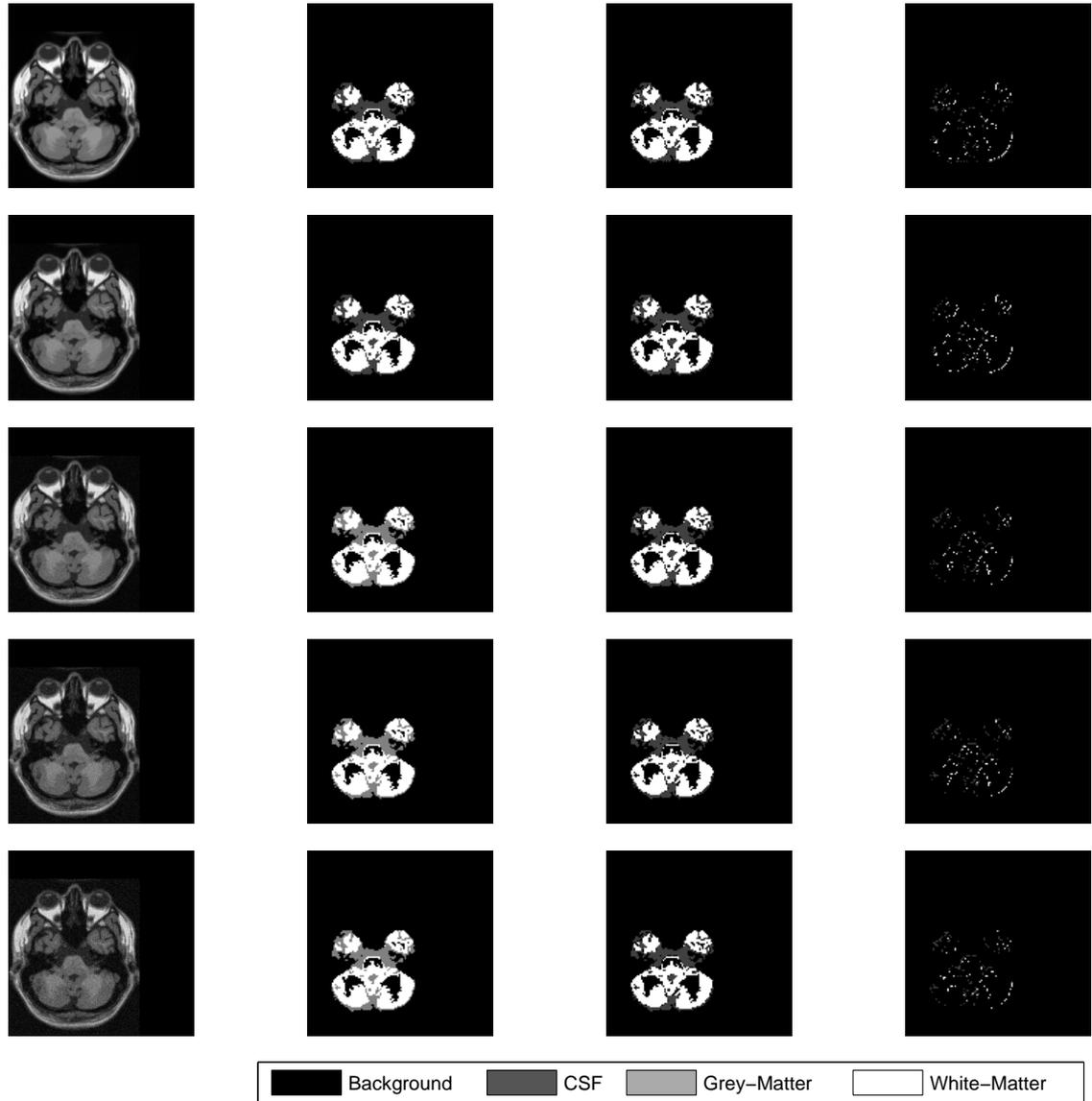


Figure 2.26: *The first, second, third, and fourth column shows the input, ground truth, segmentation and misclassification results of an axial slice at $z = 30\text{mm}$ with 1%, 3%, 5%, 7%, and 9% SNR and $INU = 0\%$ into CSF, grey-matter, and white-matter using SWA method, respectively.*

aides for identifying the RES support. The results of identified RES support for each of the eleven brain MR scans are shown in the second column of Table 2.17.

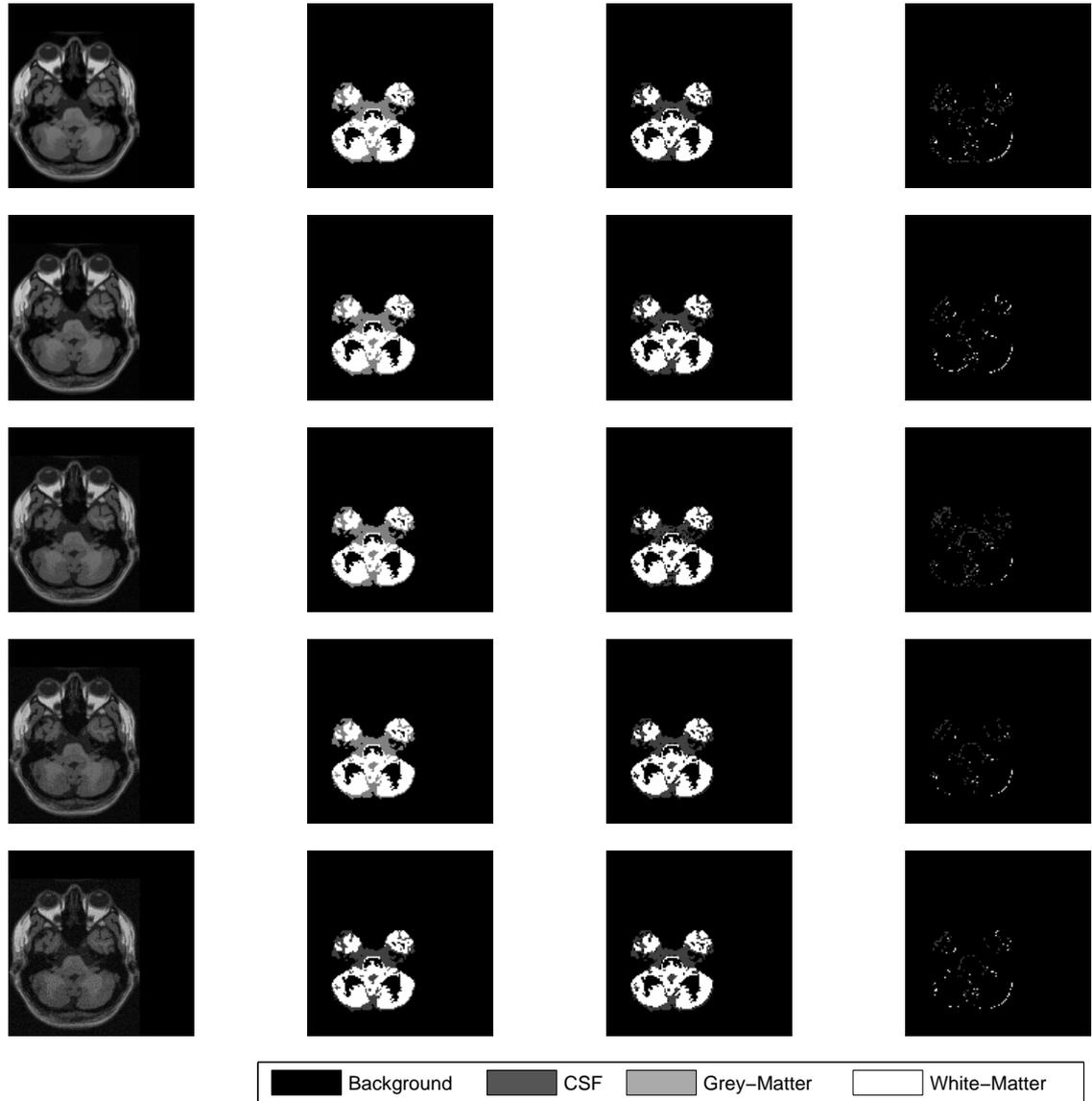
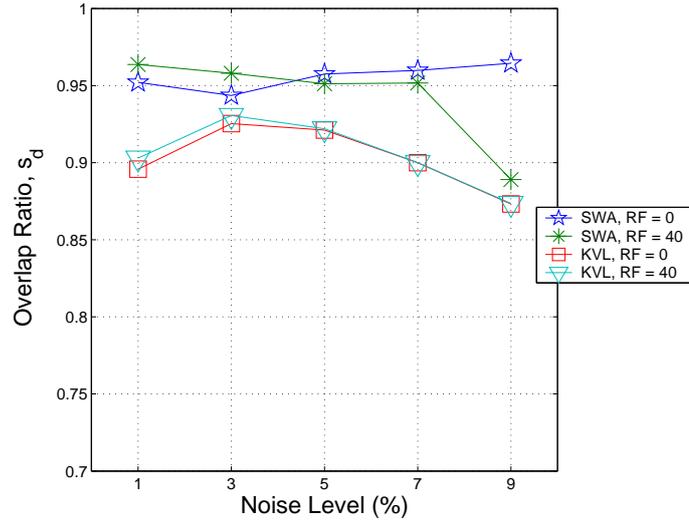
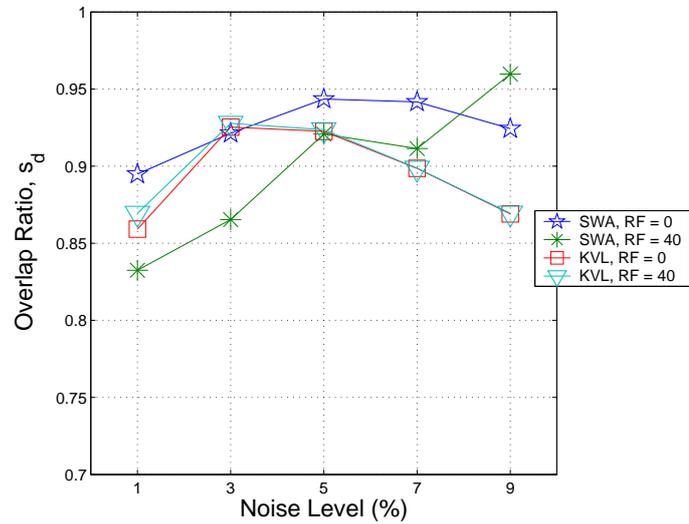


Figure 2.27: *The first, second, third, and fourth column shows the input, ground truth, segmentation and misclassification results of an axial slice at $z = 30\text{mm}$ with 1%, 3%, 5%, 7%, and 9% SNR and $INU = 40\%$ into CSF, grey-matter, and white-matter using SWA method, respectively.*

In similarity with the previous section the RES support is found to correspond to voxels of $2 \times 2 \times 2\text{mm}^3$ to $4 \times 4 \times 4\text{mm}^3$.



(a)



(b)

Figure 2.28: Segmentation results of (a) grey-matter and (b) white-matter for T-4 at RES support using SWA approach. KVL refers to the EM segmentation algorithm with bias estimation (Leemput et al., 1999).

After the RES identification, the GM classifier was used to segment the brain MR volumes into the desired three materials. The segmentation results for the CSF,

Table 2.17: Segmentation results of the brain MR volumes T-5, expressed in terms of Tanimoto similarity coefficient (s_t), into CSF, grey-matter, and white-matter using SWA approach at their respective RES support.

Test Data	RES	Overlap Ratio, $s_t(k)$		
		CSF	Grey-Matter	White-Matter
T-5(a)	1	0.76	0.58	0.39
T-5(b)	1	0.36	0.66	0.69
T-5(c)	2	0.22	0.47	0.58
T-5(d)	2	0.27	0.75	0.58
T-5(e)	2	0.62	0.68	0.68
T-5(f)	1	0.53	0.87	0.35
T-5(g)	2	0.54	0.91	0.36
T-5(h)	2	0.80	0.77	0.68
T-5(i)	1	0.86	0.46	0.70
T-5(j)	2	0.43	0.90	0.52
T-5(k)	2	0.33	0.88	0.73
Mean		0.52	0.72	0.57

grey-matter and white-matter brain tissues in each of the eleven IBSR brain volumes are presented in Figures 2.29(a), 2.29(b), 2.29(c), respectively, where they also com-

pared to the performance of different segmentation methods that were applied to the same data sets (Rajapakse and Fruggel, 1998): maximum a posteriori probability (MAP), adaptive maximum a posteriori probability (AMAP), biased maximum a posteriori probability (BMAP), fuzzy k-means (FkMN), maximum likelihood classifier (MLC) and tree-structured k-means (TSKM).

The SWA produces segmented images with overlap ratios that are generally higher than those of the other methods for the 11 test images. For CSF, the SWA overlap ratios are always significantly larger than those of other methods, irrespective of the level of difficulty of the image. For grey-matter, the SWA produces overlap ratios higher than other methods by up to 0.4 and produces a lower overlap ratio in only one, relatively easy, test image: T-5(i). For white-matter, the SWA produces overlap ratios higher or comparable to those of other methods in 7 test images, including the 5 most difficult ones, and lower overlap ratios in 4 test images. Overall, the SWA produced overlap ratios visibly superior to those of other methods in 76% (25) of these 33 tests (3 materials in 11 images) and results that are either superior or comparable to other methods in 85% (28) of the test cases. Most interestingly, the SWA always segmented the most difficult images more accurately than other methods. This is probably due to the fact that the SWA uses distinct spatial statistics for each material in the image rather than using only local voxel values (k-means) or a single set of parameters to represent spatial heterogeneity (MAP). Presumably, the criteria used to classify brain images into various levels of difficulty may have been biased more towards local voxel values (that affect class means) than towards texture. Stated differently, it appears that images classified

as "highly difficult" based on local voxel values, still contained substantial textural information which SWA successfully used to accurately segment them.

Mean overlap ratios produced by the SWA, averaged over materials, and averaged over images, are compared to those of the 6 other methods in Figures 2.29(d) and 2.29(e), respectively. Results indicate that the segmentation produced by the SWA is, on average, always better than that produced by other techniques. For the 11 test images, the material-averaged overlap ratio (mean of overlap ratios of individual materials) of SWA is at least 0.13 larger than that of other methods, corresponding to a segmentation accuracy improvement of at least 28% (the closest performing method is BMAP). Similarly, the overlap ratios produced by SWA for individual materials, averaged over the test images, are at least 0.17 (grey-matter), 0.03 (white-matter) and 0.43 (CSF) above those of other methods (Table 2.18) which indicates averaged accuracy improvements of at least 31%, 6% and 480% for grey-matter, white-matter and CSF segmentation, respectively.

Finally, it is noted that SWA results do follow a trend of increasing accuracy as the level of difficulty of images decreases (Figure 2.29(d)). However this trend is milder than in prior methods (k-means and MAP), indicating that SWA is more robust to intensity non-uniformity, noise, contrast deficiencies and other such anomalies of the source MR image.

2.6 Conclusions and Future Research

In this chapter, a new texture based segmentation method for heterogeneous biological image data was developed and evaluated. The developed segmentation

Table 2.18: Mean overlap ratio, expressed in terms of Tanimoto similarity coefficient (s_t), for eleven brain MR volumes in test suite T-5 using different segmentation methods.

Material	Mean Overlap Ratio, $s_t(k)$					
	MAP	AMAP	BMAP	FkMN	TSKM	SWA
Background	1.00	1.00	1.00	1.00	1.00	1.00
Grey-matter	0.53	0.55	0.53	0.46	0.51	0.72
White-matter	0.52	0.54	0.53	0.54	0.52	0.57
CSF	0.09	0.08	0.09	0.06	0.07	0.52

method (SWA) uses stochastic continuum concept to characterizes materials at a coarser scale by using the heterogeneity properties from finer scale(s). The SWA approach provides a quantitative approach to identify the representative elementary scale (RES) at which the material properties of macroscopic entities become homogeneous. From a pattern recognition point of view, the RES concept is analogous to the definition of "texture" scale. The RES support was identified as the scale at which the mean statistics of the four central moments stabilize. The SWA method is based on multi-dimensional Haar wavelet decomposition for texture analysis utilizing its symmetric and orthogonal properties for fast computations in arbitrary dimensions.

The feature descriptors and classifier used by the SWA method were selected by comparing the performance of 21 combinations of 7 feature vectors with 3 classifiers. The feature vectors include first-order spatial moments, entropy, and direc-

tional auto-correlation lengths. The moment-based features (\vec{M}) and the ACF-based features (\vec{Z}) were the primary choices as they not only assist in segmentation of the entities within biological image but can also be directly used as input parameters in stochastic transport models. All the feature descriptors (except ACF-based) were calculated efficiently from wavelet coefficients obtained using Haar MRA. The three investigated classifiers were the Gaussian maximum likelihood with Mahalanobis distance metric (GM), minimum-distance classifier (MD), and learning vector quantization (LVQ).

Feature vectors and classifiers were analyzed over a set of three 2-D real images (a soil cross-section, a muscle tissue and a landscape) and a set of twelve 2-D synthetic texture mosaics commonly used in the pattern recognition literature. A performance metric, ϵ_d , that considers both segmentation accuracy (leave-one-out error, ϵ_l) and feature vector size was used to compare the 21 feature vectors-classifier combinations. Results indicated that the combination of spatial moments as feature descriptors with the GM classifier provides the highest classification performance for real image data. This combination also gave very good results on synthetic texture mosaics, although results were less consistent than on real images. The Gaussian Mahalanobis classifier using spatial moments was concluded to be superior to other techniques and was selected for the SWA since the goal of the method is to segment real biological images.

The developed SWA segmentation technique was evaluated on three biomedical data sets. These test-suites varied from each other with respect to: (a) type

of sensors used for data acquisition, (b) presence of imaging artifacts in the data, and (c) spatial resolution of the data. The evaluation was both qualitative and quantitative in nature and whenever feasible the results from SWA method were also compared to contemporary segmentation methods. Results from the qualitative evaluation experiment using test suite T-3 with voxel size $0.125mm^3$ demonstrated that the SWA method was successful at segmenting skin, bone, and soft tissue, from the fresh CT scans of head. Results from the quantitative evaluation on ten simulated brain MR images with voxel size of $1mm^3$ (T-4) demonstrated that the SWA method gives consistent segmentation results even in the presence of intensity inhomogeneity and noise, and it outperforms the KVL method based on expectation-maximization technique with a bias-field correction step. Segmentation results obtained on eleven real brain MR scans with voxel size of $2mm^3$ (T-5), further demonstrated that the SWA method is superior to six segmentation methods presented in literature: maximum *a posteriori* (MAP) probability, adaptive maximum *a posteriori* probability (AMAP), biased maximum *a posteriori* probability (BMAP), fuzzy k-means (FkMN), maximum likelihood classifier (MLC), and tree-structured k-means (TSKM).

Development and evaluation results demonstrate that the SWA segmentation method is an excellent technique for segmenting heterogeneous biological image data. It is accurate, efficient and works equally well on bio-environmental and biomedical image data. Additionally, it works equally well in two-dimensions (digital photos, microscope images, and satellite images) and three dimensions (CT and MR images). It is accurate even if data is noisy and provides better efficiency-accuracy

tradeoff than twenty other combinations of feature vectors and classifiers. It outperforms six other segmentation methods presented in the literature. In addition, it directly provides, as output, the heterogeneity statistics needed as input parameters to stochastic transport models. For these reasons, the developed SWA method is concluded to be a promising tool with great potential for unifying data analysis concepts into simulation and modeling techniques and for integration into biological decision support systems where digital images are an important source of input data.

Although the novel segmentation method developed in this chapter gave excellent results for characterizing macroscopic materials in heterogeneous biological image data, it still offers several avenues for potential improvements through research. Firstly, a technique for standardizing the RES identification step could be developed by defining stronger guidelines and mathematical justification. Currently, the RES support is identified by supervised selection of the scale at which spatial statistics stabilize. This could be automated (*unsupervised*) by analyzing the statistics of the feature descriptors and identifying the scale at which the deviation of these statistics is optimal in some sense.

Secondly, one of the main assumptions of the SWA is that, all of the different macroscopic materials present in the data have the same characteristic length scale. Thus, the RES identification process selects a common representative elementary scale for all the materials. In many practical cases, materials with different characteristic length scales can coexist in the data (for example, soil with sand and gravel horizons). In such situations, it could be better if the SWA method could

use *adaptive RES supports* for segmenting or characterizing these materials at different length scales. Such an implementation could also indirectly help to resolve the *partial volume effects* common to bio-medical data sets.

Thirdly, the moment-based features descriptors used by the SWA method include only the first order moments. The generality of the method, and its accuracy for synthetic data sets, could be improved by adding *higher order statistical measures* (like, co-occurrence matrices or ACFs). However, to remain consistent with the MRA philosophy of the method, this will require the development of techniques to compute such higher-order statistics directly from the multi-resolution wavelet coefficients. In this context, the recent developments of *directional wavelets* (Antonie et al., 1999; Do, 2001; Li et al., 1997; Vandergheynst and Gobbers, 2002) may prove useful for computing directional correlation lengths from wavelet coefficients.

Fourthly, although excellent, the development and evaluation results for SWA method presented in this chapter are limited to the data sets: (a) three 2-D biological images, (b) twelve 2-D texture mosaics, and (c) twenty-two 3-D brain or head bio-medical images. It would be appropriate to extend the analysis of SWA to a much larger number of data sets and classification methods, like the Statlog project (Michie et al., 1994). Additionally, quantitative performance measures, like *area under the ROC curves*, could be used to compare different classification methods (Webb, 2002). It would also be interesting to investigate the *feature selection and feature extraction* techniques and their impact on the overall performance of the method (Bauer and Kohavi, 1999; Gama, 2000; Gama and Brazdil, 2000).

Lastly, the potential of *parallel implementation* of the SWA method would also

be an interesting topic to investigate. This will, particularly, be useful for handling large multi-dimensional data sets efficiently. It is notable that the separable Haar wavelet decomposition with symmetric and orthogonal properties makes it extremely well-suited for parallelization (Feil and Uhl, 2002; González et al., 2001; Yang and Mishra, 1998).

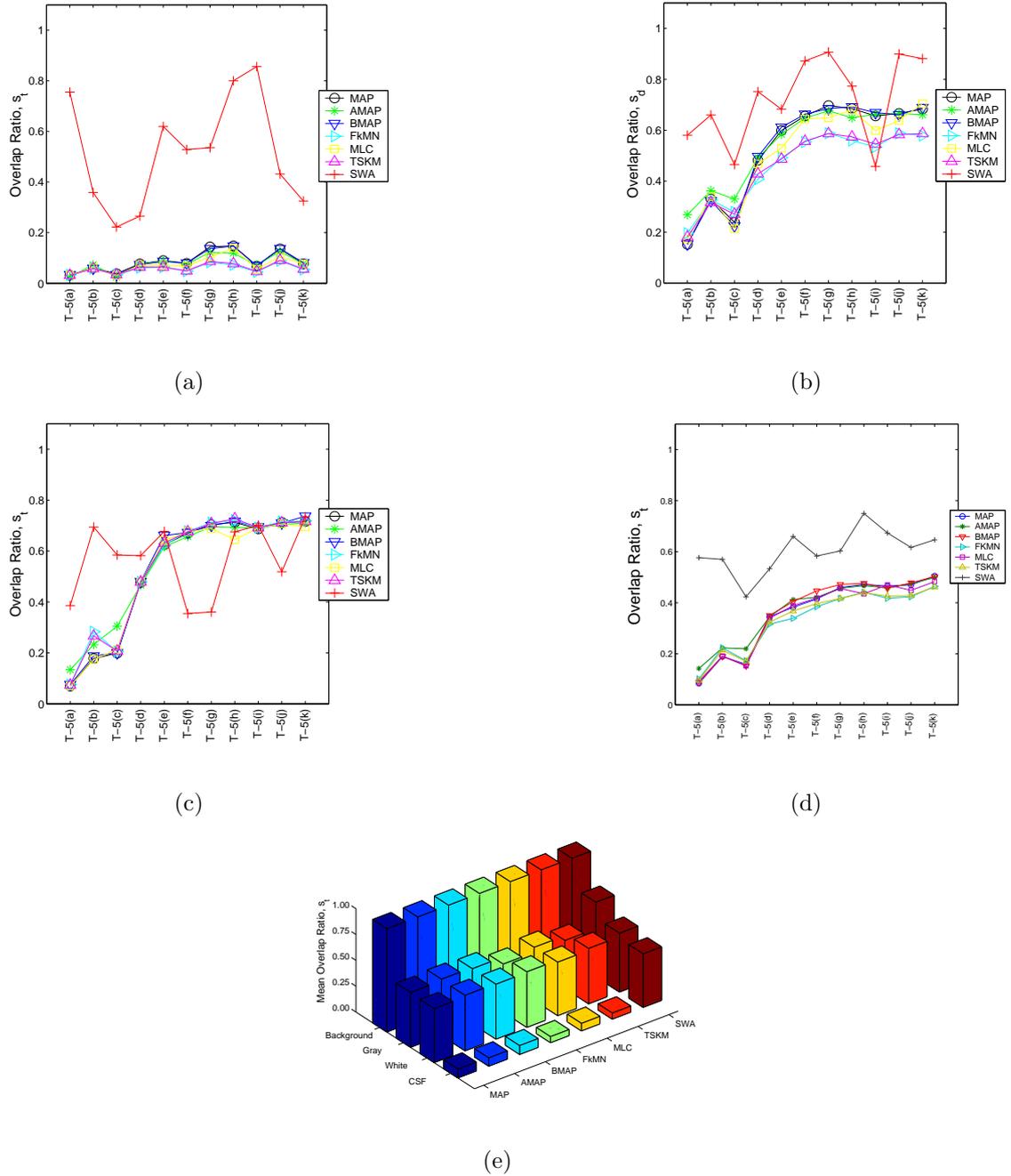


Figure 2.29: Segmentation results of: (a) CSF; (b) grey-matter; (c) white-matter for T-5 using SWA approach and various segmentation methods from IBSR. (d) Tanimoto coefficient averaged over all the brain materials for MR scans in T-5 test suite; (e) bar plot showing the mean Tanimoto coefficient over the entire T-5 test suite.

In this chapter, the next step of unstructured mesh generation of image data for use in the numerical analysis of bio-systems is developed, implemented, and analyzed. The developed mesh generation technique, IMESH, is implemented in Java SDK 1.4.1. Formal mathematical analysis is done to show the exact theoretical bounds on the quality of the elements, which is also experimentally verified. The adaptivity and efficiency of the technique is also demonstrated on a segmented 2-D landscape image and on a segmented 3-D MR brain scan. The contribution of the research described in this chapter is important in three ways. Firstly, the approach is based on a strong theoretical foundation in mesh quality up to \mathbb{R}^8 dimensions. Secondly, the unstructured mesh generation technique developed herein seamlessly integrates with the quantitative heterogeneity-adaptive segmentation method developed in Chapter 2. Lastly, the mesh generation technique developed here can be used on a wide variety of computing platforms (from a implementation stand point).

3.1 Introduction

Consider the example of the patient suffering from chronic *astrocytoma glioma*, undergoing chemotherapy treatment as described in Chapter 1. In order to develop a strategic drug delivery mechanism or treatment plan, a multi-step process (shown in Figure 1.1) has to be developed, to accurately understand (by up-scaled transport models) the effect of chemotherapy drugs.

The first step involved in this multi-step process would be to obtain quantitative information of the tumor tissue such as the biological, physical, chemical, and structural properties from biomedical images like CT and MR scans. Next, the tumor tissue (domain of interest) needs to be identified (segmented) for performing the numerical simulation. Once the domain is identified, a mesh with simpler elements needs to be generated, to facilitate the application of numerical techniques for simulating the process dynamics occurring in the bio-system.

Given the fact that the bio-systems are highly heterogeneous, the simulation process should account for this heterogeneity in their properties as much as possible for more accurate understanding of processes in these systems. In Chapter 1, it has been shown that the availability of sophisticated image sensors now permit to non-invasively capture the relevant material properties needed for modeling bio-systems. However, taking into account of the heterogeneity to very fine scales, is computationally prohibitive. The up-scaled numerical techniques provides the mathematical framework to account for the sub-scale heterogeneity by identifying an representative elementary scale (RES) support and then simulating the processes occurring in the bio-systems at that scale. Therefore, if the image data is to be used in such a numerical framework, then it should be processed to, first identify the RES support and then identify the domain of interest at the RES support.

In Chapter 2, a novel texture-based segmentation technique (SWA) rooted in up-scaled stochastic continuum concept was developed, analyzed and validated. The SWA technique permits: (a) quantitative characterization of the heterogeneous (material) properties captured by image sensors by identifying RES support at which

the different components are continuum, and (b) segmentation of the different materials at RES support by using heterogeneity statistics which form the parameters for up-scaled numerical techniques.

Before the results from the proposed segmentation can be used in up-scaled numerical modeling framework, one of the necessary steps would be to perform *mesh generation*. Mesh generation, especially unstructured, has been a part of mainstream computational geometry for quite sometime. In the last fifteen years, there has been tremendous advancement in this area, when the problem attracted computational geometers (Bern et al., 1990; Joe, 1995; Mitchell and Vavasis, 1992; Ruppert, 1993; Shewchuk, 1998; Yerry and Shephard, 1991). The current mesh generation techniques can be handled gracefully most of the complicated polygonal and polyhedral domains (Bern et al., 1990; Chew, 1989; Mitchell, 1994; Mitchell and Vavasis, 1992; Preparata and Shamos, 1990; Ruppert, 1993, 1995; Shephard and Georges, 1991). However, most existing techniques, if not all, were proposed for polygonal and convex polyhedron domain decomposition and not for image data.

In spite of the vast literature on mesh generation, there does not exist a single quality unstructured mesh generation method for image data. This has been a compelling factor for the scientific community not to use digital images capturing vital material properties in any form of numerical analysis. Even when used, in most instances, the underlying uniform grid formed by the pixels/voxels is used directly for meshing.

The aim of the research described in this chapter is to address this severe shortcoming by developing a quality guaranteed mesh generation technique for image

data. The remainder of this chapter is organized as follows. Section 3.2 starts with a review of the various types of domain decomposition (meshes) used in numerical framework . This is followed by literature review on mesh generation techniques, more specifically, unstructured mesh generation, available for polygonal domains, convex polyhedral domains and image data. After summarizing the mesh generation techniques, the main objectives of the research described in this chapter are presented in Section 3.3. In Section 3.4 the materials and methods needed to achieve the objectives are outlined. This is followed by Section 3.5 where results are presented and discussed. Section 3.6 concludes the chapter and proposes future research directions.

3.2 Literature Review

A *mesh* is a discretization of domain $\Omega(x_1, x_2, \dots, x_d)$ in \mathbb{R}^d into small simple shapes or *simplices*, typically triangles or quadrilaterals in \mathbb{R}^2 and tetrahedra or hexahedra in \mathbb{R}^3 . Meshes composed of these simple shapes are used in many applications such as computer graphics for scene rendering, numerical methods for interpolation, and geographic information systems (GIS) for terrain modeling. One of its most significant application is in the area of finite element analysis for solving partial differential equations (PDEs) that arise in domain-based numerical simulations.

Meshes used in numerical methods can (usually) be categorized as *structured* or *unstructured*. The generation of both structured and unstructured meshes can be surprisingly difficult, each posing challenges of their own. But the mesh generation literature reviewed in this chapter considers only the task of generating unstructured

meshes. Furthermore, focus is mainly on generating simplicial meshes, i.e. meshes composed of triangles in \mathbb{R}^2 and tetrahedra in \mathbb{R}^3 . Meshes with quadrilateral, hexahedral, or other non-simplicial elements are not dealt, although they comprise an interesting field of study in their own right. Since the related topics of mesh refinement, de-refinement, smoothing, improvement, and geometric problems are beyond the scope of this document they are omitted from discussion.

3.2.1 Structured and Unstructured Meshes

Structured meshes exhibit a uniform topological structure that unstructured meshes lack. A functional definition is that in a structured mesh, the indices of the neighbors of any node can be calculated using simple addition, whereas an unstructured mesh necessitates the storage of a list of each nodes neighbors. In practice, a third category of *block-structured* or *hybrid* mesh is also found, which is formed by a number of small structured meshes combined in an overall unstructured pattern. High-quality hybrid meshes enjoy the advantages of both approaches, but hybrid meshing is not yet fully automatic. Figure 3.1 illustrates an example of each. In general, structured meshes offer simplicity and easy data access, while unstructured meshes offer more convenient mesh *adaptivity*, refinement and de-refinement based on initial solution, and a better fit to the complicated domains. Unstructured mesh is also characterized by the *flexibility* it offers with respect to element sizes (i.e. it allows for grading its element size rapidly from small to large size). Unstructured meshes can provide *multi-scale resolution* and *conformity* to complex geometries, far better than structured meshes. The difference between structured and unstructured

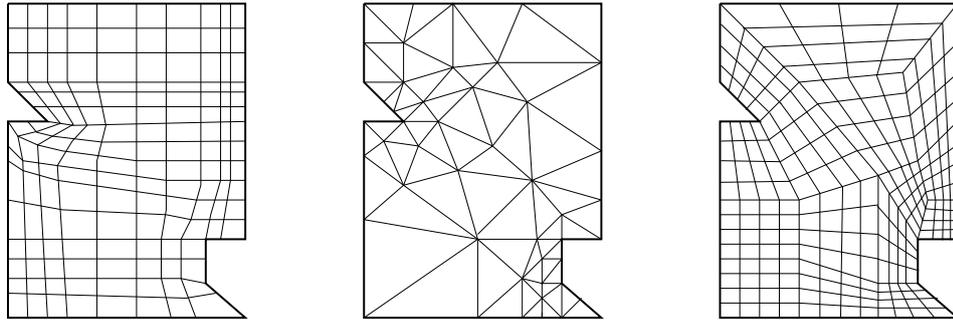


Figure 3.1: *Types of meshes: (a) structured, (b) unstructured, (c) hybrid for the simple polygonal domains.*

meshes also extends to the shape of the elements: 2-D structured meshes typically use quadrilaterals, while unstructured meshes use triangles. In 3-D the analogous element shapes are hexahedra and tetrahedra, respectively. However, there is no reason for structured and unstructured meshes to use different element shapes because it is possible to subdivide elements in order to convert between triangles and quadrilaterals and between tetrahedra and hexahedra.

3.2.2 Mesh quality for numerical methods

Many physical phenomena in science and engineering can be modeled by partial differential equations (PDEs). When these equations have complicated boundary conditions or are posed on irregularly shaped objects or domains, they usually do not admit closed-form solutions. A numerical approximation of the solution is thus necessary. Numerical methods for solving PDEs include finite difference methods (FDM), finite element methods (FEM), and the finite volume methods (FVM, also known as the control volume method). They are used to model disparate phenom-

ena such as mechanical deformation, heat transfer, fluid flow, electromagnetic wave propagation, and quantum mechanics. These methods numerically approximate the solution of a linear or nonlinear PDE by replacing the continuous system with a finite number of coupled linear or nonlinear algebraic equations. This process of *discretization* associates a variable to the finite number of points in the problem domain and can be divide into three interdependent steps: problem formulation, mesh generation, and solution of equations. Note that, although these methods result in linear systems of similar structure, the desired characteristics of meshes for these methods differ.

For FEM purposes, the shape and the quality of the elements in a FE mesh dictates the accuracy and the numerical stability of the PDE being solved. Unstructured mesh must satisfy certain conditions to properly handle the PDE for a particular domain. According to Bern et al. (1990), a FE mesh must at least satisfy the following conditions: (a) conformity - the mesh must conform to the domain being modeled; (b) adaptivity - the mesh must be fine enough to produce an acceptable approximation of the original problem, and, in parts of the domain where the solutio is complicated or changes rapidly may require large number of smaller elements than other part; (c) roundness - the elements should be relatively *well-shaped*, because elements with large or small angles can degrade the quality of the numerical solution. The last property is by far the most difficult to attain. For some methods, elements with small angles lead to ill-conditioned linear algebraic systems that are difficult to solve accurately.

In two dimensions, the finite element mesh should specifically avoid *skinny* and

cap elements. The former represents elements with small angles, whereas elements with obtuse angles (angles greater than 90°) fall under the latter category. An ideal finite element mesh should not have any skinny triangles as it affects the performance of the solver. But at times, skinny triangles could be beneficial, especially, in simulations involving anisotropic equations—the second derivative varies greatly with direction. For example, numerical solution of the *Navier-Stokes* equation with the viscosity parameter for simulating fluid dynamics. Elements with large angles are undesirable because certain numerical methods requires the center of the circumcircle of each element to lie within the element (Baker et al., 1988), so that the perpendicular bisectors of the element edges form the planar dual (Voronoi graph) of the mesh. Circumcenter of an element would lie within the element if and only if no angle of the element is obtuse. Hence, for the circumcenters of elements to be well separated from the boundary, all the angles of the element should be away from 90° . Figure 3.2 shows an example of skinny and cap triangular elements.

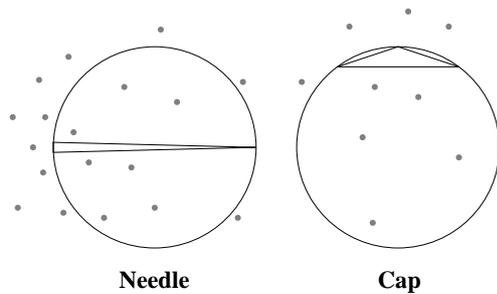


Figure 3.2: *Skinny triangles have circumcircles larger than their smallest edges. Each skinny triangle may be classified as a needle, whose longest edge is much longer than its shortest edge, or a cap, which has an angle close to 180° .*

In three-dimensions, tetrahedral elements can be ill-shaped in more ways than

their two-dimensional counterparts. Based on both *dihedral* and *solid angles* the ill-shaped tetrahedras can be classified into five types: needle, wedge, spindle, sliver, and cap (Bern et al., 1995). An example illustrating them is shown in Figure 3.3. A *needle* allows small solid angles, but not small or large dihedrals. A *wedge* allows small but not large dihedrals. A *spindle* allows large but not small dihedrals, and small but not large solid angles. A *sliver* allows large and small dihedrals, and small but not large solid angles. And a *cap* allows bad angles of all types including large solid angles. Note that a *sliver* or a *cap* can have all face angles bounded away from 0° and 180° , although the tetrahedron itself may have arbitrarily small solid angles and interior volume. An example is the sliver with vertex coordinates $(0, 0, 0)$, $(0, 1, \epsilon)$, $(1, 0, \epsilon)$, and $(1, 1, 0)$, where $\epsilon \rightarrow 0$.

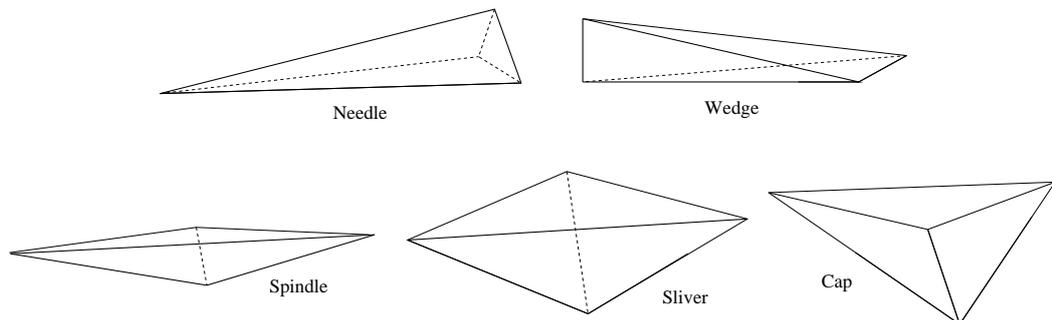


Figure 3.3: *The five types of bad tetrahedral elements.*

A wide range of element quality measures have been proposed in the computational geometry literature (Liu and Joe, 1994a; Shewchuk, 2002). Most of these quality metrics attain a maximum value for a regular simplex and a minimum value for a degenerate case. The most widely used element quality metric is the *radius-*

to-edge ratio, Θ , which is defined as the ratio of the circumradius to the shortest edge (Miller et al., 1995) and is shown in Figures 3.4(a) and 3.4(b) for a triangular and tetrahedral element, respectively. The *aspect ratio*, AR , is another element quality metric that is commonly used in the computational geometry literature (Bern et al., 1990; Ruppert, 1993). It is closely related to the radius-to-edge ratio and for a convex body it is defined as the ratio of the longest to the shortest width, which for a 2-D triangular element is equivalent to $AR = \frac{L}{H}$ (see Figure 3.4(c)).

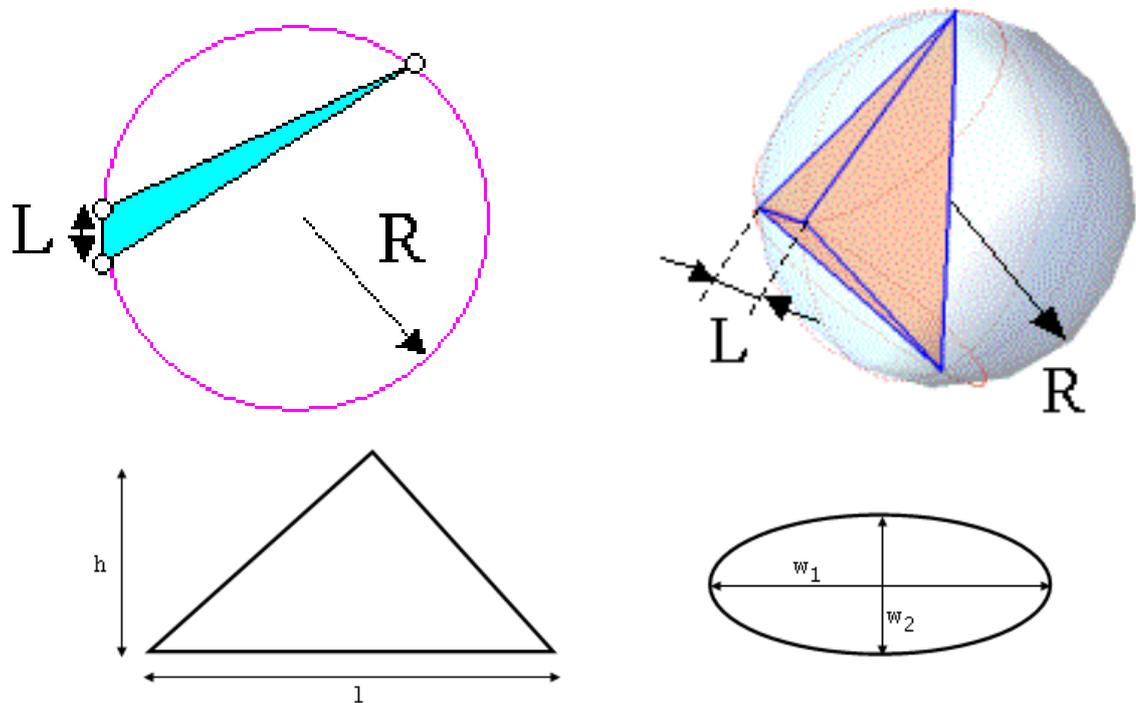


Figure 3.4: Illustration of radius-to-edge ratio defined for (a) triangular element in 2-D and (b) tetrahedral element in 3-D, and (c) definition of aspect-ratio for a triangle and a convex body.

The radius-to-edge ratio, $\Theta(\Delta_{ijk})$, for a triangle, Δ_{ijk} , in two dimensions is closely related to its sharpest angle α because $\alpha_{ikj} = \arcsin \frac{1}{2\Theta(\Delta_{ijk})}$ —the proof

follows immediately from the geometric construction shown in Figure 3.5. Hence, a bound on Θ will indirectly bound the minimum angle, and vice versa, and all two-dimensional quality guaranteed algorithms try to achieve a bound on either of these parameters. This is the main reason for the popularity of *Delaunay triangulation* for generating triangular unstructured mesh. Because among all the possible triangulations for a point set, Delaunay triangulation maximizes the minimum internal angle. However, in 3-D, the radius-to-edge ratio is considered to be a slightly weaker measure because it can not identify slivers since they are characterized by both small and large Θ values. Therefore, a bound on *Theta* for three-dimensional meshes does not guarantee the quality of unstructured tetrahedral mesh. Hence, *sliver exudation* is one of the important post-processing steps for unstructured mesh generation in three-dimensions (Baker, 1989).

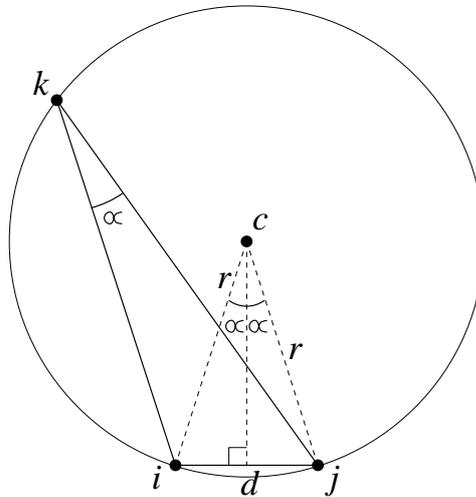


Figure 3.5: *Geometric construction to prove the relation between radius-to-edge ratio and the minimum internal angle for a two-dimensional triangular element.*

In general, elements with large radius-to-edge or aspect ratio have adverse

effects on the numerical solution by leading to ill-conditioned matrices (non-positive definite) that worsen the speed and accuracy of the solver. Speed degrades before accuracy; a triangular mesh with mild sharpest angle, say 10° , can be noticeably slower than a triangular mesh with a minimum angle of 45° . Moreover, even if the solver gives accurate answer, the large aspect ratio may result in interpolation error. As mentioned earlier, at times large aspect ratio (or skinny triangles) might be beneficial, especially when the problem being solved is anisotropic. The ideal aspect ratio of a triangle is the square root of the ratio of the largest to smallest eigenvalues of the *Hessian matrix* (Rippa, 1984). For triangular meshes, it does not make much of a difference whether long skinny elements have large angles or small angles, but if the aspect ratio exceeds the ideal ratio, the large angles will tend to be worse than small angles (Rippa, 1984).

The shape of the elements also affects another property of linear systems besides the condition number. A triangular mesh in \mathbb{R}^2 with well-shaped elements gives *symmetric M-matrix*, i.e., positive definite with negative off-diagonal entries for a finite element formulation of an equation with a Laplacian operator (∇). M-matrices are exactly those matrices that satisfy a discrete maximum principle; this desirable property rules out oscillations of the numerical method. In this case, *well-shaped* has two precise interpretations: (a) the sum of the two angles opposite to each interior edge of the mesh should be at most 180° (Babuska and Aziz, 1976). This requirement implies that no quadrilaterals are "reversed", so the triangulation must be a Delaunay or *constrained Delaunay triangulation* (CDT), and (b) depending on the boundary conditions associated with the differential equation, an M-matrix

may also require that the angle opposite to a boundary edge should measure at most 90° . In three dimensions, an unstructured tetrahedral mesh gives an M-matrix if and only if, for each edge e' in the mesh,

$$\sum_e |e| \cot \theta_e > 0 \tag{3.1}$$

where e denotes all the edges in the mesh that are opposite to e' in the tetrahedra of the FE mesh, $|e|$ denotes the length of e , and θ_e is the dihedral angle at e . All relation in equation 3.1 will be nonnegative if and only if all dihedrals in the mesh are non-obtuse.

3.2.3 Unstructured Mesh Generation Techniques

Unstructured meshes offer flexibility in fitting complicated domains, rapid grading from small to large elements, and relatively easy refinement and de-refinement. In this section, first a literature review is presented on unstructured mesh generation for PSLGs and convex polyhedra. The review on polygonal and polyhedral unstructured mesh generation includes four major approaches that have potential for application to image data, namely, conforming Delaunay triangulation, advancing front methods, constrained Delaunay triangulation, and hierarchical data-structures (quadtrees/octrees). This is followed by literature review on two mesh generation strategies developed specifically for directly handling domains in image data: energy minimization and interval volume tetrahedrization.

3.2.3.1 Conforming Delaunay Meshing

The first approach of unstructured mesh generation for a PSLG domain in 2-D, $\Omega(x_1, x_2)$, is achieved in two phases: placement of vertices followed by triangulation of the vertices. (The extra vertices, apart from the domain's original vertices that are added to the input domain in order to obtain quality FE meshes are called *Steiner points*). If the placement phase is smart enough, then the triangulation can be relatively straight forward by considering only the input vertices and the Steiner points and ignoring the edges.

In the placement phase, vertices are typically placed along the domain's boundary before adding them to the interior. In practice, the domain boundary is usually lined with enough Steiner points such that the Delaunay triangulation of all the vertices will *conform* to the domain. This requirement inspires a problem called *conforming Delaunay triangulation*, and is formally defined as: given a polygonal domain Ω , Steiner points are added such that each edge of Ω is an union of edges in the Delaunay triangulation. The algorithm by Saalfeld (1991) lines the edges of Ω with a large number of Steiner points uniformly spaced except near the endpoints. A more efficient solution (Nackmann and Srinivasan, 1991) covers the edges of Ω by disks that do not overlap other edges. Edelsbrunner and Tan (1993) gave the best theoretically robust algorithm that uses $O(n^3)$ Steiner points for an n -vertex multiple domain along with an $O(n^2)$ lower bound example.

There are several approaches for placing the Steiner points in the interior. One approach (Mavriplis, 1991) combines the vertices from a number of structured

meshes. A second approach (Baker, 1989; Barth and Jespersen., 1989) adds Steiner points in successive layers, working inwards from the domain boundary as in advancing front mesh generation. A third approach (Weatherhill and Hassan, 1994) chooses interior points at random according to some distribution, which may be interpolated from a coarse quadtree or "background" triangulation. An independent random sample is, however, likely to produce ill shaped triangles (Bern et al., 1991). Miller et al. (1995) proposed a technique that over samples and filters out points too close to previously chosen points to alleviate this problem. Deterministic methods achieve the same by defining birth and death rules that depend upon the density of neighboring points (Bossen, 1996; Shimada and Gossard, 1992).

All of these methods can also handle material and domain anisotropy. The first and second approaches, structured sub-meshes and advancing front, offer local control of element shapes and orientations. These approaches may space points improperly where structured meshes or advancing fronts collide, but this flaw can usually be corrected by filtering points and later smoothing the mesh. The third and fourth approaches trade direct control over element shapes for ease of fitting complicated geometries. Nevertheless, one can achieve anisotropy with these approaches by computing the Delaunay triangulation within a stretched space (Bossen, 1996; Castro-Diaz et al., 1995). For example, Bossen (1996) uses a "background" triangulation to define local affine transformations; Delaunay flips (described below) are then made with respect to transformed circles. Stretched Delaunay triangulations have many more large angles than ordinary Delaunay triangulations, but this should not pose a problem unless the stretching exceeds the desired amount.

The triangulation phase uses the well-known geometric structure called *Delaunay triangulation*, \mathbf{DT} , first proposed by Delaunay in 1934. For any given vertex set, $\mathbf{V} = \{v_1, v_2, \dots, v_n\} \forall v_i \in \mathbb{R}^2$, $\mathbf{DT}(\mathbf{V})$ is defined as a *graph* that satisfies *empty circle condition*. That is, a triangle T_i with vertices $\{v_a, v_b, v_c\} \forall v_i \subset \mathbf{V}$ appears in $\mathbf{DT}(\mathbf{V})$, if and only if the circumcircle of T_i encloses no other points of the vertex set $\mathbf{V}' = \mathbf{V} - \{v_a, v_b, v_c\}$. Figure 3.6 shows an example of empty circle condition.

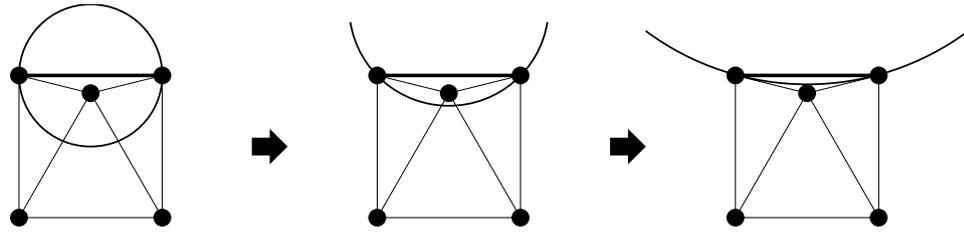


Figure 3.6: *Every triangle in a Delaunay triangulation satisfies the empty-circle-condition.*

The Delaunay triangulation of vertex set \mathbf{V} with the vertices at normal position gives a *unique triangulation* (\mathbf{T}), such that the vertices of \mathbf{T} collectively represent the input vertices, \mathbf{V} . Figure 3.7 shows Delaunay triangulation of a 2-D vertex set. Other interesting properties of the Delaunay construction include:

1. the edges in a Delaunay mesh cannot intersect each other—i.e., edges of a Delaunay mesh, \mathbf{T} , can only intersect each other at the triangle vertices (refer to Figure 3.8(a)).
2. If e is an edge of a triangulation of \mathbf{V} , then either e is locally Delaunay, or e is flippable and the edge created by flipping e is locally Delaunay (refer to Figures 3.8(b) and 3.8(c)),

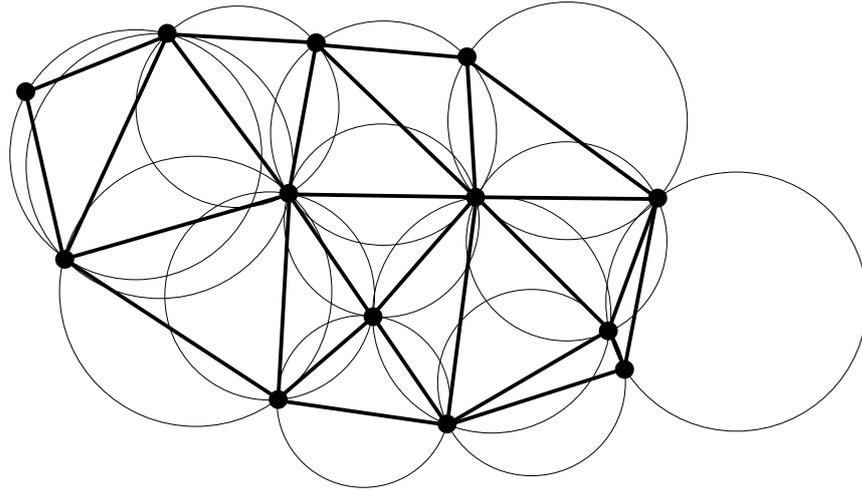


Figure 3.7: *Illustration of Delaunay triangulation of a vertex set with empty circle for each triangle.*

3. in a Delaunay triangulation \mathbf{T} all the edges of are Delaunay, and vice versa (refer to 3.8(d) and 3.8(e)),
4. the boundary formed by taking the union of all the triangles in a Delaunay mesh, \mathbf{T} , represents the *convex hull* of the input domain (\mathbf{V}),
5. Delaunay triangulation maximizes the minimum internal angle, and
6. Delaunay mesh is a dual of *Voronoi graph*.

However, there is an exception when four or more points in the vertex set are in special position or *co-planar*. The circumcircle of a triangle, whose vertices belong to the co-planar points, will pass through all the co-planar points, thus violating the empty-circle condition. In such cases, one may choose to complete the triangulation arbitrarily. Usually, the triangles that fall outside the domain Ω are discarded so that $\mathbf{DT}(\mathbf{V})$ results in a triangulation whose edges will form the convex hull of \mathbf{V} .

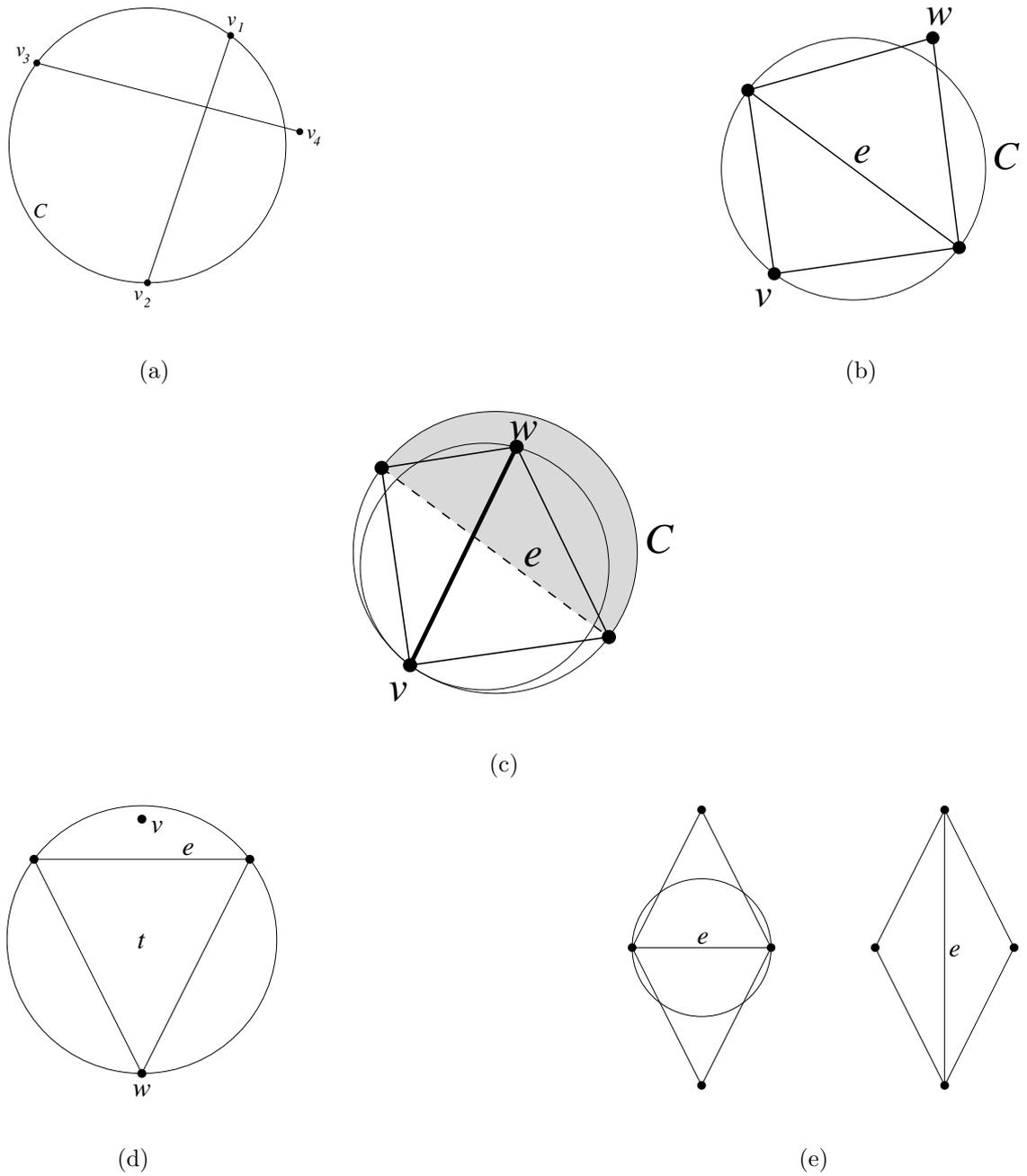


Figure 3.8: (a) If no four vertices are co-circular, two crossing edges cannot both be Delaunay, (b) case where e is locally Delaunay, and (c) case where e is not locally Delaunay. The edge created if e is flipped is locally Delaunay, (d) If the triangle t is not Delaunay, then at least one of its edges (in this case, e) is not Delaunay, and (e) Two triangulations of a vertex set: at left, e is locally Delaunay; at right, e is not.

Figure 3.9 gives an example of Delaunay triangulation for co-planar vertex set along with possible alternatives to complete the triangulation.

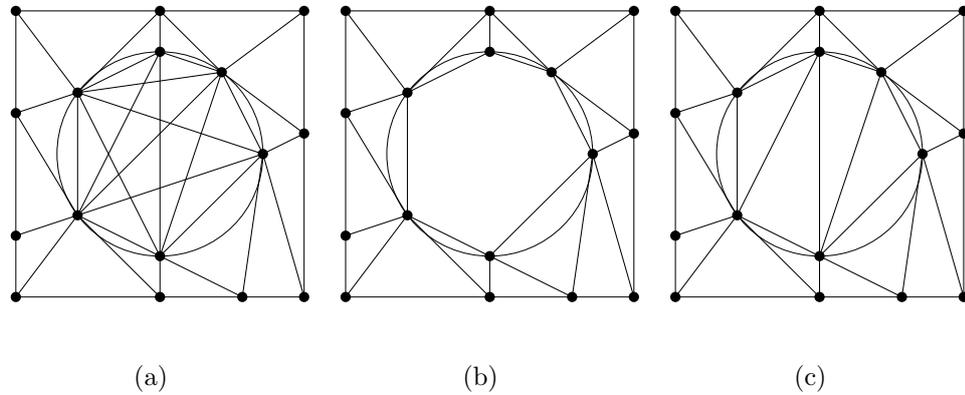


Figure 3.9: *Three ways to define the Delaunay diagram in the presence of co-circular vertices: (a) include all Delaunay edges, even if they cross, (b) exclude all crossing Delaunay edges, and (c) a subset of Delaunay edges that forms a triangulation is selected.*

There are number of Delaunay triangulation algorithms (Fortune, 1993, 1997). The following are some of the more well-known approaches: edge flipping (Sibson, 1978); incremental insertion (Bowyer, 1981; Guibas and Stolfi, 1985; Joe, 1993; Lee and Schachter, 1980); and randomized incremental insertion (Chew, 1986; Clarkson and Shor, 1989; Guibas et al., 1992; Seidel, 1993); divide-and-conquer (Chew, 1989; Guibas and Stolfi, 1985; Shamos and Hoey, 1975); alternating divide-and-conquer (Dwyer, 1987); sweep-line (Fortune, 1987); regular grid and sparse matrix (Fang and Piegl, 1992, 1993). Of these methods, some are theoretically elegant and others are relatively easy to implement. The following paragraph described *edge flipping* algorithm because of its relevance to subsequent discussions.

The edge flipping algorithm has a worst-case running time of $O(n^2)$. In spite

of being sub-optimal it performs quite well in practice. The edge flipping algorithm starts from any triangulation of V and then locally optimizes each edge. Let e be an internal (non-convex-hull) edge and Q_e be the triangulated quadrilateral formed by the triangles sharing e . Quadrilateral Q_e is *reversed* if the two angles without the diagonal total more than 180° , or equivalently, if each triangle circumcircle contains the opposite vertex as in Figure 3.10. If Q_e is reversed, it is "flipped" by exchanging e for the other diagonal. An initial triangulation can be computed by a sweep-line algorithm (Fortune, 1987). This algorithm adds the points of S by x -coordinate order. Upon each addition, the algorithm walks around the convex hull of the already-added points starting from the rightmost previous point and adding edges until the slope reverses as shown in Figure 3.11. The theorem given by Delaunay guarantees the success of edge flipping triangulation algorithm (i.e., a triangulation in which no quadrilateral is reversed must be a completion of the Delaunay triangulation). The point placement followed by Delaunay triangulation is a popular approach to mesh generation, especially in aerodynamics. The same point placement methods work fairly well: combining structured meshes, advancing front (Baker, 1989; Lo, 1991a,b), and random scattering with filtering (Weatherhill and Hassan, 1994).

Similarly, the placement phase for meshing the 3-D polyhedral domains must put sufficient number of points on the domain boundary to satisfy the conformity requirement of Delaunay tetrahedralization. The first two point placement methods, discussed for the two-dimensional case, suffer from the same problems when extended to 3-D domains, i.e., points may be improperly spaced at junctures between fronts or

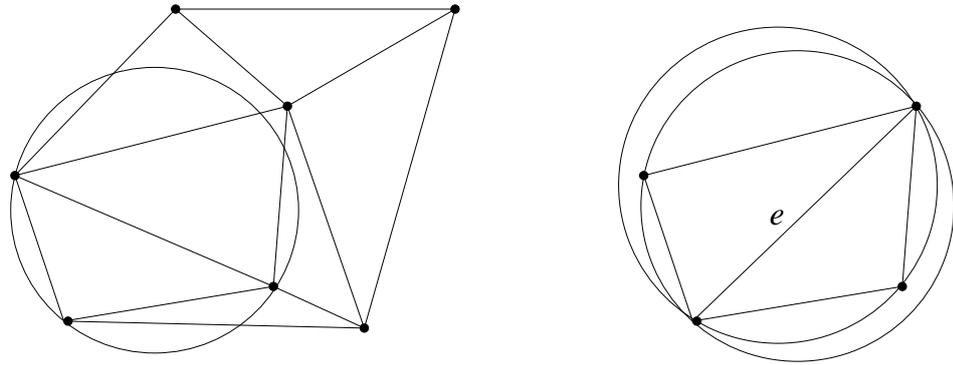


Figure 3.10: On the left is a Delaunay triangulation of vertex set and the right-side figure shows reversed/flipped quadrilateral along the edge e .

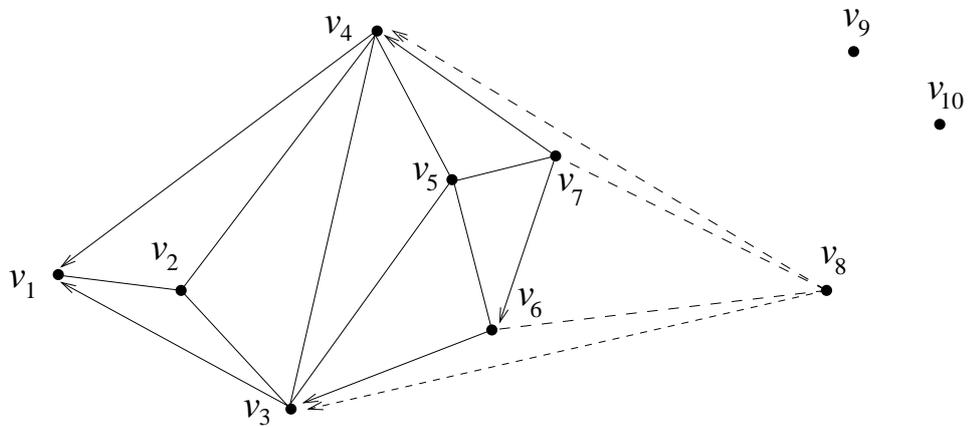


Figure 3.11: A Sweep-line algorithm for computing an initial triangulation of a vertex set.

patches. Furthermore, a new problem is encountered while extending all the three 2-D methods to 3-D even for well spaced point set because of the possible appearance of slivers in the final Delaunay tetrahedralization. For this reason, some Delaunay mesh generators include a special post-processing step that finds and removes slivers (Baker, 1989).

The triangulation phase of mesh generation is more involved for three dimensional domains. The generalization of edge flipping exchanges the two possible triangulations of five points in convex position, as shown in Figure 3.12. A flip is called a Delaunay flip if, after the flip the triangulation of the five points satisfies the empty sphere condition, that is, no circumsphere encloses a point. In \mathbb{R}^3 , it is no longer true that any tetrahedralization can be transformed into the Delaunay tetrahedralization by a sequence of Delaunay flips (Joe, 1989). Currently, it is unknown whether any tetrahedralization can be transformed into the Delaunay triangulation by arbitrary flips. Nevertheless, there are verifiably correct incremental Delaunay tetrahedralization algorithms based on edge flipping (Edelsbrunner and Shah, 1996; Joe, 1991; Rajan, 1994).

There are other practical three-dimensional Delaunay triangulation algorithms as well. Bowyer (1981) and Watson (1981) proposed incremental algorithms with reasonable *expected case* performance. Barber et al. (1996) implemented a randomized algorithm in arbitrary dimensions. This algorithm can be used to compute Delaunay triangulations through a well-known reduction given by Brown (1979) which *lifts* the Delaunay triangulation of points in \mathbb{R}^d to the convex hull of points in \mathbb{R}^{d+1} .

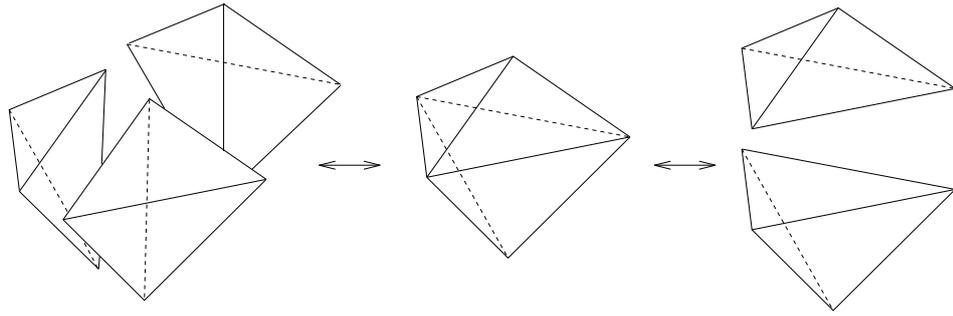


Figure 3.12: *In three dimensions, an edge flip exchanges three tetrahedra sharing an edge for two tetrahedra sharing a triangle, or vice versa.*

3.2.3.2 Advancing Front Meshing

Delaunay triangulation by itself does not generate a satisfactory mesh because of these two reasons: (1) elements of poor quality may appear in the output FE mesh and (2) input boundaries may fail to appear in the final mesh. The former problem is typically treated by adding additional vertices at either the circumcenters or the centroids of the poor quality elements. An alternative approach to address this issue is to use the advancing front approach.

Advancing front methods in 2-D begins by dividing the boundaries of the mesh into *edges*. These discretized boundaries form the initial *front*. Triangles are then generated sequentially, starting from the boundary edges or faces and working toward the center of the domain. The exposed inner faces of these elements collectively form an *advancing front*. In general, advancing front techniques requires a good deal of second-guessing. Firstly one has to ensure that the initial division of the boundaries is prudent. Later on, when the advancing fronts from the boundaries collide at the center of the domain, they have to be merged together in a manner that

does not compromise the quality of the elements. In spite of these drawbacks, advancing front methods typically create astonishingly good meshes near the domain boundaries. Mavriplis (1991) was the first to combine the Delaunay triangulation and advancing front methods. The combination makes a good deal of sense, because a Delaunay triangulation in the interior of the mesh is an useful search structure for determining how close different fronts are to each other. Some researchers have used background grids for this task. Conversely, the advancing front method may be used as a vertex placement method in Delaunay meshing. A sensible strategy might be to abandon the advancing front shortly before the fronts collide, and use a different vertex placement strategy (such as inserting vertices at circumcenters or centroids of poor quality elements) in the center of the mesh, where such strategies tend to be most effective. Figure 3.13 shows a finite element mesh generated by the advancing front method of Barth and Jespersen. (1989). The mesh is Delaunay triangulation of vertices placed along the advancing front moving outward from an airfoil. The problems associated with colliding fronts are reduced in circumstances like this, where one is meshing the exterior, rather than the interior, of an object.

As mentioned earlier, an advancing front approach places Steiner points for Delaunay triangulation. On contrary, pure advancing front mesh generators (Lo, 1985; Löhner and Parikh, 1988; Nguyen-Van-Phai, 1982) places the elements themselves, rather than just adding the Steiner points. This approach gives more direct control of element shapes, especially near the boundary, which is often a region of special interest. Löhner and Parikh (1988) based their approach on this idea by defining a desired element size (and perhaps stretching directions) at the vertices

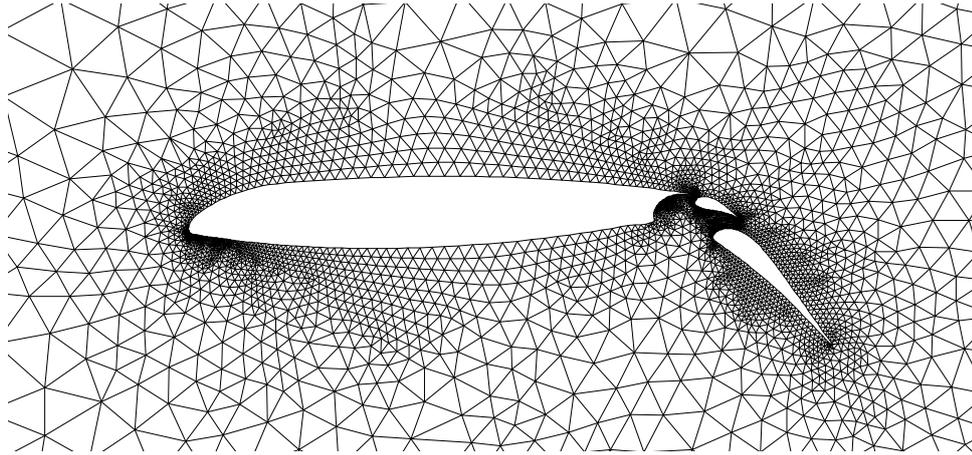


Figure 3.13: *A Sweep-line algorithm for computing an initial triangulation of vertex set.*

of a coarse "background" tetrahedron and interpolating them to the rest of the domain. Their methods starts by: (a) triangulating the boundaries of the domain; (2) constructing the initial front comprising the boundary faces; and (3) iteratively selecting a face of the front and building a tetrahedron over that face. The algorithm attempts to fill in clefts left by the last layer of tetrahedra before starting the next layer; within a layer, the algorithm chooses small faces first in order to minimize collisions. The fourth vertex of the tetrahedron will be either an already existing vertex or a vertex specially created for the tetrahedron. In the latter case, the algorithm tries to choose a smart location for the new vertex. For example, the new vertex can be placed along a normal to the base face at a distance determined by aspect ratios and length functions ultimately derived from the background triangulation (Frey et al., 1996). In either case, cleft or new vertex, the tetrahedron must be tested for collisions before final acceptance. Figure 3.14 shows the surface of a fairly isotropic tetrahedral mesh computed by an advancing front mesh gen-

erator. Marcum and Weatherill (1995) proposed an algorithm somewhere between pure advancing front and advancing front point placement followed by Delaunay triangulation. Their algorithm first starts with a coarse mesh and then uses advancing front to place additional Steiner points by subdividing the coarse tetrahedra to maintain a triangulation. The resulting mesh is then improved first by Delaunay and then by *minmax-solid-angle* flips.

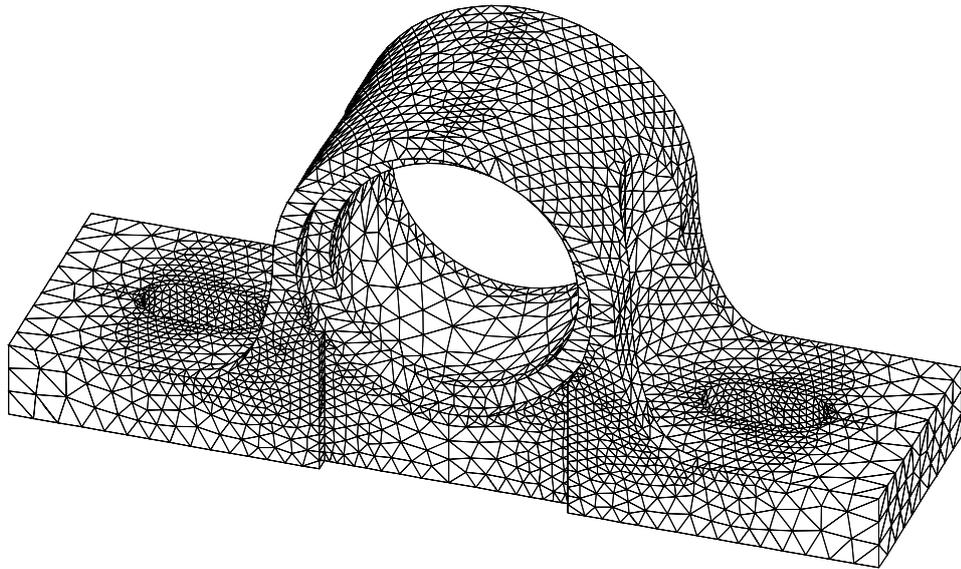


Figure 3.14: *The surface of a tetrahedral mesh computed by an advancing front generator.*

3.2.3.3 Constrained Delaunay Meshing

The *constrained Delaunay triangulation* is an alternative to discretize the polygonal domains into simplices. The constrained Delaunay triangulation of a (possibly multiple) domain Ω does not use Steiner points, but instead redefines Delaunay triangulation in order to force the edges of Ω into the final triangulation by using the

concept of *vertex-visibility*. A point p is visible to a point q in Ω if the open line segment \overrightarrow{pq} lies within Ω and does not intersect any edges or vertices of Ω . The constrained Delaunay triangulation, $\mathbf{CDT}(\Omega)$ contains each triangle not cut by an edge and satisfies an empty circumcircle condition. (Here, "empty" means that the circle does not contain any vertices visible to points inside the triangle). This visibility requirement ensures that the external proximities, where Ω wraps around to nearly touch itself, have no effect. Figure 3.15 illustrates such an example; here vertex v is not visible to any point in the interior of $\triangle abc$. The constrained De-

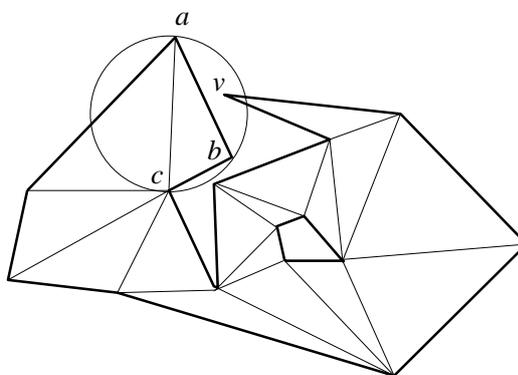


Figure 3.15: *The constrained Delaunay triangulation of a polygon with holes.*

launay triangulation can be computed by generalizing the edge flipping Delaunay algorithm. Only this time, edges of the Ω are not stored in a queue. One major problem associated with this approach for polygonal domains is obtaining an initial triangulation. (Note that discretizing polygonal domains is relatively more complex than vertex sets). The textbook by Preparata and Shamos (1990) describes an $O(n \log n)$ in-time algorithm for the initial triangulation by adding new edges to Ω and then subdividing them into easy-to-triangulate "monotone" faces.

The best approach for constrained Delaunay triangulation was first proposed by Ruppert (1993). Ruppert's algorithm, building up on the work by Chew (1989), computes $CDT(\Omega)$ at the outset and then adds Steiner points to improve the mesh, thus uniting the two phases of the Delaunay triangulation approach described earlier. In choosing this approach, the user gives up control over point placement, but obtains a more efficient mesh with fewer and "rounder" elements. The Delaunay refinement algorithm was subsequently sharpened by Mitchell (1994) and Shewchuk (1996a,b) further refined it.

The first step of Ruppert's Delaunay refinement mesh generator cuts off all vertices of the domain Ω at which the interior angle measures less than 45° . The cutting line at such a vertex v should not introduce a new small feature to Ω ; it is best to cut off an isosceles triangle whose base is about halfway from v to its closest visible neighbor. If v has degree greater than two, as might be the case in a multiple domain, then the bases of the isosceles triangles around v should match up so that no isosceles triangle receives a Steiner point on one of its legs. Next the algorithm computes the constrained Delaunay triangulation of the modified domain. The algorithm then goes through the loop given below. The last line of the loop repairs a constrained Delaunay triangulation after the addition of a new Steiner point c . To accomplish this step, there is no need to recompute the entire triangulation. The removed old triangles are exactly those with circumcircles containing c , which can be found by searching outwards from the triangle that contains c , and the new triangles that replace the removed triangles must all be incident to the new vertex c . The loop is guaranteed to halt when all angles are larger than 20° . At this point,

the cutoff isosceles triangles are returned to the domain, and the mesh is complete. Ruppert's algorithm comes with a strong theoretical guarantee: all new angles, that is, angles not present in the input, are greater than 20° , and the total number of triangles in the mesh is at most a constant times the minimum number of triangles in any such *no-small-angle* mesh. Ruppert proved the efficiency by showing that each triangle in the final mesh is within a constant factor of the *local feature size* at its vertices $(lsf_{p \in \Omega})^1$. Figures 3.17(a)–3.17(b) show the mesh computed by Chew's constrained Delaunay algorithm and by Ruppert's Delaunay refinement algorithm, respectively.

¹The $lsf_{p \in \Omega}$ is defined as the radius of the smallest circle centered at p that touches two non-adjacent edges of the boundary; this is a spacing function intrinsic to the domain.

Pseudocode 3.2.1: CDT(V)**comment:** Constrained Delaunay triangulation for vertices V **comment:** Compute Delaunay triangulation of vertex set V $T \leftarrow \text{DT}(V)$ **while** there exists a triangle $t \in T$ with any angle of $\Delta t < 20^\circ$

do	{	$c \leftarrow \text{CIRCUMCENTER}(t)$ if c is within diametric semicircle of a boundary edge, e <table style="border: none; border-collapse: collapse; margin-left: 20px;"> <tr> <td style="vertical-align: middle; padding-right: 10px;">then</td> <td style="font-size: 3em; vertical-align: middle;">{</td> <td style="padding-left: 10px;"> comment: add the midpoint (m) of e to V $m \leftarrow \text{MIDPOINT}(e)$ $V \leftarrow \text{PUSH}(m)$ </td> </tr> <tr> <td style="vertical-align: middle; padding-right: 10px;">else</td> <td style="font-size: 3em; vertical-align: middle;">{</td> <td style="padding-left: 10px;"> comment: add c to V $V \leftarrow \text{PUSH}(c)$ </td> </tr> </table>	then	{	comment: add the midpoint (m) of e to V $m \leftarrow \text{MIDPOINT}(e)$ $V \leftarrow \text{PUSH}(m)$	else	{	comment: add c to V $V \leftarrow \text{PUSH}(c)$
then	{	comment: add the midpoint (m) of e to V $m \leftarrow \text{MIDPOINT}(e)$ $V \leftarrow \text{PUSH}(m)$						
else	{	comment: add c to V $V \leftarrow \text{PUSH}(c)$						
		comment: recompute the CDT of V $\text{CDT}(V)$						

return (T)

3.2.3.4 Hierarchical Meshing

The last type of unstructured mesh generation technique for PSLGs uses *hierarchical data-structures*. A quadtree mesh generator (Baehmann et al., 1987; Bern et al., 1990; Yerry and Shephard, 1983) starts by enclosing the entire domain Ω inside an axis-aligned square ($2^n \times 2^n$ dimension). It splits this root square into four congruent

squares, and continues this recursive process of splitting until each minimal or *leaf* square intersects Ω in a simple way. Further splits may be dictated by an user-defined spacing function or balancing condition. Quadtree squares are then wrapped and cut to conform to the boundary. A final triangulation step then gives the required triangulation of the domain Ω . Figure 3.16 shows the quadtree decomposition and quadtree-based triangulation for an input vertex set.

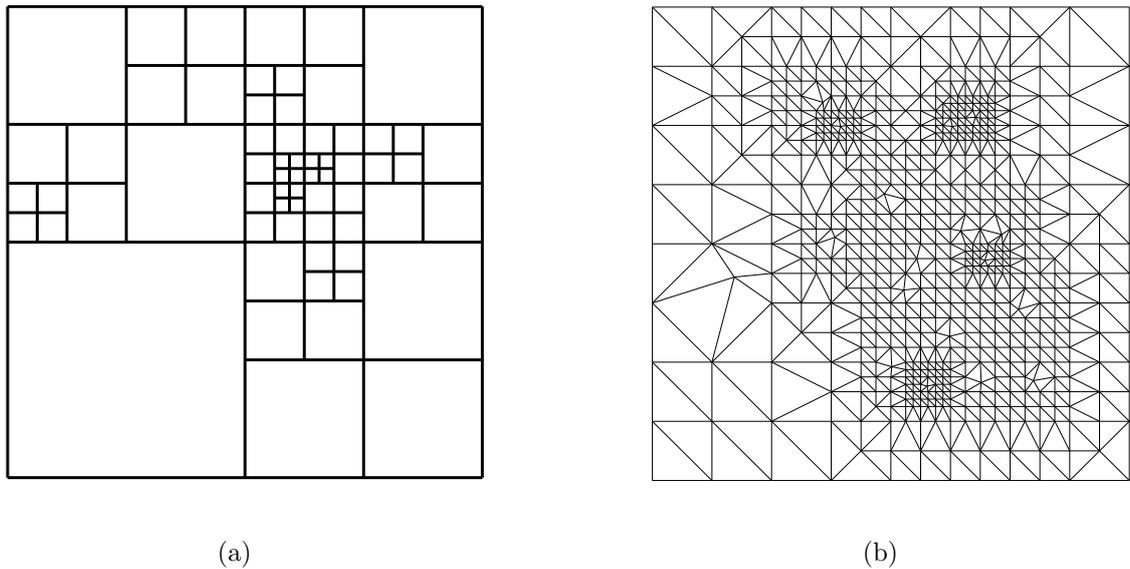


Figure 3.16: (a) A quadtree decomposition, and (b) A quadtree-based triangulation of a vertex set.

The quadtree mesh generation due to Bern et al. (1990) was the first triangulation technique that guaranteed theoretical size optimality and bounded element quality. As first presented, the algorithm assumes Ω to be a polygonal domain with holes; however, the approach can multiple and even to curved domains. Bern et al.'s provably quadtree-based mesh generation algorithm splits squares until each leaf square contains at most one connected component of the domain's boundary, with

at most one vertex. The algorithm then splits the squares near vertices two more times, so that each vertex lies within the buffer zone of equal-sized squares. This approach handles curved domains more gracefully than the Delaunay refinement algorithm approach (Ruppert, 1993), because the splitting phase can automatically adapt to the curvature of the domain.

Next, the mesh generator imposes a balance condition: no square should be adjacent to one less than one-half its size. This causes more splits to propagate across quadtree, increasing the total number of leaf squares by a constant factor of at most eight (Moore, 1995). Squares are then *wrapped* to conform to the domain. Various wrapping rules work, pseudocode 3.2.2 is one such possibility for a square b with edge length $|b|$.

Pseudocode 3.2.2: PROVABLEQUADMESH(Ω)

comment: Compute unstructured mesh for Ω using quadtree

for each vertex v of Ω

do $\left\{ \begin{array}{l} \text{let } y \text{ be the closest quadtree vertex to } v \\ \text{move } y \text{ to } v \end{array} \right.$

for each leaf nodes b still crossed by an edge e

do $\left\{ \begin{array}{l} \text{comment: 2:1 balancing of quadtree} \\ \text{move vertices of } b < \frac{|b|}{4}(e) \text{ to their closest points on } e \end{array} \right.$

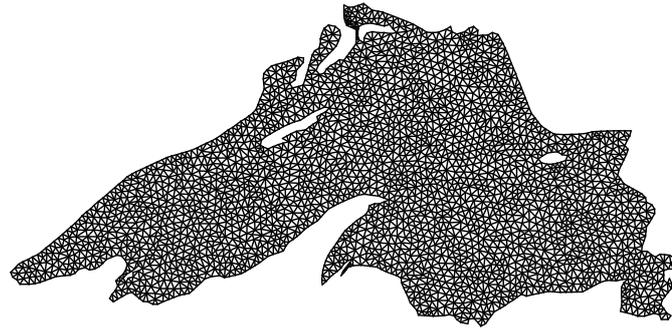
discard faces of the warped quadtree that lie outside Ω

Lastly, the cells of the wrapped quadtree are triangulated so that all angles are bounded away from 0° . Every triangle in the mesh generated will have an aspect

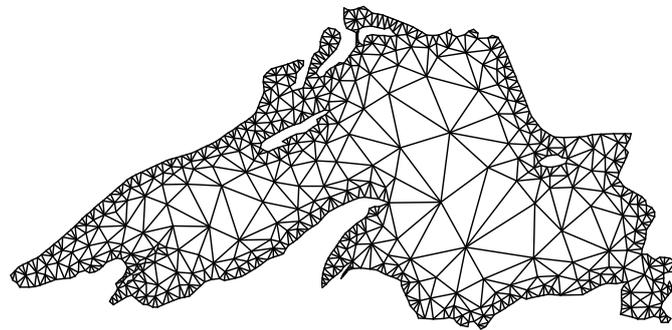
ratio greater than 4 and the number of triangles produced will be a constant factor of *optimal*–minimum number of triangles in any triangulation of the given input achieving the same aspect ratio bound. Figure 3.17(c) shows the triangular mesh from Bern et al.’s provably good mesh generation strategy for the lake Superior.

Mitchell and Vavasis (1992) improved the splitting phase by ”cloning” the square that intersects the domain in more than one connected component, so that each copy contains only a single connected component of the domain. The FE mesh from the quadtree-based algorithms exhibits preferred direction by aligning itself along the horizontal and vertical axis. If this poses a problem, the mesh improvement steps can be used to redistribute element orientation. Neugebauer and Dickmann (1996) proposed a solution to this element alignment by replacing the squares of the quadtree with rhomboid quadtrees so that the triangles in the final mesh tend to be nearly equilateral. They were able to prove that polygonal domains with polygonal holes and isolated interior points can be triangulated with all the angles between 30° and 90° –assuming there are no small input angle.

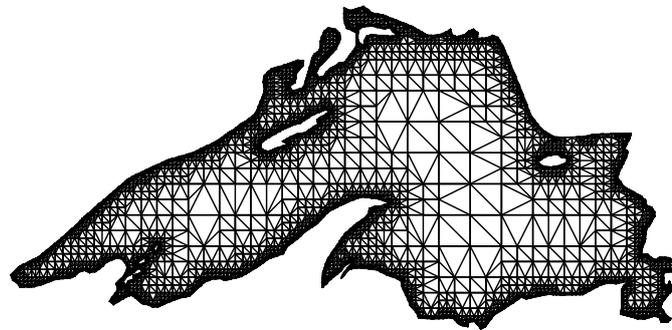
In three-dimensions, the hierarchical meshing techniques use octree data structure, which is a three-dimensional generalization of quadtree. In \mathbb{R}^3 , hierarchical data-structure based meshing strategies start by bounding the input domain Ω within a cube and then recursively splitting it into eight congruent cubes until each minimal cube intersects Ω in a simple way. As in 2-D, a balance condition ensures that no cube has neighbors less than one-half its size. The balance condition need not be explicit, but rather it may be a consequence of an intrinsic local spacing function (Vavasis, 2002).



(a)



(b)



(c)

Figure 3.17: Meshes generated for the Lake Superior as PSLG input by the: (a) Chews first constrained Delaunay algorithm, (b) Rupperts Delaunay refinement algorithm, and (c) Bern-Eppstein-Gilbert quadtree-based algorithm.

Shephard and collaborators (Schroeder and Shephard, 1990; Shephard and Georges, 1991; Yerry and Shephard, 1984, 1991) proposed *octree-based mesh generators* for polyhedral domains. The original octree-based FE mesh generator (Yerry and Shephard, 1984) tetrahedralizes leaf nodes of the octree using a collection of predefined patterns, *dictionary-method*. To keep the number of patterns manageable, the generator makes the simplifying assumption that each cube is cut by at most three facets of the input polyhedron. Figure 3.18 shows the surface mesh for a polyhedron derived from octree-based mesh generator (Yerry and Shephard, 1984). Perucchio et al. (1989) enhanced this method by proposing a more sophisticated way to conform to boundaries. Buratynski (1990) further sharpened this technique by using rectangular octrees and a new hierarchical set of wrapping rules. In all these methods, the octree is refined so that each domain edge intersects boxes of only one size. Boxes are warped to domain vertices, then edges, and finally faces. However, these techniques focused primarily on automating the mesh generation process and did not investigate the optimality and quality of the meshes.

Mitchell and Vavasis (1992) extended the 2-D mesh generator of Bern et al. (1990) to tetrahedralize a non-convex bounded polyhedral domain with holes. The generalization is not straightforward, because vertices of polyhedra may have very complicated local neighborhoods. Nevertheless, the algorithm given by Mitchell and Vavasis is guaranteed to avoid all five types of bad tetrahedra, while producing a mesh with only a constant times the minimum number of tetrahedra in any such bounded aspect ratio tetrahedralization. This optimality condition is slightly stronger than the bound proposed by Bern et al.. Currently, this is the

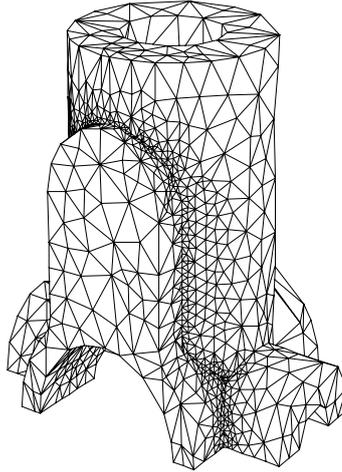


Figure 3.18: *The surface of a tetrahedral mesh derived from an octree-based mesh generator.*

only known three-dimensional mesh generation algorithm with strong theoretical guarantee. Vavasis (2002) released a modified version of this algorithm called QMG for "Quality Mesh Generator", including a simple geometric modeler and equation solver for performing FEA. The modified algorithm includes a more systematic set of wrapping rules, in particular, the new wrapping method for an octree cube cut by a single facet generalizes to any fixed dimension (Mitchell and Vavasis, 1996). Although, provably good quadtree meshing has been extended to polyhedra of arbitrary dimensionality, its generalization to more than 2-D is quite intricate and the theoretical bounds on element quality are not strong enough to be entirely satisfactory for practical applications.

3.2.3.5 Energy Minimization Image Meshing

Hale (2001) and Langer et al. (2001), instead of working with segmented image

volumes, independently proposed mesh generation techniques for working directly with raw image data. The "integrated image analysis" method proposed by Hale (2001), finds boundaries of regions within images (i.e. segmentation) and then generates finite element meshes for these regions by directly constructing a mesh that is aligned with the boundaries. The *space-filling* mesh is computed in three steps: (1)fill the space spanned by the image with a pseudo-regular lattice of atoms (vertices), where pseudo-regular means that the nominal distance between an vertex and its nearest neighbors varies smoothly and consistently with the density of the features in the image; (2)move the atoms to minimize a total potential energy, defined to be a weighted sum of an atomic potential energy and an image potential energy; and (3)connect the atoms at the least potential energy configuration using Delaunay (or some other) triangulation to form a mesh. Hale applied his mesh generation technique on 2-D and 3-D seismic image data and 3-D MR scans.

The approach proposed by Langer et al. (2001) minimizes a function E , defined as the sum of two terms for each triangular element:

$$E = \alpha E_{hom} + (1 - \alpha) E_{shape},$$

where $\alpha \in [0, 1]$ is an user tunable parameter and E_{hom} and E_{shape} are functions that depend on the element's homogeneity and shape, respectively. E attains a minima for equilateral triangles overlying a homogeneous set of pixels in the image. Thus, at any given level of refinement, the edges of the triangle approximate the interfaces in the material image as well as the length scale of the triangles. The user decides the stopping criteria for the refinement or *annealing process*. Image segmentation is

a byproduct and not an input for this approach because the homogeneity part of E automatically finds the mathematical representation of the material boundaries.

3.2.3.6 Interval Volume Tetrahedralization

The second category of image meshing techniques generate meshes for the *interval volume* bounded between the two iso-contours (Nielson and Sung, 1997). Mathematically, it can be expressed as the volume bounded between two surface Ω_a and Ω_b , where $\Omega_i = \alpha_i \leq f(x_1, x_2, x_3) \leq \beta_i$. All these methods generate unstructured meshes in two steps: (1) generate the iso-surfaces bounding the the interval volume along with the triangular surface meshes for the two iso-surfaces; and (2) generate tetrahedral mesh for the interval volume that also includes the triangular elements from the iso-surface.

The first step involving the polyhedral approximation of the surface(s) from the interval volume is generated using the *marching cubes* method Lorensen and Cline (1987). Although there exists several other techniques (like, polynomial approximation, snakes, and level set methods), marching cube method is widely used because of its multi-resolution capabilities for 3-D surface construction. The marching cubes algorithm for surface fitting is achieved by determining polygonal patch patterns in a cube-by-cube manner. Wilhelms and Geldren (1992) enhanced the original marching cubes approach by using the octree-based hierarchical data-structure. Octrees were used primarily to minimize the expensive step of visiting all the cells/voxels in the volume data. Shekhar et al. (1996) suggested an improvement to the octree-based marching cube technique by using surface tracking mechanism along with octrees

for faster surface reconstruction. The output of all these polyhedral approximation methods for input volume is a surface representation with a sub-pixel resolution. That is, while the volume data is an $O(n^3)$ volumetric space, the relevant surface model is an $O(kn^2)$ polygon set. Other approaches for surface reconstruction, from a topology stand-point, is to model the surface geometry by using mathematical functions with continuity constraints. For instance, using spline curves/surfaces (Kass et al., 1987; Shirley and Tuchman, 1990), wavelets (Gross et al., 1996), and simplicial complexes (Cignoni et al., 1994; Zhou et al., 1997). The last category of surface approximation is *Ray casting* (Garrity, 1990), which is used extensively in visualization applications but is seldom used in interval volume methods (Nielson, 1997).

The second step in all the interval volume tetrahedralization methods is to define a local mesh procedure for generating voxel-by-voxel tetrahedral elements. Frey et al. (1994) were the first to propose such a local meshing procedure, which involved two main phases for tetrahedral construction of voxels. First, the voxel is scanned to locate the intersection points along the edges (refer to Figure 1 Frey et al., 1994; Fujishiro et al., 1996, refer to Figure 6), then tetrahedrons are generated using *voxel configuration rules* based on *characteristic number*. The characteristic number of a voxel v_i of Ω_i , noted Cv_i , represents the number of pixels which belong to the class c_i (see Figures 5–7 Frey et al., 1994; Nielson and Sung, 1997, Figure 3). Their approach also accounts for mesh adaptivity, smoothing, refining and de-refining. Similar approaches, though independently, were proposed by Fujishiro et al. (1996) and Nielson and Sung (1997). The algorithm of Fujishiro et al. uses the concept

of α - *shapes* (Edelsbrunner and Mücke, 1994) and computes for each voxel the intersection of the two convex polyhedral surfaces adjusted for the ambiguous case (Fujishiro et al., 1996, see Figure 7). On the contrary, Nielson and Sung's algorithm was primarily developed for applications where the data varies linearly over the 3-D rectilinear grid. Neither of these techniques investigated the quality of the output finite element mesh.

Bajaj et al. (1999) proposed a method based on tetrahedrization of each individual prismatoid by 3-D Delaunay triangulation using advancing front method. In the first phase, their method uses a *reducing algorithm* to sub-divide the polyhedral domain into one or more prismatoids (Bajaj et al., 1996). In the second phase, the algorithm triangulates the top and the bottom iso-surfaces using 2-D constrained Delaunay triangulations and then splits the prismatoid with Steiner points into smaller prismatoids not containing any Steiner points. The implementation of their approach has an $O(n^2)$ time complexity, where n is the number of vertices (Lin, 1997). This method accounts for characterization, prevention, and post-processing of untetrahedralizable parts of the interval volume.

Zhang et al. (2003) extended the idea of dual contouring to interval volume tetrahedralization and hexahedralization for volumetric Hermite data. The dual contouring analyzes those edges that have their end points lying on different sides of the iso-surface, called *sign change edge* (Ju et al., 2002). Each sign change edge is shared by four (uniform case) or three (adaptive case) cells, and one minimizer is calculated for each of them by minimizing a predefined Quadratic Error Function

(QEF), which is formally defined as: $QEF(x) = \sum_i (n_i \cdot (x - p_i))$, where p_i and n_i represent the position and unit normal vectors of the intersection point, respectively. For each sign change edge belonging to boundary cell, a quad or a triangle is constructed by connecting the minimizers. These quads and triangles provide an approximation of the iso-surface. Hence, the approach generates an unstructured mesh that efficiently adapts to the interval volume. (Note that, especially for uniform grids, it is trivial to deal with the interior cells, as each cell/voxel can be split into into five/six tetrahedras.) This approach guarantees that the resulting three dimensional mesh is topologically equivalent to the real interval volume. In addition, the approach also uses the edge-ratio and Liu-Joe parameter (Liu and Joe, 1994b) to ensure mesh quality. Tetrahedras with bad edge-ratio and the overall mesh quality improvement with respect to Liu-Joe parameter are achieved by using the edge contraction and the smoothing-based methods, respectively, (Teng and Wong, 2000).

Wang et al. (2004) proposed *sphere carving* algorithm for the problem of interval volume tetrahedralization. The input for this algorithm also uses segmented image volume but instead it builds a tetrahedral mesh by maintaining a *genus zero surface*. The resulting tetrahedralized 3-D volume is guaranteed to have genus² zero

²Genus is a topologically invariant property of a surface defined as the largest number of non-intersecting simple closed curves that can be drawn on the surface without separating it. Roughly speaking, it is the number of *holes* in a surface. The genus of a surface, also called the geometric genus, is related to the Euler-Poincaré characteristic ϖ , by, $\varpi \equiv V - E + F = \varpi(g)$, where $\varpi(g) = 2 - 2g$ (Coxeter, 1973).

even when the segmented volume contains holes due to enclosed structures (like, ventricular or non-white matter in brain MR images) or image noise.

3.2.4 Summary

The vast majority of research on quality guaranteed unstructured mesh generation techniques has been targeted towards polygonal/polyhedral domains rather than images. The most efficient approach for generating quality-guaranteed unstructured meshes for PSLGs was found to be the method of (Bern et al., 1990) that combines a hierarchical space decomposition for triangulations.

Methods to discretize image data for FEA are extremely rare. The non-invasive property associated with imaging and recent developments in image sensors are empowering researchers with novel strategies for measuring various material properties and delineating objects. Using such data as input to numerical simulation techniques requires the development of efficient mesh generation technique adapted to image data. Such meshes could support, for example, the simulation of blood flow in arteries (Cebal and Löhner, 1999) or micro-structural analysis of bio-materials (Müller and Rügsegger, 1995). Contemporary methods for generating simplicial meshes from image data were presented by categorizing them into two classes. First category of methods integrate image segmentation and meshing in one step. The integrated image analysis based domain discretization techniques produce a small number of elements at material boundaries as it aligns elemental edges with such boundaries, thereby interleaving automatic feature isolation. However, these methods are computationally expensive because they require repeated computations of minimizing

an objective function and its partial derivatives. Another problem associated with these methods is that they are not information preserving and manipulate the image data (e.g., the annealing step of the method proposed by Langer et al. (2001)). The second category of methods for image data works with the polygonal/polyhedral surfaces bounding the interval volume. These methods preserve the topology of the domain in the image data, but they come with no theoretical guarantees on the elemental quality. Moreover, these methods also need an extra image-processing step to generate the iso-surfaces bounding the interval volume.

3.3 Objectives

The overall goal of this chapter is to develop an efficient quality-guaranteed, unstructured mesh generation technique for image data. The purpose of the technique is to enable researchers to perform transport simulations using finite element method, on spatial domains obtained by segmenting heterogeneous biological images obtained, for example digital satellite images of a landscape or magnetic resonance imaging of a body part. The specific objectives of the research described in this chapter are to:

Objective 3: Develop and implement a quality guaranteed unstructured simplicial mesh generation technique for (segmented) image data.

Objective 4: Analyze the developed unstructured simplicial mesh generation technique for elemental quality and output size.

The mesh generation technique should be integrable with the texture-based image segmentation technique developed in the previous chapter (refer to Section §2.3).

Additionally, the developed method must be capable to adapt to the heterogeneity presented in the image data, thereby generating coarser mesh for homogeneous regions and finer mesh for highly heterogeneous regions. It should also be easily scale to arbitrary dimensions so that it accommodates both 2-D images (e.g., air photos) and 3-D images (e.g., MR scans). Lastly, unlike current integrated image-analysis based mesh generation techniques (Hale, 2001; Langer et al., 2001), the developed method must also be information preserving so that material boundaries in the segmented image can be extracted exactly from the mesh, i.e., it should not destroy the original image data.

3.4 Materials and Methods

3.4.1 Development

The approach used in this work to develop the image meshing procedure consists of modifying the method of (Bern et al., 1990) so that it works with image data, and extending this algorithm to higher spatial dimensions. The meshing method of (Bern et al., 1990) was selected as a starting point because it is unstructured, efficient, quality-guaranteed, adaptive, and uses a dyadic space decomposition that is conceptually similar to that used in image segmentation techniques rooted in multi-resolution analysis (e.g., SWA segmentation method developed in Chapter 2). The 2-D algorithm of (Bern et al., 1990) consists of three steps: (1) generate a point quadtree of the input vertex set; (2) balance the point quadtree using 2:1 rule, and (3) triangulate the points within the nodes of point quadtree.

The method used to adapt, the algorithm to arbitrary dimensional image data will entail three major modifications: (1) the hierarchical data structure will be changed from point quadtree to region quadtree to accommodate image elements; (2) this data data structure will be extended to higher dimensions, and (3) a step in which vertices are extracted from the region-based data structure will be added to the algorithm, prior to meshing. The first modification needed to account for the nature of image data consists of spatially extended regions (pixels in 2-D and voxels in 3-D) rather than points. The second modification is required to ensure that the image meshing strategy works on both two-dimensional bio-images (e.g., digital microscope image, satellite images) and on three-dimensional images (e.g., CT scans and MR scans). The third modification is necessary because Delaunay meshing uses a vertex set (i.e. points) as inputs, rather than spatially extended regions (i.e., pixels or voxels).

In this study, the first two modifications will be performed by using a region 2^d -tree (explained below) to represent the input image data. This data structure is the natural equivalent of the point-quadtrees used by (Bern et al., 1990) for their 2-D input vertex set. It possesses the same density-adaptive spatial properties and coincides with the data structure used by the SWA bio-image segmentation method. The added step described as third modification above will be used to produce a vertex set from the leaf nodes of the 2^d -tree. This will be accomplished by considering all leaf node corners to be vertices which guarantees that material boundaries of the original image are preserved exactly by the mesh.

With the modifications, listed above, the image segmentation algorithm de-

veloped in this study will consist of the following steps:

1. generate a region 2^d -tree representation of the input d -dimensional images,
2. balance the 2^d -tree using 2:1 rule
3. extract a vertex set from the corners of the balance 2^d -tree's leaf nodes, and
4. use Delaunay meshing to produce a simplicial mesh from the vertex set.

The high degree of similarity between this algorithm and that of (Bern et al., 1990) suggests that the proposed image meshing method will possess similar characteristics in terms of element quality and density adaptivity.

The details of the region 2^d -tree representation and balancing used in this study are as follows. The original, d -dimensional image, S , is represented as:

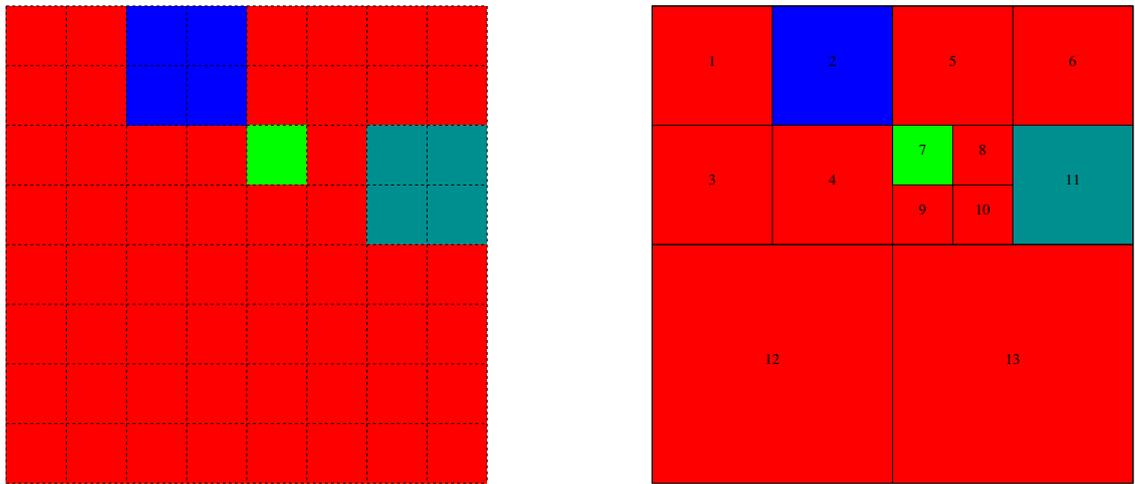
$$S(x_1, x_2, \dots, x_d : x_i \in [0, 2^{n-1}]), \quad (3.2)$$

where n is a positive integer. The input image data is resized appropriately to ensure that it respects equation 3.2. The region 2^d -tree recursively subdivides S into 2^d number of nodes until a predefined stopping criterion is attained (Samet, 1990). In our case, a leaf node is recursively split until all the pixels inside the node/block have the same value. The process of creating 2^d sons of a node is called *splitting*. Henceforth, the 2^d -tree decomposition of S will be denoted by UT . Figure 3.19(b) illustrates the quadtree (2^2 -tree) maximal block decomposition for a 2-D image of 8×8 pixels shown in Figure 3.19(a). Any node which is not split is called a *leaf* or *external* node. Conversely, an *internal* or *non-leaf* node is one that has child nodes.

In Figure 3.19(c), the leaf nodes are shown as squares and the non-leaf nodes as empty-circles. The *level* of the root node in a 2^d -tree is equal to the \log_2 of the dimension of the input image. Since the input is assumed to be a dyadic image, the root node of the 2^d -tree is always at level l_j , where $l_j \in [0, n]$, and all the children are at a level lower than their respective parent nodes. The *size* of a node at level l is equivalent to its width and is given by $s = 2^{l_j}$. The *depth* of a 2^d -tree is equal to the difference between the levels of the root node and the deepest non-leaf node. For example, depth of the quadtree shown in Figure 3.19(c) is 3, because the root node is at l_3 and the deepest non-leaf node is at l_0 .

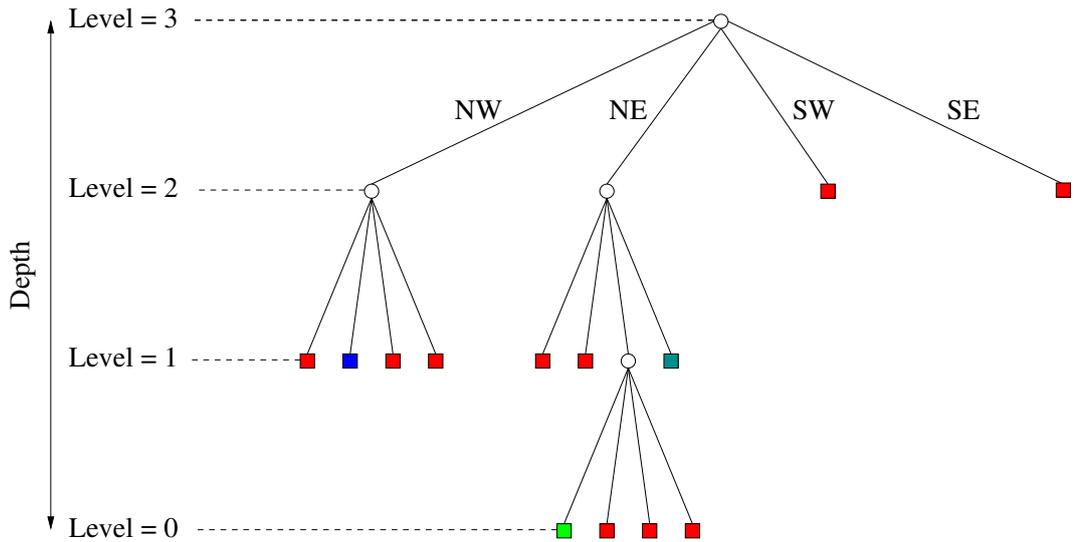
A node in a 2^d -tree has 2^d vertices or *corners*. An *edge* of a leaf node is the line connecting two neighboring corners of the leaf node in 2-D and a *facet* of a leaf node is the corresponding planar polygon in 3-D. The term *face* of a leaf node is used to refer a generic d -dimensional entity, that can represent a corner, an edge, or a facet in \mathbb{R}^0 , \mathbb{R}^1 , or \mathbb{R}^2 . The *degree* of a 2^d -tree is defined as the number of edges formed from one leaf node edge when that leaf node is split. The 2^d -trees discussed in this study are of degree two, as splitting a leaf node involves dividing its edges into two equal parts. Two leaf nodes sharing a k -dimensional face, $k \in [0, d]$, are called k -neighbors (Moore, 1995) and the special case of leaf nodes sharing a $d - 1$ face is called e -neighbors (i.e., in 2-D two leaf nodes sharing an edge and in 3-D two leaf nodes sharing a facet are called e -neighbors).

At each stage in generation of a 2^d -tree from an image, a node can be subdivided resulting in a decomposition which can be largely unbalanced. A frequently used approach to overcome this situation is to impose the *2:1 balancing rule* (Bern



(a)

(b)



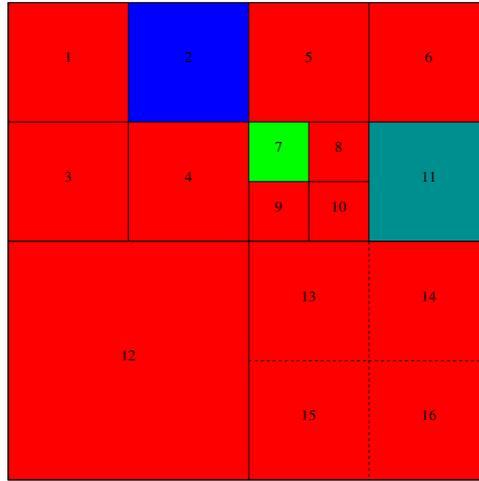
(c)

Figure 3.19: An example of (a) 2-D input image of size 8×8 pixels, (b), its maximal block decomposition, and (c) the corresponding pointer-based region quadtree structure.

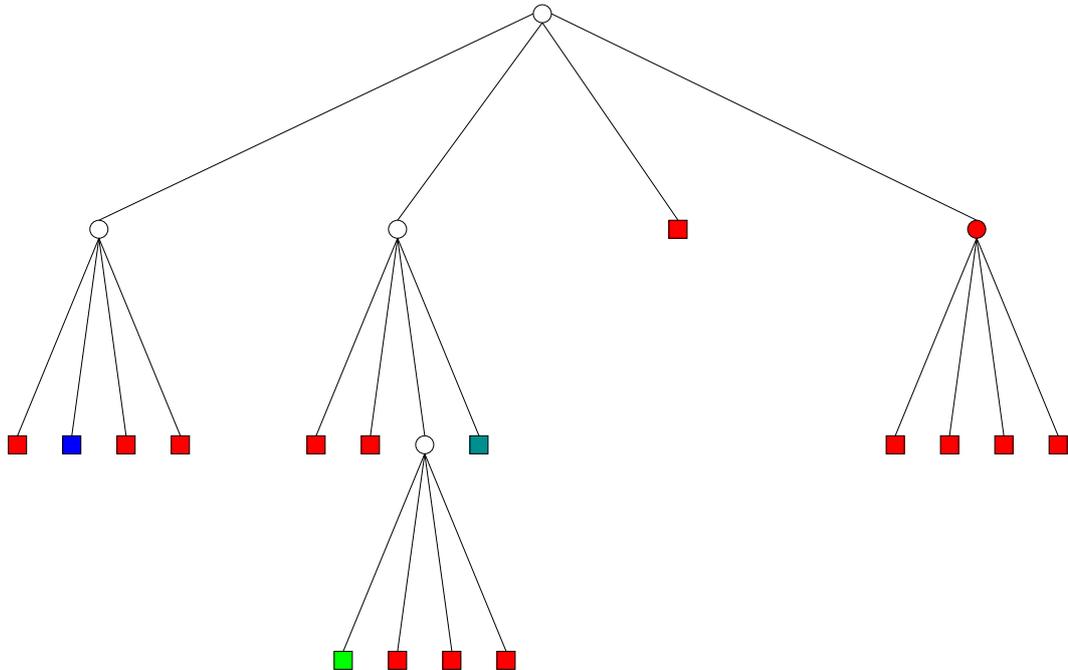
et al., 1990; Herzen and Barr, 1987; Yerry and Shephard, 1983) for converting a generic 2^d -tree (UT) to an e -balanced 2^d -trees (BT). The 2^d -tree is e -balanced if, any leaf node at level l_j contains no e -neighbors at level greater than l_{j+1} . For example, leaf nodes 9 and 10 (both at levels l_0) in Figure 3.19(b) have node 13 as an e -neighbor which is at a level l_2 . To meet the 2:1 balancing requirement, node 13 in the unbalanced quadtree is should be split into four children. The e -balanced quadtree obtained by applying this 2:1 balancing rule to the unbalanced quadtree of Figure 3.19(b) is shown in Figure 3.20. The 2:1 balancing rule can be implemented efficiently to bound the BT by $O(|UT|)$ operations (Moore, 1995).

The four step image meshing algorithm will be implemented in the Java language using the Java Software Development Kit (SDK) version 1.4.1. Java was selected over C++ and fourth generation languages (4GL) like MATLAB, IDL, and Mathematica for the following reasons: (1) Java is freely available; (2) Java runs on arbitrary platforms, from mobile devices to super computers, and; (3) Java code is platform independent, it can be written once and run on any platform with an Java interpreter. The slower speed of associated with executing Java byte code, compared to C/C++, is viewed in this study as a minor drawback relative to its advantages.

The algorithms used for generating a region 2^d -tree from image data, for e -balancing the tree and for Delaunay meshing will be obtained from the literature and implemented or translated into Java source code. In the event that an existing algorithm can be modified to improve its performance, such modification will be performed and the performance of the resulting modified algorithm will be compared to that of the original algorithm using test images prior to inclusion in the



(a)



(b)

Figure 3.20: (a) Maximal block decomposition and (b) quadtree structure after imposing the 2:1 balancing rule on the quadtree in Figure 3.19.

implementation.

The notation and conventions used for implementation are as follows. The input image co-ordinate system is assumed to be in Euclidean space with the axes in the clockwise orientation. That is, for a 2-D image the origin is at the *NW* corner, whereas a 3-D image has its origin at the *LUF* corner. This axes orientation is consistent with the conventions used in the image processing literature. The input image is assumed to be discrete approximation of a continuous function with a single value at a each location on a uniform integer grid with spacing $h = 1$. Lastly, the input image S is bounded by 2^d number of vertices or corners that are given by the following relation:

$$V \{S(x_1, x_2, \dots, x_d)\} = \left\{ \begin{array}{c} \{x_1, x_2, \dots, x_{d-1}, x_d\}, \\ \{x_1, x_2 + h, \dots, x_{d-1}, x_d\}, \\ \vdots \\ \{x_1, x_2, \dots, x_{d-1} + h, x_d\}, \\ \{x_1, x_2, \dots, x_{d-1}, x_d + h\} \end{array} \right\} \quad (3.3)$$

The information stored in the image can consist of floating-point values or integer and can be in any compression format, as long as the input conventions are meet. Figures 3.21 and 3.22 illustrates the conventions and notations listed above.

3.4.2 Analysis

The goals of the analysis are to obtain a theoretical bound and an expression for output mesh (GM) generated by the IMESH algorithm. The element quality index that will be used to analyze the quality of the unstructured mesh from the IMESH

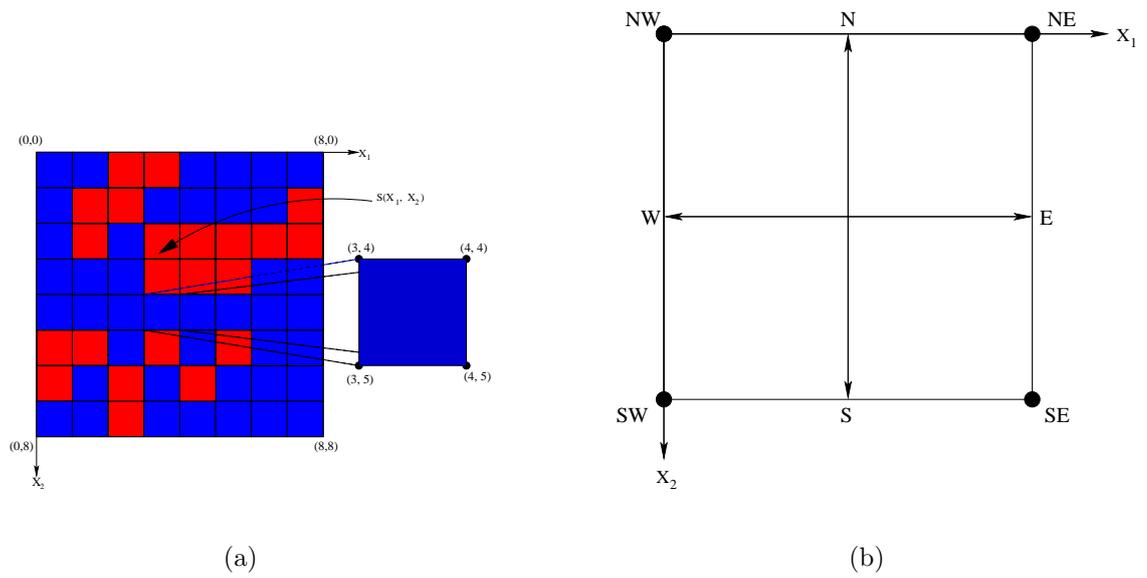


Figure 3.21: An example of (a) two-dimensional image with its coordinate system and (b) relationship between a 2-D block's four quadrants and its boundaries.

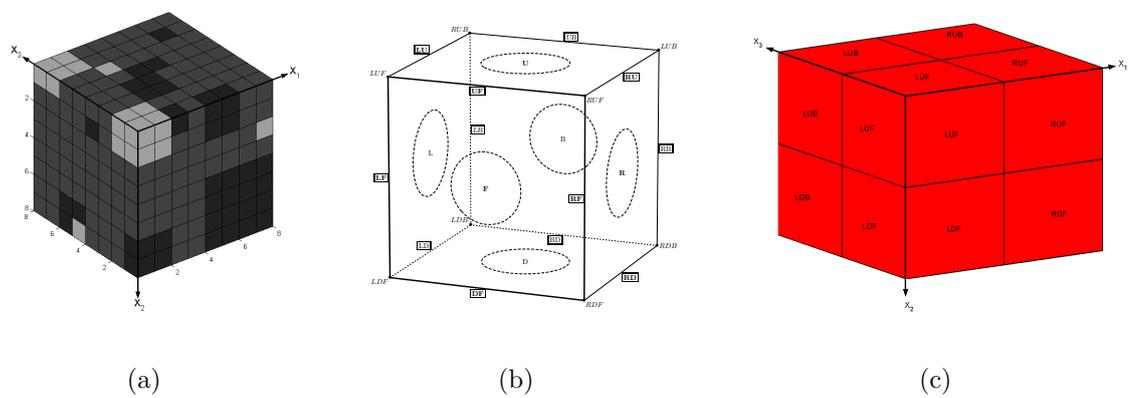


Figure 3.22: An example of (a) three-dimensional image with its coordinate system, (b) the labelling of faces, edges, and vertices for a 3-D block, and (c) labelling of octants associated with a 3-D block.

algorithm is selected from FE mesh quality measures presented in the literature (Shewchuk, 2002). The *radius-to-edge ratio* (Θ) is the preferred choice as the quality index for the analysis because it readily scales to higher-dimensional simplices. Formally, it is defined as follows: Let $QM = \{q_1, q_2, \dots, q_n\}$ be the output of the simplicial mesh, then the radius-to-edge ratio is defined for the entire mesh QM using the following relation,

$$\theta(q_i) = \frac{r_{circ}}{\bar{e}_{min}} \quad (3.4)$$

$$\Theta(QM) = \max(\theta(q_i), q_i \in QM) \quad (3.5)$$

In equation 3.4, \bar{e}_{min} is the minimum edge length of the simplex and r_{circ} is the circumradius of an element (i.e., the radius of circumscribing circle, sphere or hypersphere). Another advantage of using this quality metric is that, in two-dimensions, a bound on Θ also bounds the minimum internal angle, α_{min} , of the entire triangulation (Shewchuk, 2002, see Section 3.1). These two metrics are related by:

$$\alpha_{min} = \arcsin \frac{1}{2\Theta} \quad (3.6)$$

The worst possible radius-to-edge ratio, that can be produced by the IMESH, will be identified using a three step process: (1) all possible non-isomorphic configurations of vertices, on leaf nodes, obtainable in an e -balanced 2^d -tree will be enumerated; (2) the simplices that can be generated (by Delaunay triangulation) from the above configurations will be identified; and (3) the radius-to-edge ratio, θ , for each simplicial element will be computed. The bound on the element quality, $\Theta(GM)$, for the output mesh from IMESH will then be calculated using equation 3.4.

The enumeration of all possible non-isomorphic vertex configurations on a leaf node, λ , will be performed using a d -partite graph representation since every $(d - 1)$ -dimensional face F_i of the leaf node λ is adjacent to every other face except the opposite face. Figure 3.23 shows an example of this equivalence in two- and three-dimensions. In 2-D, the vertices of the $G(\lambda)$ geometrically represent the edges of λ and in 3-D the six vertices of the graph are geometrically equivalent to the six facets of the cube.

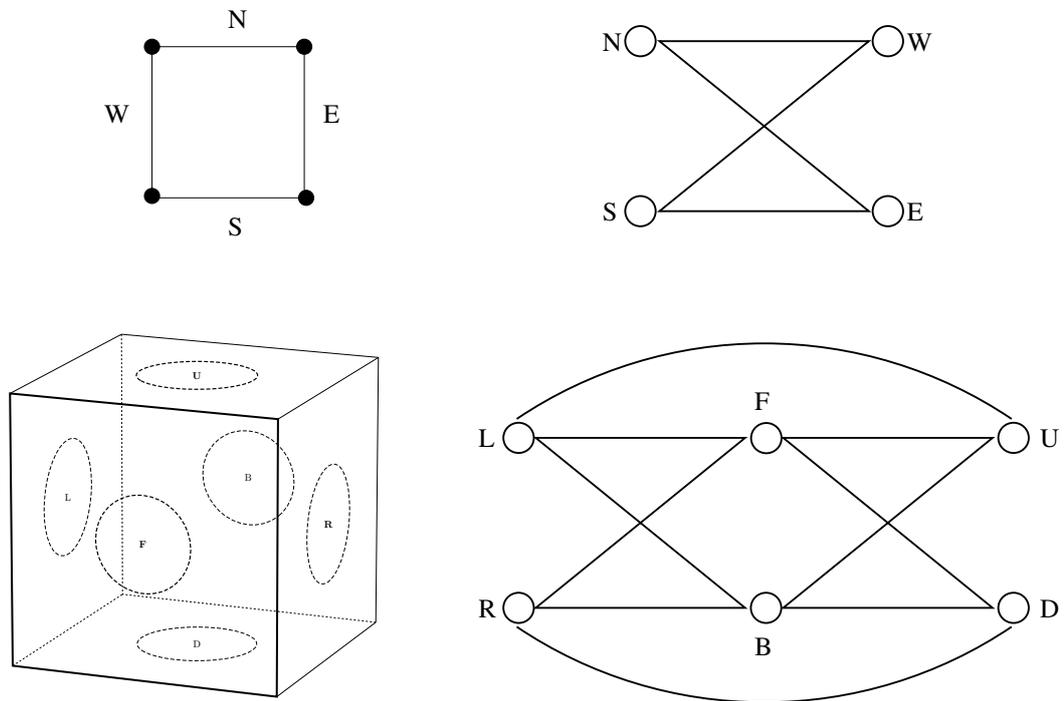


Figure 3.23: Example showing the equivalence of representing a 2-D node and a 3-D node as a d -partite graph. The vertices of the d -partite graphs (on the right) represent the edges and faces of the 2-D block and 3-D block, respectively.

Since a 2:1 balancing rule is imposed for constructing the e -balanced 2^d -trees, every leaf node, λ of size ℓ , in BT can only have e -neighbors of three possible sizes:

$\ell/2$, ℓ , and 2ℓ (Figure 3.24). These possibilities can be further reduced into two cases where the e -neighbors are of size $\ell/2$, and ℓ . Because, e -neighbors of λ with sizes $\ell/2$ and 2ℓ can be treated as one case by interchanging λ with its e -neighbor. The case when a leaf node has an e -neighbor of size $\ell/2$ will place $3^{d-1} - 2^{d-1}$ vertices along their common $(d - 1)$ -dimensional face. For example, in 2-D a new vertex ($3^1 - 2^1$) is added at the mid-point along their common edge. Similarly, in 3-D five ($3^2 - 2^2$) new vertices are added along their common face. This is indicated on the d -partite graph, $G(\lambda)$, by coloring the vertex (circle) that corresponds to the edge (face) that is split. For analysis, the graph vertices will be assigned black or white shade depending on whether it is split or not. Figure 3.25 illustrates the equivalence between leaf node and d -partite graph representations in a two and a three-dimensional situation where two e -neighbors of the central leaf node, λ , are split.

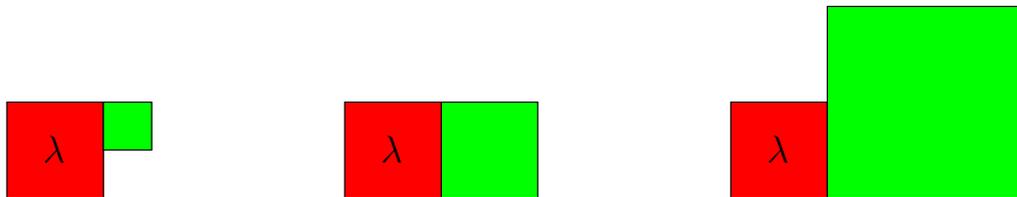


Figure 3.24: *Illustration showing a 2-D node (λ) in a e -balanced quadtree along with its possible e -neighbors. The configuration on the left shows λ having an edge neighbor of half its size. In the middle, λ shares its edge with an equal size e -neighbor. The rightmost configuration shows λ have an e -neighbor of twice its size.*

The use d -partite graph representation reduces the problem of enumerating the leaf node configurations to that of finding the non-isomorphic configurations

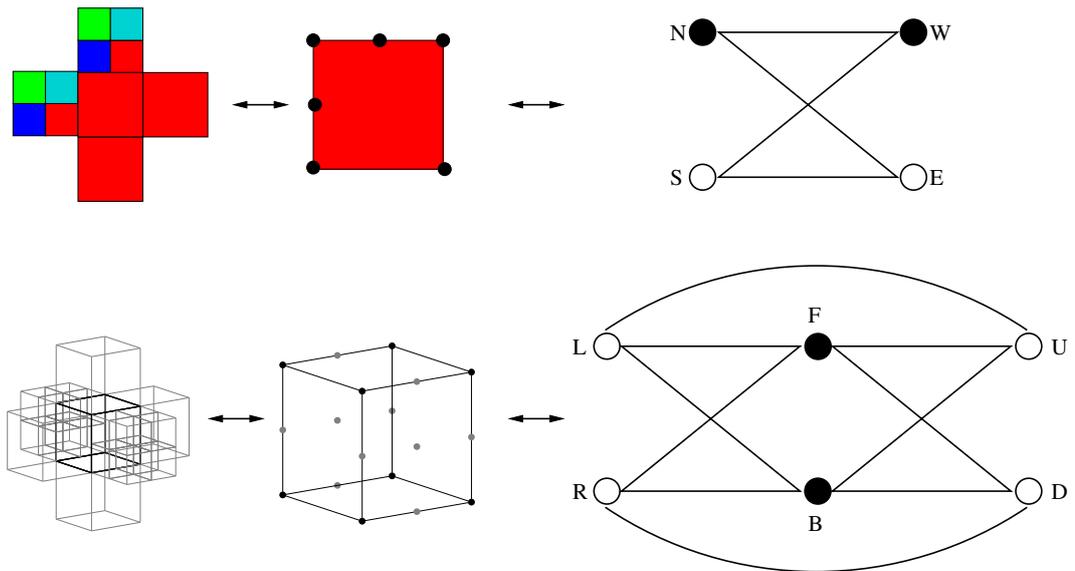


Figure 3.25: Example showing the equivalence of representing a 2-D node and a 3-D node as a d-partite graph when its e-neighbors are split. The vertices of the d-partite graphs, shown on the rightmost column, with "solid black" color corresponds to the 2-D node's edges and 3-D node's faces that are split.

of the graph. Enumeration of all the possible non-isomorphic configurations of the d -partite graphs, permits to (indirectly) obtain all possible vertex locations in BT , which in turn, permits the generation (by Delaunay triangulation) of all possible simplicial elements produced by IMESH and hence the proof for quality bounds for the algorithm.

3.5 Results

3.5.1 Development

The image meshing method (referred to as IMESH in the following) was developed and implemented in accordance with the four step algorithm presented in the Section §3.4.1. The resulting technique for computing the quality unstructured meshes QM from either a segmented or a raw multi-dimensional image S , is given in the pseudocode 3.5.1. As discussed previously the four main steps of IMESH algorithm are: (1) to convert the \mathbb{R}^d input image to a 2^d -tree representation, UT , (2) construct e -balance the 2^d -tree (BT) by imposing the 2:1 balancing rule (instrumental in fine tuning the element quality of the output mesh), (3) use the unique vertices, UV , of all the leaf nodes in the BT , and (4) use the unique vertices generated in the previous step as an input to a Delaunay triangulation technique for building the unstructured finite element mesh, QM .

Pseudocode 3.5.1: IMESH(S)

comment: Image Meshing Algorithm for image data, S

comment: Compute 2^d -tree for S

$$UT \leftarrow \text{ARRAYTOTREE}(S) \quad (1)$$

comment: Build a ϵ -balanced 2^d -tree of UT

$$BT \leftarrow \text{BALANCEDTREE}(UT) \quad (2)$$

comment: Get unique vertices of the leaf nodes in BT

$$UV \leftarrow \text{UNIQUEVERTICES}(\text{LEAFNODES}(BT)) \quad (3)$$

comment: Compute Delaunay triangulation of UV

$$UM \leftarrow \text{QDELAUNAY}(UV) \quad (4)$$

output (UM)

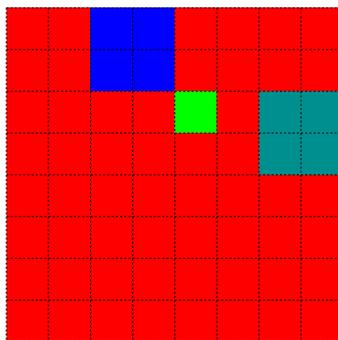
Figure 3.26 illustrates the output from each of these four steps involved in generating a quality triangular mesh for an example two-dimensional image. The source image is two-dimensional square consisting of 8×8 pixels with four intensity levels (or classes). The region-quadtrees generation step converts it to a set of 13 leaf nodes, each of which correspond to a homogenous sub-region of the input image (Figure 3.26(b)). Leaf nodes are observed to be physically smaller where the input image contained fine-scale heterogeneity and larger where the image was relatively homogeneous. The quadtree representation has preserved all of the information contained in the original image but represents it using only 13 leaf nodes rather than 64 pixels. The ϵ -balanced quadtree produced as the output of step 2 of the

algorithm is shown in Figure 3.26(b). In this example, only one leaf node had to be split in order to ϵ -balance the tree. This led to a total of 16 leaf nodes in the ϵ -balanced quadtree representation of the source image which corresponds to only one fourth of the number of pixels in that image. The unique vertices collected from the corners of leaf nodes in the ϵ -balanced tree are shown in Figure 3.26(c). A total of 27 vertices were extracted and one readily observes that the spatial density of these vertices varies over space in relation to the heterogeneity of the input image: vertex density is highest in areas where the source image has fine-scale heterogeneity and smallest where the source is spatially homogeneous. The Delaunay mesh computed over these vertices in step 4 of the algorithm is presented in Figure 3.26(d). The mesh consists of 38 triangular elements which is approximately one quarter of the 128 triangles that would be necessary to mesh the source image directly on a pixel basis. The mesh is observed to be density adaptive with a high density of small but rounder elements where the input image is to be highly heterogeneous and a low density of large quality elements where it is homogeneous. Close inspection further reveals that all material boundaries of the source image are represented by element edges in the mesh, as required by this study's objectives.

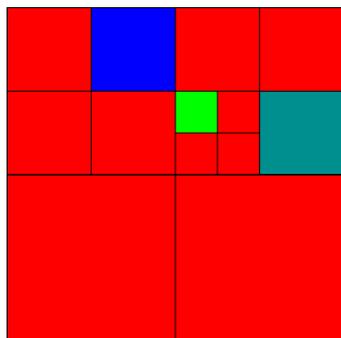
The efficiency, adaptivity, quality and boundary preservation property of the mesh are a direct result of the procedure used to generate it in the IMESH algorithm. Efficiency is primarily a result of the use of a 2^d -tree to efficiently represent the input image as a small number of leaf nodes in step 1 of the algorithm. This step, furthermore, results in a density-adapted mesh since leaf nodes are smaller where the source image has been fine scale heterogeneity and larger where it is

homogeneous. Step 2 of the algorithm mildly reduces its spatial efficiency but is key to producing a quality mesh. The quality of elements is a result of steps 2 to 4 of the algorithm (the e -balancing step using the 2:1 rule). This step results in bounds on the shape of the elements produced by IMESH that are analyzed in detail in Section §3.5.2 and guarantees that no bad simplicial elements will be generated. The boundary preservation property of the mesh results from steps 1, 3, and 4 as the quadtree respects original image boundaries and Delaunay meshing on corner vertices of leaf nodes generates elements with edges aligned to those of the leaf nodes. The efficiency, adaptivity, and quality (proved in the next section) of IMESH are hence found to be similar to those of the 2-D point-based algorithm of (Bern et al., 1990) which it extends to image data.

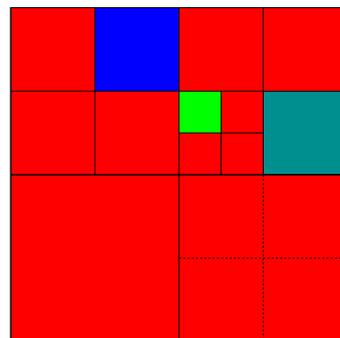
The characteristics of the IMESH algorithm are further illustrated using 2-D and 3-D segmented biological image data sources in Figures Figure 3.27(a) and 3.28(a), respectively. The result of applying IMESH to a 1024×1024 pixels segmented digital satellite photo, representing a farmed landscape and wetland is present in Figure 3.27(b). The output mesh consists of: 29716 leaf nodes in the e -balanced quadtree, 37647 unique vertices, and 75292 triangles which is 3.59% of the total triangles had each image pixel been split into two triangles. The mesh density is observed to adapt significantly to the image heterogeneity boundary with a low density of elements in homogeneous areas and a high density at material boundaries. The original locations of all material boundaries are preserved by the mesh. The smallest and the largest internal angles of the elements is 26.565° and 90° , and the worst radius-to-edge ratio is $\frac{\sqrt{5}}{2}$ (aspect-ratio is at most 2.5). Hence



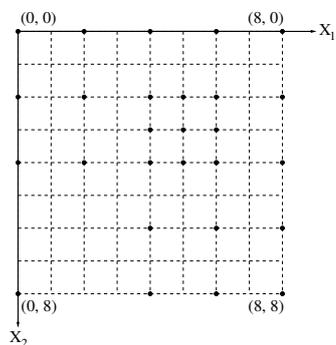
(a) Input Image



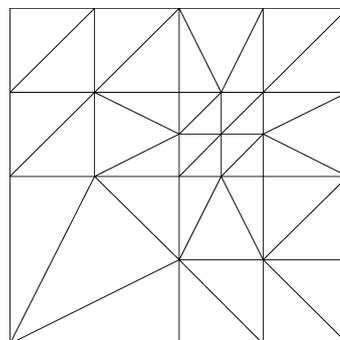
(b) Step 1



(c) Step 2



(d) Step 3



(e) Step 4

Figure 3.26: An example of applying IMESH algorithm in 2-D for: (a) input image of size 8×8 pixels, (b) its quadtree maximal block decomposition, (c) its 2 : 1 e-balanced quadtree maximal block decomposition, (d) unique vertices of the leaf nodes, and (e) output Delaunay mesh of the input image.

the output mesh is considered to be of good quality (neither a needles nor a cap shaped elements appear).



Figure 3.27: (a) *Input segmented landscape image, and (b) output mesh generated by IMESH algorithm.*

The result of applying IMESH to a $256 \times 256 \times 256$ voxels of segmented brain MR volume with three tissues, namely CSF, grey-matter, and white-matter is presented in Figure 3.28(b). The output mesh consists of 627736 leaf nodes and 2733351 tetrahedral elements and closer inspection shows that the 3-D mesh respects the material boundaries. The worst case radius-to-edge ratio is $\frac{\sqrt{29}}{4}$ and hence guarantees that the output mesh does not contain any bad tetrahedral elements, except slivers. The case of slivers appearing in the output mesh from IMESH algorithm is minimal because the locations of the vertices of a leaf nodes in a e -balanced octree is well positioned to avoid slivers. This result along with that for the 2-D biological image, confirm the expected efficiency, adaptivity, and boundary preservation properties of IMESH.

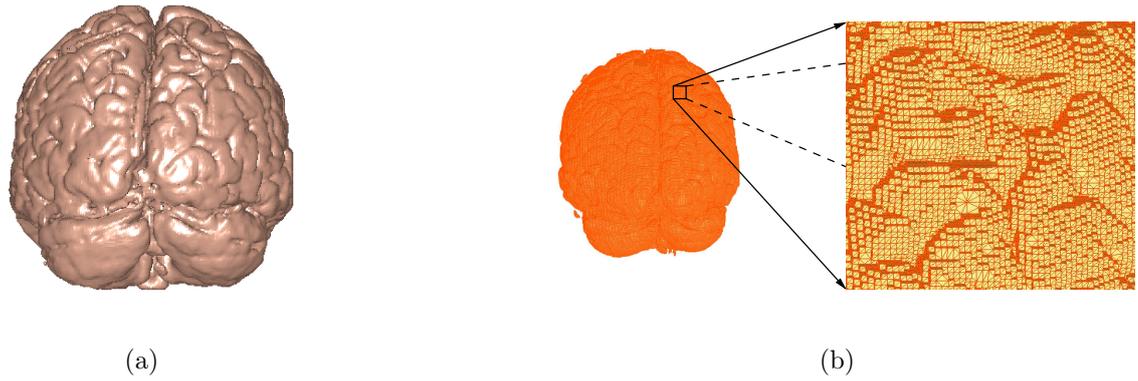


Figure 3.28: (a) *Input segmented MR brain volume with CSF, grey-matter and white-matter tissues, and (b) tetrahedral mesh generated by IMESH algorithm.*

The Java SDK 1.4.1 source code and the compiled byte code of the IMESH developed in this study can be accessed from the internet (Gudla, 2005). The details of the development are as follows. The first step of the IMESH algorithm involves constructing a pointer-based 2^d -tree with its root node as large as S . This study uses the ARRAYTOQUADTREE procedure of Samet (1989) for implementing the pointer-based 2^d -tree algorithm. Although *FD linear 2^d -trees* Samet (1989) can give compact data representation (compression), pointer-based region 2^d -trees representation is used in the implementation, as they are more efficient for tree-traversal operations (for instance, neighbor finding). Since the IMESH algorithm involves several such tree-traversal operations the selection of pointer-based 2^d -tree representation over FD linear 2^d -trees is justified. However, to accommodate the input requirements of a multi-dimensional array, the original 2^d -node data-structure (Samet, 1989, see Section §1.5)) was modified with the following changes:

- "FATHER" field of type NODE contains pointers to the node's father (parent)

and is unchanged,

- "SONS" field of type NODE contains pointers to its 2^d sons (children) and is unchanged,
- "NODETYPE" field is changed to data-type boolean, which takes a value 0 or 1 when the 2^d -tree node is non-leaf or leaf, respectively,
- "VALUE" field of data-type float is added for accommodating gray-scale pixel intensities or a floating-point values of the image data. For a non-leaf node the "VALUE" field is set to $\{\emptyset\}$,
- "LEVEL" field of data-type integer is added, which takes a value in the interval $[0, n]$ depending on the level of the node in the 2^d -tree, and
- "CORNER" field of type POINT (see pseudocode 3.5.2) is added to the data-structure to maintaining the North-West (*NW*) and Left-Upper-Front (*LUF*) integer space co-ordinates associated with the node in \mathbb{R}^2 and \mathbb{R}^3 , respectively.

Samet's ARRAYTOQUADTREE conversion algorithm was also modified appropriately to reflect the changes made to the NODE data-structure (see pseudocode 3.5.3). A detailed description and analysis of this algorithm can be found in Samet (1989, Section §4.1). The modified algorithm of converting an array to 2^d -tree, (ARRAYTOTREE), takes a (multi-dimensional) floating-point array or (segmented) image as input and returns a pointer-based region 2^d -tree, **UT**, bounding the input data.

Pseudocode 3.5.2: POINT()**comment:** Data structure for a point in \mathbb{R}^d integer x_1, x_2, \dots, x_d **Pseudocode 3.5.3:** NODE()**comment:** Data structure of a node in quadtree

{	pointer NODE FATHER
{	pointer NODE[2^d] SONS
{	boolean NODETYPE
{	float VALUE
{	integer LEVEL
{	POINT CORNER

The 2:1 e -balanced version, BT , of the 2^d -tree BT is generated using the algorithm proposed by Sivan (1996) for surface modeling except for the TRIMNODES function used at steps 1 and 2 in pseudocode 3.5.4. In the BALANCEDTREE algorithm, the TRIMNODES function serves the purpose of filtering out the leaf node(s) at levels l_n and $l_n - depth$ from ever being processed which reduces the overall processing time. This is done because if a 2^d -tree has a leaf node at l_n , then the input image/array is completely homogeneous and does not need any further balancing. Similarly, the leaf nodes in a 2^d -tree at level $l_n - depth$ will never be split during the 2:1 balancing phase. Thus, such a filtering step minimizes the number of nodes

to be processed while constructing the e -balanced 2^d -tree. (Note that in Sivan's original algorithm every leaf node is checked for 2:1 balancing condition, which in turn uses a relatively expensive step of finding the e -neighbors.) The size-filtering step is implemented very efficiently, with little overhead, by exploiting the 2^d -tree structure to collect only those leaf nodes that are between levels, $l \in \{1, 2, \dots, n-1\}$.

Pseudocode 3.5.4: BALANCEDTREE(UT)**comment:** e -balancing a pointer-based region 2^d -tree UT **comment:** Insert leaf nodes of UT into L $L \leftarrow \text{LEAFNODES}(UT)$ **comment:** Retain nodes from L whose level $\in [1, l - 1]$ $L \leftarrow \text{TRIMNODES}(L)$ (1)**while** $L \neq \{\emptyset\}$

do	{	comment: pop node l_i from L			
		$l_i \leftarrow \text{POP}(L)$			
		comment: Load the edge-neighbors of l_i into a queue, NL			
		$NL \leftarrow \text{EDGENEIGHBORS}(l_i)$			
		$BT \leftarrow \text{E-BALANCE}(l_i)$			
		$NL \leftarrow \text{TRIMNODES}(NL)$		(2)	
		while $NL \neq \{\emptyset\}$			
		do		{	comment: pop out a node, nl_i , from L
					$nl_i \leftarrow \text{POP}(NL)$
					$BT \leftarrow \text{E-BALANCE}(nl_i)$

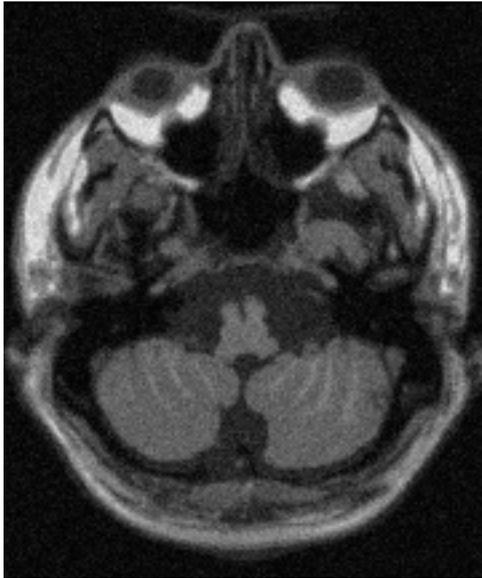
return (BT)

Four test images were used to quantitatively compare the performance of Sivan's original algorithm with its proposed modifications: (1) a 2-D raw MR brain scan (shown in Figure 3.29(a)), (2) a 2-D MR brain scan segmented into ten tissue

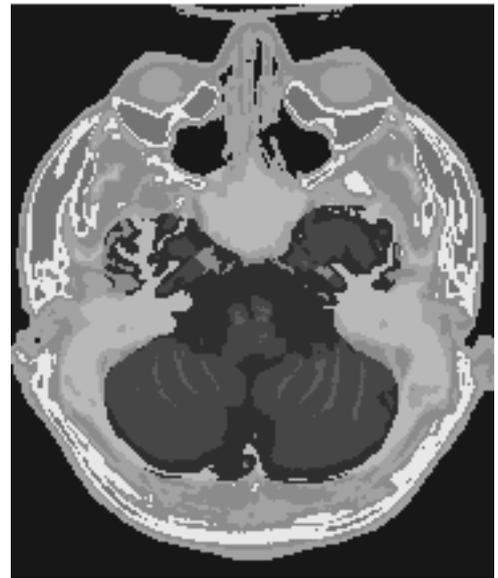
types (shown in Figure 3.29(b)), (3) a 3-D simulated raw MR brain scan with SNR = 1% and INU = 40% (shown in Figure 2.11(f)), and (4) a 3-D segmented MR brain scan from MNI-BrainWeb (shown in Figure 2.12). The ϵ -balanced 2^d -tree construction times, expressed in seconds, were obtained by implementing both the algorithms in Java SDK 1.4.1 and executing them on a SUN ULTRA-10 workstation running on Solaris 2.9 with 674 megabytes of physical RAM under typical load conditions. Experimental results of the time required for constructing ϵ -balanced 2^d -tree using Sivan's algorithm and the proposed size-filtering algorithm are shown in Table 3.1. The modified algorithm is 2 to 200 times faster than the original. The performance improvements are more pronounced for heterogeneous image data in higher dimensions. It is clearly evident that the size-filtering step speeds up the ϵ -balanced 2^d -tree construction and the modified algorithm was selected while implementing IMESH algorithm.

The unique vertex, UV , needed in step 3 of IMESH is generated from the ϵ -balanced 2^d -tree BT by applying the *quicksort* algorithm (Hoare, 1962) on the POINT data-type. The POINT data-type is declared *comparable* to facilitate the generation of unique vertices on-the-fly. The current implementation of the IMESH program is capable of generating the unique vertices in several compatible formats (e.g., *Triangle* (Shewchuk, 1996b), *detri* (Joe, 1995; Mücke, 1993), and *QHull* (Barber et al., 1996)).

The final step of the IMESH method involves Delaunay triangulation of the vertex set UV generated by collecting the unique corners of leaf nodes in BT . Although, there exists several implementations for Delaunay triangulation of point



(a)



(b)

Figure 3.29: *Test images used for quantitative comparisons of the e-balanced 2^d -tree construction using Sivan's algorithm and BalancedTree algorithm: (a) raw MR scan from MNI-BrainWeb, and (b) corresponding segmented MR scan with ten different tissues.*

Table 3.1: Comparison of constructing balanced 2^d -tree using Sivans algorithm (BQT1) and BalancedTree algorithm (BQT2).

Input Data	# Splits	Leaf Nodes Examined		Time, secs	
		BQT1	BQT2	BQT1	BQT2
2-D Raw MR (Figure 3.29(a))	72	40181	317	1.452	0.551
2-D Segmented MR (Figure 3.29(b))	567	17301	2567	0.771	0.331
3-D Raw MR (Figure 2.11(f))	1936	7263929	17680	115067.266	441.481
3-D Segmented MR (Figure 2.12)	18109	2832933	153709	23247.848	102.168

data set, the *quickhull* algorithm was chosen because it is the only known approach scalable across arbitrary dimension (Barber et al., 1996, see Figure 1)). Moreover, the quickhull algorithm is also implemented as an open source computational geometry package, called *QHull* (Barber et al., 1996) and is widely available for various computing platforms. Hence, in the final step, the output from the `UNIQUEVERTICES` function is redirected as an input to the `QDELAUNAY`³. The resulting output mesh from the `QDELAUNAY` can be viewed using the `GEOMVIEW` visualization program, which is also available from the *QHull* website. The implementation results for `IMESH` are summarized in Table 3.2. All examples presented earlier in this section were produced using this implementation (Gudla, 2005).

3.5.2 Analysis

In this section, the bounds on size and quality of the output FE mesh from the `IMESH` algorithm is presented. First, the proof for two-dimensional element quality is discussed and then generalized to higher dimensions (up to \mathbb{R}^8). Second, the proof for the bound output for the output mesh from `IMESH` algorithm in a worst-case scenario is discussed.

Result 3.5.1. *Let λ be any leaf node of a e -balanced 2^d -tree, BT , and*

$$\mu = \{e\text{-neighbors}(\lambda)\}$$

, then the number of non-isomorphic configurations, $NI(r)$, when r elements of μ

³The `QDELAUNAY` is a part of *QHull* software package. <http://www.qhull.org>

Table 3.2: Summary of implementation results for IMESH algorithm.

Step	Input	Description	Output	Function	Algorithm
1	S	Convert input image to 2^d -tree	UT	ARRAYTOTREE	Pointer-based region quadtree (Samet, 1989) extended to multi-valued and multi-dimensional image
2	UT	e -balance 2^d -tree using 2:1 rule	BT	BALANCEDTREE	Sivan (1996) algorithm with performance enhancing size-filtering
3	BT	Collect unique leaf node corner vertices	UV	UNIQUEVERTICES	Quicksort (Hoare, 1962) on POINT data type
4	UV	Generate simplicial mesh on corners	QM	QDELAUNAY	QHULL (Barber et al., 1996)

are split is

$$NI(r) = \begin{cases} \lceil \frac{r+1}{2} \rceil, & 0 \leq r \leq d \\ \lceil \frac{2d-r+1}{2} \rceil, & d < r \leq 2d \end{cases}$$

where $\lceil \cdot \rceil$ denotes ceil function.

Proof. Let F_i represent the i^{th} $(d-1)$ -dimensional face of λ and $\lambda(F_i)$ be the e -neighbor of μ along F_i , where $i \in [1, 2d]$ ⁴. If $G(\lambda)$ denotes the d -partite graph representation of λ such that its i^{th} vertex corresponding to F_i . A black or white colored vertex in the former denotes a split or un-split face in the latter. Additionally, if a binary value of 1 or 0 is assigned to black or white colored vertex in the d -partite graph, then the problem of enumerating the number of possible non-isomorphic configurations is reduced to *number partitioning*. That is, the case of enumerating the non-isomorphic leaf node configurations when its r faces are split is equivalent to partitioning the integer r into at most two parts with each part less d .

Case 1: $r = 0$ and $r = d$. These are trivial cases and each will result in only one configuration.

Case 2: $1 \leq r \leq d$. For this case, r can be partitioned into two parts in $\lceil \frac{r+1}{2} \rceil$ combinatorial ways (Comtet, 1974, pp. 115–116). Thus, the number of non-isomorphic leaf node configurations when r , $1 \leq r \leq d$ $d-1$ -dimensional faces are split is equal to $\lceil \frac{r+1}{2} \rceil$.

Case 3: $d < r < 2d$. Let $\tilde{r} = 2d - r$ be the number of faces that are not split.

This condition will reduce to the previous case by interchanging the values assigned

⁴A hyper-cube in \mathbb{R}^d has $2d$ faces each of which is a $(d-1)$ -dimensional hyper-cube.

to split and unsplit vertex of $G(\lambda)$. Hence, $NI(\tilde{r}, d) = \lceil \frac{\tilde{r}+1}{2} \rceil$. \square

An immediate consequence of Result 3.5.1 is the total number of non-isomorphic configurations, TNI , for any leaf node of BT when none or any of its e -neighbors are split. Result 3.5.2 provides an expression for TNI as a function of dimensionality d .

Result 3.5.2. *Let λ be any leaf node of BT and $\mu = \{e\text{-neighbors}(\lambda)\}$, then the total number of non-isomorphic configurations in \mathbb{R}^d , $TNI(d)$, when none or any of the elements of μ are split is given by, $TNI(d) = \frac{(d+1)(d+2)}{2}$.*

Proof. Assume $TNI(d) = \frac{(d+1)(d+2)}{2}$ to be true for \mathbb{R}^d , then using Result 3.5.1:

$$\begin{aligned} TNI(d) &= \sum_{r=0}^{d-1} \left\lceil \frac{r+1}{2} \right\rceil + \left\lceil \frac{d+1}{2} \right\rceil + \sum_{r=d+1}^{2d} \left\lceil \frac{2d-r+1}{2} \right\rceil \\ &= \left\lceil \frac{d+1}{2} \right\rceil + 2 \sum_{r=0}^d \left\lceil \frac{r+1}{2} \right\rceil \end{aligned} \quad (3.7)$$

$$TNI(d+1) = \left\lceil \frac{d+2}{2} \right\rceil + 2 \sum_{r=0}^{d-1} \left\lceil \frac{r+1}{2} \right\rceil \quad (3.8)$$

Subtracting equation 3.8 from equation 3.7 gives,

$$\begin{aligned} TNI(d+1) - TNI(d) &= \left\lceil \frac{d+1}{2} \right\rceil + \left\lceil \frac{d+2}{2} \right\rceil \\ &= d+2 \end{aligned} \quad (3.9)$$

By induction, equation (3.9) reduces to:

$$\begin{aligned} TNI(d+1) &= TNI(d) + (d+2) \\ &= \frac{(d+1)(d+2)}{2} + (d+2) \\ &= \frac{(d+2)(d+3)}{2} \end{aligned} \quad (3.10)$$

□

Table 3.3 provides an exhaustive list of Results 3.5.1 and 3.5.2 for 2-D, 3-D and 4-D. The six possible non-isomorphic configurations of a 2-D leaf node, represented as bi-partite graph, are shown in Figure 3.30. Similarly, the ten possible 3-D leaf node non-isomorphic configurations, represented as tri-partite graph, are shown in Figure 3.31. The corresponding leaf-node representations are shown in the top rows of Figures 3.32 and 3.33, respectively. The effect on a leaf node of having any of its e -neighbors are split is to add vertices along the common $(d - 1)$ -dimensional face. The node adjacency representation can be further transformed to a representation showing the leaf node with its original 2^d vertices and the extra vertices added along its faces to accommodate a split e -neighbor. This representation for both 2-D and 3-D cases are shown in the bottom row of Figures 3.32 and Figures 3.33, respectively. In these figures, the original vertices associated with the leaf node are shown in black shade whereas the newly added vertices due to the splitting of a e -neighbor are shown in gray shade. The availability of all these representations for the total non-isomorphic configurations of leaf nodes allows to prove the bounds on the elemental quality more readily than before.

Theorem 3.5.3. *In two-dimensions, the output triangular mesh, QM , from the IMESH method bounds: (a) the radius-to-edge ratio $\Theta(QM)$ to at most $\frac{\sqrt{5}}{2}$ and (b) the minimum angle α to 26.565° .*

Proof. In 2-D, it was established earlier that only six possible non-isomorphic configurations of a leaf node arrangement can appear in a balanced quadtree (refer to

Table 3.3: Exhaustive listing of the number of possible non-isomorphic configurations, up to 4-D, for a leaf node that has its $(d - 1)$ -dimensional face(s) split.

Dimension, d	NI(r)								TNI(d)	
	$r = 0$	1	2	3	4	5	6	7		8
2	1	1	2	1	1					6
3	1	1	2	2	2	1	1			10
4	1	1	2	2	3	2	2	1	1	15

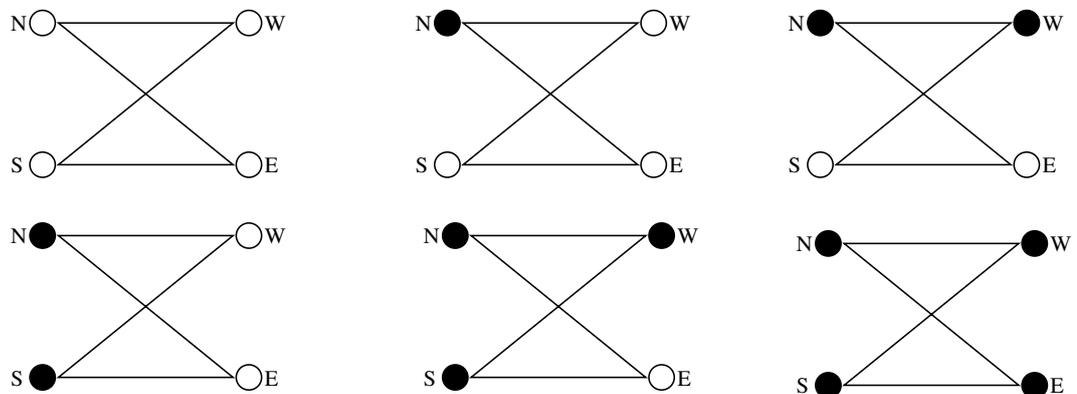


Figure 3.30: The d -partite graph representations for the six possible non-isomorphic splitting of a leaf node in two-dimensions.

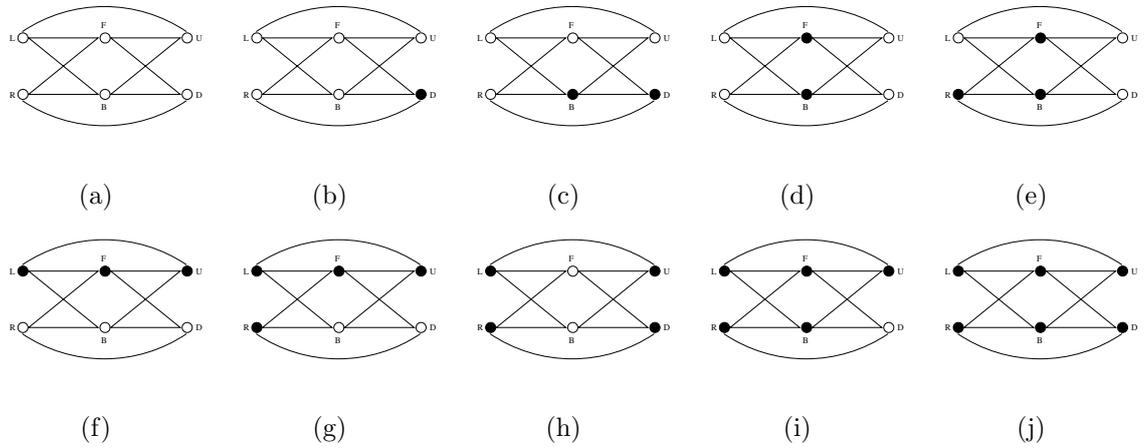


Figure 3.31: *The d-partite graph representations for the ten possible non-isomorphic splitting of a leaf node in three-dimensions.*

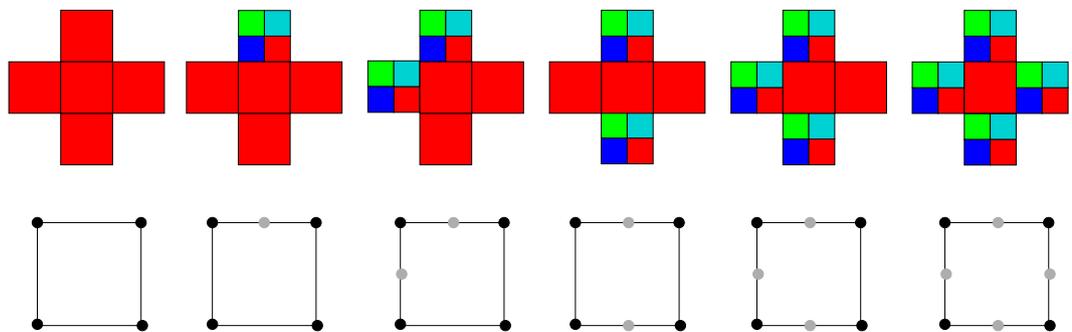


Figure 3.32: *Illustration showing the vertices of a 2-D leaf node (black circles) in a e-balanced quadtree and the extra vertices (gray circles) needed to match its split e-neighbors in each of the six possible non-isomorphic 2-D leaf node configurations.*

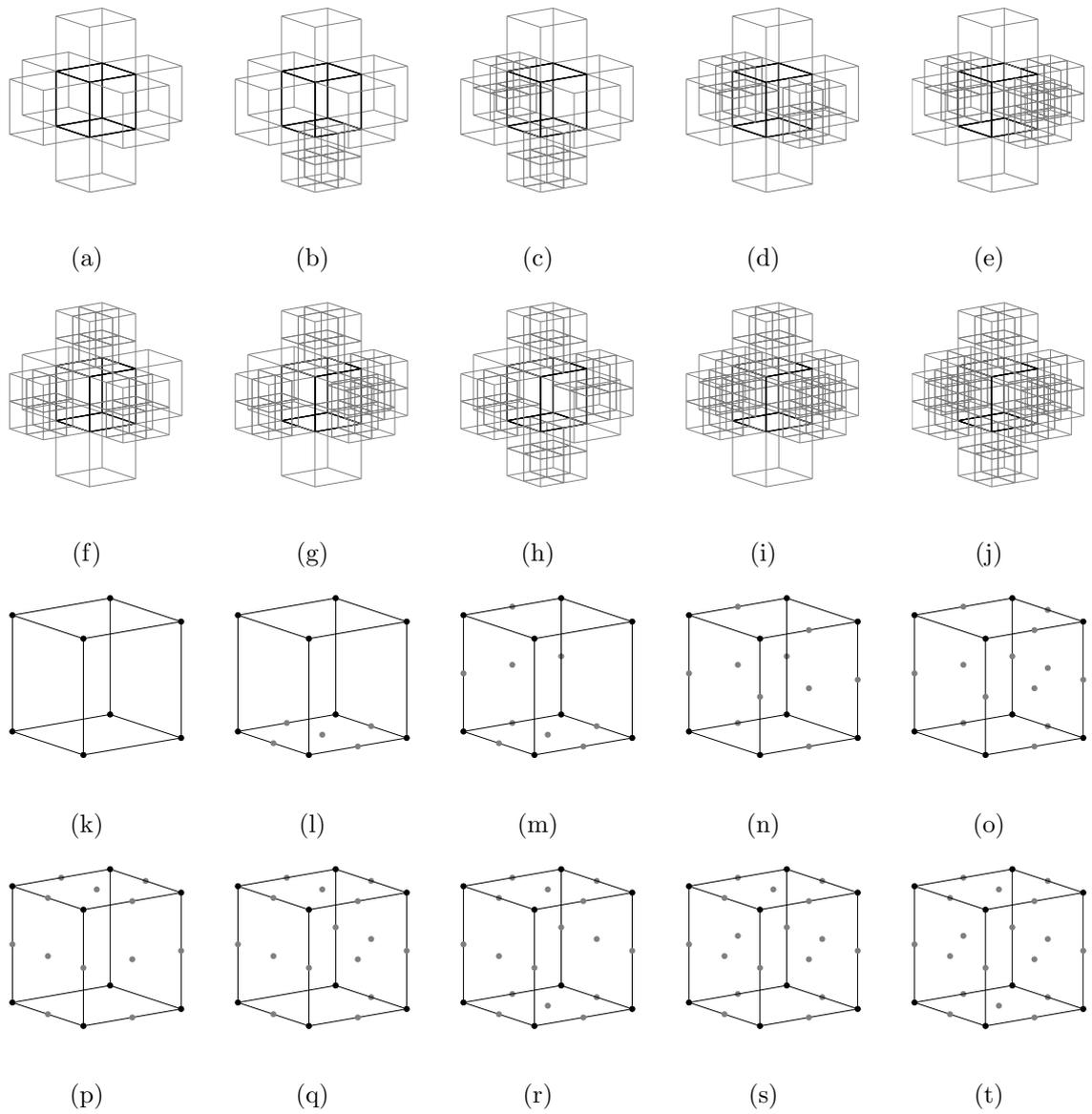


Figure 3.33: *Illustration showing the vertices of a 3-D leaf node (black circles) in a e-balanced octree and the extra vertices (gray circles) needed to match its split e-neighbors in each of the ten possible non-isomorphic 3-D leaf node configurations.*

Figure 3.32). Hence, it is sufficient to prove the quality bounds on the triangular elements obtained from Delaunay triangulation of these six types leaf nodes. The Delaunay triangulations of these six configurations are shown in Figure 3.34, where triangular elements are assigned unique colors based on their shape. Inspection of the Delaunay meshes for six leaf node configurations reveals only four template triangular elements. These four template triangles are shown in Figure 3.35 and their respective geometry and quality metrics: circumradius(r_{circ}), edge length (\bar{e}), radius-to-edge-ratio (θ), and minimum internal angles (α_{min}) are given in Table 3.4. (Note that these results are reported assuming the leaf node size to be two units.) Results indicate that the maximum radius-to-edge ratio, Θ , for the entire mesh generated by the IMESH is at most $\frac{\sqrt{5}}{2}$ and the minimum internal angle α is at least $\arcsin \frac{1}{\sqrt{5}} = 26.565^\circ$. These bounds guarantee that IMESH will not generate needle-shaped and cap-shaped elements in two-dimensions. \square



Figure 3.34: *Delaunay triangulation of six possible quadnode configurations in a balanced quadtrees decomposition. The triangles with same color in the decomposition are similar.*

The analysis for the 2-D meshes can be extended to 3-D meshes where a values of $\Theta(GM)$ guarantees the absence of all bad tetrahedral elements except slivers.

Corollary 3.5.4. *In 3-D, the output mesh (QM) from the IMESH method bounds the radius-to-edge ratio, $\Theta(QM)$, to at most $\frac{\sqrt{29}}{4}$ which is less than 2.*

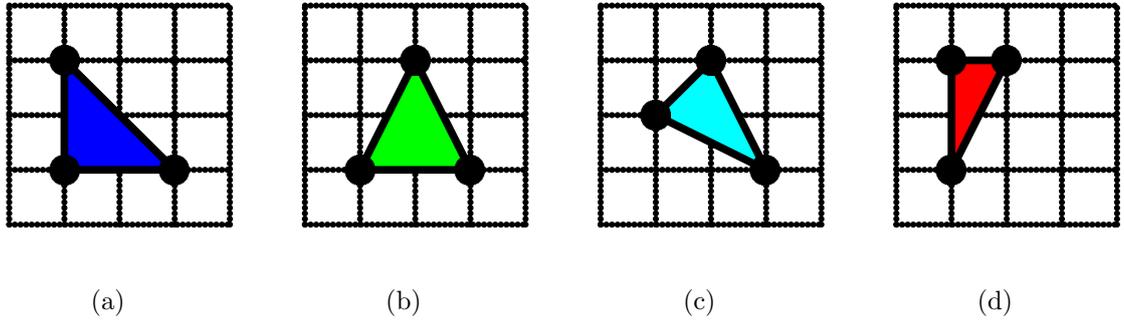


Figure 3.35: Four prototype triangles deduced from the set of Delaunay triangulation of nodes in a balanced quadtree shown in Figure 3.34.

Table 3.4: Summary of the properties of the template triangles in the 2-D mesh from IMESH method.

Template Triangle	r_{circ}	\bar{e}	θ	α_{min}
a (Figure 3.35(a))	$\sqrt{2}$	2, 2, $2\sqrt{2}$	$\frac{1}{\sqrt{2}}$	45°
b (Figure 3.35(b))	$\frac{5}{4}$	2, $\sqrt{5}$, $\sqrt{5}$	$\frac{5}{8}$	53.13°
c (Figure 3.35(c))	$\frac{5\sqrt{2}}{6}$	$\sqrt{2}$, $\sqrt{5}$, $\sqrt{5}$	$\frac{5}{6}$	36.88°
d (Figure 3.35(d))	$\frac{5}{\sqrt{2}}$	1, 2, $\sqrt{5}$	$\frac{\sqrt{5}}{2}$	26.565°

Proof. In 3-D, since there exists only ten non-isomorphic leaf node arrangements that will appear in the e -balanced octree (refer to Figure 3.33). In similarity to the two-dimensional case, the Delaunay tetrahedralization of these configurations produces only eleven (11) unique template tetrahedras shown in Figure 3.36. The respective properties (r_{crit} , \bar{e} , and θ) of all template tetrahedras are summarized in Table 3.5 and indicate a maximum radius-to-edge ratio of $\frac{\sqrt{29}}{4}$. \square

Applying this exhaustive list technique to e -balanced 2^d -trees of higher dimensions leads readily to a generalized quality metric for output mesh from IMESH algorithm.

Corollary 3.5.5. *For arbitrary dimensions, $\mathbb{R}^d, 4 \leq d \leq 8$, the output mesh QM from the IMESH method has the maximum radius-to-edge ratio, $\Theta(QM)$, bounded by $\frac{\sqrt{(d+2)^2+2^2}}{4}$.*

Proof. In higher dimensions, up to \mathbb{R}^8 , the number of non-isomorphic leaf node configurations is computed using the relationships from Results 3.5.1 and 3.5.2. Once these configurations are known, the vertices of each of the leaf node from the configurations are used for computing the Delaunay mesh using the QDELAUNAY routine. The radius-to-edge ratio for entire output mesh, Θ , in $\mathbb{R}^d, 4 \leq d \leq 8$ is then calculated for all identified unique simplicial templates. The ΘQM values so obtained for output meshes in $\mathbb{R}^d, 4 \leq d \leq 8$ are: $\frac{\sqrt{10}}{2}$, $\frac{\sqrt{53}}{4}$, $\frac{\sqrt{17}}{2}$, $\frac{\sqrt{85}}{4}$, and $\frac{\sqrt{26}}{2}$, respectively. In other words,

$$\Theta(QM) = \frac{\sqrt{(d+2)^2+2^2}}{4}, \quad 4 \leq d \leq 8 \quad (3.11)$$

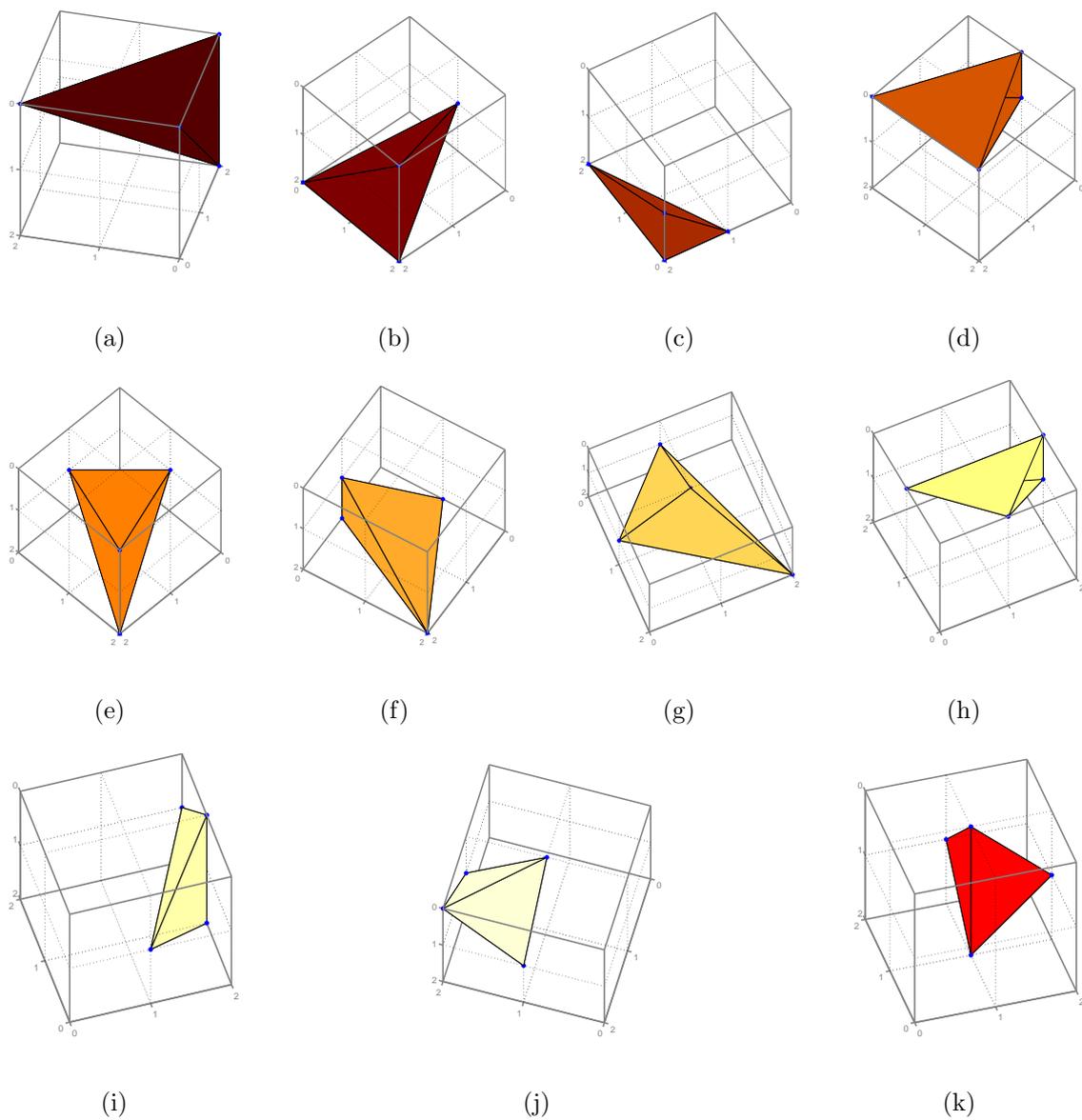


Figure 3.36: *Eleven unique prototype tetrahedra obtained from the Delaunay tetrahedralization of all non-isomorphic 3-D node configurations shown in Figures 3.33(k)–3.33(t).*

Table 3.5: Summary of the properties of the template tetrahedras in the 3-D mesh from IMESH method.

Template Tetrahedra	r_{circ}	\bar{e}	θ
a (Figure 3.36(a))	$\sqrt{3}$	$2, 2, 2, \sqrt{8}, \sqrt{8}, \sqrt{12}$	$\frac{\sqrt{3}}{2}$
b (Figure 3.36(b))	$\frac{3}{2}$	$2, 2, \sqrt{6}, \sqrt{6}, \sqrt{6}, \sqrt{8}$	$\frac{3}{4}$
c (Figure 3.36(c))	$\sqrt{\frac{3}{2}}$	$1, 1, \sqrt{2}, 2, \sqrt{5}, \sqrt{5}$	$\sqrt{\frac{3}{2}}$
d (Figure 3.36(d))	$\frac{\sqrt{29}}{4}$	$1, 2, \sqrt{5}, \sqrt{5}, \sqrt{6}, \sqrt{6}$	$\frac{\sqrt{29}}{4}$
e (Figure 3.36(e))	$\frac{\sqrt{17}}{3}$	$\sqrt{2}, 2, \sqrt{6}, \sqrt{6}, \sqrt{6}, \sqrt{6}$	$\frac{\sqrt{17}}{3\sqrt{2}}$
f (Figure 3.36(f))	$\frac{\sqrt{59}}{6}$	$1, \sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{5}, \sqrt{6}$	$\frac{\sqrt{59}}{6}$
g (Figure 3.36(g))	$\frac{\sqrt{27}}{4}$	$\sqrt{2}, \sqrt{2}, \sqrt{2}, \sqrt{6}, \sqrt{6}, \sqrt{6}$	$\frac{\sqrt{27}}{4\sqrt{2}}$
h (Figure 3.36(h))	$\frac{\sqrt{5}}{2}$	$1, \sqrt{2}, \sqrt{2}, \sqrt{3}, 2, \sqrt{5}$	$\frac{\sqrt{5}}{2}$
i (Figure 3.36(i))	$\frac{\sqrt{5}}{2}$	$\sqrt{2}, \sqrt{2}, \sqrt{3}, \sqrt{3}, 2, \sqrt{5}$	$\frac{\sqrt{5}}{2\sqrt{2}}$
j (Figure 3.36(j))	$\frac{\sqrt{3}}{2}$	$\sqrt{2}, \sqrt{2}, \sqrt{2}, \sqrt{2}, \sqrt{2}, \sqrt{2}$	$\frac{\sqrt{3}}{2\sqrt{2}}$
k (Figure 3.36(k))	1	$\sqrt{2}, \sqrt{2}, \sqrt{2}, \sqrt{2}, \sqrt{2}, 2$	$\frac{1}{\sqrt{2}}$

Combining the quality bounds obtained earlier for 2-D and 3-D with the result obtained above produces a quality bound formula for the IMESH method, valid at least for all hyper-spaces with less than nine dimensions:

$$\Theta(QM) = \frac{\sqrt{(d+2)^2 + 2^2}}{4}, \quad 2 \leq d \leq 8 \quad (3.12)$$

□

Theorem 3.5.6. *For any \mathbb{R}^d image data, $S(x_1, x_2, \dots, x_d : x_i \in [0, 2^{n-1}])$, the size of the output mesh ($|QM|$) from the IMESH method is bounded by $O(2^{nd})$.*

Proof. The worst case scenario for IMESH is that of an input image that is entirely non-homogeneous (for example, chess board). In this case, the 2^d -tree is full (no nodes are pruned) and the number of leaf nodes is 2^{nd} . Since the leaf node of a full 2^d -tree can be meshed into a constant number of simplices (e.g., two triangles in 2-D and five or six in tetrahedras in 3-D), the size of the output mesh, $|QM|$, from the IMESH algorithm is at most $O(2^{nd})$. □

3.6 Conclusions and Future Work

In this chapter, a new unstructured finite element mesh generation technique for image data was developed and analyzed. This new method, IMESH, was developed by modifying and extending the 2-D PSLG technique of Bern et al. (1990), so that it would apply to image data and arbitrary dimensions. The new algorithm consists of four steps: (1) construct a 2^d -tree for the input multi-dimensional dyadic image; (2) generate a ϵ -balanced 2^d -tree by imposing the 2:1 balancing rule; (3) generate

the unique vertices of the leaf nodes in e -balanced 2^d -tree; and (4) Delaunay mesh the unique vertices from the previous step.

The application of IMESH to a 2-D segmentation landscape image and a 3-D segmented MR brain scan demonstrated its efficiency, adaptivity, quality and boundary preservation characteristics. The method generates significantly less elements than would be produced by direct meshing of individual pixels. It generates heterogeneity adapted meshes with a large density of small elements in zones where the source image is highly heterogeneous and a low density of larger elements in homogeneous areas. All elements produced by the algorithm further respect the original material boundaries of the source image such that this image can be recovered from the mesh (i.e., the method is reversible, it is bi-directional). These characteristics of IMESH make it particularly suitable for the generation of meshes for finite element analysis of transport process in domains represented by segmented biological images.

The IMESH technique was implemented by adapting various algorithms presented in the literature for two and three-dimensions, and to improve their performance. The region quadtree data structure in Samet (1989) was modified to accommodate multi-dimensional image data and to efficiently retrieve the vertices of the leaf nodes by using the pointer-based 2^d -tree traversals. The e -balancing algorithm of Sivan was modified with a size-filtering step that improved its performance by factors of 2 to 200. The quicksort and QHULL algorithms were used as is. IMESH was implemented in Java SDK 1.4.1 to enable its use on a wide variety of computing platforms.

Exact theoretical bounds on the quality of elements produced by IMESH were derived by formal mathematical analysis and exhaustive listing of non-isomorphic leaf nodes and template simplices. Results were obtained for input image in 2 to 8 spatial dimensions and demonstrated that IMESH is guaranteed to always produce good elements (and not a single bad element) in both 2-D and 3-D.

The mesh generation method developed in this dissertation can generate multi-scale unstructured meshes and uses hierarchical data structures identical to those of the SWA image segmentation technique developed in Chapter 2. Hence, the mesh generation method and the texture-based segmentation method can be integrated seamlessly for analysis of heterogeneous bio-systems using numerical up-scaling techniques. The method can also be used as a independent tool for generating quality meshes from image data. Based on the above results, it is concluded that the four-step IMESH image meshing method is a valuable tool for developing simplicial meshes from 2-D and 3-D image data to support finite element analysis of dynamic process occurring in heterogeneous bioenvironments. It is efficient, adaptive, reversible, generates quality-guaranteed elements, can run on most modern computational platforms, and can be seamlessly integrated with the multi-scale SWA image segmentation method.

Several avenues of future research remain open for improving IMESH. The first is to minimize the total number of elements at the boundaries. The method developed in this chapter generates a large number of elements at material boundaries. This is a desirable feature when solving PDEs using finite elements, as one expects sharp changes in the numerical solution near the boundaries. However, too many

elements can overburden some computational resources and it could be valuable to investigate de-refinement strategies, as a post-processing step, to reduce the number of small elements at material interfaces.

Secondly, the FE mesh from the IMESH algorithm exhibits some level of preferred direction by aligning the elements along the principal axes of the input image. In some applications (for instance, aerodynamics simulations) this can pose problems. Investigating mesh improvements steps (Neugebauer and Dickmann, 1996) to reduce such right-angled elements would possibly be a worthwhile endeavor.

Thirdly, it would be interesting to study the computational complexity of using *dictionary method* (Gross et al., 1996), as opposed to Delaunay triangulation, for meshing of leaf nodes in the ϵ -balanced 2^d -tree. One of the possible benefits of using dictionary method would be to avoid the floating point operations associated with Delaunay construction that can be computationally demanding. Since step 3 of IMESH generates a finite number of template simplices, the triangulation of leaf node can be obtained by using look-up and conformity tables. This could lead to further computational improvements in the technique.

The IMESH method is capable of producing multi-scale meshes because of the hierarchical spatial sub-division scheme used for generating the FE mesh. However, this capability was not used in the present study. Meshes at coarser scales can be constructed efficiently by using the finer scale 2^d -tree representation of the image data along with Delaunay point deletion schemes (Devillers, 1999; Mostafavi et al., 2003) or dictionary methods. Such multi-scale meshes could then be used in conjunction with multi-grid finite element methods to produce highly efficient

simulations of bio-process transport dynamics.

Finally, the rapid advances in both algorithms and hardware appear to be approaching some intrinsic limit. Current implementations are reaching the point of diminishing returns in terms of trading off computational cost for accuracy. In this context, the IMESH method, which is intrinsically parallelizable (Bern et al., 1999; Reeve and Barragan, 2000a,b), provides an efficient solution to address this problem. The benefits from implementing a parallel version of IMESH would be enormous and would serve the scientific community even better. The availability of parallel programming libraries, like *message passing interface* (MPI, Gropp et al. (1994)) and *parallel virtual machine* (PVM, Geist et al. (1994)), should make it feasible to implement a parallel version of IMESH on high performance computing environments.

CHAPTER 4: CONCLUSION

The quantitative analysis of processes occurring in heterogeneous biological environments can benefit significantly from new development in image sensors that provide high-resolution non-invasive measurements of their spatial characteristics. One promising approach to the type of analysis consists of: (1) acquiring image data about the environment of interest; (2) segmenting the image data into the distinct materials contained in that environment; (3) developing a simplicial mesh over the segmented data, and; (4) performing a simulation of a process of interest, on the mesh, using the finite element method.

The goal of this research was to contribute to the development of this promising approach by developing computational tools that could be used to efficiently perform the second and third steps. The specific objectives were to:

Objective 1: Develop a supervised texture-based segmentation technique that can quantitatively characterize the heterogeneity of a material at a coarser scale by accounting for the heterogeneity from sub-scales.

Objective 2: Evaluate the developed technique for segmenting heterogeneous materials in biological image data.

Objective 3: Develop and implement a quality guaranteed unstructured simplicial mesh generation technique for (segmented) image data.

Objective 4: Analyze the developed unstructured simplicial mesh generation tech-

nique for elemental quality and output size.

The method used to attain these objectives was to: (1) review the literature on image segmentation methods and mesh generation techniques; (2) identify the best existing methods for these operations along with their shortcomings relative to the needs of their research; (3) develop new methods (where needed) and adapt existing techniques (where appropriate); (4) implement the new and modified methods in modern computing languages, and; (5) analyze and evaluate the new methods for efficiency, accuracy and quality over 2-D and 3-D image data of bioenvironmental and biomedical relevance.

This process resulted in a new image segmentation technique that outperforms existing methods when applied to biological image data, and a new unstructured mesh generation technique that is density adaptive and guaranteed to produce quality elements for such data. The specific contributions to knowledge made by this study were:

1. Multiresolution analysis, using Haar wavelets, can be used to efficiently identify the representative elementary scale of the materials in a heterogeneous biological image.
2. Spatial moments of pixel intensities, computed at a pre-identified representative elementary scale, form efficient feature vectors for the segmentation of heterogeneous biological images in 2-D.
3. The combination of RES identification by Haar MRA with spatial moments

as feature vectors produces a segmentation method (named SWA) that can efficiently and accurately classify heterogeneous biological images in 3-D.

4. The 2-D meshing algorithm of (Bern et al., 1990) can be extended to produce efficient heterogeneity adaptive, boundary preserving multi-dimensional meshes on segmented bio-image data. (This novel method is named IMESH).
5. The developed multi-dimensional unstructured mesh generation technique, IMESH, is guaranteed to produce quality elements in both 2-D and 3-D.

The SWA image segmentation method and the IMESH unstructured mesh generation technique meet the objectives that were set out for this research. Furthermore, they are the only tools in existence, today, that meet these objectives. It is therefore concluded that this study was successful in contributing to the development of the promising integrated approach for the quantitative analysis of processes occurring in heterogeneous biological environments described earlier. This study contributed two major components required by such an approach.

The development performed in this research should now provide significant help in studying the dynamics of various processes in complicated bio-systems. Figure 4.1 depicts possible future developments in which the SWA segmentation and IMESH are integrated with other tools to support quantitative analysis of complex bio-systems. In the bio-medical related areas such integrated systems could be used for: (a) developing better medical diagnostic tools, like early detection of tumor growth; (b) developing efficient drug delivery systems; or (c) developing novel therapies and treatment strategies. Similarly, these integrated systems should be able

to assist the soil and water resources research community: (a) to more accurately identify the hot spots in watersheds with high nutrient and non-point source pollutants concentration; (b) to more accurately understand the problems of nutrient and pollutant transport in soils systems; and (c) to prescribe best management practices for reducing the nutrient and pollutant loads in target hot spots. Such systems could also help analyze thermal process dynamics in heterogeneous food products and the population dynamics of mobile agents over heterogeneous landscapes.

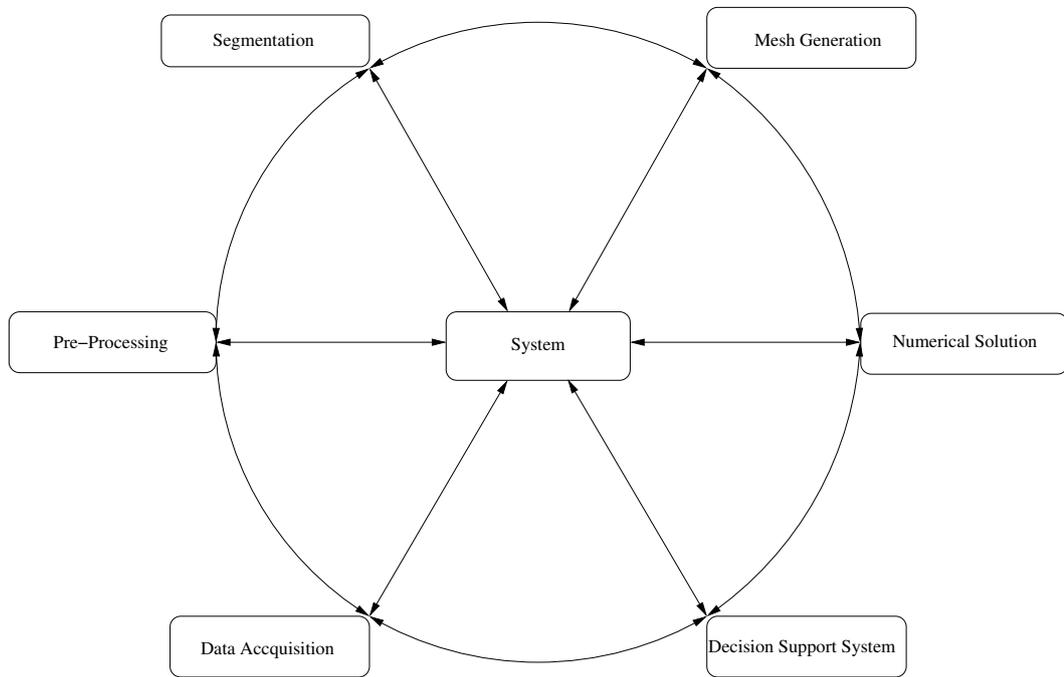


Figure 4.1: *Schematic of an integrated tool for analysis of complex bio-systems.*

APPENDIX A: EFFECT OF RES ON SWA CLASSIFICATION ACCURACY

In this appendix, the developed SWA segmentation technique in Chapter 2 is applied on the two-texture synthetic image presented earlier in Figure 2.14(a). The objective is to demonstrate the effect of the size of sample tiles (below and above the previously identified RES support of 16×16 pixels) on classification accuracy. Results are presented in Figure A.1 and they clearly demonstrate that very high error levels are attained when the tile size is very small. Increasing tile size leads to lower classification error (expressed in ϵ_l) until RES is reached, after which the error metric stabilizes.

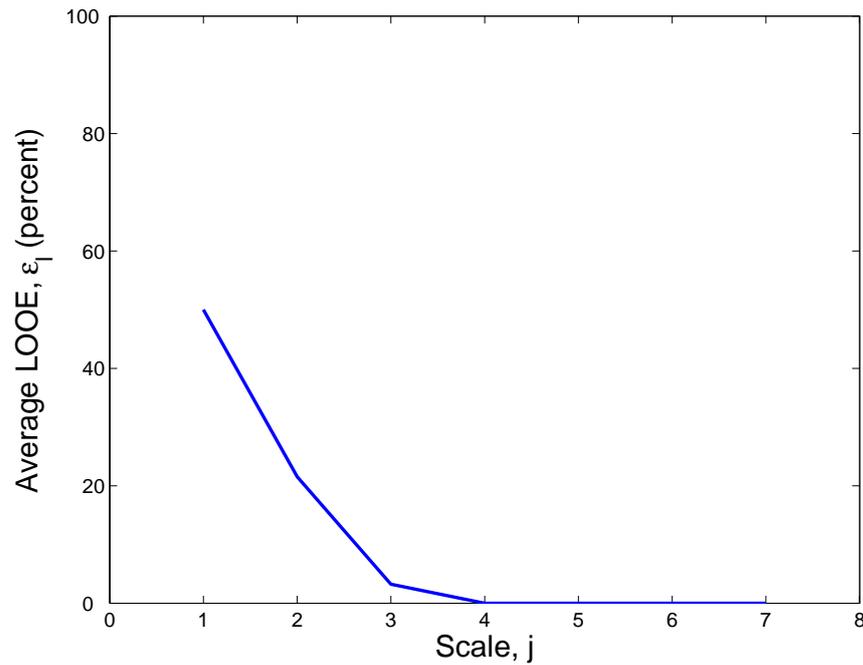


Figure A.1: *Effect of RES support on the SWA classification accuracy for the two-textured mosaic shown in Figure 2.14.*

BIBLIOGRAPHY

- Antonie, J.-P., Murezni, R., and Vandergheynst, P. (1999). Directional wavelets revisited: Cauchy wavelets and symmetry detection in patterns. *Applied and Computational Harmonic Analysis*, 6(3), 314–345.
- Archibald, R., Chan, K., Gelb, A., and Renaut, R. (2003). Improving tissue segmentation of human brain MRI through pre-processing by the Gegenbauer reconstruction method. *NeuroImage*, 20(9), 489–502.
- Babuska, I. and Aziz, A. K. (1976). On the angle condition in the finite element method. *SIAM Journal of Numerical Analysis*, 13, 214–226.
- Baehmann, P. L., Wittchen, S. L., Shephard, M. S., Grice, K. R., and Yerry, M. A. (1987). Robust, geometrically based, automatic two-dimensional mesh generation. *International Journal for Numerical Methods in Engineering*, 24, 1043–1078.
- Bajaj, C. L., Coyle, E. J., and Lin, K.-N. (1996). Arbitrary topology shape reconstruction from planar cross sections. *Graphics Models and Image Processing*, 58(6), 524–543.
- Bajaj, C. L., Coyle, E. J., and Lin, K.-N. (1999). Tetrahedral meshes from planar cross sections. *Computer Methods in Applied Mechanics and Engineering*, 179, 31–52.
- Baker, B. S., Grosse, E., and Rafferty, C. S. (1988). Nonobtuse triangulation of polygons. *Discrete Computational Geometry*, 3, 147–168.

- Baker, T. J. (1989). Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation. *Engineering with Computers (USA)*, 5(3-4), 161-175.
- Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4), 469-483.
- Barth, T. J. and Jespersen., D. C. (1989). The design and application of upwind schemes on unstructured meshes. In *Proceedings of the 27th AIAA Aerospace Sciences Meeting* (pp. Paper No. 89-0366).
- Bastian, P., Chen, Z., Ewing, R., Helmig, R., Jakobs, H., and Reichenberger, V. (2000). Numerical solution of multiphase flow in fractured porous media. In Z. Chen, R. Ewing, and Z. Shi (Eds.), *Numerical treatment of multiphase flows in porous media*, volume 152 of *Lecture Notes in Physics* (pp. 50-68). New York, NY, USA: Springer-Verlag.
- Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2), 105-139.
- Bear, J. (1972). *Dynamics of Fluids in Porous Media*. New York, NY, USA: American Elsevier Publishing Company.
- Beck, J., Sutter, A., and Ivry, R. (1987). Spatial frequency channels and perceptual grouping in texture segregation. *Computer Vision, Graphics, and Image Processing*, 37(2), 299-325.

- Bern, M. W., Chew, L. P., Eppstein, D., and Ruppert, J. (1995). Dihedral bounds for mesh generation in high dimensions. In *Proceedings of 6th Symposium on Discrete Algorithms* (pp. 189–196).: ACM and SIAM.
- Bern, M. W., Eppstein, D., and Gilbert, J. R. (1990). Provably good mesh generation. In *IEEE Symposium on Foundations of Computer Science* (pp. 231–241).
- Bern, M. W., Eppstein, D., and Teng, S.-H. (1999). Parallel construction of quadtrees and quality triangulations. *International Journal of Computational Geometry and Applications*, 9(6), 517–532.
- Bern, M. W., Eppstein, D., and Yao, F. F. (1991). The expected extremes in a Delaunay triangulation. *International Journal of Computational Geometry and Applications*, 1(1), 79–92.
- Bossen, F. J. (1996). *Anisotropic Mesh Generation with Particles*. Technical Report CMU-CS-96-134, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA.
- Bovik, A. C., Clarke, M., and Geisler, W. S. (1990). Multichannel Texture Analysis Using Localized Spatial Filters. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 12(1), 55–73.
- Bowyer, A. (1981). Computing Dirichlet tessellations. *The Computer Journal*, 24, 162–166.
- Brodatz, P. (1966). *Texture—A Photographic Album for Artists and Designers*. New York, NY, USA: Dover Publications.

- Brown, K. Q. (1979). Voronoï diagrams from convex hulls. *Information Processing Letters*, 9(5), 223–228.
- Buratynski, E. (1990). A fully automatic three-dimensional mesh generator for complex geometries. *International Journal for Numerical Methods in Engineering*, 30, 931–952.
- Burt, P. J. and Adelson, E. H. (1983). The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31(4), 532–540.
- Castro-Diaz, M. J., Hecht, F., and Mohammadi, B. (1995). New progress in anisotropic grid adaptation for inviscid and viscid flows simulations. In *Proceedings of 4th International Meshing Roundtable* (pp. 73–86). Albuquerque, NM, USA: Sandia National Laboratories.
- Cawley, G. C. and Talbot, N. L. (2003). Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. *Pattern Recognition*, 36(11), 2585–2592.
- Cebral, J. R. and Löhner, R. (1999). From medical images to CFD meshes. In *Proceedings of the 8th International Meshing Roundtable, South Lake Tahoe, CA, USA, October 10 –13, 1999* (pp. 321–331). Albuquerque, NM, USA: Sandia National Laboratories.
- Center for Morphometric Analysis, M. G. H. (2004). Internet Brain Segmentation Repository. Available from <http://www.cma.mgh.harvard.edu/ibsr>, Accessed on 7th May.

- Chang, T. and Kuo, C. C. J. (1993). A texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions of Image Processing*, 2(4), 429–441.
- Chaudhuri, B. and Sarkar, N. (1995a). Texture segmentation using fractal dimension. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 17(1), 72–77.
- Chaudhuri, B. and Sarkar, N. (1995b). Texture segmentation using fractal dimension. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 17(1), 353–370.
- Chellappa, R. and Chatterjee, S. (1985). Classification of textures using Gaussian Markov random fields. *IEEE Transactions on Acoustics Speech and Signal Processing*, 33(8), 959–963.
- Chellappa, R., Kashyap, R. L., and Manjunath, B. S. (1992). Model based texture segmentation and classification. In C. H. Chen, L. F. Pau, and P. S. P. Wang (Eds.), *Handbook of Pattern Recognition and Computer Vision* chapter 2.2, (pp. 277–310). Singapore: World Scientific Publishing Company.
- Chen, C. and Chen., D. (1996). Multiresolutional Gabor filter in texture analysis. *Pattern Recognition Letters*, 17(10), 1069–1076.
- Chew, L. P. (1986). *Building Voronoï diagrams for convex polygons in linear expected time*. Technical Report PCS-TR90-147, Department of Mathematics and Computer Science, Dartmouth College, Hanover, NH, USA.

- Chew, L. P. (1989). Constrained Delaunay triangulations. *Algorithmica*, 4, 97–108.
- Cignoni, P., Floriani, L. D., Montani, C., Puppo, E., and Scopigno, R. (1994). Multiresolution modeling and visualization of volume data based on simplicial complexes. In *Proceedings of the 1994 symposium on Volume visualization (VVS '94)*, Tysons Corner, VA, USA, October 17–18, 1994 (pp. 19–26).: ACM Press.
- Clarkson, K. L. and Shor, P. W. (1989). Applications of random sampling in computational geometry, II. *Discrete Computational Geometry*, 4(1), 387–421.
- Clunie, D. (2002). Medical image format faq—part 3: DICOM information sources. Available from <http://www.dclunie.com/medical-image-faq/html/part3.html>, Accessed on 10 August.
- Cocosco, C., Kollokian, V., Kwan, R. K.-S., and Evans, A. C. (1997). BrainWeb: Online interface to a 3D MRI simulated brain database. *NeuroImage*, 5(4), S425.
- Collins, D., Zijdenbos, A., Kollokian, V., Sled, J., Kabani, N., Holmes, C., and Evans, A. (1998). Design and construction of a realistic digital brain phantom. *IEEE Transactions on Medical Imaging*, 17(3), 463–468.
- Collins, D. L., Neelin, P., Peter, T. M., and Evans, A. C. (1994). Automatic 3D registration of MR volumetric data in standardized Talairach space. *Journal Of Computer Assisted Tomography*, 18, 192–205.
- Comtet, L. (1974). *Advanced Combinatorics: The Art of Finite and Infinite Expansions*. Dordrecht, Netherlands: Reidel Publishing Company.

- Coxeter, H. S. M. (1973). *Poincaré's Proof of Euler's Formula*, chapter 9, (pp. 165–172). Dover: New York, NY, USA, 3rd edition.
- Cross, G. R. and Jain, A. K. (1983). Markov Random Field Texture Models. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 5(1), 25–39.
- Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America*, 2, 1160–1169.
- Daugman, J. G. (1990). An information-theoretic view of analog representation in Striate cortex. In E. L. Schwartz (Ed.), *Computational Neuroscience* (pp. 403–424). Cambridge, MA, USA: MIT Press.
- Desco, M., Gispert, J., Reig, S., Santos, A., Pascau, J., Malpica, N., and Garcia-Barreno, P. (2001). Statistical segmentation of multidimensional brain datasets. In M. Sonka and K. M. Hanson (Eds.), *Proceedings of SPIE Medical Imaging 2001: Image Processing*, volume 4322 (pp. 184–193).
- Devillers, O. (1999). On deletion in delaunay triangulations. In *Proceedings of the 15th annual Symposium on Computational Geometry (SCG '99), Miami Beach, FL, USA, June 13–1, 1999* (pp. 181–188): ACM Press.
- Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3), 297–302.
- Do, M. N. (2001). *Directional Multiresolution Image Representations*. PhD thesis,

Department of Communication Systems, Swiss Federal Institute of Technology,
Lausanne, Switzerland.

Dougherty, E., Pelz, J., Sand, F., and Lent, A. (1992). Morphological image segmentation by local granulometric size distributions. *Journal of Electronic Imaging*, 1(1), 46–60.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. New York, NY, USA: Wiley-Interscience Publication, 2nd edition.

Dunn, D. F. and Higgins, W. E. (1995). Optimal Gabor filters for texture segmentation. *IEEE Transactions of Image Processing*, 4(7), 947–964.

Dwyer, R. A. (1987). A faster divide-and-conquer algorithm for constructing Delaunay triangulations. *Algorithmica*, 2, 137–151.

E., W. and Engquist, B. (2003). The heterogeneous multiscale methods. *Communications in Mathematical Sciences*, 1(1), 87–132.

Edelsbrunner, H. and Mücke, E. P. (1994). Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1), 43–72.

Edelsbrunner, H. and Shah, N. R. (1996). Incremental topological flipping works for regular triangulations. *Algorithmica*, 15, 223–241.

Edelsbrunner, H. and Tan, T. S. (1993). An upper bound for conforming Delaunay triangulations. *Discrete Computational Geometry*, 10(2), 197–213.

- Fang, T.-P. and Piegl, L. A. (1992). Algorithm for Delaunay triangulation and convex-hull computation using sparse matrix. *Computer Aided Design*, 24(8), 425–436.
- Fang, T.-P. and Piegl, L. A. (1993). Delaunay triangulation using a uniform grid. *IEEE Computer Graphics and Applications*, 13(3), 36–47.
- Farrokhnia, F. (1990). *Multi-channel Filtering Techniques for Texture Segmentation and Surface Quality Inspection*. PhD thesis, Michigan State University, East Lansing, MI, USA.
- Fatemi-Ghomi, N., Palmer, P., and Petrou, M. (1996). Performance evaluation of texture segmentation algorithms based on wavelets. In *Proceedings of the Workshop on Performance Characteristics of Vision Algorithms (ECCV'96)*, Cambridge, UK, April 1996.
- Feil, M. and Uhl, A. (2002). Wavelet packet image decomposition on mind architectures. *Real-Time Imaging*, 8(5), 399–412.
- Fitzpatrick, E. A. (1983). *Soils: Their Formation, Classification, and Distribution*. New York, NY, USA: Longman.
- Fogel, S. and Sagi, D. (1989). Gabor filters as texture discriminators. *Biological Cybernetics*, 61(2), 103–113.
- Fortune, S. (1993). Computational geometry. In R. Martin (Ed.), *Directions in Computational Geometry*. Information Geometers.

- Fortune, S. (1997). Voronoi diagrams and Delaunay triangulations. In J. E. Goodman and J. O'Rourke (Eds.), *Handbook of Discrete and Computational Geometry* chapter 20, (pp. 377–388). Boca Raton, FL, USA: CRC Press LLC.
- Fortune, S. J. (1987). A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2, 153–174.
- Freeborough, P. A. and Fox, N. C. (1998). MR image texture analysis applied to the diagnosis and tracking of Alzheimer's disease. *IEEE Transactions on Medical Imaging*, 17(3), 475–478.
- Frey, P., Sarter, B., and Gautherie, M. (1994). Fully automatic mesh generation for 3d domains based upon voxel sets. *International Journal for Numerical Methods in Engineering*, 37(16), 2735–2754.
- Frey, P. J., Borouchaki, H., and George, P.-L. (1996). Delaunay tetrahedrization using an advancing-front approach. In *Proceedings of the 5th International Meshing Roundtable* (pp. 31–43). Albuquerque, NM, USA: Sandia National Laboratories.
- Fu, K. S. (1982). *Syntactic Pattern Recognition and Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall.
- Fujishiro, I., Maeda, Y., Sato, H., and Takeshima, Y. (1996). Volumetric data exploration using interval volume. *IEEE Transactions on Visualization and Computer Graphics*, 2(2), 144–155.
- Gama, J. (2000). Combining classification algorithms. *Aicommunications*, 13(2), 135–136.

- Gama, J. and Brazdil, P. (2000). Cascade generalization. *Machine Learning*, 41(3), 315–343.
- Garrity, M. P. (1990). Ray tracing irregular volume data. In *ACM Computer Graphics (Proceedings of 1990 Workshop on Volume Visualization, San Diego, CA, USA)*, volume 24(5) (pp. 35–40).
- Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V. (1994). *PVM: Parallel virtual machine: a users' guide and tutorial for networked parallel computing*. Cambridge, MA, USA: MIT Press.
- González, P., Cabaleiro, J. C., and Pena, T. F. (2001). Parallel computation of wavelet transforms using the lifting scheme. *The Journal of Supercomputing*, 18(2).
- Greenspan, H., Goodman, R., Chellappa, R., and Anderson, C. H. (1994). Learning texture discrimination rules in a multiresolution system. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 16(9), 894–901.
- Gropp, W., Lusk, E., and Skjellum, A. (1994). *Using MPI: portable parallel programming with the message-passing interface*. Cambridge, MA, USA: MIT Press.
- Gross, M. H., Stadt, O. G., and Gatti, R. (1996). Efficient triangular surface approximations using wavelets and quadtree data structures. *IEEE Transactions on Visualization and Computer Graphics*, 2(2), 130–143.
- Gudla, P. R. (2005). Imesh: a finite element mesh generation package for im-

- ages. Available from <http://www.glue.umd.edu/~reddyg/imesh>, Accessed on 10th February.
- Guibas, L. J., Knuth, D. E., and Sharir, M. (1992). Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 7, 381–413.
- Guibas, L. J. and Stolfi, J. (1985). Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4(2), 74–123.
- Hale, D. (2001). Atomic images—a method for meshing digital images. In *Proceedings of Tenth International Meshing Roundtable* (pp. 185–196): Sandia National Laboratories.
- Haley, G. and Manjunath, B. (1995). Rotation-invariant texture classification using modified Gabor filters. In *Proceedings of Second IEEE International Conference on Image Processing*, volume 1 (pp. 262–265). Washington D.C.
- Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proceedings of IEEE*, 67(5), 768–804.
- Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6), 610–621.
- Herzen, B. v. and Barr, A. (1987). Accurate triangulations of deformed, intersecting surfaces. *Computer Graphics*, 21, 103–110.

- Hoare, C. A. R. (1962). Quicksort. *The Computer Journal*, 5(1), 10–16.
- J. Bigün and J. M. H. du Buf (1994). N-folded symmetries by complex moments in Gabor space and their application to unsupervised texture segmentation. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 16(1), 80–87.
- Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Englewood Cliffs, NJ, USA: Prentice-Hall.
- Jain, A. K. and Farrokhnia, F. (1991). Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12), 1167–1186.
- Jain, A. K. and Karu, K. (1996). Learning Texture Discrimination Masks. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 18(2), 195–205.
- Joe, B. (1989). Three-dimensional triangulations from local transformations. *SIAM Journal of Scientific and Statistical Computing*, 10(4), 718–741.
- Joe, B. (1991). Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8(2), 123–142.
- Joe, B. (1993). Construction of k -dimensional Delaunay triangulations using local transformations. *SIAM Journal of Scientific and Statistical Computing*, 14(6), 1415–1436.
- Joe, B. (1995). Construction of three-dimensional improved quality triangulations using local transformations. *SIAM Journal on Scientific Computing*, 16, 1292–1307.

- Ju, T., Losasso, F., Schaefer, S., and Warren, J. (2002). Dual contouring of hermite data. In *Proceedings of SIGGRAPH*, volume 21(3) (pp. 339–346).: ACM Press / ACM SIGGRAPH / Addison Wesley Longman.
- Julesz, B. (1975). Experiments in the visual perception of texture. *Scientific American*, 232(4), 34–43.
- Kaplan, L. (1999). Extended fractal analysis for texture classification and segmentation. *IEEE Transactions of Image Processing*, 8(11), 1572–1585.
- Kaplan, L. M. and Kuo, C.-C. J. (1995). Texture roughness analysis and synthesis via extended self-similar (ESS) model. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 17(11), 1043–1056.
- Kasparis, T., Charalampidis, D., Georgiopoulos, M., and Rolland, J. (2001). Segmentation of textured images based on fractals and image filtering. *Pattern Recognition*, 34(10), 1963–1973.
- Kass, M., Witkin, A., and Trezopoulos, D. (1987). Snakes: active contour models. *International Journal of Computer Vision*, 1, 321–331.
- Kendall, M. and Stuart, A. (1977). *The Advanced Theory of Statistics*, volume 1. New York, NY, USA: MacMillan Publishing Company, Inc., 4th edition.
- Kervrann, C. and Heitz, F. (1995). A Markov random-field model-based approach to unsupervised texture segmentation using local and global spatial statistics. *IEEE Transactions on Image Processing*, 4(6), 856–862.

- Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78, 1464–1480.
- Kwan, R. K.-S., Evans, A. C., and Pike, G. (1999). MRI simulation-based evaluation of image-processing and classification methods. *IEEE Transactions on Medical Imaging*, 18(11), 1085–97.
- Kwan, R. K.-S., Evans, A. C., and Pike, G. B. (1996). An extensible MRI simulator for post-processing evaluation. In *Visualization in Biomedical Computing-VBC'96*, volume 1131 of *Lecture Notes In Computer Sciences* (pp. 135–140). Berlin, Germany: Springer-Verlag.
- Laine, A. and Fan, J. (1993). Texture classification by wavelet packet signatures. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 15(11), 1186–1191.
- Langer, S., Fuller Jr., E., and Carter, W. (2001). OOF: An image-based finite-element analysis of material microstructures. *Computing in Science and Engineering*, 3(3), 15–23.
- Lee, D. T. and Schachter, B. J. (1980). Two algorithms for constructing Delaunay triangulations. *International Journal of Computer and Information Sciences*, 9(3), 219–242.
- Lee, T. S. (1996). Image representation using 2-D Gabor wavelets. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 18(10), 959–971.

- Leemput, K. V., Maes, F., Vandermeulen, D., and Suetens, P. (1999). Automated model-based tissue classification of MR images of the brain. *IEEE Transactions on Medical Imaging*, 18(10), 897–908.
- Leggett, G. J., Davies, M., Jackson, D., and Tendler, S. (1996). Scanning probe microscopy of biomolecules and polymeric biomaterials. *Journal of Electron Spectroscopy and Related Phenomena*, 81(3), 249–268.
- Lerski, R., Straughan, K., Shad, L., Boyce, D., Bluml, S., and Zuna, I. (1993). MR image texture analysis an approach to tissue characterisation. *Magnetic Resonance Imaging*, 11, 873–887.
- Levine, M. D. (1985). *Vision In Man and Machine*. New York, NY, USA: McGraw-Hill.
- Li, L., Mao, F., Qian, W., and Clarke, L. P. (1997). Wavelet transform for directional feature extraction in medical imaging. In *Proceedings of Fourth IEEE International Conference on Image Processing (ICIP'97)*, volume 3 (pp. 500–xxx). Washington, D.C.
- Liang, J., McInerney, T., and Terzopoulos, D. (1999). Interactive medical image segmentation with united snakes. In C. Taylor and A. C. F. Colchester (Eds.), *Proceedings of Second International Conference (MICCAI'99), Cambridge, UK, September 19-22, 1999*, volume 1679 of *Lecture Notes in Computer Science* (pp. 116–127).: Springer-Verlag.

- Lillesand, T. M. and Kiefer, R. W. (1987). *Remote Sensing and Image Interpretation*.
New York, NY, USA: John Wiley and Sons, 2nd edition.
- Lin, K.-N. (1997). *Surface and 3D triangular meshes from planar cross sections*.
PhD thesis, Computer Engineering, Purdue University, West Lafayette, IN, USA.
- Liu, A. and Joe, B. (1994a). Relationship between tetrahedron shape measures.
BIT, 34(2), 268–287.
- Liu, A. and Joe, B. (1994b). Relationship between tetrahedron shape measures.
BIT, 34, 268–287.
- Lo, S. H. (1985). A new mesh generation scheme for arbitrary planar domain.
International Journal for Numerical Methods in Engineering, 21, 1403–1426.
- Lo, S. H. (1991a). Volume discretization into tetrahedra–I. verification and orientation of boundary surfaces. *Computer and Structures*, 39(5), 493–500.
- Lo, S. H. (1991b). Volume discretization into tetrahedra–II. 3D triangulation by advancing front approach. *Computer and Structures*, 39(5), 501–511.
- Löhner, R. and Parikh, P. (1988). Three-dimensional grid generation via the advancing-front method. *International Journal for Numerical Methods in Fluids*, 8, 1135–1149.
- Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. *ACM-SIGGRAPH Computer Graphics*, 21(4), 163–169.

- Lu, C. S., Chung, P. C., and Chen, C. F. (1997). Unsupervised texture segmentation via wavelet transform. *Pattern Recognition*, 30(5), 729–742.
- Mäenpää, T., Pietikäinen, M., and Ojala, T. (2000). Texture classification by multi-predicate local binary pattern operators. In *Proceedings of Fifteenth International Conference on Pattern Recognition (ICPR'00)*, volume 3 (pp. 951–954). Barcelona, Spain.
- Malladi, R. and Sethain, J. A. (1996a). Level set and fast marching methods in image processing and computer vision. In *Proceedings of IEEE International Conference on Image Processing* Lausanne, Switzerland.
- Malladi, R. and Sethain, J. A. (1996b). An $O(n \log n)$ algorithm for shape modeling. *Proceedings of the National Academy of Sciences*, 93, 9389–9392.
- Malladi, R. and Sethain, J. A. (1996c). A unified approach to noise removal, image enhancement, and shape recovery. *IEEE Transactions on Image Processing*, 5(11), 1554–1568.
- Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 11(7), 674–693.
- Manjunath, B., Shekhar, C., and Chellappa, R. (1996). A new approach to image feature detection with applications. *Pattern Recognition*, 29(4), 627–640.
- Manjunath, B. S. and Chellappa, R. (1991). Unsupervised texture segmentation

- using Markov random field models. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 13(5), 478–482.
- Marcum, D. L. and Weatherill, N. P. (1995). Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Journal*, 33(9), 1619–1625.
- Marroquin, J., Vemuri, B., Betello, S., Calderon, F., and Fernandez-Bouzas, A. (2002). An accurate and efficient Bayesian method for automatic segmentation of brain MRI. *IEEE Transactions on Medical Imaging*, 21(8), 934–945.
- Mavriplis, D. J. (1991). *Unstructured and adaptive mesh generation for high Reynolds number viscous flows*. Technical Report 91-25, ICASE, NASA Langly Research Center.
- McKinlay, K. J., Scotchford, C. A., Grant, D. M., Oliver, J., King, J., and Brown, P. (2004). Scanning electron microscopy of biomaterials. In S. McVitie and D. McComb (Eds.), *Proceedings of the Institute of Physics Electron Microscopy and Analysis Group Conference (EMAG'03), 3–5 September, 2003*, Institute of Physics Conference Series (pp. 87–90).
- MeasTex (2004). MeasTex Image Texture Database. Available from <http://www.cssip.uq.edu.au/meastex/meastex.html>, Accessed on 13th March.
- Michie, D., Spiegelhalter, D., and Taylor, C. C., Eds. (1994). *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Series in Artificial Intelligence. New York, NY, USA: Prentice Hall.

- Miller, G. L., Talmor, D., Teng, S.-H., and Walkington, N. (1995). A Delaunay based numerical method for three dimensions: generation, formulation, and partition. In *Proceedings of 27th Annual ACM Symposium on Theory of Computing* (pp. 683–692).
- Mitchell, S. (1994). Cardinality bounds for triangulations with bounded minimum angle. In *Sixth Canadian Conference on Computational Geometry* (pp. 326–331).
- Mitchell, S. A. and Vavasis, S. A. (1992). Quality mesh generation in three dimensions. In *Symposium on Computational Geometry* (pp. 212–221).
- Mitchell, S. A. and Vavasis, S. A. (1996). An aspect ratio bound for triangulating a d -grid cut by a hyperplane. In *Proceedings of the 12th Annual ACM Symposium on Computational Geometry* (pp. 48–57).
- Montas, H. J., Moran, L. B., Peters, C., Shipman, K., Ifft, T., Felton, G., and Shirmohammadi, A. (2000a). Effectiveness of riparian buffers in small maryland watersheds. In *Proceedings of 2000 Annual International Meeting of ASAE* (pp. Paper No. 00–2182). Milwaukee, WI, USA.
- Montas, H. J., Shirmohammadi, A., Haghighi, K., and Engel, B. (2000b). Equivalence of bicontinuum and second-order transport in heterogeneous soils and aquifers. *Water Resources Research*, 36(12), 3427–3446.
- Moore, D. (1995). The cost of balancing generalized quadtrees. In *SMA '95: Proceedings of the Third Symposium on Solid Modelling and Applications* (pp. 305–312).

- Mostafavi, M. A., Gold, C., and Dakowicz, M. (2003). Delete and insert operations in voronoi/delaunay methods and applications. *Computers and Geosciences*, 29(4), 523–530.
- Mücke, E. (1993). *Shapes and Implementations in Three-Dimensional Geometry*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, IL, USA.
- Müller, R. and Rügsegger, P. (1995). Three-dimensional finite element model of non-invasively assessed trabecular bone structures. *Medical Engineering and Physics*, 17(2), 126–133.
- Nackmann, L. R. and Srinivasan, V. (1991). Point placement for delaunay triangulation of polygonal domains. In *Proceedings of 3rd Canadian Conference on Computational Geometry* (pp. 37–40).
- National Library of Medicine (2002). The Visible Human Project. Available from http://www.nlm.nih.gov/research/visible/visible_human.html, Accessed on 10th August.
- Neugebauer, F. and Dickmann, R. (1996). Improved mesh generation: not simple but good. In *Proceedings of 5th International Meshing Roundtable* (pp. 257–270). Albuquerque, NM, USA: Sandia National Laboratories.
- Nguyen-Van-Phai (1982). Automatic mesh generation with tetrahedron elements. *International Journal for Numerical Methods in Engineering*, 18, 273–289.

- Nielson, G. M. (1997). Tools for triangulations and tetrahedrizations. In G. M. Nielson, H. Hagen, and H. Müller (Eds.), *Scientific Visualization: Overviews, Methodologies, and Techniques* (pp. 429–525). Washington, DC, USA: IEEE Computer Society.
- Nielson, G. M. and Sung, J. (1997). Interval volume tetrahedrization. In *Proceedings of the 8th IEEE conference on Visualization (VIS '97), Phoenix, AZ, USA, October 18–24, 1997* (pp. 221–228).: IEEE Computer Society Press.
- Niemann, H. (1990). *Pattern Analysis and Understanding*. New York, NY, USA: Springer-Verlag, 2nd edition.
- Pachepsky, Y. A., Smettem, K. R. J., Vanderborght, J., Herbst, M., Vereecken, H., and Wösten, J. H. M. (2004). Reality and fiction of models and data in soil hydrology. In R. A. Feddes, G. H. d. Rooij, and J. C. v. Dam (Eds.), *Frontis Workshop on Unsaturated-zone modeling progress, challenges and applications, Wageningen, The Netherlands 3–5 October, 2004*, volume 6 of *Wageningen UR Frontis* (pp. 231–260).: Springer.
- Pal, N. P. and Pal, S. K. (1993). A review of image segmentation techniques. *Pattern Recognition*, 26(9), 1277–1294.
- Perucchio, R., Saxena, M., and Kela, A. (1989). Automatic mesh generation from solid models based on recursive spatial decompositions. *International Journal for Numerical Methods in Engineering*, 28, 2469–2502.

- Pitas, I. (2000). *Digital Image Processing Algorithms and Applications*. New York, NY, USA: John Wiley and Sons.
- Preparata, F. P. and Shamos, M. I. (1990). *Computational Geometry: An Introduction*. Springer-Verlag, 3rd edition.
- Rajan, V. T. (1994). Optimality of the Delaunay triangulation in R^d . *Discrete Computational Geometry*, 12, 189–202.
- Rajapakse, J. C. and Fruggel, F. (1998). Segmentation of MR images with intensity inhomogeneities. *Image and Vision Computing*, 16(3), 165–180.
- Randen, T. and Husøy, J. (1999). Filtering for texture classification: A comparative study. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 21(4), 291–310.
- Reed, T. R. and du Buf, J. M. H. (1993). A review of recent texture segmentation and feature extraction techniques. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 57(3), 359–372.
- Reed, T. R. and Wechsler, H. (1990). Segmentation of textured images and Gestalt organization using spatial/spatial-frequency representations. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 12(1), 1–12.
- Reed, T. R. and Wechsler, H. (1991). Spatial/spatial-frequency representations for image segmentation and grouping. *Image and Vision Computing*, 9(3), 175–193.
- Reeve, J. and Barragan, A. (2000a). Parallel three-dimensional finite element mesh

- generation based on octree data structures. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS'00)* Barcelona, Spain.
- Reeve, J. and Barragan, A. (2000b). A two-dimensional parallel quadtree finite element mesh generator. In E. D'Hollander, J. Joubert, F. Peters, and H. Sips (Eds.), *Parallel Computing: Fundamentals and Applications, Proceedings of the International Conference (ParCo'00)* (pp. 251–258).: Imperial College Press.
- Rippa, S. (1984). Algorithm for refining triangular grids suitable for adaptive and multigrid techniques. *International Journal for Numerical Methods in Engineering*, 20, 745–756.
- Rosenfeld, A. and Kak, A. C. (1982). *Digital Picture Processing*, volume 1. Boston, MA, USA: Academic Press, 2nd edition.
- Ruppert, J. (1993). A new and simple algorithm for quality 2-dimensional mesh generation. In *Proceedings of Fourth ACM-SIAM Symposium On Discrete Algorithms* (pp. 83–92).
- Ruppert, J. (1995). A delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18(3), 548–585.
- Saalfeld, A. (1991). Delaunay edge refinements. In *Proceedings of 3rd Canadian Conference on Computational Geometry* (pp. 33–36).
- Samet, H. (1989). *Applications of spatial data structures: computer graphics, image processing and GIS*. Reading, MA, USA: Addison-Wesley.

- Samet, H. (1990). *The design and analysis of spatial data structures*. Reading, MA, USA: Addison-Wesley.
- Schalkoff, R. J. (1992). *Pattern Recognition: Statistical, Structural, and Neural Approaches*. New York, NY, USA: John Wiley and Sons.
- Schroeder, W. J. and Shephard, M. S. (1990). A combined octree Delaunay method for fully-automatic 3-D mesh generation. *International Journal for Numerical Methods in Engineering*, 29(1), 37–55.
- Seidel, R. (1993). Backwards analysis of randomized geometric algorithms. In J. Pach (Ed.), *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics* (pp. 37–68). New York, NY, USA: Springer-Verlag.
- Serra, J. (1982). *Image Analysis and Mathematical Morphology*. Boston, MA, USA: Academic Press.
- Sethian, J. A. (1999). *Level Set methods and Fast Marching Methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*. Number 3 in Cambridge monographs on applied and computational mathematics. Cambridge, UK: Cambridge University Press, 2nd edition.
- Shamos, M. I. and Hoey, D. (1975). Closest-point problems. In *Proceedings of 16th Annual IEEE Symposium on Foundations of Computer Sciences* (pp. 151–162).
- Shekhar, R., Fayyad, E., Yagel, R., and Cornhill, J. F. (1996). Octree-based decimation of marching cubes surfaces. In *Proceedings of the 7th IEEE conference on*

- Visualization (VIS '96), San Francisco, CA, USA*, (pp. 335–342).: IEEE Computer Society Press.
- Shephard, M. S. and Georges, M. K. (1991). Automatic three-dimensional mesh generation by the finite octree technique. *International Journal for Numerical Methods in Engineering*, 32, 709–749.
- Shewchuk, J. R. (1996a). *Adaptive precision floating point arithmetic and fast robust geometric predicates*. Technical Report CMU-CS-96-140, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA.
- Shewchuk, J. R. (1996b). Triangle: Engineering a 2-d quality mesh generator and Delaunay triangulator. In *First Workshop on Applied Computational Geometry*: Association for Computing Machinery.
- Shewchuk, J. R. (1998). Tetrahedral mesh generation by delaunay refinement. In *Symposium on Computational Geometry* (pp. 86–95).
- Shewchuk, J. R. (2002). What is a good linear element? interpolation, conditioning, and quality measures. In *Proceedings of 11th International Meshing Roundtable* (pp. 115–126). Albuquerque, NM, USA: Sandia National Laboratories.
- Shimada, K. and Gossard, D. C. (1992). Computational methods for physically-based FE mesh generation. In *Proceedings of IFIP TC5/WG5.3 8th International PROLAMAT Conference on Human Aspects in Computer Integrated Manufacturing*.

- Shirley, P. and Tuchman, A. A. (1990). Polygonal approximation to direct scalar volume rendering. In *ACM Computer Graphics (Proceedings of 1990 Workshop on Volume Visualization, San Diego, CA, USA)*, volume 24(5) (pp. 63–70).
- Sibson, R. (1978). Locally equiangular triangulations. *The Computer Journal*, 21(3), 243–245.
- Sivan, R. (1996). *Surface modeling using quadtrees*. thesis, University of Maryland, College Park, MD, Center for Automation, Department of Computer Science.
- Strang, G. and Nguyen, T. (1997). *Wavelets and Filter Banks*. Wellesley, MA, USA: Wellesely-Cambridge Press, 2nd edition.
- Teng, S. H. and Wong, C. W. (2000). Unstructured mesh generation: Theory, practice, and perspectives. *International Journal of Computational Geometry and Applications*, 10(3), 227–266.
- Teuner, A., Pichler, O., and Hosticka, B. J. (1995). Unsupervised texture segmentation of images using tuned matched Gabor filters. *IEEE Transactions of Image Processing*, 4(6), 863–870.
- Tracqui, P., Cruywagen, G., Woodward, D., Bartoo, G., Murray, J., and Alvord, Jr., E. (1995). A mathematical model of glioma growth: the effect of chemotherapy on spatio-temporal growth. *Cell Proliferation*, 28, 17–31.
- Unser, M. (1995). Texture classification and segmentation using wavelet frames. *IEEE Transactions of Image Processing*, 4(11), 1549–1560.

- Unser, M. and Eden, M. (1989). Multiresolution feature extraction and selection for texture segmentation. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 11(7), 717–728.
- Vandergheynst, P. and Gobbers, J. (2002). Directional dyadic wavelet transforms: design and algorithms. *IEEE Transactions of Image Processing*, 11(4), 1057–7149.
- Vapnik, V. N. (1998). *Statistical Learning Theory*. New York, NY, USA: John Wiley and Sons, Inc.
- Vavasis, S. (2002). QMG: a finite element mesh generation package. Available from <http://www.cs.cornell.edu/Info/People/vavasis/qmg-home.html>, Accessed on 10th August.
- Vision Textures (2004). MIT Media Lab – Vision and Modelling Group. Available from <http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>, Accessed on 13 March.
- Wang, Y., Gu, X., and Yau, S.-T. (2004). Volumetric harmonic map. *Communication in Information and Systems*, 3(3), 191–202.
- Watson, D. F. (1981). Computing the n -dimensional Delaunay tessellation with applications to Voronoi polytopes. *The Computer Journal*, 24(2), 167–172.
- Weatherhill, N. P. and Hassan, O. (1994). Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering*, 37, 2005–2039.

- Webb, A. (2002). *Statistical Pattern Recognition*. New York, NY, USA: John Wiley and Sons, 2nd edition.
- Weszka, J., Deya, C., and Rosenfeld, A. (1976). A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man and Cybernetics*, 6, 269–285.
- Wilhelms, J. and Geldren, A. V. (1992). Octrees for faster iso-surface generation. *ACM Transactions on Graphics*, 11(3), 201–297.
- Woods, R., Sivapalan, M., and Duncan, M. (1995). Investigating the Representative Elementary Area concept: An approach based on field data. *Hydrological Processes*, 9(3), 291–312.
- Yang, L. and Mishra, M. (1998). Coarse-grained parallel algorithms for multi-dimensional wavelet transforms. *The Journal of Supercomputing*, 12(12).
- Yerry, M. A. and Shephard, M. S. (1983). A modified quadtree approach to finite element mesh generation. *IEEE Computer Graphics and Applications*, 3, 39–46.
- Yerry, M. A. and Shephard, M. S. (1984). Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering*, 20, 1965–1990.
- Yerry, M. A. and Shephard, M. S. (1991). Automatic three-dimensional mesh generation by the finite octree technique. *International Journal for Numerical Methods in Engineering*, 32, 704–749.

- Young, B. and Heath, J. W. (1999). *Wheater's Functional Histology : a text and color atlas*. New York, NY, USA: Churchill Livingstone, 4th edition.
- Zeng, X., Staib, L. H., Schultz, R. T., and Duncan, J. S. (1999). Segmentation and measurement of the cortex from 3D MR images using coupled surface propagation. *IEEE Transactions on Medical Imaging*, 18(10), 100–111.
- Zhang, Y., Bajaj, C. L., and Sohn, B.-S. (2003). Adaptive and quality 3D meshing from imaging data. In *Proceedings of the 8th ACM Symposium on Solid Modeling and Applications (SMA '03), Seattle, WA, USA, June 16–20, 2003* (pp. 31–43).
- Zhang, Y., Brady, M., and Smith, S. (2001). Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Transactions On Medical Imaging*, 20(1), 45–57.
- Zhou, Y., Chen, B., and Kaufman, A. (1997). Multiresolution tetrahedral framework for visualizing regular volume data. In *Proceedings of the 8th IEEE conference on Visualization (VIS '97), Phoenix, AZ, USA, October 18–24, 1997* (pp. 135–142).: IEEE Computer Society Press.
- Zhu, C. and Yang, X. (1998). Study of remote sensing image texture analysis and classification using wavelets. *International Journal of Remote Sensing*, 19(16), 3197–3203.
- Zhu, Y., Goutte, R., and Amiel, M. (1993). On the use of a two-dimensional Wigner-Ville distribution for texture segmentation. *Signal Processing*, 30(3), 329–353.

Zijdenbos, A., Dawant, B., Margolin, R., and Palmer, A. (1994). Morphometric analysis of white matter lesions in MR images: Method and validation. *IEEE Transactions on Medical Imaging*, 13(12), 716–724.